Formal Efficiency Analysis for Tree Transducer Composition

Janis Voigtländer*

Department of Computer Science, Dresden University of Technology,
01062 Dresden, Germany. E-mail: voigt@tcs.inf.tu-dresden.de

Abstract

We study the question of efficiency improvement or deterioration for a semantics-preserving program transformation technique for (lazy) functional languages, based on composition of restricted macro tree transducers. By annotating programs to reflect the intensional property "computation time" explicitly in the computed output and by manipulating such annotations, we formally prove syntactic conditions under which the composed program is guaranteed to be not less efficient than the original program with respect to the number of call-by-name reduction steps required to reach normal form. Under additional conditions the guarantee also holds for call-by-need semantics. The criteria developed can be checked automatically and efficiently, and thus are suitable for integration into an optimizing compiler.

1 Introduction

Automatic transformation of programs is a key technology in software engineering, as it enables programmers to work at a higher level of abstraction than would otherwise be possible and thus gives rise to increases in programmer productivity. Prominent among its many applications to lazy functional languages in particular is the elimination of intermediate results, with the aim of mitigating the tension between modularity—as desired for the sake of reliability, maintainability, and reusability—and efficiency of programs. Under the name *deforestation*, automatic elimination of structured intermediate results (i.e. trees) was pioneered in [Wad90] through an algorithmic instance of the *unfold/fold-technique* [BD77] that is suitable for integration into an optimizing compiler. This unfold/fold-based approach has seen many extensions—e.g. in [HJ92], [MW93], and [Chi94]—, which we will collectively refer to as *classical deforestation* in the following. A fundamental weakness of classical deforestation is its inability to eliminate intermediate results that are built inside so called *accumulating parameters*. This problem is also shared by *shortcut deforestation* techniques [GLP93, Chi99, Joh02, Sve02], based on denoting producers and consumers of intermediate results in terms of certain higher-order, polymorphic combinators.[1]

An accumulating parameter is one that may grow in recursive function calls. For example, in the following rules defining a function g with two parameters:

$$\begin{array}{lll} g\ (\delta\ v_1\ v_2) & z_1 & \rightarrow \delta\ (g\ v_1\ (g\ v_2\ z_1)) \\ g\ (\alpha\ v_1) & z_1 & \rightarrow \alpha\ (g\ v_1\ z_1) \\ g\ (\beta\ v_1) & z_1 & \rightarrow \beta\ (g\ v_1\ z_1) \\ g\ \epsilon & z_1 & \rightarrow \epsilon\ z_1 \end{array}$$

*Research supported by the "Deutsche Forschungsgemeinschaft" under grant KU 1290/2-4.

[1] Note that [Sve02] succeeds in handling accumulating parameters of **consumers**, but does not approach the more difficult problem of **producers** using accumulating parameters.

every recursive call performs a strict descent in the first parameter, whereas the second parameter is accumulating (cf. the first rule). These rules together with an initial expression $g\ v_1\ \gamma$ constitute the favored solution to the task of computing (as a monadic output tree, i.e., essentially as a list) the prefix traversal of a tree built from binary symbol δ, unary symbols α and β, and nullary symbol ϵ. In fact, accumulating style is a quite common programming idiom in functional languages. As a consequence, the problem of eliminating intermediate results produced inside accumulating parameters has received much attention [Küh98, Küh99, CDPR99, KGF02, Nis02, VK04a, Voi04]. All solutions proposed so far derive from restricting the class of input programs to certain extended schemes of primitive recursion, which can be formalized as (deterministic) *macro tree transducers* [Eng80].

A macro tree transducer (for short *mtt*) translates trees over a ranked alphabet of input symbols into trees over a ranked alphabet of output symbols. For this translation process an mtt uses a set of functions which are defined by rewrite rules. More precisely, every function is defined by pattern matching on the root symbol of its first argument t. The right-hand side of every rule may contain the other function arguments, output symbols, and recursive function calls—potentially to other functions of the mtt—, the first arguments of which must be variables that refer to subtrees of t. As a simple example of an mtt, consider the above program for prefix traversal. Since many typical functions on algebraic data types are defined by structural descent on a distinguished argument, mtts represent a large class of functional programs using accumulating parameters, which are called *context parameters* in the scope of mtts. For example, manipulation of abstract syntax trees in compilers often follows the recursion scheme of mtts [Vog91, FV98], and the "tree transformation core" of XML processing languages can be compiled into compositions of mtts [EM03a, MBPS04].

By transforming functional programs corresponding to restricted mtts into *attribute grammars* [Knu68] or *attributed tree transducers* [Fül81], performing a composition construction on that level, and transforming the result back into a functional program, [Küh98] and [CDPR99] were the first to achieve elimination of intermediate results produced inside accumulating parameters. The transformation technique consisted of a combination of constructions from [Fül81, Fra82, CF82, Gan83, GG84, Gie88] and relied on certain syntactic restrictions imposed on the original mtts, viz. *weakly single-useness* (as introduced in [Küh98]) and *context-linearity* (meaning that context parameters are never copied). In [Voi01, VK04a] a direct construction without detour to the "attribute world" was presented that is applicable to a pair of less restricted mtts. In particular, it successfully performs a composition also if the producer of the intermediate result is an arbitrary mtt, while the consumer is a *top-down tree transducer* (for short *tdtt*; [Rou70, Tha70, Eng75]), i.e., an mtt without context parameters. Thus, it captures a result from [EV85] that was used in [Küh99] as another application of tree transducer theory to accumulating parameter deforestation. In fact, none of the techniques proposed to date for eliminating intermediate results produced inside accumulating parameters handles a richer class of programs than the construction from [VK04a]. Its direct nature made it suitable for integration into an optimizing compiler [Reu03].

Beside preservation of program semantics (which for the construction from [VK04a] was proved in [Voi01, VK04b]) more intensional properties are in focus when considering program transformations. The key question usually to be asked about a source-to-source transformation implemented in a compiler is whether it will make the compiled program more efficient in some sense. Elimination of intermediate results aims in this direction by avoiding memory allocations and deallocations, but this alone is not sufficient. In fact, none of the deforestation techniques mentioned above is guaranteed, when applied to arbitrary input programs, to produce output programs that are faster—or at least not slower—than the original ones. The standard result about classical deforestation is that it does not deteriorate the efficiency of linear input programs, where efficiency is measured by counting *call-by-need* reduction steps. Examples show that for tree transducer composition linearity of input programs is neither a sufficient nor a necessary condition for efficiency nondeterioration with respect to this measure. For the special case that one of the mtts to be composed is a tdtt sufficient conditions were developed in [Küh99] and [Höf99] using rather ad-hoc reasoning. In [Voi02a] we have continued this study using a more general approach, which in the present paper is extended to the full construction from [VK04a], thus providing the first formal efficiency guarantees for general tree transducer composition.[2] The criteria obtained improve on previous results and cover many typical examples. They are decidable and we even present an efficient decision procedure.

Our strategy is based on externalizing the intensional property "computation time" by making it part of the observable program output, thus making it amenable to methods developed for reasoning about extensional properties:

- First, we enrich the original and the composed program, respectively, by annotating their rules with special symbols and adding new rules in such a way that from the output trees computed for a particular input tree one can read off the numbers of *call-by-name* reduction steps performed by the programs under consideration for that input.

- Then, we devise another annotation for the original program such that applying the composition construction to the thus annotated original program yields essentially the annotated version of the composed program counting reduction steps. What was a simple exercise for the special case considered in [Voi02a] requires some imagination for the general case. In fact, we consider the presented annotation (and the associated proof establishing its suitability) a key contribution of this paper because on an intuitive level it exposes the essence of the rather complicated construction from [VK04a] in a surprisingly elegant way, while on the technical side it allows to completely remove the details of the composition construction from concern in any further efficiency analysis.

- Left with annotated versions of the original program only, we can combine and manipulate annotations, trading those special symbols that correspond to steps of the composed program against those that correspond to steps of the original program. The aim is to arrive at an annotation for which simple conditions suffice to guarantee that for every input the number of symbols produced of the former kind is outweighed by that of symbols produced of the latter kind. Some of the involved manipulations will be safe—i.e., will not lead to an overestimation of the number of steps performed by the original program or an underestimation of the number of steps performed by the composed program—only under certain conditions that will also be recorded. Proofs will be modularized by "factoring out" what are the essential properties guaranteed

[2] Note than none of the related techniques for deforesting inside accumulating parameters [CDPR99, KGF02, Nis02, Voi04] comes with any efficiency guarantees whatsoever.

by those additional conditions, thus identifying potential levers for generalizing the analysis.

– Having established sufficient conditions for efficiency nondeterioration under call-by-name reduction, we can transfer the results to the call-by-need setting by giving conditions under which the costs measured under the two reduction strategies do not differ for the original program. In contrast to the situation for classical deforestation, where this is typically achieved by restricting input programs to be linear, the particular recursion scheme of mtts allows for more liberal conditions to be used here. In particular, we employ *basicness* [Vog87].[3]

The remainder of this paper is organized as follows. Section 2 defines necessary notions and introduces the basic concepts of mtts. Section 3 recalls the composition construction to be studied. Section 4 develops our efficiency analysis. Section 5 considers related work. Section 6 concludes. Some proofs can be found in the appendix.

2 Preliminaries

We denote by $\mathbb{N}$ the set of natural numbers including 0 and by $\mathbb{Z}$ the set of all integers. We set $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$. For every $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. The mapping *max* gives for every finite, nonempty subset of $\mathbb{N}$ the maximum of that subset's elements.

Let S be a set. We denote by S^* the set of finite sequences of elements of S, where ε denotes the empty sequence. The power set of S is denoted by $\mathcal{P}(S)$. If S is finite, then the number of its elements is denoted by $|S|$. The cartesian product of sets $S_1, \ldots, S_n$ is denoted by $S_1 \times \cdots \times S_n$. The n-fold cartesian product of one set S is denoted by S^n. Given a binary relation $\Rightarrow\, \in \mathcal{P}(S^2)$, we write $s \Rightarrow s'$ for $(s, s') \in\, \Rightarrow$. We denote by $\Rightarrow^n$ (with $n \in \mathbb{N}$) the n-fold composition, by $\Rightarrow^?$ the reflexive, and by $\Rightarrow^*$ the reflexive, transitive closure of $\Rightarrow$, respectively (all in $\mathcal{P}(S^2)$). If for every $s \in S$ there exists at most one $s' \in S$ with $s \Rightarrow s'$, then $\Rightarrow$ is called *deterministic*. If for every $s, s_1, s_2 \in S$ with $s \Rightarrow^* s_1$ and $s \Rightarrow^* s_2$ there exists $s' \in S$ with $s_1 \Rightarrow^* s'$ and $s_2 \Rightarrow^* s'$, then $\Rightarrow$ is called *confluent*. Determinism implies confluence. If there is no infinite chain $s_1 \Rightarrow s_2 \Rightarrow s_3 \Rightarrow \cdots$, then $\Rightarrow$ is called *terminating*. If for a given $s \in S$ there is no s' with $s \Rightarrow s'$, then s is called *irreducible* with respect to $\Rightarrow$. If $s \Rightarrow^* s'$ and s' is irreducible w.r.t. $\Rightarrow$, then s' is called a *normal form* of s w.r.t. $\Rightarrow$. If $\Rightarrow$ is confluent and terminating, then every $s \in S$ has a unique normal form, denoted by $nf(\Rightarrow, s)$.

We denote by *id* the identity function on every set. We use two symmetric versions of function composition, namely for every sets S_1, S_2, S_3, and functions $f : S_1 \longrightarrow S_2$ and $g : S_2 \longrightarrow S_3$ we define $f; g = g \cdot f : S_1 \longrightarrow S_3$ such that for every $s \in S_1$: $(g \cdot f)(s) = g(f(s))$. For every function $f : S \longrightarrow S$ and $n \in \mathbb{N}$ the n-fold composition of f is denoted by f^n, where $f^0 = id$. The product of functions $f_1 : S_1 \longrightarrow S'_1, \ldots, f_n : S_n \longrightarrow S'_n$ is denoted by $f_1 \times \cdots \times f_n : S_1 \times \cdots \times S_n \longrightarrow S'_1 \times \cdots \times S'_n$ and defined by $(f_1 \times \cdots \times f_n)(s_1, \ldots, s_n) = (f_1(s_1), \ldots, f_n(s_n))$. For two functions $f : S \longrightarrow \mathbb{Z}$ and $g : S \longrightarrow \mathbb{Z}$ we write $f \leq g$ iff $f(s) \leq g(s)$ for every $s \in S$, and we write $f - g : S \longrightarrow \mathbb{Z}$ for the function that maps every $s \in S$ to $f(s) - g(s)$.

[3]Note that in [Voi02a], rather than first studying call-by-name efficiency and then transferring the results to the call-by-need setting, the same conditions were used from the outset to study only call-by-need efficiency.

2.1 Trees, substitutions, and rewrite systems

A *ranked alphabet* is a pair $(\Sigma, rank_\Sigma)$, where Σ is a finite set of symbols and $rank_\Sigma$ assigns to each of these symbols a natural number, its *rank*. In the following, we drop the $rank_\Sigma$-function from the denotation and only mention Σ when referring to a ranked alphabet. For every $p \in \mathbb{N}$ we define $\Sigma^{(p)} = \{\sigma \in \Sigma \mid rank_\Sigma(\sigma) = p\}$. The rank p of a symbol σ is also denoted by writing $\sigma^{(p)}$. A *nullary* symbol is one of rank 0; a *unary* symbol one of rank 1. For the sake of brevity, quantifications over a symbol in a ranked alphabet implicitly quantify also its rank. E.g., we write "for every $\sigma \in \Sigma^{(p)}$" instead of "for every $p \in \mathbb{N}$, $\sigma \in \Sigma^{(p)}$" and "there exists $f \in F^{(r+1)}$" instead of "there exist $r \in \mathbb{N}$ and $f \in F^{(r+1)}$". For a ranked alphabet Σ we denote the set of all its ranks as $rank(\Sigma) = \{p \in \mathbb{N} \mid \exists \sigma \in \Sigma.\ rank_\Sigma(\sigma) = p\}$.

We use several sets of lowercase variables. We denote by U the set $\{u_1, u_2, u_3, \ldots\}$ of variables, and for every $p \in \mathbb{N}$ by U_p the finite set $\{u_1, \ldots, u_p\} \subseteq U$; analogously for V, Y, Z, and $Y' = \{y'_1, y'_2, y'_3, \ldots\}$.

For a ranked alphabet Σ and a set X of variables disjoint from Σ we define the set $T_\Sigma(X)$ of *trees over* Σ *indexed by* X as the smallest set $T \subseteq (\Sigma \cup X \cup \{(,)\})^*$ such that (i) $X \subseteq T$ and (ii) for every $\sigma \in \Sigma^{(p)}$ and $t_1, \ldots, t_p \in T$: $(\sigma\ t_1 \cdots t_p) \in T$. If readability allows, outer brackets of trees are omitted. Note that every symbol $\sigma \in \Sigma^{(p)}$ can also be considered as a function of type $T_\Sigma(X)^p \longrightarrow T_\Sigma(X)$, mapping $(t_1, \ldots, t_p)$ to $\sigma\ t_1 \cdots t_p$. We denote the set $T_\Sigma(\emptyset)$ by T_Σ.

We need the *set of paths* in a tree, the *label at a path in a tree*, the *subtree at a path in a tree*, and the *number of occurrences of some symbol* $s \in \Sigma \cup X$ *in a tree*, given by the functions *paths*, *lab*, *sub*, and $|\,.\,|_s$, which are defined, with $x \in X$, $\sigma \in \Sigma^{(p)}$, $i \in \mathbb{N}_+$, $\pi \in (\mathbb{N}_+)^*$, and $t, t_1, \ldots, t_p \in T_\Sigma(X)$, as follows:

$$
\begin{aligned}
&paths : T_\Sigma(X) \longrightarrow \mathcal{P}((\mathbb{N}_+)^*)\\
&paths(x) &&= \{\varepsilon\}\\
&paths(\sigma\ t_1 \cdots t_p) &&= \{\varepsilon\} \cup \{i\pi \mid i \in [p], \pi \in paths(t_i)\}
\end{aligned}
$$

$$
\begin{aligned}
&lab : \{(t, \pi) \mid t \in T_\Sigma(X), \pi \in paths(t)\} \longrightarrow \Sigma \cup X\\
&lab(x, \varepsilon) &&= x\\
&lab(\sigma\ t_1 \cdots t_p, \varepsilon) &&= \sigma\\
&lab(\sigma\ t_1 \cdots t_p, i\pi) &&= lab(t_i, \pi)
\end{aligned}
$$

$$
\begin{aligned}
&sub : \{(t, \pi) \mid t \in T_\Sigma(X), \pi \in paths(t)\} \longrightarrow T_\Sigma(X)\\
&sub(t, \varepsilon) &&= t\\
&sub(\sigma\ t_1 \cdots t_p, i\pi) &&= sub(t_i, \pi)
\end{aligned}
$$

$$
\begin{aligned}
&|\,.\,|_s : T_\Sigma(X) \longrightarrow \mathbb{N}\\
&|x|_s = 0, \text{ if } x \neq s && |s|_s = 1, \text{ if } s \in X\\
&|\sigma\ t_1 \cdots t_p|_s = \sum_{i \in [p]} |t_i|_s, \text{ if } \sigma \neq s && |s\ t_1 \cdots t_p|_s = 1 + \sum_{i \in [p]} |t_i|_s, \text{ if } s \in \Sigma^{(p)}.
\end{aligned}
$$

The label $lab(t, \varepsilon)$ is called the *root symbol* of the tree t.

We use two kinds of substitution over trees, written postfix and left-associative. For some $n \in \mathbb{N}$, a ranked alphabet Σ', and a set X' of variables such that $(\Sigma \cup \Sigma') \cap (X \cup X') = \emptyset$, the first-order substitution $.\,[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n]$ for pairwise different variables $x_1, \ldots, x_n \in X$ and trees $t'_1, \ldots, t'_n \in T_{\Sigma'}(X')$, and the second-order substitution

- $[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n]$ for pairwise different symbols $\sigma_1 \in \Sigma^{(p_1)}, \ldots, \sigma_n \in \Sigma^{(p_n)}$ and functions $f_1 : T_{\Sigma\cup\Sigma'}(X)^{p_1} \longrightarrow T_{\Sigma\cup\Sigma'}(X), \ldots, f_n : T_{\Sigma\cup\Sigma'}(X)^{p_n} \longrightarrow T_{\Sigma\cup\Sigma'}(X)$, are defined, with $x \in X$, $\sigma \in \Sigma^{(p)}$, and $t_1, \ldots, t_p \in T_\Sigma(X)$, as follows:

$$
\begin{aligned}
&\text{\textbullet}\,[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n] : T_\Sigma(X) \longrightarrow T_{\Sigma\cup\Sigma'}((X \setminus \{x_1, \ldots, x_n\}) \cup X') \\
x[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n] &= x, && \text{if } x \notin \{x_1, \ldots, x_n\} \\
x_i[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n] &= t'_i, && \text{for } i \in [n] \\
(\sigma\ t_1 \cdots t_p)[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n] &= \sigma\ t_1[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n] \\
&\quad \cdots \\
&\quad t_p[x_1, \ldots, x_n \leftarrow t'_1, \ldots, t'_n]
\end{aligned}
$$

$$
\begin{aligned}
&\text{\textbullet}\,[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n] : T_\Sigma(X) \longrightarrow T_{\Sigma\cup\Sigma'}(X) \\
x[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n] &= x \\
(\sigma\ t_1 \cdots t_p)[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n] &= \sigma\ t_1[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n] \\
&\quad \cdots \\
&\quad t_p[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n], \text{ if } \sigma \notin \{\sigma_1, \ldots, \sigma_n\} \\
(\sigma_i\ t_1 \cdots t_p)[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n] &= f_i(t_1[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n], \\
&\quad \ldots, \\
&\quad t_p[\sigma_1, \ldots, \sigma_n \twoheadleftarrow f_1, \ldots, f_n]), \text{ for } i \in [n], p = p_i.
\end{aligned}
$$

We also use alternative notations similar to set comprehensions, e.g. $t[x_i \leftarrow t'_i \mid i \in [n]]$, and appropriate multi-line notations for long substitutions. Note that substitution binds stronger than tree forming, i.e., "$\sigma\ t[\cdots]$" for a unary symbol σ means "$\sigma\ (t[\cdots])$" rather than "$(\sigma\ t)[\cdots]$".

Let Σ be a ranked alphabet and X a set of variables, where $\Sigma \cap X = \emptyset$. A *rewrite rule* over Σ and X is a rule of the form $lhs \to rhs$ with $lhs, rhs \in T_\Sigma(X)$ such that the left-hand side lhs does not contain two occurrences of the same variable and every variable occurring in the right-hand side rhs is also contained in lhs. A set R of rewrite rules over Σ and X is called a *rewrite system* (over Σ and X)[4] [DJ90, BN98]. For every $\Sigma' \supseteq \Sigma$ it induces a binary *reduction relation* $\Rightarrow_R \subseteq T_{\Sigma'} \times T_{\Sigma'}$ such that $t \Rightarrow_R t'$ iff R contains a rule $lhs \to rhs$, there is a tree $c \in T_{\Sigma'}(\{x\})$ (with $x \notin \Sigma' \cup X$) that contains x exactly once, and there exist $n \in \mathbb{N}$, trees $t_1, \ldots, t_n \in T_{\Sigma'}$, and pairwise different variables $x_1, \ldots, x_n \in X$ such that $t = c[x \leftarrow lhs[x_1, \ldots, x_n \leftarrow t_1, \ldots, t_n]]$ and $t' = c[x \leftarrow rhs[x_1, \ldots, x_n \leftarrow t_1, \ldots, t_n]]$.

2.2 Macro tree transducers and syntactic restrictions

Definition 2.1 (macro tree transducer, RHS)
A *macro tree transducer* (for short *mtt*) M is a tuple $(F, \Sigma, \Delta, e, R)$ consisting of:

- a ranked alphabet F of *states*, where $F^{(0)} = \emptyset$ and $F \neq \emptyset$,
- a ranked alphabet Σ of *input symbols*, where $\Sigma^{(0)} \neq \emptyset$ and $F \cap \Sigma = \emptyset$,
- a ranked alphabet Δ of *output symbols*, where $\Delta^{(0)} \neq \emptyset$ and $F \cap \Delta = \emptyset$,
- an *initial expression* $e \in RHS(F, \Delta, U_1, \emptyset)$, and

[4] When Σ and X are clear from the context, we only mention R. Note that a rewrite system over Σ and X is also a rewrite system over $\Sigma' \supseteq \Sigma$ and $X' \supseteq X$ if $\Sigma' \cap X' = \emptyset$.

– a set R containing for every $f \in F^{(r+1)}$ and $\sigma \in \Sigma^{(p)}$ exactly one *rule* of the form $f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \rightarrow rhs_{M,f,\sigma}$, where $rhs_{M,f,\sigma} \in RHS(F, \Delta, U_p, Y_r)$.

Here for variable sets X and X' the set $RHS(F, \Delta, X, X')$ is the smallest set $RHS \subseteq T_{F\cup\Delta}(X \cup X')$ satisfying the following conditions:

– $X' \subseteq RHS$,

– for every $\delta \in \Delta^{(q)}$ and $\phi_1, \ldots, \phi_q \in RHS$: $(\delta\ \phi_1 \cdots \phi_q) \in RHS$, and

– for every $f \in F^{(r+1)}$, $x \in X$, $\phi_1, \ldots, \phi_r \in RHS$: $(f\ x\ \phi_1 \cdots \phi_r) \in RHS$. □

Note that R in the above definition is a rewrite system over $F \cup \Sigma \cup \Delta$ and $U \cup Y$. For $\sigma \in \Sigma$, every rule in R of the form $f\ (\sigma\ \cdots)\ \cdots \rightarrow \cdots$ for some $f \in F$ is called a *σ-rule*. A subtree of the form $(f\ t\ \cdots)$ is referred to as a *call* to f on t. The first argument of a state f is called *recursion argument*; the others, *context parameters*. Correspondingly, variables from U are called *recursion variables* and variables from Y are called *context variables*. However, the actual variable names used in mtt rules are not fixed to come from U_p and Y_r for some $p, r \in \mathbb{N}$; consistent renaming is allowed. For example, we later use recursion variables from V and context variables from Z for the second mtt in the composition construction (with states in G, input symbols in Δ, and output symbols in Ω), in the way we have already done for the introductory example.

Example 2.2 (two mtts, to be used as running example)
Let $\Sigma_{mon} = \{\alpha^{(1)}, \beta^{(1)}, \epsilon^{(0)}\}$, $\Delta_{tree} = \{\delta^{(2)}, \alpha^{(1)}, \beta^{(1)}, \epsilon^{(0)}\}$, and $\Omega_{mon} = \{\delta^{(1)}, \alpha^{(1)}, \beta^{(1)}, \epsilon^{(1)}, \gamma^{(0)}\}$. We define the mtts $M_{spine} = (\{f_1^{(2)}, f_2^{(3)}\}, \Sigma_{mon}, \Delta_{tree}, f_1\ u_1\ \epsilon, R_{spine})$ with the set of rules

$$
\begin{array}{rlll}
R_{spine}: & f_1\ (\alpha\ u_1) & y_1 & \rightarrow \alpha\ (f_1\ u_1\ y_1) \\
& f_1\ (\beta\ u_1) & y_1 & \rightarrow f_2\ u_1\ y_1\ (\beta\ \epsilon) \\
& f_1\ \epsilon & y_1 & \rightarrow y_1 \\
& f_2\ (\alpha\ u_1) & y_1\ y_2 & \rightarrow \alpha\ (f_1\ u_1\ (\delta\ y_1\ y_2)) \\
& f_2\ (\beta\ u_1) & y_1\ y_2 & \rightarrow f_2\ u_1\ y_1\ (\beta\ (\beta\ y_2)) \\
& f_2\ \epsilon & y_1\ y_2 & \rightarrow \delta\ y_1\ y_2
\end{array}
$$

and $M_{pfx} = (\{g^{(2)}\}, \Delta_{tree}, \Omega_{mon}, g\ v_1\ \gamma, R_{pfx})$, where R_{pfx} contains the rules for g given in the introduction. □

The semantics of an mtt is a function from trees over the input ranked alphabet to trees over the output ranked alphabet. It is given by substituting the input tree for u_1 (or for v_1 if recursion variables are drawn from V) in the initial expression of an mtt and then computing the normal form of this expression with respect to the reduction relation induced by the mtt's set of rules. This normal form exists and is unique because the rules of any mtt induce a confluent and terminating reduction relation (cf. e.g. [FV98]).

Definition 2.3 (semantics of an mtt)
The *tree transduction induced by an mtt* $M = (F, \Sigma, \Delta, e, R)$ is the total function $\tau_M : T_\Sigma \longrightarrow T_\Delta$ that assigns to every tree $t \in T_\Sigma$ the value $nf(\Rightarrow_R, e[u_1 \leftarrow t])$. □

It is easy to see that during the reduction of $e[u_1 \leftarrow t]$ with $\Rightarrow_R$ only well-formed terms as defined in the following can occur.

Definition 2.4 (well-formed terms)
Let $M = (F, \Sigma, \Delta, e, R)$ be an mtt. The set WFT_M of *well-formed terms for* M is the smallest set $WFT \subseteq T_{F \cup \Delta \cup \Sigma}$ satisfying the following conditions:

- for every $f \in F^{(r+1)}$, $t \in T_\Sigma$, and $\phi_1, \ldots, \phi_r \in WFT$: $(f\ t\ \phi_1 \cdots \phi_r) \in WFT$, and
- for every $\delta \in \Delta^{(q)}$ and $\phi_1, \ldots, \phi_q \in WFT$: $(\delta\ \phi_1 \cdots \phi_q) \in WFT$. $\square$

The tree transduction induced by the mtt M_{spine} from Example 2.2 maps every element of $T_{\Sigma_{mon}}$ to one of $T_{\Delta_{tree}}$ as depicted in Figure 1. While on first sight that transformation might not seem to be most useful[5], it highlights several capabilities of mtts. Firstly, there is the possibility of spatially separating the output generated for adjacent parts of the input, realized here by directly reproducing encountered α-symbols at the top of the output whereas β-symbols are accumulated in context parameters. Secondly, there is the "finite control" aspect of using different states to remember the last input symbol read and so recognize the beginning and ending of β-"blocks". Finally, note that an mtt can reverse an order found in the input, manifested in our example in that the output produced for the top-most β-block appears lowest in the final output, and vice versa.

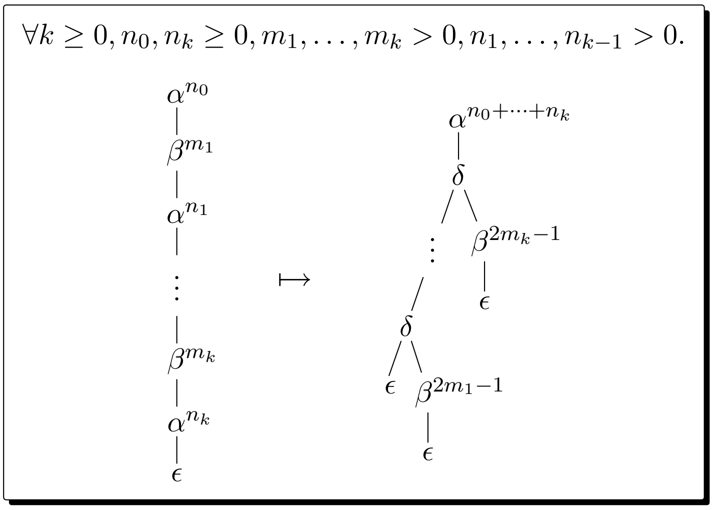

Figure 1: $\tau_{M_{spine}}$

Moreover, the mtts M_{spine} and M_{pfx} serve as a good running example because they seem to constitute about the simplest reasonable composition pair experiencing all the phenomena we later want to illustrate when developing our efficiency analysis. Compared to a more straight-forward looking example, it also has the advantage of requiring a quite disciplined, general approach rather than tempting us to found the analysis on too specific assumptions about the program to be transformed. In particular, M_{spine} is complex enough to defeat simplistic analysis attempts based solely on the size of—i.e. the number of symbols in—the intermediate result compared to that of the original input.[6]

We introduce some syntactic restrictions of mtts that are used throughout the paper.

[5] For more practical examples see [VK04a] and also [KV01], where it was demonstrated how typical functions on polymorphic data types and some higher-order functions like *map* can be viewed as mtts.

[6] For every input the size increase realized by $\tau_{M_{spine}}$ lies between (including) 1 and 2, and every rational number in that interval occurs as size increase for infinitely many different input trees.

Definition 2.5 (syntactic restrictions of mtts)
An mtt $M = (F, \Sigma, \Delta, e, R)$ is:

- a *top-down tree transducer* (for short *tdtt*) if $F = F^{(1)}$
- *basic* if e and the right-hand sides of the rules in R do not contain any nested calls, i.e. subtrees of the form $(f\ \cdots (f'\ \cdots) \cdots)$ with $f, f' \in F$
- *weakly single-use* if for every $\sigma \in \Sigma^{(p)}$, $i \in [p]$, and $f \in F \setminus F^{(1)}$ at most one call to f on u_i occurs in the set of all right-hand sides of σ-rules in R[7]
- *recursion-linear* if for every $f \in F$, $\sigma \in \Sigma^{(p)}$, and $i \in [p]$: $|rhs_{M,f,\sigma}|_{u_i} \leq 1$
- *context-linear*[8] if for every $f \in F^{(r+1)}$, $\sigma \in \Sigma$, and $k \in [r]$: $|rhs_{M,f,\sigma}|_{y_k} \leq 1$
- *linear* if it is recursion- and context-linear
- *atmost* if it is recursion-linear and it is context-linear or basic
- *recursion-nondeleting* if for every $f \in F$, $\sigma \in \Sigma^{(p)}$, and $i \in [p]$: $|rhs_{M,f,\sigma}|_{u_i} \geq 1$
- *context-nondeleting* if for every $f \in F^{(r+1)}$, $\sigma \in \Sigma$, and $k \in [r]$: $|rhs_{M,f,\sigma}|_{y_k} \geq 1$
- *nondeleting* if it is recursion-nondeleting and context-nondeleting
- *atleast* if it is recursion-nondeleting and it is context-nondeleting or basic
- *producing* if for every $f \in F$ and $\sigma \in \Sigma$ it exists $\delta \in \Delta$ with $|rhs_{M,f,\sigma}|_\delta \geq 1$. $\square$

Note that every tdtt is basic, weakly single-use, and context-linear and -nondeleting. Further note that atmostness generalizes linearity, while atleastness generalizes nondeletingness. For tdtts the restrictions recursion-linear, linear, and atmost coincide; as do recursion-nondeleting, nondeleting, and atleast.

Example 2.6 (syntactic restrictions checked for the mtts from Example 2.2)
The mtt M_{spine} is basic, recursion- and context-linear, and recursion- and context-nondeleting. It is thus also linear, atmost, nondeleting, and atleast, but it is is not a tdtt, not weakly single-use (due to two calls each to f_1 and f_2 on u_1 in α- and β-rules, respectively), and not producing (due to the rule for f_1 at ϵ). The mtt M_{pfx} fulfills all the properties from Definition 2.5, except for being a tdtt or basic. $\square$

The following two simple lemmata will be used extensively throughout the paper.

Lemma 2.7 (auxiliary)
Let $M = (F, \Sigma, \Delta, e, R)$ be an mtt. For every $t \in T_\Sigma$, $f \in F^{(r'+1)}$, $r \in \mathbb{N}$, $k \in [r]$, and $\theta_1, \ldots, \theta_{r'} \in T_{F \cup \Sigma \cup \Delta}(Y_r)$:

$$|nf(\Rightarrow_R, f\ t\ \theta_1 \cdots \theta_{r'})|_{y_k} = \sum_{k' \in [r']} |nf(\Rightarrow_R, f\ t\ y_1 \cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}. \quad \square$$

[7] In [Küh98] and [VK04a] this condition was required for all states, not just for nonunary ones, and an analogous condition was imposed also on the initial expression (the "right-hand side of the rule for the initial synthesized function at the root symbol" in the terminology of [Küh98]). It was, however, observed in [VK04a] that "weakly single-use" can be generalized as done here, without requiring any change to the presented construction or to its correctness proof.

[8] This restriction was called *non-copying* in [VK04a].

The proof (by induction on t, using a nested induction over the structure of right-hand sides of rules of M) can be found in the appendix.

Lemma 2.8 (properties of context-linear and -nondeleting mtts, resp.) Let $M = (F, \Sigma, \Delta, e, R)$ be an mtt. For every $t \in T_\Sigma$, $f \in F^{(r+1)}$, and $k \in [r]$:

1. if M is context-linear, then: $|nf(\Rightarrow_R, f\ t\ y_1 \cdots y_r)|_{y_k} \leq 1$
2. if M is context-nondeleting, then: $|nf(\Rightarrow_R, f\ t\ y_1 \cdots y_r)|_{y_k} \geq 1$. □

The proof (by induction on t, using a nested induction over the structure of right-hand sides of rules of M, as well as Lemma 2.7) can be found in the appendix. Actually, we only need to explicitly prove point 1 of the lemma; the proof of point 2 is completely dual in the sense in which the definition of "context-nondeleting" is dual to that of "context-linear" by requiring $|rhs_{M,f,\sigma}|_{y_k} \geq 1$ rather than $|rhs_{M,f,\sigma}|_{y_k} \leq 1$ for every $f \in F^{(r+1)}$, $\sigma \in \Sigma$, and $k \in [r]$.

3 Tree transducer composition

In [Voi01, VK04a] a direct composition construction is presented that given two mtts M_1 and M_2, where the output ranked alphabet of M_1 is the input ranked alphabet of M_2 and where one of M_1 and M_2 is a tdtt or M_1 is context-linear and M_2 is weakly single-use, produces an mtt $\overline{M_1;M_2}$ such that $\tau_{\overline{M_1;M_2}} = \tau_{M_1} ; \tau_{M_2}$. Before recalling the formal construction (with notations adapted to the setting of the present paper), we briefly consider its ingredients on an intuitive level. For a more detailed step-by-step development of the underlying ideas, consult Section 4 of [VK04a]. The brave-hearted may also want to study the formal correctness proof supplied in [VK04b].

The key to avoiding the production and consumption of an intermediate result is that whenever a state g of M_2 would recurse on the output generated by a state f of M_1, the final output is instead computed more directly through a call to a new state $\overline{fg}$[9] of $\overline{M_1;M_2}$ on f's recursion argument. This means that every nesting of the form

$$g\ (f\ t\ \phi_1 \cdots \phi_r)\ \eta_1 \cdots \eta_s \tag{1}$$

is replaced by an appropriate call to $\overline{fg}$ on t. Given that the (intermediate) output generated by f is essentially built up by patching together right-hand sides of M_1's rules at the symbols occurring in t, the rules for such a new state $\overline{fg}$ are obtained by "translating" the right-hand sides of rules for f using M_2, starting with g. More precisely, for every potential root symbol σ of t, the right-hand side $rhs_{\overline{M_1;M_2},\overline{fg},\sigma}$ is determined from $g\ rhs_{M_1,f,\sigma}\ z_1 \cdots z_s$—where $z_1 \cdots z_s$ are the context variables of $\overline{fg}$ that correspond to the context parameters of the outer g-call above—by exhaustively rewriting with the rules of M_2. Thus, the symbols in $rhs_{M_1,f,\sigma}$ that would otherwise build part of the intermediate data structure passed at run time from the inner f-call to the outer g-call in (1) are consumed at transformation time. Since $rhs_{M_1,f,\sigma}$ may contain recursive calls, one may during the rewriting also encounter situations corresponding to a nesting like (1) above (possibly for some $f' \neq f$ and $g' \neq g$). To ensure that all parts of $rhs_{M_1,f,\sigma}$—including those in context parameter positions of recursive calls—are effectively reached by the transformation, any call to a new paired state replacing such a nesting takes as arguments beside

[9] Here $\overline{fg}$ is used as a compacter notation for the pair (f, g).

the unchanged context parameters of the outer state from M_2 also "translated" versions of the context parameters of the inner state from M_1 with all states of M_2. Under this strategy, the replacement for a call like (1) takes the following form (assuming that M_2 has exactly $\mu \in \mathbb{N}_+$ states $g_1, \ldots, g_\mu$):

$$\overline{fg}\ t\ (g_1\ \phi_1\ \boxed{\cdots})\ \cdots\ (g_\mu\ \phi_1\ \boxed{\cdots})\ \cdots\ (g_1\ \phi_r\ \boxed{\cdots})\ \cdots\ (g_\mu\ \phi_r\ \boxed{\cdots})\ \eta_1 \cdots \eta_s \tag{2}$$

Since $\overline{fg}$ is thus provided with precomputed translations of f's context parameters, any call to some state of M_2 on some context variable of f surfacing during the processing of $g\ rhs_{M_1,f,\sigma}\ z_1 \cdots z_s$ can be resolved by simply selecting the corresponding parameter of $\overline{fg}$, modeled by a rewrite system *Pre* in the formal construction below.

The crux in this strategy of sending the states of M_2 into the context parameters of calls to M_1's states, however, is to determine what should go into the boxes in the replacement (2) for (1) above. Here the preconditions on M_1 and M_2 enter the picture. If one of the two mtts is actually a tdtt, then no information is missing because either $r = 0$ (if M_1 is a tdtt) or none of the $g_1, \ldots, g_\mu$ takes any context parameters (if M_2 is a tdtt). Otherwise, context-linearity of M_1 and weakly single-useness of M_2 ensure that the context parameter values with which the states $g_1, \ldots, g_\mu$ may reach the context parameters of the inner f-call during reduction of (1) are unambiguous. Moreover, these values can themselves be computed—without leaving the recursion scheme of mtts—by new states $\overline{k_f l_{g'}}$[10] for every $k \in [r]$, $g' \in \{g_1, \ldots, g_\mu\}$, and context parameter position l of g', as indicated in Figure 2. Hence, these auxiliary states can be used to provide the information missing in (2). In the formal construction the precise shape of the thus completed replacement for (1) is specified in the rewrite system *Pair*, built up using the function $nest_f$ to arrange a repeated nesting of $g_1, \ldots, g_\mu$-translations of f's context parameters interspersed with appropriate calls to $\overline{k_f l_{g'}}$-states. The reasons why this nesting is necessary but can be kept of finite depth are discussed at length in [VK04a]. There it is also motivated which context parameters the auxiliary states require and how their rules can be constructed following an intricate scheme of "walking upwards" from occurrences of context variables in M_1's right-hand sides, using *par*-functions that assemble information about context parameters of recursive calls in M_2's rules at symbols encountered on the way to the root.

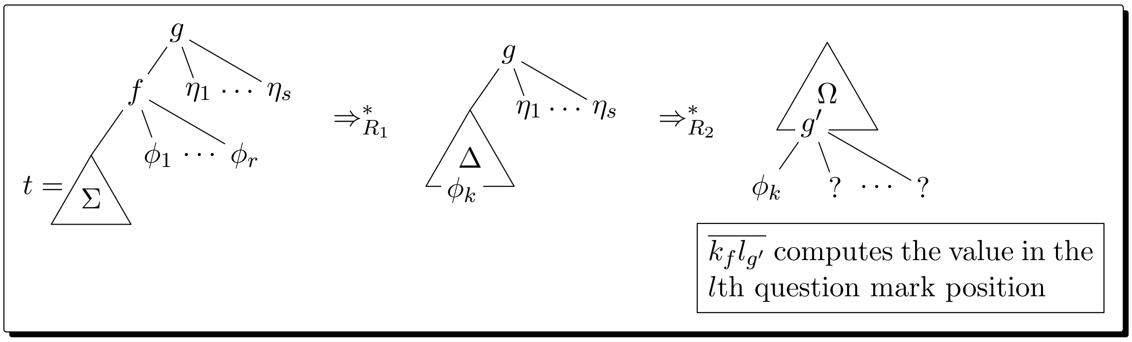

Figure 2: Role of the auxiliary states.

The following construction differs from Construction 5.1 in [VK04a] only in that the required dummy nullary symbol *nil* is explicitly added to the output ranked alphabet

[10] Here $\overline{k_f l_{g'}}$ is used as a compacter notation for the pair of pairs $((k, f), (l, g'))$.

rather than randomly picking an existing one. It may appear in the right-hand sides of rules of $\overline{M_1;M_2}$, but never in a tree returned by $\tau_{\overline{M_1;M_2}}$.

Construction 3.1 (direct composition of restricted mtts)
Let $M_1 = (F, \Sigma, \Delta, e_{M_1}, R_1)$ and $M_2 = (G, \Delta, \Omega, e_{M_2}, R_2)$ be mtts such that one of the two is a tdtt or M_1 is context-linear and M_2 is weakly single-use. Without loss of generality, assume that F and G are disjoint and that M_1 uses recursion variables from U and context variables from Y, whereas M_2 uses V and Z, respectively. Let $\mu = |G|$ and fix some ordering of the states in G such that $G = \{g_1, \ldots, g_\mu\}$. For every $n \in [\mu]$ let $s_n \in \mathbb{N}$ be such that $g_n \in G^{(s_n+1)}$. Additionally, set $r_{max} = max(rank(F)) - 1$, $Z_G = \{z_{g_1,1}, \ldots, z_{g_\mu,s_\mu}\}$[11], and let $nil \notin \Omega$ be some arbitrary symbol. The mtt $\overline{M_1;M_2} = (H, \Sigma, \Omega \cup \{nil^{(0)}\}, e_{\overline{M_1;M_2}}, \overline{R_1;R_2})$ is obtained as follows:

$$\begin{aligned} H = \;& \{\overline{fg}^{(r*\mu+s+1)} & \mid f \in F^{(r+1)}, g \in G^{(s+1)}\} \\ \cup\;& \{\overline{k_f l_g}^{(r*\mu+|Z_G|+1)} & \mid f \in F^{(r+1)}, g \in G^{(s+1)}, k \in [r], l \in [s]\}\,, \end{aligned}$$

$e_{\overline{M_1;M_2}} = nf(\Rightarrow_{R_2 \cup Pair}, e_{M_2}[v_1 \leftarrow e_{M_1}])$, and

$\overline{R_1;R_2}$ contains

– for every $f \in F^{(r+1)}$, $g \in G^{(s+1)}$, and $\sigma \in \Sigma^{(p)}$ the rule[12]

$$\overline{fg}\ (\sigma\ u_1 \cdots u_p)\ y_{1,g_1} \cdots y_{r,g_\mu}\ z_1 \cdots z_s \to nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, g\ rhs_{M_1,f,\sigma}\ z_1 \cdots z_s)\,, \text{ and}$$

– for every $f \in F^{(r+1)}$, $g \in G^{(s+1)}$, $k \in [r]$, $l \in [s]$, and $\sigma \in \Sigma^{(p)}$ the rule

$$\overline{k_f l_g}\ (\sigma\ u_1 \cdots u_p)\ y_{1,g_1} \cdots y_{r,g_\mu}\ z_{g_1,1} \cdots z_{g_\mu,s_\mu} \to nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, \varrho)\,,$$

where $\varrho = nil$ if $rhs_{M_1,f,\sigma}$ does not contain the context variable y_k;
otherwise, $\varrho = par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g, l)$, where $\pi \in paths(rhs_{M_1,f,\sigma})$ is the path of the unique occurrence[13] of y_k in $rhs_{M_1,f,\sigma}$.

The rewrite systems Pre (over $G \cup Y_{r_{max}} \cup \{y_{k,g}^{(0)} \mid k \in [r_{max}], g \in G\}$ and Z) and $Pair$ (over $F \cup G \cup H \cup \{nil\}$ and $\{u\} \cup Y' \cup Z$) used above are defined as follows[14]:

$$\begin{array}{lll} Pre\ :\ g\ y_k\ z_1 \cdots z_s & \to y_{k,g} & \forall g \in G^{(s+1)}, k \in [r_{max}] \\ \hline Pair\ :\ g\ (f\ u\ y'_1 \cdots y'_r)\ z_1 \cdots z_s \to & & \forall g \in G^{(s+1)}, f \in F^{(r+1)} \\ \quad \overline{fg}\ u\ nest_f(1, g_1, \emptyset)[z_{g,l} \leftarrow z_l \mid l \in [s], & & \\ \qquad\qquad z_{g',l} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l \in [s']] & & \\ \qquad \cdots & & \\ \qquad nest_f(r, g_\mu, \emptyset)[z_{g,l} \leftarrow z_l \mid l \in [s], & & \\ \qquad\qquad z_{g',l} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l \in [s']] & & \\ \qquad z_1 \cdots z_s & & \end{array}$$

[11] The enumeration $z_{g_1,1}, \ldots, z_{g_\mu,s_\mu}$ resolves to $z_{g_1,1}, \ldots, z_{g_1,s_1}, \ldots, z_{g_\mu,1}, \ldots, z_{g_\mu,s_\mu}$, and correspondingly for other enumerations doubly indexed from $(g_1, 1)$ to (g_μ, s_μ) below.

[12] The enumeration $y_{1,g_1} \cdots y_{r,g_\mu}$ resolves to $y_{1,g_1} \cdots y_{1,g_\mu} \cdots y_{r,g_1} \cdots y_{r,g_\mu}$, and correspondingly for other enumerations doubly indexed from $(1, g_1)$ to (r, g_μ), or from $(1, g_1)$ to (r', g_μ), below.

[13] Note that if the ranges of the above quantifications of f, g, k, and l are nonempty, then neither M_1 nor M_2 can be a tdtt and hence M_1 is context-linear by the construction's preconditions.

[14] The combined reduction relations $\Rightarrow_{R_2 \cup Pair}$ and $\Rightarrow_{R_2 \cup Pre \cup Pair}$ are confluent and terminating, hence the above use of normal forms with respect to them is justified.

For every $f \in F^{(r+1)}$ the function

$$nest_f : [r] \times G \times \mathcal{P}([r] \times G) \longrightarrow T_{G \cup H \cup \{nil\}}(\{u\} \cup Y'_r \cup Z_G)$$

is defined by setting for every $k \in [r]$, $g' \in G^{(s'+1)}$, and $\mathcal{C} \subseteq [r] \times G$:

$$nest_f(k, g', \mathcal{C}) = \begin{cases} g'\ y'_k\ (\overline{k_f 1_{g'}}\ u\ nest_f(1, g_1, \mathcal{C} \cup \{(k,g')\})\ \cdots\ nest_f(r, g_\mu, \mathcal{C} \cup \{(k,g')\}) \\ \qquad z_{g_1,1} \cdots z_{g_\mu,s_\mu}) \\ \qquad \cdots \\ \qquad (\overline{k_f s'_{g'}}\ u\ nest_f(1, g_1, \mathcal{C} \cup \{(k,g')\})\ \cdots\ nest_f(r, g_\mu, \mathcal{C} \cup \{(k,g')\}) \\ \qquad z_{g_1,1} \cdots z_{g_\mu,s_\mu}) & \text{if } (k,g') \notin \mathcal{C} \\ nil & \text{otherwise.} \end{cases}$$

Using these functions, we additionally define for every $p \in \mathbb{N}$, $r \in \mathbb{N}_+$ with $(r+1) \in rank(F)$, and $\phi \in RHS(F, \Delta, U_p, Y_r)$ the function

$$\begin{aligned} par_{M_2,\phi} : \{(\pi, g, l) \mid \pi \in paths(\phi), lab(\phi,\pi) \notin U_p, g \in G^{(s+1)}, l \in [s]\} \\ \longrightarrow T_{F \cup G \cup H \cup \Delta \cup \Omega \cup \{nil\}}(U_p \cup Y_r \cup Z_G) \end{aligned}$$

by induction over the prefix-order of paths in ϕ as follows. For every $g \in G^{(s+1)}$ and $l \in [s]$:

- $par_{M_2,\phi}(\varepsilon, g, l) = z_{g,l}$
- For every $j \in \mathbb{N}_+$ and $\pi j \in paths(\phi)$ with $lab(\phi, \pi j) \notin U_p$ we perform a case distinction.

  $\underline{lab(\phi,\pi) = \delta}$ for some $\delta \in \Delta^{(q)}$ and $j \in [q]$:
  If any of the δ-rules of M_2 contains a call to g on v_j as follows:

  $$g'\ (\delta\ v_1 \cdots v_q)\ z_1 \cdots z_{s'} \to \cdots (g\ v_j\ \psi_1 \cdots \psi_s) \cdots ,$$

  where $g' \in G^{(s'+1)}$ and $\psi_1, \dots, \psi_s \in RHS(G, \Omega, V_q, Z_{s'})$, then[15]:

  $$\begin{aligned} par_{M_2,\phi}(\pi j, g, l) = \psi_l[&v_1, \dots, v_q \leftarrow sub(\phi, \pi 1), \dots, sub(\phi, \pi q), \\ &z_1, \dots, z_{s'} \leftarrow par_{M_2,\phi}(\pi, g', 1), \dots, par_{M_2,\phi}(\pi, g', s')]. \end{aligned}$$

  If no such call exists in the δ-rules of M_2, then $par_{M_2,\phi}(\pi j, g, l) = nil$.

  $\underline{lab(\phi,\pi) = f}$ for some $f \in F^{(r'+1)}$, $1 \le j-1 \le r'$, and $lab(\phi, \pi 1) = u_i \in U_p$:

  $$\begin{aligned} par_{M_2,\phi}(\pi j, g, l) = (\overline{(j-1)_f l_g}\ u\ &nest_f(1, g_1, \{(j-1, g)\}) \\ &\cdots \\ &nest_f(r', g_\mu, \{(j-1, g)\}) \\ &z_{g_1,1} \cdots z_{g_\mu,s_\mu}) \\ [u &\leftarrow u_i, \\ y'_1, \dots, y'_{r'} &\leftarrow sub(\phi, \pi 2), \dots, sub(\phi, \pi(r'+1)), \\ z_{g_1,1}, \dots, z_{g_\mu,s_\mu} &\leftarrow par_{M_2,\phi}(\pi, g_1, 1), \dots, par_{M_2,\phi}(\pi, g_\mu, s_\mu)]. \quad \square \end{aligned}$$

[15] Note that if the ranges of the above quantifications of r, g, and l are nonempty, then neither M_1 nor M_2 can be a tdtt and hence M_2 is weakly single-use by the construction's preconditions. Moreover, $g \in G^{(s+1)}$ and $l \in [s]$ imply that g is a nonunary state, i.e., the weakly single-use condition applies to g-calls, so that g' and the $\psi_1, \dots, \psi_s$ are uniquely determined.

The following theorem was proved in [Voi01]; an even more detailed account can be found in [VK04b].

Theorem 3.2 (correctness of Construction 3.1; Theorem 5.2 of [VK04a])
$\tau_{M_1}\,;\tau_{M_2} = \tau_{\overline{M_1;M_2}}$ □

To provide some additional intuition about the construction, we observe the exact roles that the states of the resulting mtt play by describing the output they compute in terms of computations using the combined reduction relation $\Rightarrow_{R_1 \cup R_2}$ induced by the original mtts[16].

Observation 3.3 ("extensional semantics" for the states of $\overline{M_1;M_2}$)
For every $t \in T_\Sigma$, $f \in F^{(r+1)}$, and $g \in G^{(s+1)}$ we have:

$$nf(\Rightarrow_{\overline{R_1;R_2}}, \overline{fg}\; t\; y_{1,g_1} \cdots y_{r,g_\mu}\; z_1 \cdots z_s) = nf(\Rightarrow_{Pre}, nf(\Rightarrow_{R_1 \cup R_2}, g\; (f\;\; t\; y_1 \cdots y_r)\; z_1 \cdots z_s)).$$

Moreover, for every $k \in [r]$ and $g' \in G^{(s'+1)}$, if $nf(\Rightarrow_{R_1 \cup R_2}, g\; (f\;\; t\; y_1 \cdots y_r)\; z_1 \cdots z_s)$ contains a subtree of the form $(g'\;\, y_k\;\, \varrho_1 \cdots \varrho_{s'})$, then for every $l \in [s']$:

$$nf(\Rightarrow_{\overline{R_1;R_2}}, \overline{k_f l_{g'}}\; t\; y_{1,g_1} \cdots y_{r,g_\mu}\;\; z_{g_1,1} \cdots z_{g_\mu,s_\mu}) = nf(\Rightarrow_{Pre}, \varrho_l)[z_1, \ldots, z_s \leftarrow z_{g,1}, \ldots, z_{g,s}].$$

Proof
By combining statements **I**a and **I**b of Lemma A.15, Definition A.6(1), and Theorem A.4 in [VK04b]. □

For a detailed protocol of the presented construction in action for two concrete mtts, consult Section 5.2 of [VK04a]. Here we give only the result of applying it to our running example.

Example 3.4 (composition result for the mtts from Example 2.2)
Composition of the mtts M_{spine} and M_{pfx} yields the mtt $\overline{M_{spine};M_{pfx}} = (\{\overline{f_1g}^{(3)}, \overline{f_2g}^{(4)}, \overline{1_{f_1}1_g}^{(3)}, \overline{1_{f_2}1_g}^{(4)}, \overline{2_{f_2}1_g}^{(4)}\}, \Sigma_{mon}, \Omega_{mon} \cup \{nil^{(0)}\}, e_{\overline{M_{spine};M_{pfx}}}, \overline{R_{spine};R_{pfx}})$ with rules

$$
\begin{array}{lllll}
\overline{R_{spine};R_{pfx}}: & \overline{f_1g} & (\alpha\; u_1) & y_{1,g}\; z_1 & \to \alpha\; (\overline{f_1g}\; u_1\; y_{1,g}\; z_1) \\
 & \overline{f_1g} & (\beta\; u_1) & y_{1,g}\; z_1 & \to \overline{f_2g}\; u_1\; y_{1,g}\; (\beta\; (\epsilon\; (\overline{2_{f_2}1_g}\; u_1\; y_{1,g}\; nil\; z_1)))\; z_1 \\
 & \overline{f_1g} & \epsilon & y_{1,g}\; z_1 & \to y_{1,g} \\
 & \overline{f_2g} & (\alpha\; u_1) & y_{1,g}\; y_{2,g}\; z_1 & \to \alpha\; (\overline{f_1g}\; u_1\; (\delta\; y_{1,g})\; z_1) \\
 & \overline{f_2g} & (\beta\; u_1) & y_{1,g}\; y_{2,g}\; z_1 & \to \overline{f_2g}\; u_1\; y_{1,g}\; (\beta\; (\beta\; y_{2,g}))\; z_1 \\
 & \overline{f_2g} & \epsilon & y_{1,g}\; y_{2,g}\; z_1 & \to \delta\; y_{1,g} \\
 & \overline{1_{f_1}1_g} & (\alpha\; u_1) & y_{1,g}\; z_{g,1} & \to \overline{1_{f_1}1_g}\; u_1\; nil\; z_{g,1} \\
 & \overline{1_{f_1}1_g} & (\beta\; u_1) & y_{1,g}\; z_{g,1} & \to \overline{1_{f_2}1_g}\; u_1\; nil\; (\beta\; (\epsilon\; (\overline{2_{f_2}1_g}\; u_1\; nil\; nil\; z_{g,1})))\; z_{g,1} \\
 & \overline{1_{f_1}1_g} & \epsilon & y_{1,g}\; z_{g,1} & \to z_{g,1} \\
 & \overline{1_{f_2}1_g} & (\alpha\; u_1) & y_{1,g}\; y_{2,g}\; z_{g,1} & \to y_{2,g} \\
 & \overline{1_{f_2}1_g} & (\beta\; u_1) & y_{1,g}\; y_{2,g}\; z_{g,1} & \to \overline{1_{f_2}1_g}\; u_1\; nil\; (\beta\; (\beta\; y_{2,g}))\; z_{g,1} \\
 & \overline{1_{f_2}1_g} & \epsilon & y_{1,g}\; y_{2,g}\; z_{g,1} & \to y_{2,g} \\
 & \overline{2_{f_2}1_g} & (\alpha\; u_1) & y_{1,g}\; y_{2,g}\; z_{g,1} & \to \overline{1_{f_1}1_g}\; u_1\; nil\; z_{g,1} \\
 & \overline{2_{f_2}1_g} & (\beta\; u_1) & y_{1,g}\; y_{2,g}\; z_{g,1} & \to \overline{2_{f_2}1_g}\; u_1\; y_{1,g}\; nil\; z_{g,1} \\
 & \overline{2_{f_2}1_g} & \epsilon & y_{1,g}\; y_{2,g}\; z_{g,1} & \to z_{g,1}
\end{array}
$$

[16] Since F and G are assumed to be disjoint, $\Rightarrow_{R_1 \cup R_2}$ is confluent. It is also terminating.

and initial expression $e_{\overline{M_{spine};M_{pfx}}} = \overline{f_1 g}\ u_1\ (\epsilon\ (\overline{1_{f_1}1_g}\ u_1\ nil\ \gamma))\ \gamma$. $\square$

The mtt obtained in the previous example is unnecessarily complicated because the composition construction has introduced some context parameters that are superfluous in the sense that they will never (for no possible input tree) influence the output generated by a state. This is the case for parameter z_1 of $\overline{f_1 g}$, parameters $y_{2,g}$ and z_1 of $\overline{f_2 g}$, parameter $y_{1,g}$ of $\overline{1_{f_1}1_g}$, parameters $y_{1,g}$ and $z_{g,1}$ of $\overline{1_{f_2}1_g}$, and parameters $y_{1,g}$ and $y_{2,g}$ of $\overline{2_{f_2}1_g}$. While unavoidable for the sake of generality of the composition construction, such superfluous parameters pose no serious problems in practice. In Section 4.1 of [Voi01] we have developed a post-processing—based on a straight-forward fixpoint construction—that detects and removes all superfluous context parameters of an mtt. For $\overline{M_{spine};M_{pfx}}$ it yields an mtt with states of reduced ranks, initial expression $\overline{f_1 g}\ u_1\ (\epsilon\ (\overline{1_{f_1}1_g}\ u_1\ \gamma))$, and rules

$$
\begin{array}{lllll}
\overline{R_{spine};R_{pfx}}' : & \overline{f_1 g} & (\alpha\ u_1) & y_{1,g} & \to \alpha\ (\overline{f_1 g}\ u_1\ y_{1,g}) \\
& \overline{f_1 g} & (\beta\ u_1) & y_{1,g} & \to \overline{f_2 g}\ u_1\ y_{1,g} \\
& \overline{f_1 g} & \epsilon & y_{1,g} & \to y_{1,g} \\
& \overline{f_2 g} & (\alpha\ u_1) & y_{1,g} & \to \alpha\ (\overline{f_1 g}\ u_1\ (\delta\ y_{1,g})) \\
& \overline{f_2 g} & (\beta\ u_1) & y_{1,g} & \to \overline{f_2 g}\ u_1\ y_{1,g} \\
& \overline{f_2 g} & \epsilon & y_{1,g} & \to \delta\ y_{1,g} \\
& \overline{1_{f_1}1_g} & (\alpha\ u_1) & z_{g,1} & \to \overline{1_{f_1}1_g}\ u_1\ z_{g,1} \\
& \overline{1_{f_1}1_g} & (\beta\ u_1) & z_{g,1} & \to \overline{1_{f_2}1_g}\ u_1\ (\beta\ (\epsilon\ (\overline{2_{f_2}1_g}\ u_1\ z_{g,1}))) \\
& \overline{1_{f_1}1_g} & \epsilon & z_{g,1} & \to z_{g,1} \\
& \overline{1_{f_2}1_g} & (\alpha\ u_1) & y_{2,g} & \to y_{2,g} \\
& \overline{1_{f_2}1_g} & (\beta\ u_1) & y_{2,g} & \to \overline{1_{f_2}1_g}\ u_1\ (\beta\ (\beta\ y_{2,g})) \\
& \overline{1_{f_2}1_g} & \epsilon & y_{2,g} & \to y_{2,g} \\
& \overline{2_{f_2}1_g} & (\alpha\ u_1) & z_{g,1} & \to \overline{1_{f_1}1_g}\ u_1\ z_{g,1} \\
& \overline{2_{f_2}1_g} & (\beta\ u_1) & z_{g,1} & \to \overline{2_{f_2}1_g}\ u_1\ z_{g,1} \\
& \overline{2_{f_2}1_g} & \epsilon & z_{g,1} & \to z_{g,1}
\end{array}
$$

Figure 3 shows a computation of this simplified mtt for a particular input. Since the elimination of superfluous context parameters has no effect with respect to the abstract efficiency measure we are going to use in the next section, our analysis will concern the original output program of the composition construction rather than its post-processed form.[17] Clearly, this does not hinder the application of positive analysis results because the post-processed program could only be even more efficient in practice.

4 Formal efficiency analysis

We want to formally relate the efficiency of a composed program obtained by Construction 3.1 to the efficiency of the original program. That is, we want to compare the cost of a computation starting from an instantiation of $e_{\overline{M_1;M_2}}$ and using rules from $\overline{R_1;R_2}$ to the cost of the computation starting from the corresponding instantiation of $e_{M_2}[v_1 \leftarrow e_{M_1}]$ and using rules from $R_1 \cup R_2$.

From the construction it is obvious that intermediate data structures (over the ranked alphabet Δ) produced by M_1 in the original program do not appear in the composed

[17] We also do not consider the *elimination of copy-states* from Section 4.2 of [Voi01] here.

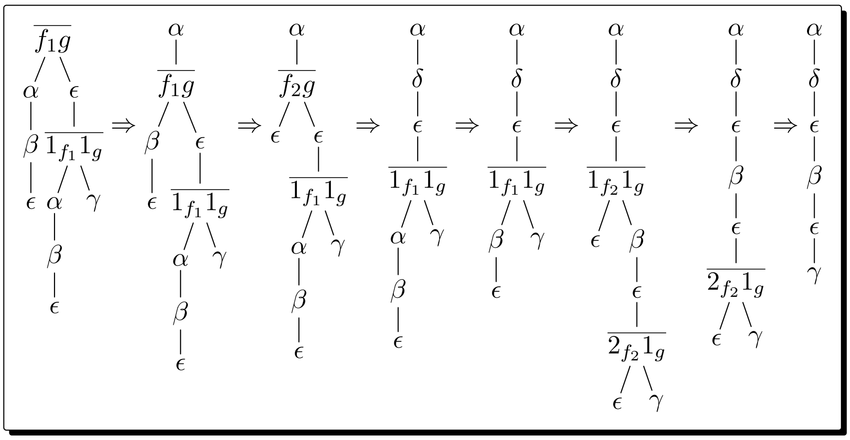

Figure 3: Reduction sequence using $\Rightarrow_{\overline{R_{spine};R_{pfx}}'}$.

program. Thus, no memory cells for this intermediate result have to be allocated in the heap and later be deallocated by the garbage collector. To make sure that this beneficial effect on **space** usage is not a Pyrrhic victory, the composed program should not take more **time** to compute its output than the original program. A reasonable notion for the "cost" of a computation to consider is therefore the number of reduction steps required to reach the final output for a given input. While the reduction relation used to provide the semantics of an mtt in Section 2.2 is in general nondeterministic (though confluent), implementations of functional languages are typically based on deterministic reduction strategies. Since we are interested in applying tree transducer composition to nonstrict languages, we study *call-by-name* (also known as *normal order* or *leftmost-outermost*) reduction. Later, in Section 4.9, we will provide simple conditions on M_1 and M_2 under which the gained insights can also be applied to the setting of *call-by-need* reduction.

Even though the pure number of reduction steps is a somewhat coarse efficiency measure, abstracting from the fact that not every step necessarily requires the same amount of time in a concrete implementation, it is usually sufficient in practice. For example, classical deforestation is successfully applied in a production compiler [vvS02], despite the fact that the only known formal efficiency statement regarding this transformation is that it does not increase the number of call-by-need reductions steps for linear programs (the number of call-by-name steps, respectively, for arbitrary programs).

4.1 Call-by-name reduction

In this subsection we define call-by-name reduction relations for the composed and for the original program, respectively. We show that they are adequate restrictions of the corresponding nondeterministic reduction relations and use them to define cost functions as base for our analysis.

For the remainder of the paper, let $M_1 = (F, \Sigma, \Delta, e_{M_1}, R_1)$ and $M_2 = (G, \Delta, \Omega, e_{M_2}, R_2)$ be two fixed mtts to which Construction 3.1 is applicable, yielding the mtt $\overline{M_1;M_2} = (H, \Sigma, \Omega \cup \{nil\}, e_{\overline{M_1;M_2}}, \overline{R_1;R_2})$. We use the notions and variable naming conventions from Construction 3.1.

In Definition 2.4 we described the set of terms that can occur for a given mtt during a computation starting from an instantiation of its initial expression. The deterministic reduction relation for the composed mtt $\overline{M_1;M_2}$ needs only to be defined for such well-formed terms.

Definition 4.1 (call-by-name restriction of $\Rightarrow_{\overline{R_1;R_2}}$)
The *call-by-name reduction relation induced by* $\overline{R_1;R_2}$, denoted by $\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}$, is the smallest relation $\Rightarrow \subseteq WFT^2_{\overline{M_1;M_2}}$ satisfying the following conditions:

- for every $h \in H^{(n+1)}$, $\sigma \in \Sigma^{(p)}$, $t_1, \ldots, t_p \in T_\Sigma$, and $\varrho_1, \ldots, \varrho_n \in WFT_{\overline{M_1;M_2}}$:

  $$h\ (\sigma\ t_1 \cdots t_p)\ \varrho_1 \cdots \varrho_n \Rightarrow rhs_{\overline{M_1;M_2},h,\sigma}[u_1, \ldots, u_p, x_1, \ldots, x_n \leftarrow t_1, \ldots, t_p, \varrho_1, \ldots, \varrho_n]\,,$$

  where $x_1, \ldots, x_n$ are the context variables (in correct order) from the left-hand side of the rule for h at σ in $\overline{R_1;R_2}$, and

- for every $\omega \in \Omega^{(n)}$, $b \in [n]$, $\varrho_1, \ldots, \varrho_{b-1} \in T_{\Omega\cup\{nil\}}$, $\varrho_b \Rightarrow \varrho'_b$, and $\varrho_{b+1}, \ldots, \varrho_n \in WFT_{\overline{M_1;M_2}}$:

  $$\omega\ \varrho_1 \cdots \varrho_n \Rightarrow \omega\ \varrho_1 \cdots \varrho_{b-1}\ \varrho'_b\ \varrho_{b+1} \cdots \varrho_n. \quad \square$$

The defined reduction relation is unmistakably call-by-name because no reduction takes place under function calls. That it is in fact deterministic and also sound and complete in the sense of computing the same normal forms as $\Rightarrow_{\overline{R_1;R_2}}$ is shown below in Lemma 4.3. An example call-by-name computation for the mtt $\overline{M_{spine};M_{pfx}}$ would look essentially like in Figure 3, apart from some additional clutter due to the superfluous context parameters.

For defining the deterministic reduction relation for the original program consisting of the mtts M_1 and M_2, it is useful to first describe the kind of terms that can occur during a computation starting from $e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]]$ for some $t \in T_\Sigma$.

Definition 4.2 ($WFT_{M_1;M_2}$, call-by-name restriction of $\Rightarrow_{R_1\cup R_2}$)
The set $WFT_{M_1;M_2}$ of *well-formed terms for the sequential composition of* M_1 *and* M_2 is the smallest set $WFT \subseteq T_{G\cup\Omega\cup F\cup\Delta\cup\Sigma}$ satisfying the following conditions:

- for every $g \in G^{(s+1)}$, $\phi \in WFT_{M_1}$, $\varrho_1, \ldots, \varrho_s \in WFT$: $(g\ \phi\ \varrho_1 \cdots \varrho_s) \in WFT$, and
- for every $\omega \in \Omega^{(n)}$ and $\varrho_1, \ldots, \varrho_n \in WFT$: $(\omega\ \varrho_1 \cdots \varrho_n) \in WFT$.

The *call-by-name reduction relation induced by* R_1 *and* R_2, denoted by $\overset{cbn}{\Rightarrow}_{R_1\cup R_2}$, is the smallest relation $\Rightarrow \subseteq WFT_{M_1;M_2} \times WFT_{M_1;M_2}$ satisfying the following conditions:

- for every $g \in G^{(s+1)}$, $f \in F^{(r+1)}$, $\sigma \in \Sigma^{(p)}$, $t_1, \ldots, t_p \in T_\Sigma$, $\phi_1, \ldots, \phi_r \in WFT_{M_1}$, and $\varrho_1, \ldots, \varrho_s \in WFT_{M_1;M_2}$:

  $$\begin{aligned} &g\ (f\ (\sigma\ t_1 \cdots t_p)\ \phi_1 \cdots \phi_r)\ \varrho_1 \cdots \varrho_s \\ \Rightarrow\ &g\ rhs_{M_1,f,\sigma}[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \phi_1, \ldots, \phi_r]\ \varrho_1 \cdots \varrho_s\,, \end{aligned}$$

- for every $g \in G^{(s+1)}$, $\delta \in \Delta^{(q)}$, $\phi_1, \ldots, \phi_q \in WFT_{M_1}$, and $\varrho_1, \ldots, \varrho_s \in WFT_{M_1;M_2}$:

  $$g\ (\delta\ \phi_1 \cdots \phi_q)\ \varrho_1 \cdots \varrho_s \Rightarrow rhs_{M_2,g,\delta}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \phi_1, \ldots, \phi_q, \varrho_1, \ldots, \varrho_s]\,,\ \text{and}$$

- for every $\omega \in \Omega^{(n)}$, $b \in [n]$, $\varrho_1, \ldots, \varrho_{b-1} \in T_\Omega$, $\varrho_b \Rightarrow \varrho'_b$, and $\varrho_{b+1}, \ldots, \varrho_n \in WFT_{M_1;M_2}$:

  $$\omega\ \varrho_1 \cdots \varrho_n \Rightarrow \omega\ \varrho_1 \cdots \varrho_{b-1}\ \varrho'_b\ \varrho_{b+1} \cdots \varrho_n. \quad \square$$

The reduction relation defined is call-by-name because reduction under a function call takes only place when absolutely required to proceed with the outer call. Figure 4 shows an example for the mtts M_{spine} and M_{pfx} with the same input as in Figure 3. The shaded areas correspond to memory cells that have to be allocated for the intermediate data structure. Note that no such need arose during the reduction using the transformed program in Figure 3 (only cells for the final output needed to be allocated there, which also happens here), which moreover was shorter by one step. The challenge in our efficiency analysis will be to make general statements about how the lengths of such reduction sequences for the original and the composed program compare to each other, without knowing the actual input tree provided.

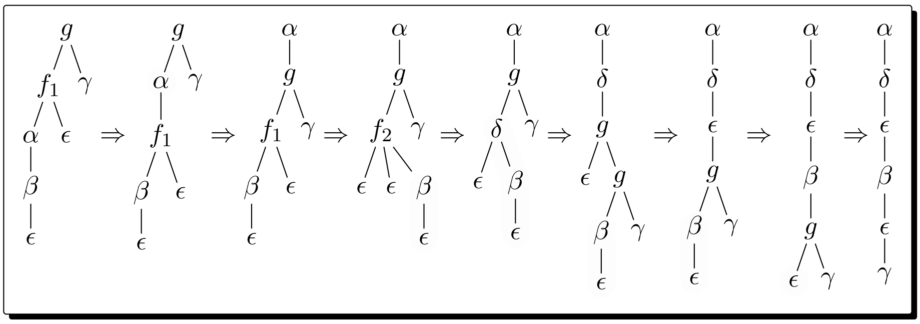

Figure 4: Reduction sequence using $\overset{cbn}{\Rightarrow}_{R_{spine} \cup R_{pfx}}$.

But first we should convince ourselves that $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ and $\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}$ are adequate restrictions of $\Rightarrow_{R_1 \cup R_2}$ and $\Rightarrow_{\overline{R_1;R_2}}$, respectively. Taking into account that $e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]] \in WFT_{M_1;M_2}$ and $e_{\overline{M_1;M_2}}[u_1 \leftarrow t] \in WFT_{\overline{M_1;M_2}}$ for every $t \in T_\Sigma$, this is established by the following lemma.

Lemma 4.3 (adequacy of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ and $\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}$ on well-formed terms)

1. $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ is a deterministic subrelation of $\Rightarrow_{R_1 \cup R_2}$.
2. For every $\varrho \in WFT_{M_1;M_2}$: $nf(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}, \varrho) = nf(\Rightarrow_{R_1 \cup R_2}, \varrho)$.
3. $\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}$ is a deterministic subrelation of $\Rightarrow_{\overline{R_1;R_2}}$.
4. For every $\varrho \in WFT_{\overline{M_1;M_2}}$: $nf(\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}, \varrho) = nf(\Rightarrow_{\overline{R_1;R_2}}, \varrho)$.

Proof

1. By straightforward induction over the recursive structure of the definition of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$, one can show that $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \subseteq \Rightarrow_{R_1 \cup R_2}$. This inclusion implies that elements of T_Ω are irreducible w.r.t. $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$. The latter implies that for every $\omega\ \varrho_1 \cdots \varrho_n$ there can be at most one $b \in [n]$ such that $\varrho_1, \ldots, \varrho_{b-1} \in T_\Omega$ and $\varrho_b \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \varrho_b'$, which ensures determinism of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$.
2. Point 1 implies that $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ is confluent and terminating, thus $nf(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}, \varrho)$ exists for every $\varrho \in WFT_{M_1;M_2}$. Induction over the structure of elements of $WFT_{M_1;M_2}$ shows that the only such elements which are irreducible w.r.t. $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ are in T_Ω and hence are also irreducible w.r.t. $\Rightarrow_{R_1 \cup R_2}$. The equality of the normal forms follows from this by $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \subseteq \Rightarrow_{R_1 \cup R_2}$.

The proof of points 3 and 4 is analogous. □

Since $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ is deterministic, for every $t \in T_\Sigma$ there is exactly one sequence

$$e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]] \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \cdots \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} nf(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}, e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]])\,,$$

and likewise for $\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}$ and $e_{\overline{M_1;M_2}}[u_1 \leftarrow t]$. The functions $cbn_{M_1;M_2} : T_\Sigma \longrightarrow \mathbb{N}$ and $cbn_{\overline{M_1;M_2}} : T_\Sigma \longrightarrow \mathbb{N}$ associating to every input tree the number of call-by-name steps required by the original and the composed program, respectively, to compute the corresponding output tree, are then given as follows:

$$cbn_{M_1;M_2}(t) = c \;\text{ if } e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]] \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^c nf(\Rightarrow_{R_1 \cup R_2}, e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]])$$
$$= \tau_{M_2}(\tau_{M_1}(t)) = \tau_{\overline{M_1;M_2}}(t) =$$
$$cbn_{\overline{M_1;M_2}}(t) = c' \;\text{ if } e_{\overline{M_1;M_2}}[u_1 \leftarrow t] \qquad \left(\overset{cbn}{\Rightarrow}_{\overline{R_1;R_2}}\right)^{c'} \qquad nf(\Rightarrow_{\overline{R_1;R_2}}, e_{\overline{M_1;M_2}}[u_1 \leftarrow t]).$$

We aim to find conditions under which for every $t \in T_\Sigma$ we have $cbn_{\overline{M_1;M_2}}(t) \leq cbn_{M_1;M_2}(t)$. Actually, it would also be satisfactory to find some small constant $d \in \mathbb{N}$ such that for every possible input tree the following holds:

$$cbn_{\overline{M_1;M_2}}(t) \leq cbn_{M_1;M_2}(t) + d.$$

This would mean that, whatever the input is, the composed program performs at most d additional reduction steps but avoids the allocation and eventual deallocation of an intermediate data structure of arbitrary size.

4.2 Outline of the analysis

In the following subsections we will work toward a theorem (Theorem 4.40) providing the desired conditions. As a guide through the jungle of various lemmata proved on the way, Figure 5 shows all proof dependencies for Theorem 4.40.[18] To allow a tracing of the flow of potential "imprecision" along these dependencies, lemmata providing inequational rather than equational statements are marked with $\leq$. In addition, shading is used to indicate a grouping of lemmata and theorems according to their nature or object of study. More specifically, the two lemmata in group (i) show that certain annotations of the original and the composed program, respectively, can be used to reflect the numbers of performed call-by-name reduction steps in output trees. The members of group (ii) study the composition construction itself and provide a way to remove its working details from our concern for the remainder of the analysis. Left with annotated versions of the original program only, the lemmata in group (iii) establish how annotations can be combined and manipulated in a safe manner, i.e., without overestimating costs incurred by the original program or underestimating costs incurred by the composed program. In particular, a certain rewrite system $\mathcal{E}_{M_1,M_2}$ will be used arbitrarily often on annotated versions of the first mtt's rule

[18]Note that there is also an interesting dependency on the correctness statement from Theorem 3.2.

right-hand sides and initial expression. That manipulation process can be stopped when we have reached an annotation which is simple enough to establish an input-independent upper bound for the cost count; corresponding conditions are contained in Lemma 4.38. Throughout the analysis, essential use is made of certain statements about context-linear, context-nondeleting, atmost, and atleast mtts, respectively, and about $\mathcal{E}_{M_1,M_2}$. These statements are established by the lemmata in group (iv). The remaining (unshaded) lemmata are of an auxiliary nature. The figure also lists the three definitions required—in addition to the material presented in this paper so far—to grasp the content of Theorem 4.40 right away, in case the reader wants to jump to the main result of our study immediately rather than going the long way through the successive steps leading to it. Taking the latter, steeper trail is rewarded by Theorem 4.19, encountered halfway, which guarantees call-by-name efficiency nondeterioration by the composition construction for the case that one of the involved mtts is a tdtt. In Section 4.9 we combine Theorems 4.19 and 4.40 with conditions under which there is no difference between the numbers of call-by-name and call-by-need steps, respectively, required by the original program. This leads to Theorem 4.44, containing conditions for efficiency nondeterioration with respect to call-by-need reduction. We also recall corresponding results from [Voi02a] about the situation that at least one of the two mtts involved in the composition is a tdtt.

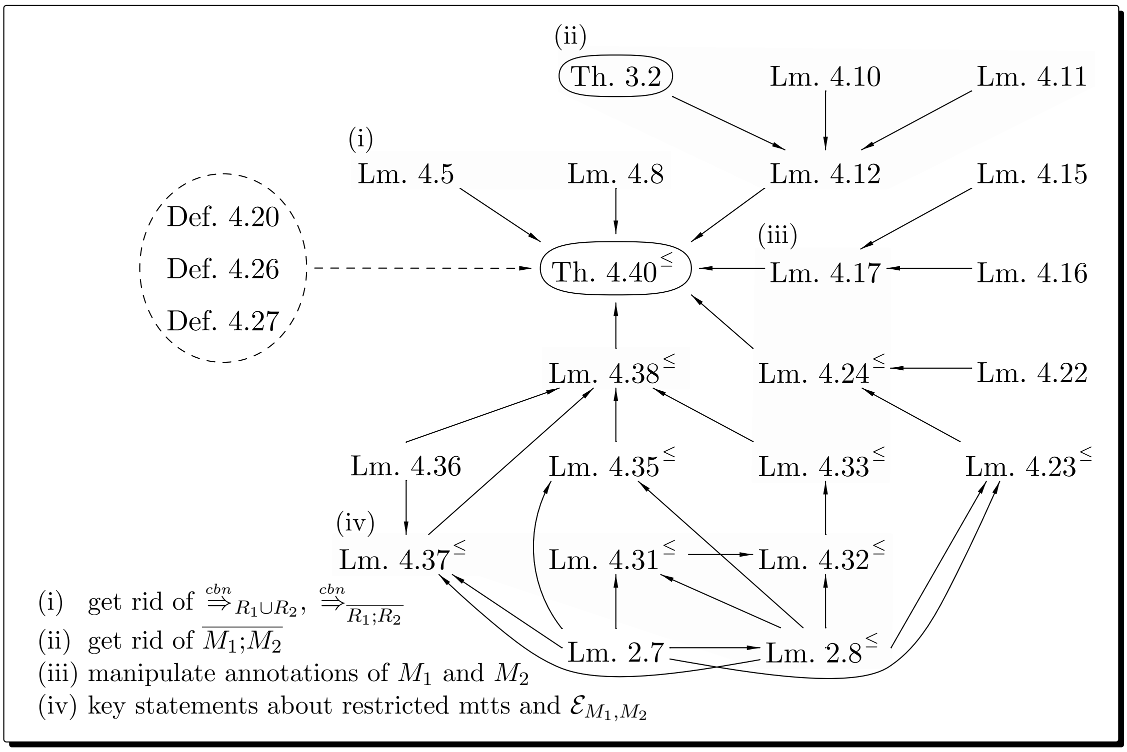

Figure 5: Dependencies for Theorem 4.40.

4.3 Annotating programs to reflect computational costs

Required computation time is an intensional property of a program, i.e., it cannot be determined by solely observing which output is generated for which input according to the program's (extensional) semantics. This implies that methods developed to reason about extensional properties of programs are not readily applicable to study it. For example, the correctness proof for our composition construction in [VK04b] makes heavy use of reordering of normal form computations, justified by the fact that the reduction relation induced by the rules of any mtt is confluent. But since the efficiency analysis we are aiming at here is concerned with a particular fixed reduction strategy, any offhand use of confluence is denied us now.

A seemingly naïve—but often quite effective, see related work in Section 5.3—strategy in such a situation is to externalize the intensional property, e.g. by transforming or annotating programs, thus making it more accessible. To do so, we enrich the output generated by a program, explicitly indicating every performed reduction step (every "tick" of computation time). In the following, special unary symbols $\diamond$, $\bullet$, $\circ$, and $\star$, assumed not to occur in $\Sigma \cup \Delta \cup \Omega \cup F \cup G$, are used to this end.

The natural way to keep track of reduction steps is by generating a dedicated tick-symbol whenever some rewrite rule is applied, e.g. by replacing every rule

$$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \rightarrow rhs$$

by a corresponding one with an additional symbol on top of its right-hand side:

$$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \rightarrow \diamond\ rhs.$$

The dedicated symbol then appears scattered over the output tree generated by computing with such annotated rules. Compared to an attempt of gathering all the information about performed reduction steps in one place, e.g. at the root of the complete output or as an additional function result, this has the advantage that the origin of computational costs is remembered. In particular, discarding a piece of output produced inside an unused argument position also removes from consideration any computational costs associated with it. Moreover, the locational information associated with the positions of $\diamond$-symbols is important if the output produced by the annotated rules above is not yet the final output. Recall that in the original program—computing with $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ on an instantiation of $e_{M_2}[v_1 \leftarrow e_{M_1}]$—the rules R_1 are used to produce an intermediate result then consumed by states of M_2. The latter must therefore not only record their own reduction steps properly in the final output, but also propagate information about the computational costs of M_1 for producing the intermediate result. By simply including in their count any $\diamond$-symbols they come across, they can count exactly those reduction steps of M_1 that produce parts of the intermediate result actually demanded during computation of the final output.

The following definition gives annotated versions of M_1 and M_2 along these lines. Lemma 4.5 then establishes that the composition of the tree transductions induced by these annotated mtts produces, for every input, as many $\bullet$-symbols as the original program performs call-by-name steps.

Definition 4.4 ($M_1^{\rightarrow\diamond}$ and $M_2^{\diamond\rightarrow\bullet}$)
The mtts $M_1^{\rightarrow\diamond} = (F, \Sigma, \Delta \cup \{\diamond\}, e_{M_1}, R_1^{\rightarrow\diamond})$ and $M_2^{\diamond\rightarrow\bullet} = (G, \Delta \cup \{\diamond\}, \Omega \cup \{\bullet\}, e_{M_2}, R_2^{\diamond\rightarrow\bullet})$ have rules as follows:

$R_1^{\to\diamond}$:		
$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r$	$\to \diamond\ rhs_{M_1,f,\sigma}$	$\forall f \in F^{(r+1)}, \sigma \in \Sigma^{(p)}$
$R_2^{\diamond\to\bullet}$:		
$g\ (\delta\ v_1 \cdots v_q)\ z_1 \cdots z_s$	$\to \bullet\ rhs_{M_2,g,\delta}$	$\forall g \in G^{(s+1)}, \delta \in \Delta^{(q)}$
$g\ (\diamond\ v_1)\ z_1 \cdots z_s$	$\to \bullet\ (g\ v_1\ z_1 \cdots z_s)$	$\forall g \in G^{(s+1)}$

Note that the initial expressions remain unchanged. □

Lemma 4.5 ($M_1^{\to\diamond}$ and $M_2^{\diamond\to\bullet}$ count the call-by-name steps of M_1 and M_2)
$cbn_{M_1;M_2} = \tau_{M_1^{\to\diamond}}\,;\tau_{M_2^{\diamond\to\bullet}}\,;|\,.\,|_\bullet$ □

The proof (by an induction on the length of call-by-name reduction sequences, using another induction over the structure of elements of $WFT_{M_1;M_2}$) can be found in the appendix.

Example 4.6 (counting call-by-name steps for the mtts from Example 2.2)
According to Definition 4.4, the mtts $M_{spine}^{\to\diamond}$ and $M_{pfx}^{\diamond\to\bullet}$ have rules

$$
\begin{array}{llll}
R_{spine}^{\to\diamond}: & f_1\ (\alpha\ u_1)\ y_1 & & \to \diamond\ (\alpha\ (f_1\ u_1\ y_1)) \\
 & f_1\ (\beta\ u_1)\ y_1 & & \to \diamond\ (f_2\ u_1\ y_1\ (\beta\ \epsilon)) \\
 & f_1\ \epsilon & y_1 & \to \diamond\ y_1 \\[1ex]
 & f_2\ (\alpha\ u_1)\ y_1\ y_2 & & \to \diamond\ (\alpha\ (f_1\ u_1\ (\delta\ y_1\ y_2))) \\
 & f_2\ (\beta\ u_1)\ y_1\ y_2 & & \to \diamond\ (f_2\ u_1\ y_1\ (\beta\ (\beta\ y_2))) \\
 & f_2\ \epsilon & y_1\ y_2 & \to \diamond\ (\delta\ y_1\ y_2)
\end{array}
$$

and

$$
\begin{array}{llll}
R_{pfx}^{\diamond\to\bullet}: & g\ (\delta\ v_1\ v_2) & z_1 & \to \bullet\ (\delta\ (g\ v_1\ (g\ v_2\ z_1))) \\
 & g\ (\alpha\ v_1) & z_1 & \to \bullet\ (\alpha\ (g\ v_1\ z_1)) \\
 & g\ (\beta\ v_1) & z_1 & \to \bullet\ (\beta\ (g\ v_1\ z_1)) \\
 & g\ \epsilon & z_1 & \to \bullet\ (\epsilon\ z_1) \\
 & g\ (\diamond\ v_1) & z_1 & \to \bullet\ (g\ v_1\ z_1)
\end{array}
$$

respectively. Figure 6 shows an example computation using these annotated rules, starting from the same input as in Figure 4. Note that the number of $\bullet$-symbols in the final output is the same as the number of call-by-name reduction steps required in Figure 4. Further note that this final output is independent of the used reduction strategy: any other order of applying rules from $R_{spine}^{\to\diamond}$ and $R_{pfx}^{\diamond\to\bullet}$ would have yielded the same normal form! □

For the composed program things are simpler than above because it consists of only a single mtt, so no propagation of annotated symbols from an intermediate to the final result is necessary. We just need to put some (other) special symbol on top of the right-hand side of every rule of $\overline{M_1;M_2}$. The following definition and lemma should thus hold no surprises.

Definition 4.7 ($\overline{M_1;M_2}^{\to\circ}$)
The mtt $\overline{M_1;M_2}^{\to\circ} = (H, \Sigma, \Omega \cup \{nil, \circ\}, e_{\overline{M_1;M_2}}, \overline{R_1;R_2}^{\to\circ})$ has rules as follows:

$\overline{R_1;R_2}^{\to\circ}$:		
$h\ (\sigma\ u_1 \cdots u_p) \cdots$	$\to \circ\ rhs_{\overline{M_1;M_2},h,\sigma}$	$\forall h \in H, \sigma \in \Sigma^{(p)}$

where on the left-hand sides the same context variables occur (in the same order) as in

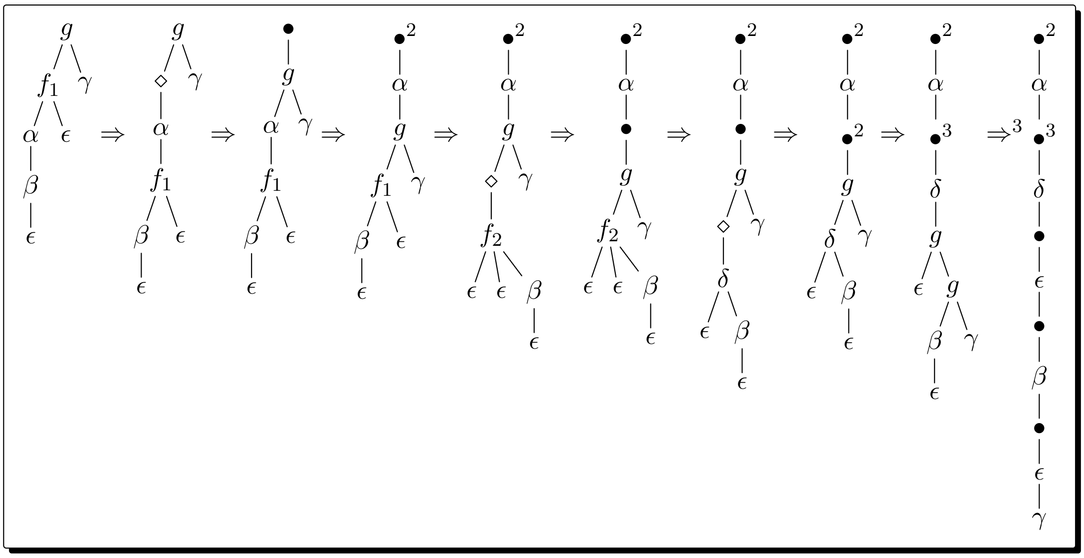

Figure 6: Reduction sequence using $\Rightarrow_{R^{\to\diamond}_{spine} \cup R^{\diamond\to\bullet}_{pfx}}$.

the corresponding rules of $\overline{M_1;M_2}$. $\square$

Lemma 4.8 ($\overline{M_1;M_2}^{\to\circ}$ counts the call-by-name steps of $\overline{M_1;M_2}$)
$cbn_{\overline{M_1;M_2}} = \tau_{\overline{M_1;M_2}^{\to\circ}} \,;\, |\centerdot|_\circ$ $\square$

The proof is analogous to—but slightly simpler than—that of Lemma 4.5; we omit it.

Through the characterization of $cbn_{M_1;M_2}$ and $cbn_{\overline{M_1;M_2}}$ in terms of $\tau_{M_1^{\to\diamond}}$, $\tau_{M_2^{\diamond\to\bullet}}$, and $\tau_{\overline{M_1;M_2}^{\to\circ}}$ we have for now freed ourselves from concerns about a particular reduction strategy (though we will return to this issue in Section 4.9 when considering the call-by-need case). Since the tree transduction induced by an mtt is defined without imposing call-by-name or another policy on reductions computing the normal forms of instantiations of its initial expression, we can reason about the outputs generated by the mtts $M_1^{\to\diamond}$, $M_2^{\diamond\to\bullet}$, and $\overline{M_1;M_2}^{\to\circ}$ with much more ease than about the lengths of very specific reduction sequences involving M_1, M_2, and $\overline{M_1;M_2}$. The externalization will thus prove fruitful.

4.4 Pushing annotation through composition

One hurdle for analyzing the relation between the number of $\circ$-symbols produced by $\overline{M_1;M_2}^{\to\circ}$ and the number of $\bullet$-symbols produced by $M_1^{\to\diamond}$ followed by $M_2^{\diamond\to\bullet}$ is that the rule structure of the former mtt seems rather unconnected to that of the latter two on first sight. But of course, there is a tight connection in that $M_1^{\to\diamond}$ and $M_2^{\diamond\to\bullet}$ are annotated versions of the original mtts M_1 and M_2 while $\overline{M_1;M_2}^{\to\circ}$ was obtained by annotating the rules produced by our composition construction, given those original mtts as input. The most promising approach to exploit this connection is by trying to get an alternative perspective on $\overline{M_1;M_2}^{\to\circ}$: rather than viewing it as an annotated version of the composed mtt, can we also view it as the composition of annotated versions of the original mtts? This would allow us to shift further attention completely to the level of (annotations of) the original program; after all the base on which we aim to formulate conditions for efficiency nondeterioration.

The quest for appropriately annotated versions of M_1 and M_2 whose composition according to Construction 3.1 would differ from $\overline{M_1;M_2}$ only by an added $\circ$-symbol on top of every (non-*nil*[19]) rule right-hand side required a more thorough analysis of the internal workings of the composition construction and the interplay between its ingredients than the necessarily cursory explanations in Section 3. The result, however, is surprisingly simple and elegant, and comes with a pretty straightforward proof that it indeed has the desired characteristic. Moreover, it sheds new light on the roles of the states of the composed mtt as identified in Observation 3.3.

Recall that a state $\overline{fg}$ of $\overline{M_1;M_2}$ essentially computes the output of state g of M_2 run on an intermediate result produced by state f of M_1. It consumes its recursion input with the same granularity as f does, namely by pattern-matching against the root symbol. Hence, to chart steps of $\overline{fg}$ in terms of the original program, it makes sense to mark portions of the intermediate result produced by a single step of f (reusing the $\diamond$-symbol):

$$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \rightarrow \diamond\ (\cdots)$$

and to record the consumption of such a portion by g as follows in the final output:

$$g\ (\diamond\ v_1)\ z_1 \cdots z_s \rightarrow \circ\ (g\ v_1\ z_1 \cdots z_s).$$

So far, the approach is the same as in [Voi02a] for the special case that one of the involved mtts is a tdtt; but for $\overline{k_f l_{g'}}$-states, which were not present in that special case, things become considerably more intricate. They are used to determine the values of context parameters of M_2's state g' on reaching the positions of context parameters of M_1's state f in an intermediate result produced by f for a given input. To model such behavior, we explicitly record positions in the intermediate result where a context parameter of f was used by marking all context variables in rule right-hand sides of f with yet another dedicated symbol $\star$. The cost associated in the composed program with accessing a context parameter of g' on a thus marked context parameter of a state of M_1 is reflected by adding the following rule:

$$g'\ (\star\ v_1)\ z_1 \cdots z_{s'} \rightarrow g'\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_{s'}).$$

Its effect is illustrated in Figure 7; compare that to Figure 2. The chosen scheme ensures that if some of the context parameters of g' are not required for its computation on a concrete instantiation of v_1, e.g. ϕ_k in the figure, then the $\circ$-symbols stored in the corresponding positions will not be included in the final count as they will be discarded along with the values they mark. That this is exactly what we want hinges on the fact that a $\overline{k_f l_{g'}}$-computation is necessary only if the corresponding value is really required in the replacement (2) for (1) in Section 3.

Summarizing, we obtain the following annotated versions of M_1 and M_2.

Definition 4.9 ($M_1^{\rightarrow\diamond\star}$ and $M_2^{\diamond\star\rightarrow\circ}$)
The mtts $M_1^{\rightarrow\diamond\star} = (F, \Sigma, \Delta \cup \{\diamond, \star\}, e_{M_1}, R_1^{\rightarrow\diamond\star})$ and $M_2^{\diamond\star\rightarrow\circ} = (G, \Delta \cup \{\diamond, \star\}, \Omega \cup \{\circ\}, e_{M_2}, R_2^{\diamond\star\rightarrow\circ})$ have rules as follows:

[19] Rules with the dummy symbol *nil* as right-hand side are guaranteed not to have an impact on the outcome of the composed mtt's computation. Hence, it is irrelevant how the corresponding right-hand side of the annotated version looks like.

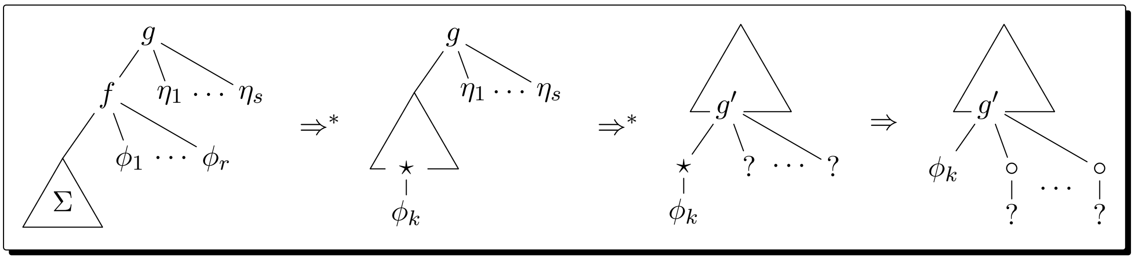

Figure 7: Effects of the $\star$-annotation and associated rules for the second mtt.

$R_1^{\rightarrow\diamond\star}$:		
$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \rightarrow \diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]]$		$\forall f \in F^{(r+1)}, \sigma \in \Sigma^{(p)}$
$R_2^{\diamond\star\rightarrow\circ}$:		
$g\ (\delta\ v_1 \cdots v_q)$	$z_1 \cdots z_s \rightarrow rhs_{M_2,g,\delta}$	$\forall g \in G^{(s+1)}, \delta \in \Delta^{(q)}$
$g\ (\diamond\ v_1)$	$z_1 \cdots z_s \rightarrow \circ\ (g\ v_1\ z_1 \cdots z_s)$	$\forall g \in G^{(s+1)}$
$g\ (\star\ v_1)$	$z_1 \cdots z_s \rightarrow g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s)$	$\forall g \in G^{(s+1)}$

Note that $M_1^{\rightarrow\diamond\star}$ is a tdtt iff M_1 is; likewise for context-linearity. Further, $M_2^{\diamond\star\rightarrow\circ}$ is a tdtt iff M_2 is; likewise for weakly single-useness. $\square$

By the remarks in Definition 4.9, applicability of Construction 3.1 to M_1 and M_2 implies its applicability also to $M_1^{\rightarrow\diamond\star}$ and $M_2^{\diamond\star\rightarrow\circ}$, yielding the mtt $\overline{M_1^{\rightarrow\diamond\star};M_2^{\diamond\star\rightarrow\circ}} = (H, \Sigma, \Omega \cup \{nil, \circ\}, e_{\overline{M_1^{\rightarrow\diamond\star};M_2^{\diamond\star\rightarrow\circ}}}, \overline{R_1^{\rightarrow\diamond\star};R_2^{\diamond\star\rightarrow\circ}})$, which we aim to show equivalent to $\overline{M_1;M_2}^{\rightarrow\circ}$.

Note that between the two applications of Construction 3.1 to M_1 and M_2, respectively to $M_1^{\rightarrow\diamond\star}$ and $M_2^{\diamond\star\rightarrow\circ}$, the $nest_f$-functions and the rewrite systems Pre and $Pair$ are unchanged because their definitions only depend on F and G. The following two auxiliary lemmata study the impact of the differences which do exist between the rule sets of the mtts involved in those two invocations of the composition construction.

Lemma 4.10 (auxiliary)
For every $f \in F^{(r+1)}$, $g \in G^{(s+1)}$, and $\sigma \in \Sigma$:

$$\begin{aligned} & nf(\Rightarrow_{R_2^{\diamond\star\rightarrow\circ} \cup Pre \cup Pair}, g\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]]\ z_1 \cdots z_s) \\ = {} & nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, g\ rhs_{M_1,f,\sigma}\ z_1 \cdots z_s). \quad \square \end{aligned}$$

The proof (of a more general statement by induction over the structure of right-hand sides of rules of M_1, using nested inductions over the structure of right-hand sides of rules of M_2 and over the reversed subset-order on the type of the third argument of $nest_f$-functions) can be found in the appendix. It relies on the facts that, for every $\delta \in \Delta$, the δ-rules do not differ between $R_2^{\diamond\star\rightarrow\circ}$ and R_2, that $rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]]$ contains no $\diamond$-symbols, that it contains $\star$-symbols only atop context variables y_k, and that a reduction

$$g\ (\star\ y_k)\ \varrho_1 \cdots \varrho_s \Rightarrow_{R_2^{\diamond\star\rightarrow\circ}} g\ y_k\ (\circ\ \varrho_1)\ \cdots\ (\circ\ \varrho_s) \Rightarrow_{Pre} y_{k,g}$$

for arbitrary g, y_k, and $\varrho_1, \ldots, \varrho_s$ has the same outcome as a $\Rightarrow_{Pre}$-reduction for a corresponding call to g on (unannotated) y_k.

Lemma 4.11 (auxiliary)
For every $f \in F^{(r+1)}$, $g \in G^{(s+1)}$, $\sigma \in \Sigma$, $\pi \in paths(rhs_{M_1,f,\sigma})$ with $lab(rhs_{M_1,f,\sigma}, \pi) \in Y_r$,

and $l \in [s]$:

$$nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ}, rhs_{M_1^{\to\diamond\star},f,\sigma}}(1\pi, g, l)) \\ = nf(\Rightarrow_{R_2\cup Pre\cup Pair}, par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g, l)). \quad \square$$

The proof (of a slightly more general statement by induction over the prefix-order of paths in $rhs_{M_1,f,\sigma}$, using a nested induction over the structure of right-hand sides of rules of M_2 and two statements from the proof of Lemma 4.10) can be found in the appendix. Beside the facts that, for every $\delta \in \Delta$, the δ-rules are the same in $R_2^{\diamond\star\to\circ}$ and R_2, that no $\star$-symbols occur between the root of $rhs_{M_1^{\to\diamond\star},f,\sigma} = \diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]]$ and the one at path 1π, and that the use of substitution $[y_k \leftarrow \star\ y_k \mid k \in [r]]$ makes for no difference between computations using states from G with reduction relations $\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}$ and $\Rightarrow_{R_2\cup Pre\cup Pair}$, respectively (see above), the key is that the recursive calls in $R_2^{\diamond\star\to\circ}$-rules at $\diamond$ are performed with unchanged context parameters.

Rather than stating the aimed-at syntactic equivalence of $\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}$ and $\overline{M_1;M_2}^{\to\circ}$ (which does not strictly hold due to possible differences regarding dummy rules with right-hand side *nil*), we now prove a statement on the level of induced tree transductions. Moreover, we use the fact that $\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}$ was obtained by our **semantics-preserving** composition construction from $M_1^{\to\diamond\star}$ and $M_2^{\diamond\star\to\circ}$ to express the semantics of $\overline{M_1;M_2}^{\to\circ}$, counting the call-by-name steps of the composed program, directly in terms of these annotations of the original program. This implies that we do not even need to perform Construction 3.1 to analyze the efficiency of a composed program resulting from it. While the externalization by annotation in the previous subsection freed us from further concerns about the specifics of call-by-name reduction, the following lemma thus eliminates any necessity to consider the composition construction itself in the further development.

Lemma 4.12 ($M_1^{\to\diamond\star}$ and $M_2^{\diamond\star\to\circ}$ simulate $\overline{M_1;M_2}^{\to\circ}$)
$\tau_{\overline{M_1;M_2}^{\to\circ}} = \tau_{M_1^{\to\diamond\star}} ; \tau_{M_2^{\diamond\star\to\circ}}$

Proof
We first prove that for every $h \in H$ and $\sigma \in \Sigma$, if $rhs_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}},h,\sigma} \neq nil$, then

$$rhs_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}},h,\sigma} = rhs_{\overline{M_1;M_2}^{\to\circ},h,\sigma},$$

by case distinction on h. On the one hand, for every $f \in F^{(r+1)}$ and $g \in G^{(s+1)}$:

$rhs_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}},\overline{fg},\sigma}$
$\quad =$ (by Construction 3.1)
$nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ rhs_{M_1^{\to\diamond\star},f,\sigma}\ z_1 \cdots z_s)$
$\quad =$ (by $rhs_{M_1^{\to\diamond\star},f,\sigma} = \diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]]$, cf. Definition 4.9)
$nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ (\diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]])\ z_1 \cdots z_s)$
$\quad =$ (by $\Rightarrow_{R_2^{\diamond\star\to\circ}}$, using that $rhs_{M_2^{\diamond\star\to\circ},g,\diamond} = \circ\ (g\ v_1\ z_1 \cdots z_s)$ by Definition 4.9, and by normal form)
$\circ\ nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]]\ z_1 \cdots z_s)$
$\quad =$ (by Lemma 4.10)
$\circ\ nf(\Rightarrow_{R_2\cup Pre\cup Pair}, g\ rhs_{M_1,f,\sigma}\ z_1 \cdots z_s)$
$\quad =$ (by Construction 3.1)
$\circ\ rhs_{\overline{M_1;M_2},\overline{fg},\sigma}$

$= $ (by Definition 4.7)
$rhs_{\overline{M_1;M_2}^{\to\circ},\overline{fg},\sigma}$

On the other hand, for every $f \in F^{(r+1)}$, $g \in G^{(s+1)}$, $k \in [r]$, and $l \in [s]$:

– If $rhs_{M_1,f,\sigma}$ does not contain the context variable y_k, then also

$$rhs_{M_1^{\to\diamond\star},f,\sigma} = \diamond\ rhs_{M_1,f,\sigma}[y_{k'} \leftarrow \star\ y_{k'} \mid k' \in [r]]$$

(cf. Definition 4.9) does not contain y_k, and hence by Construction 3.1:

$$rhs_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}},\overline{k_f l_g},\sigma} = nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, nil) = nil.$$

– Otherwise, if $lab(rhs_{M_1,f,\sigma}, \pi) = y_k$—and thus $sub(rhs_{M_1^{\to\diamond\star},f,\sigma}, 1\pi) = \star\ y_k$—for some $\pi \in paths(rhs_{M_1,f,\sigma})$, then:

$rhs_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}},\overline{k_f l_g},\sigma}$
$= $ (by Construction 3.1)
$nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ},rhs_{M_1^{\to\diamond\star},f,\sigma}}(1\pi1, g, l))$
$= $ (by definition of $par_{M_2^{\diamond\star\to\circ},rhs_{M_1^{\to\diamond\star},f,\sigma}}$ in Construction 3.1,
using that $rhs_{M_2^{\diamond\star\to\circ},g,\star} = g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s)$ by Definition 4.9)
$nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \circ\ par_{M_2^{\diamond\star\to\circ},rhs_{M_1^{\to\diamond\star},f,\sigma}}(1\pi, g, l))$
$= $ (by normal form)
$\circ\ nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ},rhs_{M_1^{\to\diamond\star},f,\sigma}}(1\pi, g, l))$
$= $ (by Lemma 4.11)
$\circ\ nf(\Rightarrow_{R_2\cup Pre\cup Pair}, par_{M_2,rhs_{M_1,f,\sigma}}(\pi, g, l))$
$= $ (by Construction 3.1)
$\circ\ rhs_{\overline{M_1;M_2},\overline{k_f l_g},\sigma}$
$= $ (by Definition 4.7)
$rhs_{\overline{M_1;M_2}^{\to\circ},\overline{k_f l_g},\sigma}$

The lemma follows from an instantiation of Theorem 3.2,

$$\tau_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}} = \tau_{M_1^{\to\diamond\star}}\,;\tau_{M_2^{\diamond\star\to\circ}} : T_\Sigma \longrightarrow T_{\Omega\cup\{\circ\}}\,,$$

if for every $t \in T_\Sigma$ we can show that

$$\tau_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}}(t) = nf(\Rightarrow_{\overline{R_1^{\to\diamond\star};R_2^{\diamond\star\to\circ}}}, e_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}}[u_1 \leftarrow t])$$

is equal to

$$\tau_{\overline{M_1;M_2}^{\to\circ}}(t) = nf(\Rightarrow_{\overline{R_1;R_2}^{\to\circ}}, e_{\overline{M_1;M_2}}[u_1 \leftarrow t]).$$

Since the range of $\tau_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}}$ contains no trees with *nil*-symbols, it must be possible to obtain the former normal form by applying only rules from $\overline{R_1^{\to\diamond\star};R_2^{\diamond\star\to\circ}}$ whose right-hand side is not equal to *nil*. But as shown above, all such rules are also contained in

$\overline{R_1;R_2}^{\to\circ}$.[20] This implies that

$$e_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}}[u_1 \leftarrow t] \Rightarrow^*_{\overline{R_1;R_2}^{\to\circ}} \tau_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}}(t).$$

Hence, the desired equality follows from

$$\begin{aligned}
& e_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}}} \\
& \quad = \text{(by Construction 3.1 and Definition 4.9)} \\
& nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pair}, e_{M_2}[v_1 \leftarrow e_{M_1}]) \\
& \quad = \text{(since } e_{M_2}[v_1 \leftarrow e_{M_1}] \in T_{F\cup G\cup\Delta\cup\Omega}(U_1) \text{ contains no } \diamond\text{- or } \star\text{-symbols} \\
& \qquad\quad \text{and since, for every } \delta \in \Delta, \text{ the } \delta\text{-rules in } R_2^{\diamond\star\to\circ} \text{ coincide with} \\
& \qquad\quad \text{those in } R_2 \text{ by Definition 4.9)} \\
& nf(\Rightarrow_{R_2\cup Pair}, e_{M_2}[v_1 \leftarrow e_{M_1}]) \\
& \quad = \text{(by Construction 3.1)} \\
& e_{\overline{M_1;M_2}}
\end{aligned}$$

and the fact that an element of $T_{\Omega\cup\{\circ\}}$ contains no further $\overline{R_1;R_2}^{\to\circ}$-redices. □

To appreciate the value of the previous lemma (in combination with Lemma 4.8), consider the following for the mtts from Example 2.2.

Example 4.13 (counting steps for composed program in running example) The number of $\circ$-symbols produced using the annotated rules

$$\begin{array}{rlll}
R_{spine}^{\to\diamond\star}: & f_1\ (\alpha\ u_1)\ y_1 & \to \diamond\ (\alpha\ (f_1\ u_1\ (\star\ y_1))) \\
& f_1\ (\beta\ u_1)\ y_1 & \to \diamond\ (f_2\ u_1\ (\star\ y_1)\ (\beta\ \epsilon)) \\
& f_1\ \epsilon\ \ y_1 & \to \diamond\ (\star\ y_1) \\[4pt]
& f_2\ (\alpha\ u_1)\ y_1\ y_2 & \to \diamond\ (\alpha\ (f_1\ u_1\ (\delta\ (\star\ y_1)\ (\star\ y_2)))) \\
& f_2\ (\beta\ u_1)\ y_1\ y_2 & \to \diamond\ (f_2\ u_1\ (\star\ y_1)\ (\beta\ (\beta\ (\star\ y_2)))) \\
& f_2\ \epsilon\ \ y_1\ y_2 & \to \diamond\ (\delta\ (\star\ y_1)\ (\star\ y_2))
\end{array}$$

of $M_{spine}^{\to\diamond\star}$ (cf. Definition 4.9) and the rules of $M_{pfx}^{\diamond\star\to\circ}$ (i.e. the rules from R_{pfx} plus $g\ (\diamond\ v_1)\ z_1 \to \circ\ (g\ v_1\ z_1)$ and $g\ (\star\ v_1)\ z_1 \to g\ v_1\ (\circ\ z_1)$) in the example computation in Figure 8 is the same as the number of call-by-name reduction steps required in Figure 3 for the same input. In fact, there even exists a more fine-grained correspondence regarding how many call-by-name reduction steps are required in Figure 3 to produce which part of the output. Namely, one step is required there to output the top-most α-symbol, after two further steps one obtains (in one go) one δ- and one ϵ-symbol. Only after three more steps the next chunk of the output is produced (β and ϵ), before one last step gives the final γ. This information is exactly coded in the distribution of $\circ$-symbols over the output tree in Figure 8. The same kind of correspondence existed already between Figures 6 and 4. □

Summarizing Lemmata 4.5, 4.8, and 4.12, we have that $M_1^{\to\diamond\star}$ and $M_2^{\diamond\star\to\circ}$ count the call-by-name steps of the composed program while $M_1^{\to\diamond}$ and $M_2^{\diamond\to\bullet}$ count the call-by-name steps of the original one. To determine the relative efficiency of the composed vs.

[20] By Construction 3.1 the left-hand sides of the rules in $\overline{R_1^{\to\diamond\star};R_2^{\diamond\star\to\circ}}$ coincide with the left-hand sides of the $\overline{R_1;R_2}$-rules and thus by Definition 4.7 also with those of the $\overline{R_1;R_2}^{\to\circ}$-rules. That the right-hand sides also coincide for $h \in H$ and $\sigma \in \Sigma$ with $rhs_{\overline{M_1^{\to\diamond\star};M_2^{\diamond\star\to\circ}},h,\sigma} \neq nil$ has been shown above.

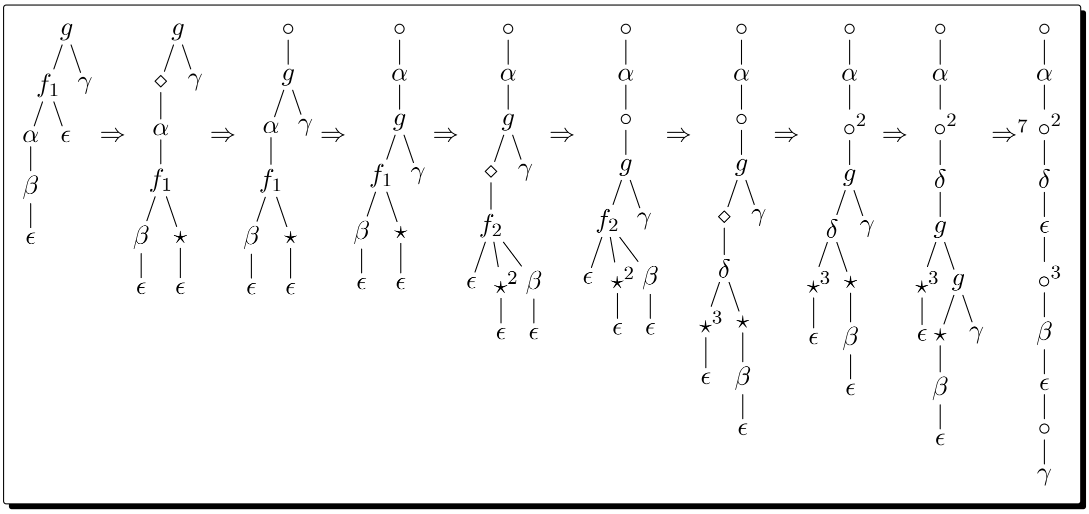

Figure 8: Reduction sequence using $\Rightarrow_{R^{\to\diamond\star}_{spine}\cup R^{\diamond\star\to\circ}_{pfx}}$.

the original program for a particular input, all we need to do is to compare the numbers of $\circ$- and $\bullet$-symbols, respectively, in the outputs produced by those different annotations of the mtts M_1 and M_2 (e.g., compare the values computed in Figures 8 and 6). In the remainder of the paper we follow one particular road to syntactic conditions guaranteeing efficiency nondeterioration that opens up from here. Since clearly we do not want to compare runtimes for just particular inputs, we are seeking sufficient conditions on a pair of mtts M_1 and M_2 under which the number of $\circ$-symbols in the output of $M_1^{\to\diamond\star}$ and $M_2^{\diamond\star\to\circ}$ is less or equal to the number of $\bullet$-symbols in the output of $M_1^{\to\diamond}$ and $M_2^{\diamond\to\bullet}$ (plus some small constant) for every input.

4.5 Combining annotations

To compare the number of $\circ$-symbols produced by $M_1^{\to\diamond\star}$ and $M_2^{\diamond\star\to\circ}$ to the number of $\bullet$-symbols produced by $M_1^{\to\diamond}$ and $M_2^{\diamond\to\bullet}$, it is beneficial to work with only one annotated version for each of M_1 and M_2. Quite naturally, "overlaying" the different annotation schemes leads to the following rules for $f \in F^{(r+1)}$, $\sigma \in \Sigma^{(p)}$, $g \in G^{(s+1)}$, and $\delta \in \Delta^{(q)}$:

$$
\begin{array}{lll}
f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r & \to & \diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]] \\
g\ (\delta\ v_1 \cdots v_q)\ \ z_1 \cdots z_s & \to & \bullet\ rhs_{M_2,g,\delta} \\
g\ (\diamond\ v_1)\qquad\ \ z_1 \cdots z_s & \to & \circ\ (\bullet\ (g\ v_1\ z_1 \cdots z_s)) \\
g\ (\star\ v_1)\qquad\ \ z_1 \cdots z_s & \to & g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s)
\end{array}
$$

Since we are only interested in whether there are more occurrences of one of the symbols $\circ$ and $\bullet$ than of the other, rather than in their absolute numbers, the adjacent $\circ$ and $\bullet$ in the g-rules at $\diamond$ can be omitted. One interpretation for this is that there indeed exists a one-to-one correspondence between steps of $\overline{fg}$ in the composed program and steps of f in the original one.[21]

[21] In particular, this means that if the composed program contains no $\overline{k_f l_g}$-states—which is the case exactly if one of the two original mtts is a tdtt—, then the composed program does for no input perform more call-by-name steps than the original program, as formally established in Theorem 4.19 below. More

Since the new g-rules at $\diamond$ will simply delete any $\diamond$-symbols appearing in the intermediate result, we can equivalently annotate as follows:

$$\begin{array}{lll} f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r & \to rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]] & \\ g\ (\delta\ v_1 \cdots v_q) & z_1 \cdots z_s \to \bullet\ rhs_{M_2,g,\delta} & \\ g\ (\star\ v_1) & z_1 \cdots z_s \to g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s) & \end{array}$$

For the later analysis it will be beneficial to keep the δ-rules unchanged from M_2, i.e. devoid of any annotation. To achieve this, we instead of producing a $\bullet$-symbol whenever such a rule is applied arrange for every $\delta \in \Delta$ in the intermediate result to be explicitly marked with a $\bullet$-symbol that is then reproduced separately using appropriate new rules. This leads to the following annotated versions of M_1 and M_2.

Definition 4.14 ($M_1^{\to\star\bullet}$ and $M_2^{\star\bullet\to\circ\bullet}$)
The mtts $M_1^{\to\star\bullet} = (F, \Sigma, \Delta \cup \{\star, \bullet\}, e_{M_1^{\to\star\bullet}}, R_1^{\to\star\bullet})$ and $M_2^{\star\bullet\to\circ\bullet} = (G, \Delta \cup \{\star, \bullet\}, \Omega \cup \{\circ, \bullet\}, e_{M_2}, R_2^{\star\bullet\to\circ\bullet})$ are given as follows:

$R_1^{\to\star\bullet}$:

$$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \to rhs_{M_1,f,\sigma}[\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta][y_k \leftarrow \star\ y_k \mid k \in [r]] \quad \forall f \in F^{(r+1)}, \sigma \in \Sigma^{(p)}$$

$$e_{M_1^{\to\star\bullet}} = e_{M_1}[\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]$$

$R_2^{\star\bullet\to\circ\bullet}$:

$$\begin{array}{lll} g\ (\delta\ v_1 \cdots v_q) & z_1 \cdots z_s \to rhs_{M_2,g,\delta} & \forall g \in G^{(s+1)}, \delta \in \Delta^{(q)} \\ g\ (\star\ v_1) & z_1 \cdots z_s \to g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s) & \forall g \in G^{(s+1)} \\ g\ (\bullet\ v_1) & z_1 \cdots z_s \to \bullet\ (g\ v_1\ z_1 \cdots z_s) & \forall g \in G^{(s+1)} \end{array}$$

Note that $M_2^{\star\bullet\to\circ\bullet}$ is context-linear iff M_2 is. □

To show that the output computed by $M_1^{\to\star\bullet}$ and $M_2^{\star\bullet\to\circ\bullet}$ correctly reflects the difference between the number of $\circ$-symbols produced by $M_1^{\to\diamond\star}$ and $M_2^{\diamond\star\to\circ}$ on the one hand and the number of $\bullet$-symbols produced by $M_1^{\to\diamond}$ and $M_2^{\diamond\to\bullet}$ on the other hand, we first need two auxiliary lemmata. The first one relates the outputs produced for a particular input by $M_1^{\to\diamond}$ and $M_1^{\to\star\bullet}$ to that produced for the same input by $M_1^{\to\diamond\star}$.

Lemma 4.15 (auxiliary)
For every $t \in T_\Sigma$:

1. $\tau_{M_1^{\to\diamond}}(t) = \tau_{M_1^{\to\diamond\star}}(t)[\star \twoheadleftarrow id]$
2. $\tau_{M_1^{\to\star\bullet}}(t) = \tau_{M_1^{\to\diamond\star}}(t)[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]$.

Proof
Point 1 is immediate from the facts that $M_1^{\to\diamond\star}$ and $M_1^{\to\diamond}$ have the same initial expression containing no $\star$-symbols and that for every $f \in F$ and $\sigma \in \Sigma$, $rhs_{M_1^{\to\diamond},f,\sigma} = rhs_{M_1^{\to\diamond\star},f,\sigma}[\star \twoheadleftarrow id]$. Similarly, point 2 follows from the facts that $e_{M_1^{\to\star\bullet}}$ can be obtained from the initial expression of $M_1^{\to\diamond\star}$ (e_{M_1}, containing no $\diamond$-symbols) via the substitutions $[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]$ and that for every $f \in F$ and $\sigma \in \Sigma$, $rhs_{M_1^{\to\star\bullet},f,\sigma} = rhs_{M_1^{\to\diamond\star},f,\sigma}[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]$. □

precisely, the amount of improvement corresponds to the number of steps performed by states of M_2 in the original program.

The second one relates the numbers of ∘- and •-symbols produced by $M_2^{\diamond\star\to\circ}$, $M_2^{\diamond\to\bullet}$, and $M_2^{\star\bullet\to\circ\bullet}$ for differently annotated versions of the same intermediate result, using the function $|\,.\,|_{\circ-\bullet} = |\,.\,|_\circ - |\,.\,|_\bullet$.

Lemma 4.16 (auxiliary)
For every $t' \in T_{\Delta\cup\{\diamond,\star\}}$:

$$|\tau_{M_2^{\diamond\star\to\circ}}(t')|_\circ - |\tau_{M_2^{\diamond\to\bullet}}(t'[\star \twoheadleftarrow id])|_\bullet = |\tau_{M_2^{\star\bullet\to\circ\bullet}}(t'[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta])|_{\circ-\bullet}. \quad \square$$

The proof (of a more general statement by induction over the structure of elements of $T_{\Delta\cup\{\diamond,\star\}}$, using a nested induction over the structure of right-hand sides of rules of—and of the initial expression of—M_2) can be found in the appendix; it holds no big surprises.

Plugging the two auxiliary lemmata together, we obtain the following.

Lemma 4.17 ($M_1^{\to\star\bullet}$ and $M_2^{\star\bullet\to\circ\bullet}$ count the difference of ∘- and •-symbols)
$\tau_{M_1^{\to\diamond\star}}\,;\tau_{M_2^{\diamond\star\to\circ}}\,;|\,.\,|_\circ - \tau_{M_1^{\to\diamond}}\,;\tau_{M_2^{\diamond\to\bullet}}\,;|\,.\,|_\bullet = \tau_{M_1^{\to\star\bullet}}\,;\tau_{M_2^{\star\bullet\to\circ\bullet}}\,;|\,.\,|_{\circ-\bullet}$

Proof
For every $t \in T_\Sigma$:

$$\begin{aligned}
&|\tau_{M_2^{\diamond\star\to\circ}}(\tau_{M_1^{\to\diamond\star}}(t))|_\circ - |\tau_{M_2^{\diamond\to\bullet}}(\tau_{M_1^{\to\diamond}}(t))|_\bullet\\
&\quad = \text{ (by Lemma 4.15(1))}\\
&|\tau_{M_2^{\diamond\star\to\circ}}(\tau_{M_1^{\to\diamond\star}}(t))|_\circ - |\tau_{M_2^{\diamond\to\bullet}}(\tau_{M_1^{\to\diamond\star}}(t)[\star \twoheadleftarrow id])|_\bullet\\
&\quad = \text{ (by Lemma 4.16 with } t' = \tau_{M_1^{\to\diamond\star}}(t))\\
&|\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\diamond\star}}(t)[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta])|_{\circ-\bullet}\\
&\quad = \text{ (by Lemma 4.15(2))}\\
&|\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\star\bullet}}(t))|_{\circ-\bullet} \quad \square
\end{aligned}$$

Example 4.18 (instantiation of Lemma 4.17 to our running example)
Compare the output computed using the annotated rules

$$\begin{array}{lllll}
R_{spine}^{\to\star\bullet}: & f_1\ (\alpha\ u_1) & y_1 & \to & \bullet\ (\alpha\ (f_1\ u_1\ (\star\ y_1)))\\
& f_1\ (\beta\ u_1) & y_1 & \to & f_2\ u_1\ (\star\ y_1)\ (\bullet\ (\beta\ (\bullet\ \epsilon)))\\
& f_1\ \epsilon & y_1 & \to & \star\ y_1\\
& f_2\ (\alpha\ u_1) & y_1\ y_2 & \to & \bullet\ (\alpha\ (f_1\ u_1\ (\bullet\ (\delta\ (\star\ y_1)\ (\star\ y_2)))))\\
& f_2\ (\beta\ u_1) & y_1\ y_2 & \to & f_2\ u_1\ (\star\ y_1)\ (\bullet\ (\beta\ (\bullet\ (\beta\ (\star\ y_2)))))\\
& f_2\ \epsilon & y_1\ y_2 & \to & \bullet\ (\delta\ (\star\ y_1)\ (\star\ y_2))
\end{array}$$

of $M_{spine}^{\to\star\bullet}$ (and the new initial expression $e_{M_{spine}^{\to\star\bullet}} = f_1\ u_1\ (\bullet\ \epsilon)$, cf. Definition 4.14) and the rules of $M_{pfx}^{\star\bullet\to\circ\bullet}$ (i.e. the rules from R_{pfx} plus $g\ (\star\ v_1)\ z_1 \to g\ v_1\ (\circ\ z_1)$ and $g\ (\bullet\ v_1)\ z_1 \to \bullet\ (g\ v_1\ z_1)$) in Figure 9 to that from Figures 6 and 8 (for the same input). The continuity is that there is one more •-symbol than there are ∘-symbols. $\square$

It is easy to see that the output computed by $M_1^{\to\star\bullet}$ and $M_2^{\star\bullet\to\circ\bullet}$ will contain no ∘-symbols if one of the two is a tdtt. This yields a first useful result of our efficiency analysis, albeit for a very special case.

Theorem 4.19 (efficiency nondeterioration if M_1 or M_2 is a tdtt)
If either of M_1 and M_2 is a tdtt, then $cbn_{\overline{M_1;M_2}}(t) \leq cbn_{M_1;M_2}(t)$ for every $t \in T_\Sigma$.

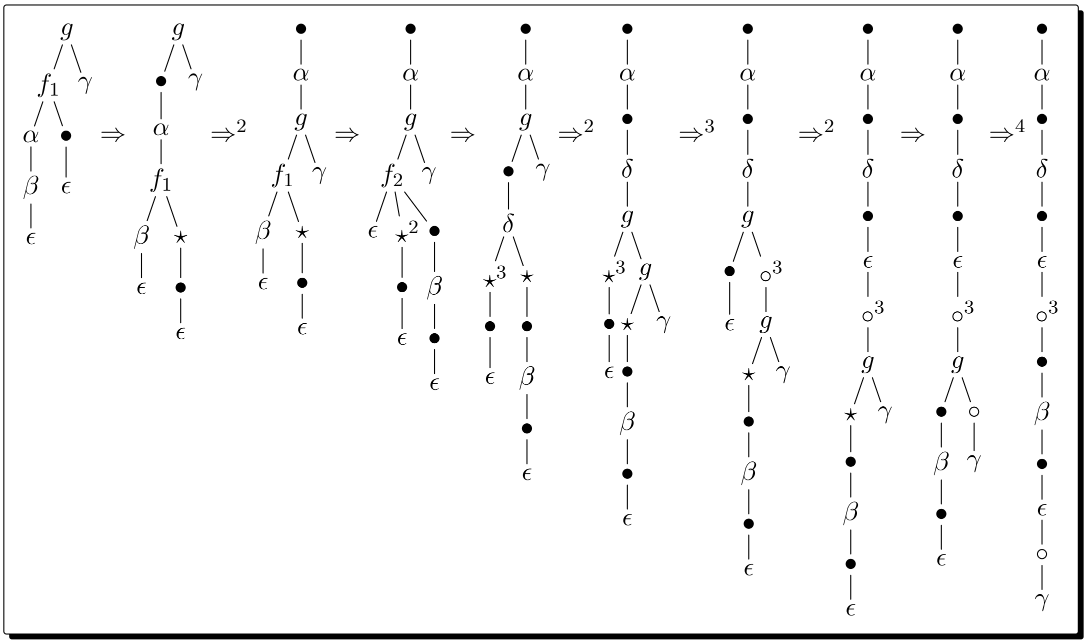

Figure 9: Reduction sequence using $\Rightarrow_{R^{\to\star\bullet}_{spine}\cup R^{\star\bullet\to\circ\bullet}_{pfx}}$.

Proof
We calculate

$$
\begin{aligned}
& cbn_{\overline{M_1;M_2}} - cbn_{M_1;M_2} \\
& \quad = \text{ (by Lemmata 4.8 and 4.5)} \\
& \tau_{\overline{M_1;M_2}^{\to\circ}} \,;\, |\,\cdot\,|_\circ - \tau_{M_1^{\to\diamond}} \,;\, \tau_{M_2^{\diamond\to\bullet}} \,;\, |\,\cdot\,|_\bullet \\
& \quad = \text{ (by Lemma 4.12)} \\
& \tau_{M_1^{\to\diamond\star}} \,;\, \tau_{M_2^{\diamond\star\to\circ}} \,;\, |\,\cdot\,|_\circ - \tau_{M_1^{\to\diamond}} \,;\, \tau_{M_2^{\diamond\to\bullet}} \,;\, |\,\cdot\,|_\bullet \\
& \quad = \text{ (by Lemma 4.17)} \\
& \tau_{M_1^{\to\star\bullet}} \,;\, \tau_{M_2^{\star\bullet\to\circ\bullet}} \,;\, |\,\cdot\,|_{\circ-\bullet}\,,
\end{aligned}
$$

which implies

$$cbn_{\overline{M_1;M_2}}(t) - cbn_{M_1;M_2}(t) = |\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\star\bullet}}(t))|_\circ - |\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\star\bullet}}(t))|_\bullet$$

for every $t \in T_\Sigma$.

The only possibility for $\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\star\bullet}}(t))$ to contain $\circ$-symbols is via applications of $\star$-rules of $M_2^{\star\bullet\to\circ\bullet}$. If M_1 is a tdtt, then $\tau_{M_1^{\to\star\bullet}}(t)$ contains no $\star$-symbols because there are no context variables in M_1's rule right-hand sides in this case. If M_2 is a tdtt, then the $\star$-rules of $M_2^{\star\bullet\to\circ\bullet}$ produce no $\circ$-symbols at all because $G = G^{(1)}$ in this case. In both cases we obtain $|\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\star\bullet}}(t))|_\circ = 0$. Since $|\tau_{M_2^{\star\bullet\to\circ\bullet}}(\tau_{M_1^{\to\star\bullet}}(t))|_\bullet$ is always nonnegative, the theorem follows. $\square$

4.6 Unifying the treatment of $\circ$- and $\bullet$-symbols

For the general case, i.e. when neither of the two mtts to be composed is a tdtt, we have reduced our analysis problem to comparing—for each input—the numbers of $\circ$- and

•-symbols appearing in the final output produced by $M_1^{\rightarrow\star\bullet}$ and $M_2^{\star\bullet\rightarrow\circ\bullet}$. While the emergence of •-symbols by adding one atop every symbol from Δ in the intermediate result and then simply reproducing them using rules

$$g\ (\bullet\ v_1)\ z_1 \cdots z_s \rightarrow \bullet\ (g\ v_1\ z_1 \cdots z_s)$$

for $g \in G^{(s+1)}$ is already quite handy, the completely different manner in which ∘-symbols are produced, namely through rules of the form

$$g\ (\star\ v_1)\ z_1 \cdots z_s \rightarrow g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s) \tag{3}$$

from ⋆-symbols in the intermediate result, is somewhat unwieldy. Clearly, it would facilitate the comparison if ∘-symbols were also to appear directly in right-hand sides of rules of the first mtt and to be reproduced by the second mtt using rules analogous to the ones for •. A first step in that direction is to move all ∘-symbols out of the context parameter positions in ⋆-rules as in the following replacement for (3):

$$g\ (\star\ v_1)\ z_1 \cdots z_s \rightarrow \underbrace{\circ\ (\cdots \circ}_{s \text{ times}}\ (g\ v_1\ z_1 \cdots z_s) \cdots) \tag{4}$$

We have to be careful, though, because this might change the total number of ∘-symbols produced. If, for example, g has the potential to delete one of its context parameters, then the ∘-symbol in that position in the right-hand side of (3) might be discarded while the corresponding outwards moved symbol in (4) is unnecessarily included in the count. Conversely, if g may copy one of its context parameters, then a ∘-symbol that is rightfully counted several times when using (3) would be counted only once when using (4) instead. While overestimating the number of ∘-symbols and thus the number of call-by-name steps performed by the composed program is acceptable when looking for sufficient (rather than necessary) conditions guaranteeing efficiency nondeterioration through our construction, any underestimation in that respect is of course unsafe. It is avoided by performing the replacement of (3) with (4) only if $M_2^{\star\bullet\rightarrow\circ\bullet}$ is context-linear.

The next step is to directly have sequences of ∘-symbols instead of single ⋆-symbols in the right-hand sides of rules of the first mtt and to reproduce those ∘-symbols one by one with an appropriate rule of the second mtt. Since states in G can have different numbers of context parameters, to be on the safe side we have to replace every ⋆ by s_{max} times ∘, where s_{max} is the maximal such number. This yields for every $f \in F^{(r+1)}$ and $\sigma \in \Sigma^{(p)}$ the following rule:

$$f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \rightarrow rhs_{M_1,f,\sigma}[\delta \Leftarrow \bullet \cdot \delta \mid \delta \in \Delta][y_k \leftarrow \circ^{s_{max}}\ y_k \mid k \in [r]].$$

In the next subsection our analysis will proceed by finding conditions under which pairs of ∘- and •-symbols appearing in one and the same annotated rule right-hand side of M_1 can be caused to cancel each other out without compromising safety. To gain flexibility in this endeavor, it would be beneficial if the ∘-symbols in question were not always fixed to appear cascaded above all context variables. Without changing the computed annotated intermediate result (as proved in Lemma 4.22 below), this can be achieved by, e.g., moving some of the s_{max} symbols from the top of all occurrences of context variable y_k in every right-hand side for f to the kth context parameter position of every call to f throughout right-hand sides or the initial expression. To provide for maximal freedom of choice, the amount of ∘-symbols to be relocated thus can differ for different f and k, as modeled in the following definition through parameterization over the mapping κ.

Definition 4.20 ($M_1^{\to\circ\bullet,\kappa}$ and $M_2^{\circ\bullet\to\circ\bullet}$)
Let $s_{max} = max(rank(G)) - 1$ and let $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\ldots,s_{max}\}$ be arbitrary. For every $f \in F^{(r+1)}$ and $k \in [r]$ abbreviate $\kappa_{f,k} = \kappa(f,k)$ and $\overline{\kappa}_{f,k} = s_{max} - \kappa(f,k)$. The mtts $M_1^{\to\circ\bullet,\kappa} = (F, \Sigma, \Delta \cup \{\circ,\bullet\}, e_{M_1^{\to\circ\bullet,\kappa}}, R_1^{\to\circ\bullet,\kappa})$ and $M_2^{\circ\bullet\to\circ\bullet} = (G, \Delta \cup \{\circ,\bullet\}, \Omega \cup \{\circ,\bullet\}, e_{M_2}, R_2^{\circ\bullet\to\circ\bullet})$ are given as follows:

$R_1^{\to\circ\bullet,\kappa}$:

$$
\begin{array}{ll}
f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \to rhs_{M_1,f,\sigma} & \forall f \in F^{(r+1)}, \sigma \in \Sigma^{(p)} \\
\qquad [\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta] & \\
\qquad [f' \twoheadleftarrow f' \cdot (id \times \circ^{\kappa_{f',1}} \cdots \times \circ^{\kappa_{f',r'}}) & \\
\qquad\qquad \mid f' \in F^{(r'+1)}] & \\
\qquad [y_k \leftarrow \circ^{\overline{\kappa}_{f,k}}\ y_k \mid k \in [r]] &
\end{array}
$$

$$e_{M_1^{\to\circ\bullet,\kappa}} = e_{M_1}[\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta][f' \twoheadleftarrow f' \cdot (id \times \circ^{\kappa_{f',1}} \cdots \times \circ^{\kappa_{f',r'}}) \mid f' \in F^{(r'+1)}]$$

$R_2^{\circ\bullet\to\circ\bullet}$:

$$
\begin{array}{lll}
g\ (\delta\ v_1 \cdots v_q) & z_1 \cdots z_s \to rhs_{M_2,g,\delta} & \forall g \in G^{(s+1)}, \delta \in \Delta^{(q)} \\
g\ (\circ\ v_1) & z_1 \cdots z_s \to \circ\ (g\ v_1\ z_1 \cdots z_s) & \forall g \in G^{(s+1)} \\
g\ (\bullet\ v_1) & z_1 \cdots z_s \to \bullet\ (g\ v_1\ z_1 \cdots z_s) & \forall g \in G^{(s+1)}
\end{array}
$$

Note that $M_2^{\circ\bullet\to\circ\bullet}$ is recursion-linear iff M_2 is; likewise for recursion-nondeletingness, context-linearity and -nondeletingness, and basicness, and hence also for atmostness and atleastness. □

Example 4.21 ($M_{spine}^{\to\circ\bullet,\kappa}$ for $smax = 1$ and a concrete κ)
Choose κ so that $\kappa_{f_1,1} = \kappa_{f_2,2} = 1$ and $\kappa_{f_2,1} = 0$, and hence $\overline{\kappa}_{f_1,1} = \overline{\kappa}_{f_2,2} = 0$ and $\overline{\kappa}_{f_2,1} = 1$. Then, the mtt $M_{spine}^{\to\circ\bullet,\kappa} = (\{f_1^{(2)}, f_2^{(3)}\}, \Sigma_{mon}, \Delta_{tree} \cup \{\circ,\bullet\}, e_{M_{spine}^{\to\circ\bullet,\kappa}}, R_{spine}^{\to\circ\bullet,\kappa})$ has the rules

$$
\begin{array}{rllll}
R_{spine}^{\to\circ\bullet,\kappa}: & f_1\ (\alpha\ u_1) & y_1 & \to & \bullet\ (\alpha\ (f_1\ u_1\ (\circ\ y_1))) \\
& f_1\ (\beta\ u_1) & y_1 & \to & f_2\ u_1\ y_1\ (\circ\ (\bullet\ (\beta\ (\bullet\ \epsilon)))) \\
& f_1\ \epsilon & y_1 & \to & y_1 \\
& f_2\ (\alpha\ u_1) & y_1\ y_2 & \to & \bullet\ (\alpha\ (f_1\ u_1\ (\circ\ (\bullet\ (\delta\ (\circ\ y_1)\ y_2))))) \\
& f_2\ (\beta\ u_1) & y_1\ y_2 & \to & f_2\ u_1\ (\circ\ y_1)\ (\circ\ (\bullet\ (\beta\ (\bullet\ (\beta\ y_2))))) \\
& f_2\ \epsilon & y_1\ y_2 & \to & \bullet\ (\delta\ (\circ\ y_1)\ y_2)
\end{array}
$$

and the initial expression $e_{M_{spine}^{\to\circ\bullet,\kappa}} = f_1\ u_1\ (\circ\ (\bullet\ \epsilon))$. □

To show that—irrespective of the choice of κ—the difference between the numbers of $\circ$- and $\bullet$-symbols in the output computed by $M_1^{\to\circ\bullet,\kappa}$ and $M_2^{\circ\bullet\to\circ\bullet}$ for context-linear M_2 is greater or equal to the corresponding difference in the output produced by $M_1^{\to\star\bullet}$ and $M_2^{\star\bullet\to\circ\bullet}$, we first need two auxiliary lemmata.

Lemma 4.22 (auxiliary)
For every $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\ldots,s_{max}\}$ and $t \in T_\Sigma$:

$$\tau_{M_1^{\to\circ\bullet,\kappa}}(t) = \tau_{M_1^{\to\star\bullet}}(t)[\star \twoheadleftarrow \circ^{s_{max}}]. \quad \square$$

The proof (of a more general statement by induction over the structure of elements of T_Σ, using a nested induction over the structure of right-hand sides of rules of—and of the initial expression of—M_1, annotated using $[\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]$) can be found in the appendix. The key observation is that when a call $f\ t\ (\circ^{\kappa_{f,1}}\ \theta_1)\ \cdots\ (\circ^{\kappa_{f,r}}\ \theta_r)$ is reduced using a rule from $R_1^{\rightarrow\circ\bullet,\kappa}$, additional ∘-symbols are put on top of each context parameter by applying $\circ^{\overline{\kappa}_{f,k}}$ to $(\circ^{\kappa_{f,k}}\ \theta_k)$ for every $k \in [r]$. Since $\overline{\kappa}_{f,k} + \kappa_{f,k} = s_{max}$ for every $k \in [r]$, this corresponds to the $\circ^{s_{max}}$ substituted eventually for the single ⋆-symbol put onto each context parameter when reducing a call to f on t using a rule from $R_1^{\rightarrow\star\bullet}$.

Lemma 4.23 (auxiliary)
If M_2 is context-linear, then for every $t' \in T_{\Delta\cup\{\star,\bullet\}}$:

$$|\tau_{M_2^{\star\bullet\rightarrow\circ\bullet}}(t')|_{\circ-\bullet} \leq |\tau_{M_2^{\circ\bullet\rightarrow\circ\bullet}}(t'[\star \twoheadleftarrow \circ^{s_{max}}])|_{\circ-\bullet}.$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then equality holds. □

The proof (of a more general statement by induction over the structure of elements of $T_{\Delta\cup\{\star,\bullet\}}$, using a nested induction over the structure of right-hand sides of rules of—and of the initial expression of—M_2, as well as Lemmata 2.7 and 2.8 for the mtt M_2) can be found in the appendix. The key idea is to quantify how often the ∘-symbols produced in a step

$$g\ (\star\ t_1')\ \varrho_1 \cdots \varrho_s \Rightarrow_{R_2^{\star\bullet\rightarrow\circ\bullet}} g\ t_1'\ (\circ\ \varrho_1)\ \cdots\ (\circ\ \varrho_s)$$

will appear in the final output in terms of the number of occurrences of $z_1, \ldots, z_s$ in

$$nf(\Rightarrow_{R_2}, g\ t_1'[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_s)\,,$$

which can be approximated using Lemma 2.8, and to compare this to the number of ∘-symbols produced when reducing a call to g on $(\star\ t_1')[\star \twoheadleftarrow \circ^{s_{max}}]$ using (s_{max} times) the rule for g at ∘ in $R_2^{\circ\bullet\rightarrow\circ\bullet}$.

Note that in Lemma 4.23 our count gets imprecise for the first time through the use of an inequation. This stems from the (potential) proliferation of ∘-symbols for the sake of advancing in the analysis by forging the annotation scheme into a more tractable form, as discussed at the beginning of this subsection. To keep some grip on imprecisions introduced by overestimating the number of ∘-symbols—or underestimating the number of •-symbols—, we will make a habit of recording additional conditions on M_1 and M_2 under which given approximations are exact. For practical use in showing that a composed program is more efficient than the original one, a count based on inequations (in the proper direction) is sufficient, though possibly overly pessimistic.

Lemma 4.24 ($M_1^{\rightarrow\circ\bullet,\kappa}$ and $M_2^{\circ\bullet\rightarrow\circ\bullet}$ compute a safe approximation)
For every $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0, \ldots, s_{max}\}$, if M_2 is context-linear, then:

$$\tau_{M_1^{\rightarrow\star\bullet}}\,;\tau_{M_2^{\star\bullet\rightarrow\circ\bullet}}\,;|\,.\,|_{\circ-\bullet} \leq \tau_{M_1^{\rightarrow\circ\bullet,\kappa}}\,;\tau_{M_2^{\circ\bullet\rightarrow\circ\bullet}}\,;|\,.\,|_{\circ-\bullet}.$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then equality holds.

Proof
For every $t \in T_\Sigma$:

$$\begin{aligned}
&|\tau_{M_2^{\star\bullet\rightarrow\circ\bullet}}(\tau_{M_1^{\rightarrow\star\bullet}}(t))|_{\circ-\bullet}\\
&\quad\leq \text{ (by Lemma 4.23 with } t' = \tau_{M_1^{\rightarrow\star\bullet}}(t))\\
&|\tau_{M_2^{\circ\bullet\rightarrow\circ\bullet}}(\tau_{M_1^{\rightarrow\star\bullet}}(t)[\star \twoheadleftarrow \circ^{s_{max}}])|_{\circ-\bullet}\\
&\quad= \text{ (by Lemma 4.22)}\\
&|\tau_{M_2^{\circ\bullet\rightarrow\circ\bullet}}(\tau_{M_1^{\rightarrow\circ\bullet,\kappa}}(t))|_{\circ-\bullet}
\end{aligned}$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then the above inequality can be strengthened to equality by Lemma 4.23. □

Example 4.25 (instantiation of Lemma 4.24 to our running example)
To observe the effect of the transition from $M_1^{\to\star\bullet}$ and $M_2^{\star\bullet\to\circ\bullet}$ to $M_1^{\to\circ\bullet,\kappa}$ and $M_2^{\circ\bullet\to\circ\bullet}$ for our running example with the concrete κ from Example 4.21, compare the computation in Figure 10 to that from Figure 9. In particular, note how as a consequence of replacing (3) with (4), ∘-symbols have "floated upwards" in the computed output, as indicated by the dashed arrows. In fact, the difference between the numbers of ∘- and •-symbols is the same in both outputs, as expected since M_{pfx} is not only context-linear but also context-nondeleting, and since all its states have the same rank as there is only one. □

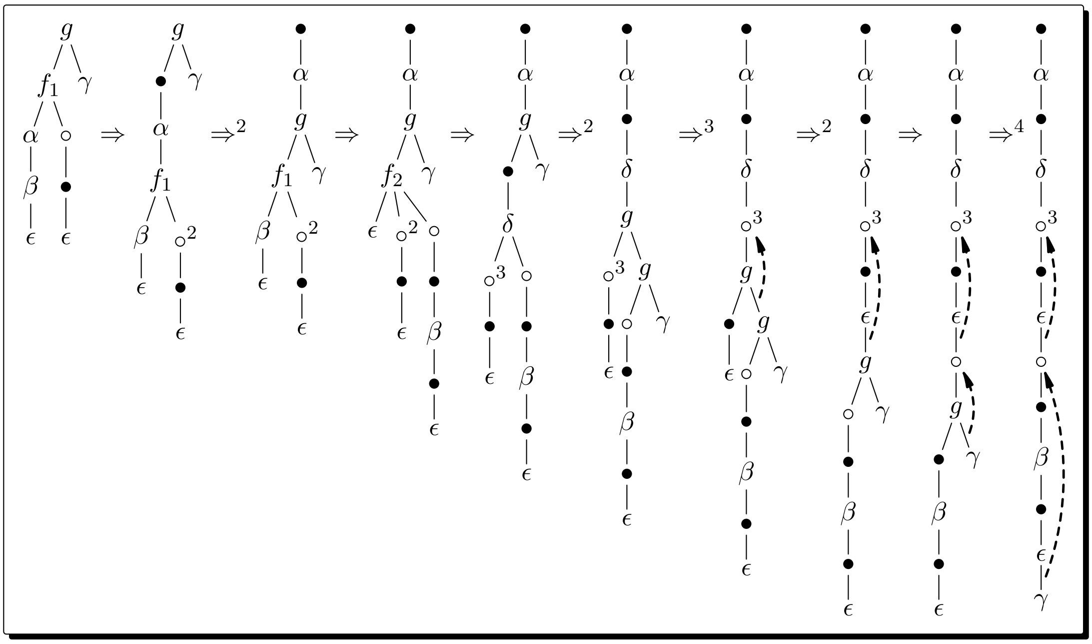

Figure 10: Reduction sequence using $\Rightarrow_{R_{spine}^{\to\circ\bullet,\kappa} \cup R_{pfx}^{\circ\bullet\to\circ\bullet}}$.

4.7 Moving •-symbols to cancel out ∘-symbols

Our ultimate goal is now to put a constant upper bound, for every input, on the difference between the numbers of ∘- and •-symbols reproduced by $M_2^{\circ\bullet\to\circ\bullet}$ from an intermediate result produced by some $M_1^{\to\circ\bullet,\kappa}$. That could in principle be achieved by correspondingly bounding only the number of ∘-symbols because the subtracted number of •-symbols is always nonnegative[22]. Clearly, it is unlikely that there is indeed a constant upper bound for the number of ∘-symbols because the number of applications of rules of $M_1^{\to\circ\bullet,\kappa}$ in whose right-hand sides such symbols appear depends on the input, and so consequently does the number of ∘-symbols in the final output. We would stand much better chances if all ∘-symbols were eliminated from the rules of the mtt producing the annotated intermediate result. Fortunately, we can make this happen by continuing to manipulate the first mtt's

[22]It could then either be neglected or used to attempt a further analysis **by how much** the composed program is more efficient than the original one, rather than just **whether** it is.

annotation, as long as this does not lead to an underestimation of the difference between the numbers of ∘- and •-symbols. One obviously safe manipulation is to once again remove pairs of adjacent ∘- and •-symbols. If such symbols do not appear next to each other, we can attempt to first bring them closer together by moving either kind of symbols. The following definition lists the sensible choices for moving •-symbols around in annotated right-hand sides of rules of the first mtt.[23]

Definition 4.26 (symmetric up- and down-rules for •)
For every $\delta \in \Delta^{(q)}$ and $j \in [q]$ rewrite rules $\downarrow_{\delta,j}$ and $\uparrow_{\delta,j}$, and for every $f \in F^{(r+1)}$ and $k \in [r]$ rewrite rules $\downarrow_{f,k}$ and $\uparrow_{f,k}$, are given as follows:

$$\begin{array}{lll} \downarrow_{\delta,j} & : \bullet\ (\delta\ v_1 \cdots v_q) & \to \delta\ v_1 \cdots (\bullet\ v_j) \cdots v_q \\ \uparrow_{\delta,j} & : \delta\ v_1 \cdots (\bullet\ v_j) \cdots v_q & \to \bullet\ (\delta\ v_1 \cdots v_q) \\ \downarrow_{f,k} & : \bullet\ (f\ u\ v_1 \cdots v_r) & \to f\ u\ v_1 \cdots (\bullet\ v_k) \cdots v_r \\ \uparrow_{f,k} & : f\ u\ v_1 \cdots (\bullet\ v_k) \cdots v_r & \to \bullet\ (f\ u\ v_1 \cdots v_r) \end{array} \qquad \square$$

Such rules may not be applied unconstrained. If, for example, $M_2^{\circ\bullet\to\circ\bullet}$ is not recursion-linear, then the application of a $\downarrow_{\delta,j}$-rule to move a •-symbol below an output symbol in the right-hand side of some rule of the first mtt—and as a consequence also in the annotated intermediate result—can lead to that •-symbol being counted more often after the manipulation than it was previously, in case the jth recursion variable occurs more than once in some δ-rule of M_2. That is, in the presence of a rule

$$g\ (\delta\ v_1\ v_2) \to \omega\ (g\ v_1)\ (g\ v_1)$$

of M_2 an application of the $\downarrow_{\delta,1}$-rule can have the effect that an $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$-reduction

$$g\ (\bullet\ (\delta\ \theta_1\ \theta_2)) \Rightarrow \bullet\ (g\ (\delta\ \theta_1\ \theta_2)) \Rightarrow \bullet\ (\omega\ (g\ \theta_1)\ (g\ \theta_1))$$

for some $\theta_1, \theta_2 \in T_{\Delta\cup\{\circ,\bullet\}}$ is replaced by the following:

$$g\ (\delta\ (\bullet\ \theta_1)\ \theta_2) \Rightarrow \omega\ (g\ (\bullet\ \theta_1))\ (g\ (\bullet\ \theta_1)) \Rightarrow^2 \omega\ (\bullet\ (g\ \theta_1))\ (\bullet\ (g\ \theta_1))\,,$$

thus overestimating the number of •-symbols. In addition to avoiding unlegitimate duplication of a •-symbol through several recursive calls on the subtree into which it was moved, also more subtle ways of duplication via context parameters must be barred. If, for example, M_2 has a rule

$$g\ (\delta\ v_1\ v_2) \to g'\ v_1\ (g\ v_2) \tag{5}$$

then an application of the $\downarrow_{\delta,2}$-rule can have the effect that an $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$-reduction

$$g\ (\bullet\ (\delta\ \theta_1\ \theta_2)) \Rightarrow \bullet\ (g\ (\delta\ \theta_1\ \theta_2)) \Rightarrow \bullet\ (g'\ \theta_1\ (g\ \theta_2))$$

for some $\theta_1, \theta_2 \in T_{\Delta\cup\{\circ,\bullet\}}$ is replaced by the following:

$$g\ (\delta\ \theta_1\ (\bullet\ \theta_2)) \Rightarrow g'\ \theta_1\ (g\ (\bullet\ \theta_2)) \Rightarrow g'\ \theta_1\ (\bullet\ (g\ \theta_2)).$$

If g' happens to copy its context parameter, then one •-symbol is turned into more than one. To avoid this problem, we can either outlaw nested calls such as the one in the right-hand side of (5), or require that context parameters are never copied. Syntactic restrictions

[23] It turns out that dually moving also ∘-symbols does not add strength to the analysis.

of mtts to these effects are (either of) basicness and context-linearity from Definition 2.5; the proper combination with recursion-linearity to guarantee a safe application of $\downarrow_{\delta,j}$-rules is given by what we dubbed **atmost** there. Completely dually, M_2 must be **atleast** in order to apply $\uparrow_{\delta,j}$-rules safely. In the case of $\downarrow_{f,k}$- and $\uparrow_{f,k}$-rules for a state $f \in F$, we additionally need to require context-linearity and -nondeletingness, respectively, of the first mtt to prevent unlegitimate proliferation of $\bullet$-symbols in the annotated intermediate result already. The following definition captures the gained insights by specifying a permissible subset $\mathcal{E}_{M_1,M_2}$ of the possible rewrite rules for moving $\bullet$-symbols and eliminating $\circ$-symbols.

Definition 4.27 ($\mathcal{E}_{M_1,M_2}$)
The rewrite system $\mathcal{E}_{M_1,M_2}$ contains the following rules:

- $\bullet\ (\circ\ v) \to v$,
- $\circ\ (\bullet\ v) \to v$,
- $\downarrow_{\delta,j}$ for every $\delta \in \Delta^{(q)}$ and $j \in [q]$ if M_2 is atmost,
- $\uparrow_{\delta,j}$ for every $\delta \in \Delta^{(q)}$ and $j \in [q]$ if M_2 is atleast,
- $\downarrow_{f,k}$ for every $f \in F^{(r+1)}$ and $k \in [r]$ if M_1 is context-linear and M_2 is atmost, and
- $\uparrow_{f,k}$ for every $f \in F^{(r+1)}$ and $k \in [r]$ if M_1 is context-nondeleting and M_2 is atleast. □

Example 4.28 ($\mathcal{E}_{M_{spine},M_{pfx}}$, cf. Example 2.2)
Since M_{spine} is context-linear and context-nondeleting and M_{pfx} is atmost and atleast, the rewrite system $\mathcal{E}_{M_{spine},M_{pfx}}$ contains all the following rules:

$$
\begin{array}{lll@{\qquad}lll}
 & \bullet\ (\circ\ v) & \to v & \uparrow_{\alpha,1} : & \alpha\ (\bullet\ v_1) & \to \bullet\ (\alpha\ v_1) \\
 & \circ\ (\bullet\ v) & \to v & \uparrow_{\beta,1} : & \beta\ (\bullet\ v_1) & \to \bullet\ (\beta\ v_1) \\
\downarrow_{\delta,1} : & \bullet\ (\delta\ v_1\ v_2) & \to \delta\ (\bullet\ v_1)\ v_2 & \downarrow_{f_1,1} : & \bullet\ (f_1\ u\ v_1) & \to f_1\ u\ (\bullet\ v_1) \\
\downarrow_{\delta,2} : & \bullet\ (\delta\ v_1\ v_2) & \to \delta\ v_1\ (\bullet\ v_2) & \downarrow_{f_2,1} : & \bullet\ (f_2\ u\ v_1\ v_2) & \to f_2\ u\ (\bullet\ v_1)\ v_2 \\
\downarrow_{\alpha,1} : & \bullet\ (\alpha\ v_1) & \to \alpha\ (\bullet\ v_1) & \downarrow_{f_2,2} : & \bullet\ (f_2\ u\ v_1\ v_2) & \to f_2\ u\ v_1\ (\bullet\ v_2) \\
\downarrow_{\beta,1} : & \bullet\ (\beta\ v_1) & \to \beta\ (\bullet\ v_1) & \uparrow_{f_1,1} : & f_1\ u\ (\bullet\ v_1) & \to \bullet\ (f_1\ u\ v_1) \\
\uparrow_{\delta,1} : & \delta\ (\bullet\ v_1)\ v_2 & \to \bullet\ (\delta\ v_1\ v_2) & \uparrow_{f_2,1} : & f_2\ u\ (\bullet\ v_1)\ v_2 & \to \bullet\ (f_2\ u\ v_1\ v_2) \\
\uparrow_{\delta,2} : & \delta\ v_1\ (\bullet\ v_2) & \to \bullet\ (\delta\ v_1\ v_2) & \uparrow_{f_2,2} : & f_2\ u\ v_1\ (\bullet\ v_2) & \to \bullet\ (f_2\ u\ v_1\ v_2)
\end{array}
$$
□

We will later allow the rewrite rules from $\mathcal{E}_{M_1,M_2}$ to be applied arbitrarily often to rule right-hand sides, but also to the initial expression, of annotated versions of M_1. For the sake of proving that this indeed does not lead to an underestimation of the difference between the numbers of $\circ$- and $\bullet$-symbols in the final output, however, it is more convenient to first consider the application of at most one rule from $\mathcal{E}_{M_1,M_2}$ to the initial expression and each right-hand side, though not necessarily the same rule everywhere. This is described by a relation $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}$ on a set containing all $M_1^{\to\circ\bullet,\kappa}$ and all mtts possibly obtained from them via the rewrite rules in question.

Definition 4.29 ($\mathcal{M}_1^{\to\circ\bullet}$ and $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}} \subseteq \mathcal{M}_1^{\to\circ\bullet} \times \mathcal{M}_1^{\to\circ\bullet}$)
Let $\mathcal{M}_1^{\to\circ\bullet}$ be the set of all mtts $M = (F, \Sigma, \Delta \cup \{\circ, \bullet\}, e, R)$ for which $e[\circ, \bullet \twoheadleftarrow id, id] = e_{M_1}$ and $rhs_{M,f,\sigma}[\circ, \bullet \twoheadleftarrow id, id] = rhs_{M_1,f,\sigma}$ for every $f \in F$ and $\sigma \in \Sigma$. Note that every $M_1^{\to\circ\bullet} \in \mathcal{M}_1^{\to\circ\bullet}$ is context-linear iff M_1 is; likewise for context-nondeletingness. For mtts $M_1^{\to\circ\bullet} = (F, \Sigma, \Delta \cup \{\circ, \bullet\}, e_{M_1^{\to\circ\bullet}}, R_1^{\to\circ\bullet})$ and $M_1^{\to\circ\bullet\prime} = (F, \Sigma, \Delta \cup \{\circ, \bullet\}, e_{M_1^{\to\circ\bullet\prime}}, R_1^{\to\circ\bullet\prime})$ in

$\mathcal{M}_1^{\to\circ\bullet}$ we write $M_1^{\to\circ\bullet} \rightsquigarrow_{\mathcal{E}_{M_1,M_2}} M_1^{\to\circ\bullet\prime}$ if $e_{M_1^{\to\circ\bullet}} \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} e_{M_1^{\to\circ\bullet\prime}}$ and $rhs_{M_1^{\to\circ\bullet},f,\sigma} \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} rhs_{M_1^{\to\circ\bullet\prime},f,\sigma}$ for every $f \in F$ and $\sigma \in \Sigma$. ☐

Example 4.30 (rewriting $M_{spine}^{\to\circ\bullet,\kappa}$ from Example 4.21 with $\rightsquigarrow_{\mathcal{E}_{M_{spine},M_{pfx}}}$)
By rewriting the initial expression and each right-hand side of a rule of $M_{spine}^{\to\circ\bullet,\kappa}$ with at most one of the rules given in Example 4.28, we obtain $M_{spine}^{\to\circ\bullet,\kappa} \rightsquigarrow_{\mathcal{E}_{M_{spine},M_{pfx}}} M_{spine}^{\to\circ\bullet\prime} = (\{f_1^{(2)}, f_2^{(3)}\}, \Sigma_{mon}, \Delta_{tree} \cup \{\circ, \bullet\}, f_1\ u_1\ (\circ\ (\bullet\ \epsilon)), R_{spine}^{\to\circ\bullet\prime})$ with rules

$$
\begin{array}{lllll}
R_{spine}^{\to\circ\bullet\prime}: & f_1\ (\alpha\ u_1) & y_1 & \to & \bullet\ (\alpha\ (f_1\ u_1\ (\circ\ y_1))) \\
 & f_1\ (\beta\ u_1) & y_1 & \to & f_2\ u_1\ y_1\ (\beta\ (\bullet\ \epsilon)) \\
 & f_1\ \epsilon & y_1 & \to & y_1 \\
 & f_2\ (\alpha\ u_1) & y_1\ y_2 & \to & \bullet\ (\alpha\ (f_1\ u_1\ (\circ\ (\delta\ (\bullet\ (\circ\ y_1))\ y_2)))) \\
 & f_2\ (\beta\ u_1) & y_1\ y_2 & \to & f_2\ u_1\ (\circ\ y_1)\ (\circ\ (\bullet\ (\beta\ (\bullet\ (\beta\ y_2))))) \\
 & f_2\ \epsilon & y_1\ y_2 & \to & \delta\ (\bullet\ (\circ\ y_1))\ y_2
\end{array}
$$
☐

To show that rewriting an annotated version of M_1 according to $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}$ does not lead to a decrease in the difference between the numbers of ∘- and •-symbols reproduced by $M_2^{\circ\bullet\to\circ\bullet}$ from the annotated intermediate result, we first need to prove two lemmata.

The first one captures how the omission of •-symbols from an annotated intermediate result is reflected in the final output produced by a state of an atmost and of an atleast M_2, respectively.[24] In fact, the properties given here motivated the choice of names "atmost" and "atleast".

Lemma 4.31 (essential properties of atmost and atleast mtts, respectively)
For every $\theta, \theta' \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'$, $g \in G^{(s+1)}$, and $\eta_1, \ldots, \eta_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$:

1. if M_2 is atmost, then:
$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta'\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \leq |\theta|_\bullet - |\theta'|_\bullet$
2. if M_2 is atleast, then:
$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta'\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \geq |\theta|_\bullet - |\theta'|_\bullet$. ☐

The proof (of more general statements by induction on θ, using a nested induction over the structure of right-hand sides of rules of M_2, as well as Lemmata 2.7 and 2.8 for the mtt $M_2^{\circ\bullet\to\circ\bullet}$) can be found in the appendix. It uses $|\,.\,|_{\bullet-\circ} = |\,.\,|_\bullet - |\,.\,|_\circ$ instead of $|\,.\,|_{\circ-\bullet}$ to later enable a "recycling" of the performed inductions for the proof of Lemma 4.37. Another dimension of proof reuse is that again, as in the case of Lemma 2.8, we only need to explicitly prove the first statement, whereas the proof of the second one is completely dual (in the sense in which the definition of "atleast" is dual to that of "atmost" by replacing each of recursion- and context-linearity by the corresponding nondeletingness).

The previous lemma is used in the following one to establish in which sense the ↑- and ↓-rules we have chosen to put into $\mathcal{E}_{M_1,M_2}$ are permissible.

Lemma 4.32 (essential properties of the rewrite system $\mathcal{E}_{M_1,M_2}$)
For every $g \in G^{(s+1)}$ and $\eta_1, \ldots, \eta_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$:

[24] The different orders in which θ and θ' appear in the left- and right-hand sides of the given inequations might appear counter-intuitive on first sight. This is caused by the fact that •-symbols are counted negatively by $|\,.\,|_{\circ-\bullet}$ but positively by $|\,.\,|_\bullet$.

1. for every $\delta \in \Delta^{(q)}$, $j \in [q]$, and $\theta_1, \ldots, \theta_q \in T_{\Delta \cup \{\circ,\bullet\}}$:

   (a) if $\downarrow_{\delta,j} \in \mathcal{E}_{M_1,M_2}$, then:
   $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta_1 \cdots \theta_q)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$
   $\leq 1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta_1 \cdots (\bullet\ \theta_j) \cdots \theta_q)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

   (b) if $\uparrow_{\delta,j} \in \mathcal{E}_{M_1,M_2}$, then:
   $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta_1 \cdots \theta_q)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$
   $\geq 1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta_1 \cdots (\bullet\ \theta_j) \cdots \theta_q)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

2. for every $M_1^{\to\circ\bullet} = (F, \Sigma, \Delta \cup \{\circ, \bullet\}, e_{M_1^{\to\circ\bullet}}, R_1^{\to\circ\bullet}) \in \mathcal{M}_1^{\to\circ\bullet}$, $f \in F^{(r+1)}$, $k \in [r]$, $t \in T_\Sigma$, and $\theta_1, \ldots, \theta_r \in T_{\Delta \cup \{\circ,\bullet\}}$:

   (a) if $\downarrow_{f,k} \in \mathcal{E}_{M_1,M_2}$, then:
   $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots \theta_r)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$
   $\leq 1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots (\bullet\ \theta_k) \cdots \theta_r)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

   (b) if $\uparrow_{f,k} \in \mathcal{E}_{M_1,M_2}$, then:
   $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots \theta_r)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$
   $\geq 1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots (\bullet\ \theta_k) \cdots \theta_r)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$.

Proof
Since from $\downarrow_{\delta,j} \in \mathcal{E}_{M_1,M_2}$ follows that M_2 is atmost, and from $\uparrow_{\delta,j} \in \mathcal{E}_{M_1,M_2}$ follows that M_2 is atleast, the points 1a and 1b follow from Lemma 4.31 with $\theta = \delta\ \theta_1 \cdots (\bullet\ \theta_j) \cdots \theta_q$ and $\theta' = \delta\ \theta_1 \cdots \theta_q$, using that $\theta \Rightarrow_{\{\bullet\ v \to v\}} \theta'$ and $|\theta|_\bullet - |\theta'|_\bullet = 1$.

Since from $\downarrow_{f,k} \in \mathcal{E}_{M_1,M_2}$ follows that M_1 is context-linear and M_2 is atmost, and from $\uparrow_{f,k} \in \mathcal{E}_{M_1,M_2}$ follows that M_1 is context-nondeleting and M_2 is atleast, the points 2a and 2b follow from Lemma 4.31 with $\theta = nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots (\bullet\ \theta_k) \cdots \theta_r)$ and $\theta' = nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots \theta_r)$, using three auxiliary statements:

1. $\theta \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'$,
2. $|\theta|_\bullet - |\theta'|_\bullet \leq 1$ if M_1, and hence also $M_1^{\to\circ\bullet}$, is context-linear, and
3. $|\theta|_\bullet - |\theta'|_\bullet \geq 1$ if M_1, and hence also $M_1^{\to\circ\bullet}$, is context-nondeleting.

These statements, in turn, follow immediately from the following equalities for θ and θ' (which hold because the $\theta_1, \ldots, \theta_r \in T_{\Delta \cup \{\circ,\bullet\}}$ are normal forms w.r.t. $\Rightarrow_{R_1^{\to\circ\bullet}}$):

$$\begin{aligned} \theta &= nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ y_1 \cdots y_r)[y_1, \ldots, y_r \leftarrow \theta_1, \ldots, \bullet\ \theta_k, \ldots, \theta_r] \\ \theta' &= nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ y_1 \cdots y_r)[y_1, \ldots, y_r \leftarrow \theta_1, \ldots, \theta_r] \end{aligned}$$

and Lemma 2.8 for the mtt $M_1^{\to\circ\bullet}$. □

Now, we are in a position to prove that rewriting according to $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}$ does not lead to an unlegitimate underestimation of the number of $\circ$-symbols or overestimation of the number of $\bullet$-symbols.

Lemma 4.33 (applying $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}$ leads to a safe approximation)
If $M_1^{\to\circ\bullet} \rightsquigarrow_{\mathcal{E}_{M_1,M_2}} M_1^{\to\circ\bullet\prime}$, then:

$$\tau_{M_1^{\to\circ\bullet}} \,;\, \tau_{M_2^{\circ\bullet\to\circ\bullet}} \,;\, |\,.\,|_{\circ-\bullet} \leq \tau_{M_1^{\to\circ\bullet\prime}} \,;\, \tau_{M_2^{\circ\bullet\to\circ\bullet}} \,;\, |\,.\,|_{\circ-\bullet}.$$

If for every $\downarrow$- or $\uparrow$-rule used for rewriting the initial expression or right-hand sides of rules of $M_1^{\to\circ\bullet}$, $\mathcal{E}_{M_1,M_2}$ also contains the symmetric rule, then equality holds. □

The proof (of a more general statement by induction over the structure of elements of T_Σ, using nested inductions over the structure of right-hand sides of rules of—and of the initial expression of—$M_1^{\to\circ\bullet}$, and over the structure of right-hand sides of rules of—and of the initial expression of—M_2, as well as Lemma 4.32) can be found in the appendix. At its heart is an analysis of the different ways in which a rule right-hand side or initial expression of $M_1^{\to\circ\bullet}$ can be rewritten (or not) using $\Rightarrow^?_{\mathcal{E}_{M_1,M_2}}$. This leads to eleven cases to consider, but a lot of reuse is possible in resolving the associated proof obligations. Another interesting reuse aspect is the proof of the statement about equality in case that only ⇃- or ↾-rules are used whose symmetric version is also contained in $\mathcal{E}_{M_1,M_2}$ (cf. the appendix for details).

Example 4.34 (instantiation of Lemma 4.33 to our running example)
To observe the effect of the concrete application of $\rightsquigarrow_{\mathcal{E}_{M_{spine},M_{pfx}}}$ in Example 4.30, compare the computation in Figure 11 to that from Figure 10. In particular, note how as a consequence of rewriting the right-hand side of the rule for f_2 at ϵ, one $\bullet$-symbol has "floated downwards" in the computed output, as indicated by the dashed arrow, and how the removal of adjacent $\circ$- and $\bullet$-symbols in the right-hand side of the rule for f_1 at β has lead to the elimination of one such pair also from the computed output at the $\times$-marked position. In fact, the difference between the numbers of $\circ$- and $\bullet$-symbols is the same in both outputs, as expected since $\mathcal{E}_{M_{spine},M_{pfx}}$ contains all potential ⇃- and ↾-rules. □

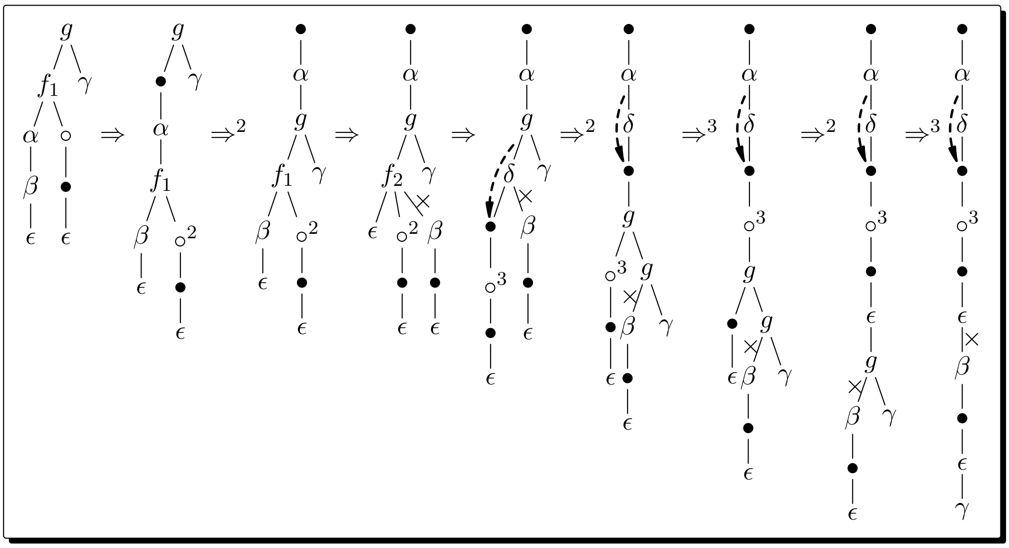

Figure 11: Reduction sequence using $\Rightarrow_{R_{spine}^{\to\circ\bullet\prime}\cup R_{pfx}^{\circ\bullet\to\circ\bullet}}$.

4.8 Constantly bounding the number of ∘-symbols

As motivated at the beginning of the previous subsection, we want to manipulate some $M_1^{\to\circ\bullet,\kappa}$ (by eliminating $\circ$-symbols via $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}$) until we obtain an $M_1^{\to\circ\bullet} \in \mathcal{M}_1^{\to\circ\bullet}$ for which we can put a constant upper bound on $|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\circ$ for every $t \in T_\Sigma$. The most natural approach to establishing such an upper bound is to first consider conditions under which the number of $\circ$-symbols in the indermediate result $\tau_{M_1^{\to\circ\bullet}}(t)$ is constantly bounded. Requiring that no $\circ$-symbols occur anymore in right-hand sides of rules of $M_1^{\to\circ\bullet}$

is not quite enough because there might still be $\circ$-symbols present in its initial expression. In that case, we additionally impose context-linearity to be sure that such $\circ$-symbols are not duplicated during the computation.

Lemma 4.35 (bounding the number of $\circ$-symbols produced by $M_1^{\to\circ\bullet}$)
Let $M_1^{\to\circ\bullet} = (F, \Sigma, \Delta \cup \{\circ, \bullet\}, e_{M_1^{\to\circ\bullet}}, R_1^{\to\circ\bullet}) \in \mathcal{M}_1^{\to\circ\bullet}$ be such that the right-hand sides of the rules in $R_1^{\to\circ\bullet}$ contain no $\circ$-symbols. If $|e_{M_1^{\to\circ\bullet}}|_\circ = 0$ or M_1 is context-linear, then for every $t \in T_\Sigma$:

$$|\tau_{M_1^{\to\circ\bullet}}(t)|_\circ \leq |e_{M_1^{\to\circ\bullet}}|_\circ.$$

If $|e_{M_1^{\to\circ\bullet}}|_\circ = 0$ or additionally M_1 is context-nondeleting, then equality holds. $\square$

The proof (of a more general statement by induction over the structure of elements of T_Σ, using a nested induction over the structure of right-hand sides of rules of—and of the initial expression of—$M_1^{\to\circ\bullet}$, as well as Lemmata 2.7 and 2.8 for the mtt $M_1^{\to\circ\bullet}$) can be found in the appendix. The key observation is that if the right-hand sides of the rules in $R_1^{\to\circ\bullet}$ contain no $\circ$-symbols, then any $\circ$-symbols appearing in the output of a call to a state from F can only originate from copies of that call's context parameters, i.e., we can always decompose as follows:

$$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots \theta_r)|_\circ = \sum_{k \in [r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ y_1 \cdots y_r)|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ.$$

Given a constant bound for the number of $\circ$-symbols in $\tau_{M_1^{\to\circ\bullet}}(t)$, we could obtain a corresponding bound for $\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))$ if we knew that $M_2^{\circ\bullet\to\circ\bullet}$ duplicates $\circ$-symbols found in its input by at most a constant factor. The latter is obviously the case if no such symbols are found in the input at all.

Lemma 4.36 ($M_2^{\circ\bullet\to\circ\bullet}$ does not create $\circ$-symbols out of nothing)
For every $\theta \in T_{\Delta\cup\{\circ,\bullet\}}$, if $|\theta|_\circ = 0$, then $|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\theta)|_\circ = 0$.

Proof
Since the initial expression $e_{M_2} \in RHS(G, \Omega, V_1, \emptyset)$ of $M_2^{\circ\bullet\to\circ\bullet}$ contains no $\circ$-symbols, such symbols in the output $\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\theta) = nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, e_{M_2}[v_1 \leftarrow \theta])$ could only originate from right-hand sides of rules in $R_2^{\circ\bullet\to\circ\bullet}$. But the only such right-hand sides containing $\circ$-symbols are those of the rules at input symbol $\circ$, which however are never applied if $|\theta|_\circ = 0$. $\square$

If $M_2^{\circ\bullet\to\circ\bullet}$'s input **does** contain $\circ$-symbols, then we want to be sure that each of them is reproduced in the output at most a bounded number of times. One reason for unbounded reproduction could be recursion-nonlinearity because then a given $\circ$-symbol could be reached arbitrarily many times, depending on its depth in the input tree. Another source for unbounded duplication could be that the output of a call that is nested inside a context parameter position of another call is duplicated when the outer call copies its parameters. This can be avoided by requiring basicness or context-linearity, just as we did in the previous subsection when discussing conditions for the safe applicability of $\downarrow_{\delta,j}$-rules. In fact, the restriction atmost is once again what we are looking for. It guarantees that each $\circ$-symbol found in the input is reproduced in the output at most as many times as the input is plugged into the initial expression of $M_2^{\circ\bullet\to\circ\bullet}$.

Lemma 4.37 (bounding the number of copies of ∘-symbols made by $M_2^{\circ\bullet\to\circ\bullet}$)
If M_2 is atmost, then for every $\theta \in T_{\Delta\cup\{\circ,\bullet\}}$:

$$|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\theta)|_\circ \leq |e_{M_2}|_{v_1} * |\theta|_\circ.$$

If additionally M_2 is atleast, then equality holds. □

The proof (by essentially "recycling" the inductions from the proof of Lemma 4.31 and additionally using Lemma 4.36 for $\theta[\circ \twoheadleftarrow id]$) can be found in the appendix. The close connection of this lemma to Lemma 4.31, which might not be apparent on first sight, stems from the fact that the omission of ∘-symbols from an input tree for an atmost and for an atleast $M_2^{\circ\bullet\to\circ\bullet}$, respectively, is reflected in the final output just in the same manner as in the case of •-symbols.

Together with Lemma 4.33, the previous three lemmata can under appropriate preconditions be used to safely approximate the difference between the numbers of ∘- and •-symbols reproduced by $M_2^{\circ\bullet\to\circ\bullet}$ from an intermediate result produced by $M_1^{\to\circ\bullet,\kappa}$ for an input tree t in terms of $|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\bullet$ and the initial expressions of $M_1^{\to\circ\bullet}$ and $M_2^{\circ\bullet\to\circ\bullet}$ for some $M_1^{\to\circ\bullet}$ obtained from $M_1^{\to\circ\bullet,\kappa}$ via applications of $\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}$. Since the still input-dependent, nonnegative value $|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\bullet$ occurs as subtrahend only, it can later be dropped.

Lemma 4.38 (the final puzzle piece)
Let $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\ldots,s_{max}\}$ and $M_1^{\to\circ\bullet} = (F, \Sigma, \Delta\cup\{\circ,\bullet\}, e_{M_1^{\to\circ\bullet}}, R_1^{\to\circ\bullet}) \in \mathcal{M}_1^{\to\circ\bullet}$ be such that $M_1^{\to\circ\bullet,\kappa} \rightsquigarrow^*_{\mathcal{E}_{M_1,M_2}} M_1^{\to\circ\bullet}$ and the right-hand sides of the rules in $R_1^{\to\circ\bullet}$ contain no ∘-symbols. If $|e_{M_1^{\to\circ\bullet}}|_\circ = 0$ or if M_1 is context-linear and M_2 is atmost, then for every $t \in T_\Sigma$:

$$|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet,\kappa}}(t))|_{\circ-\bullet} \leq |e_{M_2}|_{v_1} * |e_{M_1^{\to\circ\bullet}}|_\circ - |\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\bullet.$$

If for every ↓- or ↑-rule used in the rewriting process to obtain $e_{M_1^{\to\circ\bullet}}$ and $R_1^{\to\circ\bullet}$, $\mathcal{E}_{M_1,M_2}$ also contains the symmetric rule, and if $|e_{M_1^{\to\circ\bullet}}|_\circ = 0$ or additionally M_1 is context-nondeleting and M_2 is atleast, then equality holds.

Proof
From repeated applications of Lemma 4.33 follows by reflexivity and transitivity of $\leq$:

$$\tau_{M_1^{\to\circ\bullet,\kappa}} \,;\, \tau_{M_2^{\circ\bullet\to\circ\bullet}} \,;\, |\,.\,|_{\circ-\bullet} \leq \tau_{M_1^{\to\circ\bullet}} \,;\, \tau_{M_2^{\circ\bullet\to\circ\bullet}} \,;\, |\,.\,|_{\circ-\bullet}\,,$$

and hence for every $t \in T_\Sigma$:

$$|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet,\kappa}}(t))|_{\circ-\bullet} \leq |\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\circ - |\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\bullet.$$

If for every ↓- or ↑-rule used in the rewriting process to obtain $e_{M_1^{\to\circ\bullet}}$ and $R_1^{\to\circ\bullet}$, $\mathcal{E}_{M_1,M_2}$ also contains the symmetric rule, then the three occurrences of $\leq$ above can be replaced by $=$. Now, we reason as follows:

- If $|e_{M_1^{\to\circ\bullet}}|_\circ = 0$, then $|\tau_{M_1^{\to\circ\bullet}}(t)|_\circ = 0$ by Lemma 4.35, and as a consequence $|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\circ = 0 = |e_{M_2}|_{v_1} * |e_{M_1^{\to\circ\bullet}}|_\circ$ by Lemma 4.36 with $\theta = \tau_{M_1^{\to\circ\bullet}}(t)$.
- If M_1 is context-linear and M_2 is atmost, then:

$|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\circ$
$\quad\leq$ (by Lemma 4.37 with $\theta = \tau_{M_1^{\to\circ\bullet}}(t)$)
$|e_{M_2}|_{v_1} * |\tau_{M_1^{\to\circ\bullet}}(t)|_\circ$
$\quad\leq$ (by Lemma 4.35, using that the factor $|e_{M_2}|_{v_1} \geq 0$)
$|e_{M_2}|_{v_1} * |e_{M_1^{\to\circ\bullet}}|_\circ$

If additionally M_1 is context-nondeleting and M_2 is atleast, then Lemmata 4.35 and 4.37 give equalities instead of inequalities in the above calculation. □

Example 4.39 (further rewriting $M_{spine}^{\to\circ\bullet\prime}$ from Example 4.30 with $\rightsquigarrow^*_{\mathcal{E}_{M_{spine},M_{pfx}}}$)
By rewriting the initial expression and each right-hand side of a rule of $M_{spine}^{\to\circ\bullet\prime}$ with arbitrary many of the rules given in Example 4.28, we obtain $M_{spine}^{\to\circ\bullet\prime} \rightsquigarrow^*_{\mathcal{E}_{M_{spine},M_{pfx}}} M_{spine}^{\to\circ\bullet} = (\{f_1^{(2)}, f_2^{(3)}\}, \Sigma_{mon}, \Delta_{tree} \cup \{\circ, \bullet\}, e_{M_{spine}^{\to\circ\bullet}}, R_{spine}^{\to\circ\bullet})$ with $e_{M_{spine}^{\to\circ\bullet}} = f_1\ u_1\ \epsilon$ and rules

$$
\begin{array}{rllll}
R_{spine}^{\to\circ\bullet}: & f_1\ (\alpha\ u_1) & y_1 & \to & \alpha\ (f_1\ u_1\ y_1) \\
& f_1\ (\beta\ u_1) & y_1 & \to & f_2\ u_1\ y_1\ (\beta\ (\bullet\ \epsilon)) \\
& f_1\ \epsilon & y_1 & \to & y_1 \\
& f_2\ (\alpha\ u_1) & y_1\ y_2 & \to & \alpha\ (f_1\ u_1\ (\delta\ y_1\ y_2)) \\
& f_2\ (\beta\ u_1) & y_1\ y_2 & \to & f_2\ u_1\ y_1\ (\beta\ (\beta\ y_2)) \\
& f_2\ \epsilon & y_1\ y_2 & \to & \delta\ y_1\ y_2
\end{array}
$$

For illustration, Figure 12 gives a sequence of rewritings performed on the right-hand side of the rule for f_2 at β.

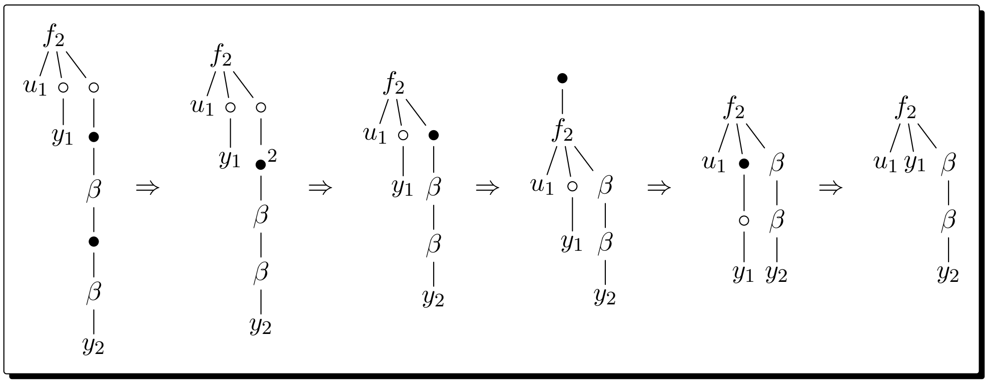

Figure 12: Rewriting $rhs_{M_{spine}^{\to\circ\bullet,\kappa},f_2,\beta} = rhs_{M_{spine}^{\to\circ\bullet\prime},f_2,\beta}$ using $\Rightarrow_{\mathcal{E}_{M_{spine},M_{pfx}}}$.

As predicted by Lemma 4.38, given that $|e_{M_{spine}^{\to\circ\bullet}}|_\circ = 0$ and that $\mathcal{E}_{M_{spine},M_{pfx}}$ contains all potential $\downarrow$- and $\uparrow$-rules, the negative number of $\bullet$-symbols in the output computed in Figure 13 using the above rules is equal to the difference of the numbers of $\circ$- and $\bullet$-symbols in the output computed in Figure 10. □

Finally, we can prove the main theorem of our efficiency analysis.

Theorem 4.40 (putting together the pieces)
If M_2 is context-linear and it exists a mapping $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\dots,s_{max}\}$ such that

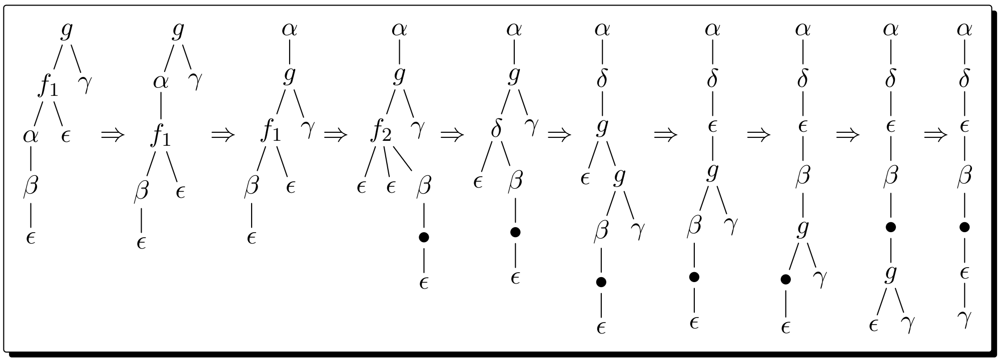

Figure 13: Reduction sequence using $\Rightarrow_{R^{\to\circ\bullet}_{spine} \cup R^{\circ\bullet\to\circ\bullet}_{pfx}}$.

- the right-hand sides of all rules in $R_1^{\to\circ\bullet,\kappa}$ can be rewritten with $\Rightarrow^*_{\mathcal{E}_{M_1,M_2}}$ such that all $\circ$-symbols are eliminated, and
- $e_{M_1^{\to\circ\bullet,\kappa}}$ can be rewritten with $\Rightarrow^*_{\mathcal{E}_{M_1,M_2}}$ such that all $\circ$-symbols are eliminated, or M_1 is context-linear and M_2 is atmost,

then there exists a constant $d \in \mathbb{N}$ such that $cbn_{\overline{M_1;M_2}}(t) \leq cbn_{M_1;M_2}(t) + d$ for every $t \in T_\Sigma$.

Proof
First, we recall

$$cbn_{\overline{M_1;M_2}} - cbn_{M_1;M_2} = \tau_{M_1^{\to\star\bullet}} ; \tau_{M_2^{\star\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet}$$

from the proof of Theorem 4.19. Since M_2 is context-linear, we have

$$\tau_{M_1^{\to\star\bullet}} ; \tau_{M_2^{\star\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet} \leq \tau_{M_1^{\to\circ\bullet,\kappa}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet}$$

for every $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\ldots,s_{max}\}$ by Lemma 4.24, i.e.,

$$cbn_{\overline{M_1;M_2}}(t) - cbn_{M_1;M_2}(t) \leq |\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet,\kappa}}(t))|_{\circ-\bullet}$$

for every such κ and every $t \in T_\Sigma$. The statements about the particular κ whose existence is a precondition of the theorem imply that there is an mtt $M_1^{\to\circ\bullet} = (F, \Sigma, \Delta \cup \{\circ,\bullet\}, e_{M_1^{\to\circ\bullet}}, R_1^{\to\circ\bullet})$ meeting the requirements of Lemma 4.38. As a consequence, we obtain

$$|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet,\kappa}}(t))|_{\circ-\bullet} \leq |e_{M_2}|_{v_1} * |e_{M_1^{\to\circ\bullet}}|_\circ - |\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\bullet$$

for every $t \in T_\Sigma$. Since $|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\tau_{M_1^{\to\circ\bullet}}(t))|_\bullet$ is always nonnegative, the previous two inequations imply, with $d = |e_{M_2}|_{v_1} * |e_{M_1^{\to\circ\bullet}}|_\circ$,

$$cbn_{\overline{M_1;M_2}}(t) - cbn_{M_1;M_2}(t) \leq d$$

for every $t \in T_\Sigma$. $\square$

Example 4.41 (call-by-name analysis outcome for our running example) Given that M_{pfx} is context-linear, and in the light of the rule right-hand sides and the initial expression of the mtt $M_{spine}^{\to\circ\bullet}$ from Example 4.39, the previous theorem yields that there exists a constant $d \in \mathbb{N}$ such that $cbn_{\overline{M_{spine};M_{pfx}}}(t) \leq cbn_{M_{spine};M_{pfx}}(t) + d$ for every $t \in T_{\Sigma_{mon}}$. In fact, the constant here is 0 because $|e_{M_{spine}^{\to\circ\bullet}}|_\circ = 0$. Since for the mtts M_{spine} and M_{pfx} the Lemmata 4.24 and 4.38 yield equalities instead of inequalities only (as observed in Examples 4.25 and 4.39), we could even conclude the stronger statement that for a given input t the composed program is more efficient than the original program by exactly $|\tau_{M_{pfx}^{\circ\bullet\to\circ\bullet}}(\tau_{M_{spine}^{\to\circ\bullet}}(t))|_\bullet$ steps. This is witnessed, for a concrete input, by the lengths of the call-by-name reduction sequences in Figures 3 and 4 on the one hand and the output computed in Figure 13 on the other hand. By analyzing $|\tau_{M_{pfx}^{\circ\bullet\to\circ\bullet}}(\tau_{M_{spine}^{\to\circ\bullet}}(t))|_\bullet$, it is also not difficult to see that in general the amount of improvement corresponds to the number of β-blocks in the input tree (i.e. k in a decomposition as in Figure 1). $\square$

Note that in the course of our analysis up to this point, we have in a sense “tunneled” all uses of context-linearity through Lemma 2.8. This means that if we were to use a syntactically more flexible restriction of mtts that still guarantees point 1 of Lemma 2.8, then Theorem 4.40 would still hold with respect to this generalized notion of “context-linear”.[25] In [EM99] the desired property is called *finite copying in the parameters with parameter copying bound* 1, and following the strategy from the proof of Lemma 4.10(ii) in [EM03b] it is decidable for an mtt whether or not it has the property. While the applicability of Construction 3.1 itself would clearly not be increased through such a transition from “static” to “dynamic” context-linearity (because the use of “context-linear” in Construction 3.1 explicitly refers to the syntactic notion from Definition 2.5), the class of programs—among those to which Construction 3.1 applies—for which our analysis guarantees efficiency nondeterioration could be enlarged by more than it might seem at first glance. This is so because also the definition of “atmost” could refer to the generalized notion of context-linearity without requiring any new proof to preserve Theorem 4.40, and so could the definition of $\mathcal{E}_{M_1,M_2}$. The other lemmata from group (iv) in Figure 5 open similar possibilities for generalizing the analysis results. In particular, alternative definitions of “atmost” and “atleast” complying to Lemmata 4.31 and 4.37 would also be acceptable, as would a more liberal inclusion of $\uparrow$- and $\downarrow$-rules into $\mathcal{E}_{M_1,M_2}$ so long as Lemma 4.32 holds.

In the next subsection we study additional preconditions on M_1 and M_2 under which the results from Theorems 4.19 and 4.40 also hold when considering call-by-need rather than call-by-name reduction.

4.9 From call-by-name to call-by-need

Call-by-need differs from call-by-name reduction by representing terms as directed acyclic graphs rather than as trees to avoid—through sharing—repeated evaluation of the same subexpression. This improvement comes to bear only in situations where a call-by-name step would duplicate a subexpression containing a function call by substituting it for several occurrences of the same variable in the right-hand side of a rewrite rule. The

[25] The only thing that needs to be established additionally is that for such a generalized notion it would still be the case that context-linearity of M_1 implies that the inequation from point 1 of Lemma 2.8 holds also for every mtt in $\mathcal{M}_1^{\to\circ\bullet}$, and correspondingly so for M_2 and $M_2^{\circ\bullet\to\circ\bullet}$. But this is pretty obvious.

standard way to transfer insights about call-by-name efficiency to the setting of call-by-need is thus to require all programs to be linear. Essentially, this is what [Küh99] does, and it also underlies accounts of efficiency nondeterioration for classical deforestation.

Due to the known particular recursion scheme of mtts, we can do a bit better. First, we describe shapes of well-formed terms for the original program which ensure that in a call's context parameters no further calls occur unless context-linearity of the corresponding mtt guarantees that they are not duplicated during its reduction.

Definition 4.42 (WFT'_{M_1} and $WFT'_{M_1;M_2}$)
The set WFT'_{M_1} of *well-formed terms for M_1 without duplicable calls in context parameters* is the smallest set $WFT' \subseteq WFT_{M_1}$ satisfying the following conditions:

- for every $f \in F^{(r+1)}$, $t \in T_\Sigma$, $\phi_1, \ldots, \phi_r \in WFT'$, if $\phi_1, \ldots, \phi_r \in T_\Delta$ or M_1 is context-linear, then: $(f\ t\ \phi_1 \cdots \phi_r) \in WFT'$, and
- for every $\delta \in \Delta^{(q)}$ and $\phi_1, \ldots, \phi_q \in WFT'$: $(\delta\ \phi_1 \cdots \phi_q) \in WFT'$.

The set $WFT'_{M_1;M_2}$ of *well-formed terms for the original program without duplicable calls in context parameters* is the smallest set $WFT' \subseteq WFT_{M_1;M_2}$ satisfying the following conditions:

- for every $g \in G^{(s+1)}$, $\phi \in WFT'_{M_1}$, $\varrho_1, \ldots, \varrho_s \in WFT'$, if $\varrho_1, \ldots, \varrho_s \in T_\Omega$ or M_2 is context-linear, then: $(g\ \phi\ \varrho_1 \cdots \varrho_s) \in WFT'$, and
- for every $\omega \in \Omega^{(n)}$ and $\varrho_1, \ldots, \varrho_n \in WFT'$: $(\omega\ \varrho_1 \cdots \varrho_n) \in WFT'$. $\square$

The following lemma shows that during call-by-name reduction of $e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]]$ for some $t \in T_\Sigma$ only this particular kind of well-formed terms can occur if each of the involved mtts is context-linear or basic.

Lemma 4.43 ()
If M_1 is context-linear or basic and M_2 is context-linear or basic, then:

1. for every $t \in T_\Sigma$ we have $e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]] \in WFT'_{M_1;M_2}$, and
2. for every $\varrho \in WFT'_{M_1;M_2}$ and $\varrho' \in WFT_{M_1;M_2}$ with $\varrho \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \varrho'$ we have $\varrho' \in WFT'_{M_1;M_2}$.

Proof

1. This is immediate from the structure of the inititial expressions of the involved mtts, in particular that initial expressions of basic mtts do not contain nested calls.
2. This is shown by a straightforward induction over the structure of elements of $WFT'_{M_1;M_2}$ (with nested inductions over the structure of right-hand sides of rules of M_1 and M_2), using the definition of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ and that right-hand sides of rules of basic mtts do not contain nested calls. $\square$

From the previous lemma and the definitions of "atmost" and $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ follows that for context-linear or basic M_1 and atmost M_2 every rewriting taking place (possibly deep inside some term) in the reduction sequence

$$e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]] \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^* \tau_{M_2}(\tau_{M_1}(t))$$

has one of the following two forms (with $f \in F^{(r+1)}$, $\sigma \in \Sigma^{(p)}$, $g \in G^{(s+1)}$, and $\delta \in \Delta^{(q)}$):

– $f\ (\sigma\ t_1 \cdots t_p)\ \phi_1 \cdots \phi_r \rightarrow rhs_{M_1,f,\sigma}[u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\phi_1,\ldots,\phi_r]$, where
  - $t_1,\ldots,t_p \in T_\Sigma$ and
  - $\phi_1,\ldots,\phi_r \in T_\Delta$ or $|rhs_{M_1,f,\sigma}|_{y_k} \leq 1$ for every $k \in [r]$, or

– $g\ (\delta\ \phi_1 \cdots \phi_q)\ \varrho_1 \cdots \varrho_s \rightarrow rhs_{M_2,g,\delta}[v_1,\ldots,v_q,z_1,\ldots,z_s \leftarrow \phi_1,\ldots,\phi_q,\varrho_1,\ldots,\varrho_s]$, where
  - $|rhs_{M_2,g,\delta}|_{v_j} \leq 1$ for every $j \in [q]$
  - $\varrho_1,\ldots,\varrho_s \in T_\Omega$ or $|rhs_{M_2,g,\delta}|_{z_l} \leq 1$ for every $l \in [s]$.

It is thus easy to see that for such mtts no subexpression containing a call to some state is ever duplicated during the above call-by-name reduction sequence. Therefore, the number of reduction steps in this sequence, $cbn_{M_1;M_2}(t)$, does not differ from the one that would be obtained under a call-by-need strategy. Since, moreover, a call-by-need strategy always (in particular for the composed program) performs at most as many reduction steps as a call-by-name strategy, a combination of Theorems 4.19 and 4.40 yields the following.

Theorem 4.44 (conditions for call-by-need efficiency nondeterioration)
If

– M_1 is context-linear or basic,

– M_2 is atmost,

– M_1 is a tdtt or M_2 is context-linear,

and it exists a mapping $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\ldots,s_{max}\}$ such that

– the right-hand sides of all rules in $R_1^{\rightarrow\circ\bullet,\kappa}$ can be rewritten with $\Rightarrow^*_{\mathcal{E}_{M_1,M_2}}$ such that all $\circ$-symbols are eliminated, and

– $e_{M_1^{\rightarrow\circ\bullet,\kappa}}$ can be rewritten with $\Rightarrow^*_{\mathcal{E}_{M_1,M_2}}$ such that all $\circ$-symbols are eliminated, or M_1 is context-linear,

then there exists a constant $d \in \mathbb{N}$ such that for every input the composed program performs not more call-by-need steps than d more than the original program does. □

Example 4.45 (call-by-need analysis outcome for our running example)
Given that M_{spine} is context-linear and M_{pfx} is linear, and in the light of the rule right-hand sides of the mtt $M_{spine}^{\rightarrow\circ\bullet}$ from Example 4.39, the previous theorem yields that there exists a constant $d \in \mathbb{N}$ such that for every input $\overline{M_{spine};M_{pfx}}$ performs not more call-by-need steps than d more than the program consisting of expression $e_{M_{pfx}}[v_1 \leftarrow e_{M_{spine}}]$ and rules $R_{spine} \cup R_{pfx}$ does. Following the discussion from Example 4.41, the constant here is 0 and we could even conclude the stronger statement that the composed program is more efficient than the original program by at least[26] as many steps as there are β-blocks in the input tree. □

[26] We would have "exactly" instead of "at least" here if we convinced ourselves that for $\overline{M_{spine};M_{pfx}}$ there is no difference between the numbers of call-by-name and call-by-need steps, respectively, on every input.

Note that Theorem 4.44 in particular indicates a call-by-need efficiency nondeterioration if M_1 is a tdtt and M_2 is atmost, or if M_1 is context-linear or basic and M_2 is a linear tdtt.[27] It is easy to see—from Theorem 4.19—that actually $d = 0$ in those two special cases. They correspond to Theorem 1 of [Voi02a], except for the different formalization of initial expressions in that paper. The latter also contains results obtained by only considering the steps performed by the **second** mtt in the original program, which allows weaker restrictions than context-linearity or basicness of M_1 and atmostness of M_2 to still guarantee, using a different annotation, that call-by-need steps are counted. The same approach could be followed if neither of the involved mtts is a tdtt. We refrain from doing so here, and only provide the outcome for the special case from [Voi02a].

Theorem 4.46 (essentially Theorem 2 of [Voi02a])
If either of M_1 and M_2 is a tdtt, M_2 is context-linear or basic, $e_{M_1} = f\ u_1\ \theta_1 \cdots \theta_r$ and $e_{M_2} = g\ v_1\ \eta_1 \cdots \eta_s$ for some $f \in F^{(r+1)}$, $g \in G^{(s+1)}$, $\theta_1, \ldots, \theta_r \in T_\Delta$, and $\eta_1, \ldots, \eta_s \in T_\Omega$, and it exists a $\lambda \in \{0, 1\}$ such that the right-hand sides of all rules in

$$R_1^{\to\circ\bullet,\lambda}:\ f\ (\sigma\ u_1 \cdots u_p)\ y_1 \cdots y_r \to \circ^{(1-\lambda)}\ rhs_{M_1,f,\sigma} \qquad \forall f \in F^{(r+1)}, \sigma \in \Sigma^{(p)}$$
$$[\delta\ \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]$$
$$[f' \twoheadleftarrow \circ^\lambda \cdot f' \mid f' \in F]$$

can be rewritten with $\Rightarrow^*_{\mathcal{E}_{M_1,M_2}}$ such that all $\circ$-symbols are eliminated, then for every input the composed program performs not more call-by-need steps than the original program does. $\square$

Note that in [Voi02a] no $\downarrow_{f,k}$-rules are available to rewrite right-hand sides of rules in $R_1^{\to\circ\bullet,\lambda}$. But since by Definition 4.26 these rules are contained in $\mathcal{E}_{M_1,M_2}$ only if M_1 is context-linear and M_2 is atmost, this does not make a difference, given that under these conditions we already know—by specialization of Theorem 4.44 to the case that either of the involved mtts is a tdtt, see above—that the composed program never performs more call-by-need steps than the original program does.

Theorems 4.44 and 4.46 are not the end of the road. For example, the latter could be generalized by using a **mapping** λ from F to $\{0, 1\}$—similar to the parameterization of κ over context parameters of states of M_1 in Definition 4.20—, or one could indeed consider the approximation of counting only the steps of the second mtt in the original program, but therefor under weaker a priori restrictions, also for the general case of Construction 3.1. However, the theorems are already quite powerful in practice, both with respect to typical examples and in comparison to previous formal results regarding the effect of tree transducer composition on program efficiency (cf. Section 5.2).

4.10 Effectiveness and efficiency of the analysis

At the heart of applying Theorems 4.40, 4.44, and 4.46 to establish call-by-name or call-by-need efficiency nondeterioration for a particular application of our composition construction is the need to check, given a rewrite system $\mathcal{E}$ containing $\bullet\ (\circ\ v) \to v$, $\circ\ (\bullet\ v) \to v$, and some of the rules from Definition 4.26[28], whether certain $\circ$- and $\bullet$-annotated versions

[27] In both cases, there is only a trivial choice for the mapping κ because its source is an empty set if M_1 is a tdtt, while its target is a singleton set if M_2 is a tdtt. It is easy to see that in each case neither $e_{M_1^{\to\circ\bullet,\kappa}}$ nor the right-hand sides of rules in $R_1^{\to\circ\bullet,\kappa}$ contain any $\circ$-symbols.

[28] We consider an arbitrary $\mathcal{E}$ of this kind rather than the fixed $\mathcal{E}_{M_1,M_2}$ from Definition 4.27 for two reasons. Firstly, we want to allow for potential generalizations as discussed at the end of Section 4.8.

of rule right-hand sides and the initial expression of M_1 can be rewritten with $\Rightarrow_{\mathcal{E}}^*$ such that all $\circ$-symbols are eliminated.[29] That is, we have to be able to check the following predicate for some $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$:

$$\exists \phi' \in RHS(F, \Delta \cup \{\bullet\}, U, Y).\ \phi \Rightarrow_{\mathcal{E}}^* \phi' \tag{6}$$

Since in general $\Rightarrow_{\mathcal{E}}$ is neither confluent nor terminating, we cannot simply solve this task by computing a normal form with respect to it. However, given that none of the rewrite rules potentially in $\mathcal{E}$ introduces new symbols, we can enumerate all the (finitely many) trees to which a ϕ rewrites via $\Rightarrow_{\mathcal{E}}^*$ and check whether at least one of them is free of $\circ$-symbols. This means that the conditions from our theorems are decidable. Clearly, it does not yet mean that they can be checked efficiently, as would be desirable for inclusion into an optimizing compiler.

In the remainder of this subsection we develop a procedure deciding (6) in only a single traversal over ϕ, along with a sketched correctness proof. Actually, we will decide the following more general predicate for $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$ and $d \in \mathbb{Z}$:

$$clean_{\mathcal{E}}(\phi, d) = \exists \phi' \in RHS(F, \Delta \cup \{\bullet\}, U, Y).\ \begin{cases} \phi \Rightarrow_{\mathcal{E}}^* \bullet^d\ \phi' & \text{if } d > 0 \\ \bullet^{-d}\ \phi \Rightarrow_{\mathcal{E}}^* \phi' & \text{otherwise.} \end{cases}$$

Note that (6) coincides with $clean_{\mathcal{E}}(\phi, 0)$. Our aim is to characterize $clean_{\mathcal{E}}$ inductively, in terms of the structure of its first argument.

First, we observe some properties of $\Rightarrow_{\mathcal{E}}^*$. On the one hand, for every $\phi, \phi' \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$:

$$\bullet\ \phi \Rightarrow_{\mathcal{E}}^* \bullet\ \phi' \ \text{ iff } \ \phi \Rightarrow_{\mathcal{E}}^* \phi' \tag{7}$$

The "if"-direction is obvious, the "only-if"-direction is proved by induction on the length of reduction sequences, analyzing the first (if existing) rewrite step that involves the $\bullet$-symbol on top. On the other hand, for every $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$, $\phi' \in RHS(F, \Delta \cup \{\bullet\}, U, Y)$, and $n \in \mathbb{N}$:

$$\circ\ \phi \Rightarrow_{\mathcal{E}}^* \phi' \ \text{ iff } \ \phi \Rightarrow_{\mathcal{E}}^* \bullet\ \phi' \tag{8}$$

and

$$\bullet^{n+1}\ (\circ\ \phi) \Rightarrow_{\mathcal{E}}^* \phi' \ \text{ iff } \ \bullet^n\ \phi \Rightarrow_{\mathcal{E}}^* \phi' \tag{9}$$

In both cases the "if"-direction is obvious, while the "only-if"-direction requires some analysis of how the $\circ$-symbol on top of ϕ is eliminated.

For every $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$ and $d \in \mathbb{Z}$ we get

$$clean_{\mathcal{E}}(\bullet\ \phi, d) \ \text{ iff } \ clean_{\mathcal{E}}(\phi, d-1) \tag{10}$$

from (7), and

$$clean_{\mathcal{E}}(\circ\ \phi, d) \ \text{ iff } \ clean_{\mathcal{E}}(\phi, d+1) \tag{11}$$

from (8) and (9), respectively, by essentially only expanding the definition of $clean_{\mathcal{E}}$.

Secondly, as discussed in [Voi02a], we might consider dropping all $\uparrow$-rules to analyze efficiency in the case that the final output of the mtts to be composed is consumed only partially.

[29] Note that applying Theorem 4.19 requires only a simple syntactic check, namely whether either of M_1 and M_2 is a tdtt. Similarly, applying Theorems 4.40, 4.44, and 4.46 requires syntactic checks for some of the restrictions introduced in Definition 2.5. The quantifications over κ and λ, respectively, in these theorems cause no problems because in each case there is only a finite number of possibilities to check.

Things are a bit more complicated for the cases $\delta\ \phi_1 \cdots \phi_q$ and $f\ u_i\ \phi_1 \cdots \phi_r$. For every $\delta \in \Delta^{(q)}$, $\phi_1, \ldots, \phi_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$, and $n, m \in \mathbb{N}$, one can prove that

$$\exists \phi' \in RHS(F, \Delta \cup \{\bullet\}, U, Y).\ \bullet^n\ (\delta\ \phi_1 \cdots \phi_q) \Rightarrow^*_{\mathcal{E}} \bullet^m\ \phi'$$

iff

$$\begin{aligned} \exists d_1, \ldots, d_q \in \mathbb{Z}.\ m - n \leq \sum_{j \in [q]} d_j \wedge \forall j \in [q].\ & clean_{\mathcal{E}}(\phi_j, d_j) \\ & \wedge (\downarrow_{\delta,j} \in \mathcal{E} \vee 0 \leq d_j) \\ & \wedge (\uparrow_{\delta,j} \in \mathcal{E} \vee 0 \geq d_j). \end{aligned}$$

The "if"-direction is proved by producing a ϕ' with the requested properties by first rewriting all the ϕ_j in $\bullet^n\ (\delta\ \phi_1 \cdots \phi_q)$ with $d_j \geq 0$ as far as possible due to $clean_{\mathcal{E}}(\phi_j, d_j)$, then moving d_j $\bullet$-symbols up from all positions with $d_j > 0$, moving $-d_j$ $\bullet$-symbols down into all positions with $d_j < 0$, and finally eliminating all $\circ$-symbols in those positions, taking advantage of $clean_{\mathcal{E}}(\phi_j, d_j)$ for them. The "only-if"-direction is proved by induction on the length of reduction sequences, analyzing the first (if existing) rewrite step that involves the top-most δ-symbol. Additionally, the proof uses law (10) and the fact that for every $\phi, \phi' \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$ and $d \in \mathbb{Z}$, $\phi \Rightarrow^*_{\mathcal{E}} \phi'$ and $clean_{\mathcal{E}}(\phi', d)$ imply $clean_{\mathcal{E}}(\phi, d)$.

From the just proved equivalence we easily get a characterization of $clean_{\mathcal{E}}(\delta\ \phi_1 \cdots \phi_q, d)$ for $d \in \mathbb{Z}$, in terms of the equivalence's right-hand side with $m - n$ replaced by d. An analogous characterization of $clean_{\mathcal{E}}(f\ u_i\ \phi_1 \cdots \phi_r, d)$ is obtained likewise. Using these together with laws (10) and (11), we can prove (by a pretty straight-forward structural induction) that for every $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y)$ and $d \in \mathbb{Z}$,

$$clean_{\mathcal{E}}(\phi, d) \ \text{ iff } \ d \leq sup_{\mathcal{E}}(\phi)$$

holds, where the function $sup_{\mathcal{E}}$ is defined as follows:[30]

$$\begin{aligned} & sup_{\mathcal{E}} : RHS(F, \Delta \cup \{\circ, \bullet\}, U, Y) \longrightarrow \mathbb{Z} \cup \{-\infty\} \\ & sup_{\mathcal{E}}(y_k) &&= 0 \\ & sup_{\mathcal{E}}(\bullet\ \phi_1) &&= sup_{\mathcal{E}}(\phi_1) + 1 \\ & sup_{\mathcal{E}}(\circ\ \phi_1) &&= sup_{\mathcal{E}}(\phi_1) - 1 \\ & sup_{\mathcal{E}}(\delta\ \phi_1 \cdots \phi_q) &&= \sum_{j \in [q]} \begin{cases} -\infty & \text{if } \downarrow_{\delta,j} \notin \mathcal{E} \wedge 0 > sup_{\mathcal{E}}(\phi_j) \\ 0 & \text{if } \uparrow_{\delta,j} \notin \mathcal{E} \wedge 0 \leq sup_{\mathcal{E}}(\phi_j) \\ sup_{\mathcal{E}}(\phi_j) & \text{otherwise} \end{cases} \\ & sup_{\mathcal{E}}(f\ u_i\ \phi_1 \cdots \phi_r) &&= \sum_{k \in [r]} \begin{cases} -\infty & \text{if } \downarrow_{f,k} \notin \mathcal{E} \wedge 0 > sup_{\mathcal{E}}(\phi_k) \\ 0 & \text{if } \uparrow_{f,k} \notin \mathcal{E} \wedge 0 \leq sup_{\mathcal{E}}(\phi_k) \\ sup_{\mathcal{E}}(\phi_k) & \text{otherwise} \end{cases} \end{aligned}$$

Clearly, $sup_{\mathcal{E}}(\phi)$ can be computed in linear time with respect to the size of ϕ, so we have an efficient procedure deciding (6).

5 Comparison to related work

In [VK04a] the presented composition construction was compared with classical and short-cut deforestation on a qualitative level. Here we relate it to (other) composition techniques

[30] The usual interpretation of arithmetic operations and comparison relations on $\mathbb{Z} \cup \{-\infty\}$ is used, in particular $-\infty + a = -\infty$ for every $a \in \mathbb{Z} \cup \{-\infty\}$ and $d > -\infty$ for every $d \in \mathbb{Z}$.

for tree transducers (and for attribute grammars). We also discuss previous work on efficiency analysis for tree transducer composition, and for functional programs in general.

5.1 Tree transducer composition

Composition of tree transducers was first considered in [Rou70] for two tdtts, to be composed into a single one. Using a (reversible) decomposition of mtts into tdtts and certain substitution devices [Eng80], the composition of tdtts was reused in [Eng81] to compose a tdtt and an mtt (in this order) into a single mtt. A more direct construction for this composition was later given in [Küh99]. In [EV85] an mtt and a tdtt (in this order) are composed into a single mtt. Our Construction 3.1 generalizes all these composition results. More precisely, if both M_1 and M_2 are tdtts, then it corresponds to the construction in the proof of Theorem 2 in [Rou70]; if M_1 is a tdtt, then it corresponds to Transformation 11 in [Küh99]; and if M_2 is a tdtt, then it corresponds to the construction in the proof of Theorem 4.12 in [EV85].

The potential application of tree transducer composition for optimizing functional programs was first recognized in [Küh97] and [Küh98]. These were also the first works to deal with the composition of two mtts neither of which is a tdtt. This was done using an indirection via attributed tree transducers (for short *atts*). Essentially the same approach was independently developed in [CDPR98, CDPR99], though not in the syntactical framework of tree transducers. The key idea is to transform the functions from the original program—corresponding to restricted mtts—into atts or attribute grammars (either using a further indirection via tdtts and substitution devices, or using a direct construction in the spirit of [CF82]), to compose these using a construction based on ideas from [Fül81, Gan83, GG84, Gie88], and to transform the result back into an mtt using a construction from [Fra82]. Our direct construction is applicable to more programs because the restrictions on the original mtts are less severe. More precisely, the technique from [Küh97, Küh98] requires M_1 to be context-linear and weakly single-use, and M_2 to be weakly single-use, using a stronger notion of "weakly single-use" than we do (as indicated in Footnote 7). In [KV01] the indirect transformation was generalized by weakening the condition on M_2 to a certain *restricted-use* restriction. This condition is incomparable with—i.e., neither strictly more nor strictly less general than—our weaker notion of "weakly single-use", but it is not too hard to see that Construction 3.1 also applies if M_1 is (only) context-linear and M_2 fulfills a straight-forward generalization (by dropping the condition on the initial expression from the definition of "restricted-use") of both "restricted-use" and our notion of "weakly single-use".

Interestingly, the annotation scheme from Definition 4.9 works also for the indirect transformation via atts, i.e., an analogon of Lemma 4.12 holds, implying that any conditions for efficiency nondeterioration derived using our approach apply to the indirect transformation as well. Put differently, we obtain that in terms of the number of call-by-name reduction steps the output programs produced by the indirect and the direct composition construction, respectively, perform identically. This gives a negative answer to the question, implicitly raised in [KV01], whether the fact that the output programs can be syntactically different provides a practical motivation for holding on to the indirect transformation via atts, even though a direct construction (with the obvious attendant benefits, e.g. for an implementor) is available. With respect to call-by-name efficiency, it is now clear that there is no point in doing so. Any advantage that a program obtained by the indirect construction might, on the other hand, hypothetically enjoy under call-by-need evaluation would have to come from a somehow better use of sharing than in the program produced by the direct construction. But such an advantage is highly unlikely if the transformation result is to be an mtt in both cases. Instead, one would have to deviate from the principle that the final program should be an mtt again, by using a different mapping from atts back to functional programs. The most promising candidate would be the mapping into a circular program applied in [Joh87] and [KS87]. We conjecture that this would lead to an overall transformation corresponding to our *lazy composition* transformation from [Voi02b, Voi04] (cf. also the related technique in [Nis02]).

In Theorem 2 of [Man03] the impressive result is proved that a composition of arbitrary many mtts can be transformed into a single one if the function computed by the composition is of linear size increase, which is decidable. The heart of the associated algorithm is to compose two mtts, the first of which is *finite copying* [EM99] and the second of which is recursion-linear. This is again done using an indirection via atts, applying essentially the composition construction from [Küh98] (cum grano salis, as expressed in [Man03], because the atts are additionally equipped with regular look-ahead). It would be interesting to investigate a direct composition construction also for this setting. Formal efficiency analysis of the final mtt vs. the original program, however, would be complicated by the fact that before we even get hands at the finite copying and the recursion-linear mtt to compose, a fair number of constructions from [Man02], [EV85], and [EM03b] have to be applied. Note that neither is the composition technique from [Man03] more general than our Construction 3.1, nor the other way round. While in [Man03] more than two mtts can be composed and there are no a priori syntactic restrictions on their rules, our construction can also handle mtts the composition of which computes a function that is not of linear size increase. Further note that there is not much room for generalizing the result from [Man03]: the introduction of [KV94] describes a tree transduction that is of **quadratic** size increase, can be computed by the composition of two mtts, but not by a single mtt.[31]

5.2 Efficiency analysis for tree transducer composition

Theorems 4.44 and 4.46 are sufficient to formally establish call-by-need efficiency nondeterioration for many typical examples of tree transducer composition for which previously efficiency nondeterioration was assured by ad-hoc reasoning or by experiments only. In particular, this is the case for the various such examples considered in [VK04a]. Our results also capture all the examples for tree transducer composition from [KV01], where the pure elimination of intermediate results was considered as success, without regard to actual numbers of reduction steps. Moreover, they improve on formal efficiency consid-

[31] That the mentioned tree transduction is of quadratic size increase can be established by a straightforward structural induction. That it can be computed by the composition of two mtts is apparent because in [KV94] it is computed by a *macro attributed tree transducer* (for short *matt*), which by a result of that paper can be decomposed into two atts, each of which can be transformed into an mtt following [Fra82]. That it cannot be computed by a single mtt can be considered folklore, in particular in light of the fact that in [KV94] it is used as motivating example for generalizing mtts by introducing the more powerful concept of matts. As an aside: a variation of this example (in which the path is added nondeterministically to an arbitrary leaf rather than to all leafs of the input tree) indicates that the result from [Man03] about the collapse of the composition hierarchy for **nondeterministic** mtts under the assumptions that the relation computed by a composition is a function and is of linear size increase cannot be generalized by dropping the first assumption.

erations performed in [Küh99] and [Höf99] for the special case that one of the mtts to be composed is a tdtt. More precisely, the respective results of [Küh99], namely that the composed program is at least as efficient as the original program if M_1 is a linear tdtt and M_2 is linear (Corollary 21), or if M_1 is linear, but not a tdtt, and M_2 is a linear tdtt with exactly one state (Theorem 23), are covered and generalized by Theorem 1 of [Voi02a], and thus by Theorem 4.44. The same is true for Claim 3.13[32] and Conjecture 3.21 of [Höf99], together stating an efficiency improvement if M_1 is a tdtt and M_2 is linear. Theorem 4.34 of [Höf99] states an efficiency improvement for the case that M_1 is producing and context-nondeleting, and M_2 is a nondeleting tdtt with exactly one state that is not linear. Theorem 2 of [Voi02a]—corresponding to Theorem 4.46 from the present paper—gives (with $\lambda = 0$, using the fact that all $\uparrow$-rules are in $\mathcal{E}_{M_1,M_2}$ if M_1 is context-nondeleting and M_2 is nondeleting) a more general result in that the conditions on M_2 to have only one state and not to be linear can be dropped. While [Küh99] and [Höf99] additionally considered the efficiency of the program obtained by classical deforestation as opposed to tree transducer composition, such a comparison was not a concern of the current study. However, since in the case that the first composition partner is a tdtt the tree transducer composition technique is equivalent to classical deforestation (cf. [Küh99]), our results also establish call-by-need efficiency nondeterioration through classical deforestation for some nonlinear programs, namely if M_1 is a tdtt and M_2 is atmost, or if M_1 is a producing tdtt and M_2 is context-linear or basic (cf. Section 4.6 of [Voi02a]).

5.3 Efficiency analysis of functional programs

The idea to externalize an intensional property of a computation is not new. For example, in [Ros89] for every first-order functional program a step-counting version is produced that—when called with the same arguments—returns the number of call-by-value reduction steps performed by the original program. Then, an abstract interpretation of this step-counting program is used to express the worst-case complexity (in dependence on the input size) by recurrence equations. The ultimate goal is to transform these equations into a closed expression, which unfortunately is feasible only for subclasses of programs. A similar approach is followed in [LG01] but generalized by using more detailed measures than just the number of reduction steps and by using program transformations to make the analysis more accurate.

These techniques cannot easily be adapted to languages with nonstrict semantics because the number of call-by-name (or call-by-need) reduction steps performed for a certain program expression is not a compositional property. Since the computation time invested in evaluating a function argument depends on how much of the argument's value is actually "needed", the total cost of a function application cannot be understood solely from the cost properties of its ingredients in isolation. In [Wad88] step-counting versions computing upper bounds for the number of call-by-need steps are derived (inspired by [Bje87]) by formally describing this "need" using strictness analysis. In [San90] this technique is extended by considering a higher-order language and also deriving lower bounds.

To avoid the problem of having to compute exact demand information, which is un-

[32] Actually, this is Theorem 3.13 in [Höf99], but its proof is flawed by depending on a lemma which is wrong. Unfortunately, this affects also Theorem 3.16 of [Höf99], which moreover cannot be repaired using a different proof as there is a counterexample with a producing tdtt and a context-nondeleting mtt whose composition is less efficient than the original program.

feasible in general, [San95] developed a more operational approach, leading to a theory of call-by-name cost equivalence, an axiomatization of which is given by annotating functional programs with a special identity function that represents a single “tick” of computation time and by stating a set of laws that can be used to derive statements about the relative efficiency of program expressions. With a sharp eye, some of these laws can be recognized in the annotation schemes (for counting the call-by-name steps of the original and of the composed program) from Section 4.3. In [GS01] syntactic means similar to tick-symbols—so called *space gadgets*—are used in reasoning about asymptotic (i.e., only up to constant factors) **space** complexity.

While the papers mentioned in this subsection so far are concerned with deriving explicit cost bounds for a given program, or with comparing the efficiency of two concrete programs that compute the same result but are otherwise unrelated (at least not related in a formal, systematic way), our aim was to analyze on a general level the relative efficiency for pairs of two closely connected programs, namely the input and output programs of Construction 3.1. In this respect our aim is similar to that of [San96], where a call-by-name improvement relation (a nonsymmetric counterpart to the cost equivalence of [San95]) is used to prove the correctness—and, more as a side product, also call-by-name efficiency nondeterioration—of unfold/fold-transformations. A key difference is that, unlike unfold/fold-transformations, our construction cannot be understood as a sequence of locally equivalence-preserving steps, but rather must be considered as a whole in one go. As a consequence, also the “bookkeeping” of tick-symbols for the transformed program cannot be performed in a similarly modular fashion as in [San96]. Instead, we had to very carefully analyze the composition construction to invent the annotation scheme in Section 4.4 so that eventually Lemma 4.12 could be proved. The same aspect prevents the application—to our construction—of the call-by-need variant of the approach from [San96] developed in [MS99].

6 Conclusion

We have developed sufficient conditions for efficiency improvement by tree transducer composition in lazy functional languages. The key was to annotate programs thus that their outputs reflect the numbers of performed reduction steps. Together with the presented “decomposition” of the respective annotation of the composed program in terms of a suitable annotation of the original program, this allows to study the effect of the composition construction on call-by-name efficiency based solely on the outputs computed by two different annotated versions of the original program. The most general and exact way to exploit this base would be to try and develop a procedure deciding for every input program whether or not the number of symbols corresponding to steps of the original program outweighs the number of symbols corresponding to steps of the composed program in the outputs produced for arbitrary input trees. Given that relatively little is known about decidability problems—such as semantic equivalence—for mtts, this seems a bit overambitious as such a decision procedure might not even exist. Instead, driven by practical motivation, we have made compromises on generality and accepted pessimistic approximations at some places for the sake of proceeding with the analysis. Nevertheless, the theorems obtained are rather powerful in many relevant cases. The relatively simple approach to transferring the results from the call-by-name to the call-by-need setting adds to the mentioned compromises and pessimistic approximations—since we essentially compare call-by-name efficiency of the composed program to call-by-need efficiency of the original program, and only under additional conditions—, but not more so than in comparable efficiency studies for other transformation techniques. In fact, it seems questionable whether a more faithful analysis of the call-by-need efficiency of the composed program is possible at all, at least using an "externalization" approach, given that the composition construction itself proceeds intrinsically call-by-name and loses a certain amount of sharing that cannot be recovered without leaving the framework of mtts.[33]

The sufficient conditions established in our theorems are suitable to be checked in an optimizing compiler. In fact, for the special case that one of the mtts to be composed is a tdtt the conditions (from [Voi02a]) have already been implemented in [Reu03], using a slightly less general variant of the procedure presented in Section 4.10. It is planned to integrate also the more general conditions developed in the present paper. For a more fine-grained analysis it would be worthwhile to investigate efficiency measures where not every reduction step is assumed to incur exactly the same cost. Experiments using the implementation from [Reu03] can hopefully provide help in determining necessary weights.

Acknowledgments

I would like to thank Armin Kühnemann for his comments on a draft version, for (literally) proof reading, and for spotting the z that needed to be a y. Thanks are also due to Sebastian Maneth for pointing me to the decidability of the finite copying in the parameters property.

References

[BD77] R.M. Burstall and J. Darlington. A transformation system for developing recursive programs. *J. ACM*, 24:44–67, 1977.

[Bje87] B. Bjerner. Complexity analysis of programs in type theory. Chalmers University of Technology, Göteborg, 1987.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[CDPR98] L. Correnson, E. Duris, D. Parigot, and G. Roussel. Symbolic composition. Technical Report 3348, Unité de recherche INRIA Rocquencourt, 1998.

[CDPR99] L. Correnson, E. Duris, D. Parigot, and G. Roussel. Declarative program transformation: A deforestation case-study. In *Principles and Practice of Declarative Programming, Proceedings*, volume 1702 of *LNCS*, pages 360–377. Springer-Verlag, 1999.

[CF82] B. Courcelle and P. Franchi-Zannettacci. Attribute grammars and recursive program schemes. *Theoret. Comput. Sci.*, 17:163–191, 235–257, 1982.

[33] Note that the lazy composition transformation from [Voi02b, Voi04], which does just that, poses other problems for efficiency analysis because the circular nature of the resulting programs prevents a meaningful direct comparison of numbers of reduction steps and because tupling and untupling operations add to the runtime costs.

[Chi94] W.N. Chin. Safe fusion of functional expressions II: Further improvements. *J. Funct. Prog.*, 4:515–555, 1994.

[Chi99] O. Chitil. Type inference builds a short cut to deforestation. In *International Conference on Functional Programming, Proceedings*, pages 249–260. ACM Press, 1999.

[DJ90] N. Dershowitz and J.P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6, pages 243–320. Elsevier Science Publishers B.V., 1990.

[EM99] J. Engelfriet and S. Maneth. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Inform. and Comput.*, 154:34–91, 1999.

[EM03a] J. Engelfriet and S. Maneth. A comparison of pebble tree transducers with macro tree transducers. *Acta Informatica*, 39:613–698, 2003.

[EM03b] J. Engelfriet and S. Maneth. Macro tree translations of linear size increase are MSO definable. *SIAM J. Comput.*, 32:950–1006, 2003.

[Eng75] J. Engelfriet. Bottom-up and top-down tree transformations — A comparison. *Math. Syst. Theory*, 9:198–231, 1975.

[Eng80] J. Engelfriet. Some open questions and recent results on tree transducers and tree languages. In R.V. Book, editor, *Formal language theory; perspectives and open problems*, pages 241–286. Academic Press, 1980.

[Eng81] J. Engelfriet. Tree transducers and syntax directed semantics. Technical Report Memorandum 363, Technische Hogeschool Twente, 1981.

[EV85] J. Engelfriet and H. Vogler. Macro tree transducers. *J. Comput. Syst. Sci.*, 31:71–146, 1985.

[Fra82] P. Franchi-Zannettacci. *Attributs sémantiques et schémas de programmes*. PhD thesis, Université de Bordeaux I, 1982.

[Fül81] Z. Fülöp. On attributed tree transducers. *Acta Cybernetica*, 5:261–279, 1981.

[FV98] Z. Fülöp and H. Vogler. *Syntax-Directed Semantics — Formal Models Based on Tree Transducers*. Monographs in Theoretical Computer Science, An EATCS Series. Springer-Verlag, 1998.

[Gan83] H. Ganzinger. Increasing modularity and language-independency in automatically generated compilers. *Sci. of Comput. Prog.*, 3:223–278, 1983.

[GG84] H. Ganzinger and R. Giegerich. Attribute coupled grammars. In *Symposium on Compiler Construction, Proceedings*, pages 157–170. ACM Press, 1984.

[Gie88] R. Giegerich. Composition and evaluation of attribute coupled grammars. *Acta Informatica*, 25:355–423, 1988.

[GLP93] A. Gill, J. Launchbury, and S.L. Peyton Jones. A short cut to deforestation. In *Functional Programming Languages and Computer Architecture, Proceedings*, pages 223–232. ACM Press, 1993.

[GS01] J. Gustavsson and D. Sands. Possibilities and limitations of call-by-need space improvement. In *International Conference on Functional Programming, Proceedings*, pages 265–276. ACM Press, 2001.

[HJ92] G.W. Hamilton and S.B. Jones. Extending deforestation for first order functional programs. In *1991 Glasgow Workshop on Functional Programming, Proceedings*, Series of Workshops in Computing, pages 134–145. Springer-Verlag, 1992.

[Höf99] M. Höff. Vergleich von Verfahren zur Elimination von Zwischenergebnissen bei funktionalen Programmen. Master's thesis, Dresden University of Technology, 1999.

[Joh87] T. Johnsson. Attribute grammars as a functional programming paradigm. In *Functional Programming Languages and Computer Architecture, Proceedings*, volume 274 of *LNCS*, pages 154–173. Springer-Verlag, 1987.

[Joh02] P. Johann. A generalization of short-cut fusion and its correctness proof. *Higher-Order and Symb. Comp.*, 15:273–300, 2002.

[KGF02] K. Kakehi, R. Glück, and Y. Futamura. An extension of shortcut deforestation for accumulative list folding. *IEICE Trans. on Inform. and Syst.*, E85-D:1372–1383, 2002.

[Knu68] D.E. Knuth. Semantics of context-free languages. *Math. Syst. Theory*, 2:127–145, 1968. Corrections *Ibid.*, 5:95–96, 1971.

[KS87] M.F. Kuiper and S.D. Swierstra. Using attribute grammars to derive efficient functional programs. In *Computing Science in the Netherlands, Proceedings*. SION, 1987.

[Küh97] A. Kühnemann. *Berechnungsstärken von Teilklassen primitiv-rekursiver Programmschemata*. PhD thesis, Dresden University of Technology, 1997.

[Küh98] A. Kühnemann. Benefits of tree transducers for optimizing functional programs. In *Foundations of Software Technology and Theoretical Computer Science, Proceedings*, volume 1530 of *LNCS*, pages 146–157. Springer-Verlag, 1998.

[Küh99] A. Kühnemann. Comparison of deforestation techniques for functional programs and for tree transducers. In *Functional and Logic Programming, Proceedings*, volume 1722 of *LNCS*, pages 114–130. Springer-Verlag, 1999.

[KV94] A. Kühnemann and H. Vogler. Synthesized and inherited functions — A new computational model for syntax-directed semantics. *Acta Informatica*, 31:431–477, 1994.

[KV01] A. Kühnemann and J. Voigtländer. Tree transducer composition as deforestation method for functional programs. Technical Report TUD-FI01-07, Dresden University of Technology, 2001.

[LG01] Y.A. Liu and G. Gómez. Automatic accurate cost-bound analysis for high-level languages. *IEEE Trans. on Comp.*, 50:1295–1309, 2001.

[Man02] S. Maneth. The complexity of compositions of deterministic tree transducers. In *Foundations of Software Technology and Theoretical Computer Science, Proceedings*, volume 2556 of *LNCS*, pages 265–276. Springer Verlag, 2002.

[Man03] S. Maneth. The macro tree transducer hierarchy collapses for functions of linear size increase. In *Foundations of Software Technology and Theoretical Computer Science, Proceedings*, volume 2914 of *LNCS*, pages 326–337. Springer-Verlag, 2003.

[MBPS04] S. Maneth, A. Berlea, T. Perst, and H. Seidl. XML type checking with macro tree transducers. Technical Report TUM-I0407, Technische Universität München, 2004.

[MS99] A.K. Moran and D. Sands. Improvement in a lazy context: An operational theory for call-by-need. In *Principles of Programming Languages, Proceedings*, pages 43–56. ACM Press, 1999.

[MW93] S. Marlow and P. Wadler. Deforestation for higher-order functions. In *1992 Glasgow Workshop on Functional Programming, Proceedings*, Series of Workshops in Computing, pages 154–165. Springer-Verlag, 1993.

[Nis02] S. Nishimura. Deforesting in accumulating parameters via type-directed transformations. In *Asian Workshop on Programming Languages and Systems, Informal Proceedings*, 2002.

[Reu03] S. Reuther. Implementing tree transducer composition for the Glasgow Haskell Compiler. Master's thesis, Dresden University of Technology, 2003.

[Ros89] M. Rosendahl. Automatic complexity analysis. In *Functional Programming Languages and Computer Architecture, Proceedings*, pages 144–156. ACM Press, 1989.

[Rou70] W.C. Rounds. Mappings and grammars on trees. *Math. Syst. Theory*, 4:257–287, 1970.

[San90] D. Sands. Complexity analysis for a lazy higher-order language. In *European Symposium on Programming, Proceedings*, volume 432 of *LNCS*, pages 361–376. Springer-Verlag, 1990.

[San95] D. Sands. A naïve time analysis and its theory of cost equivalence. *J. Logic and Comput.*, 5:495–541, 1995.

[San96] D. Sands. Total correctness by local improvement in the transformation of functional programs. *ACM Trans. on Prog. Lang. and Systems*, 18:175–234, 1996.

[Sve02] J. Svenningsson. Shortcut fusion for accumulating parameters & zip-like functions. In *International Conference on Functional Programming, Proceedings*, pages 124–132. ACM Press, 2002.

[Tha70] J.W. Thatcher. Generalized^2 sequential machine maps. *J. Comput. Syst. Sci.*, 4:339–367, 1970.

[VK04a] J. Voigtländer and A. Kühnemann. Composition of functions with accumulating parameters. *J. Funct. Prog.*, 14:317–363, 2004.

[VK04b] J. Voigtländer and A. Kühnemann. Proof appendix of [VK04a], 2004. Available from *http://wwwtcs.inf.tu-dresden.de/∼voigt/JFP-appendix.pdf*.

[Vog87] H. Vogler. Basic tree transducers. *J. Comput. Syst. Sci.*, 34:87–128, 1987.

[Vog91] H. Vogler. Functional description of the contextual analysis in block-structured programming languages: A case study of tree transducers. *Sci. of Comput. Prog.*, 16:251–275, 1991.

[Voi01] J. Voigtländer. Composition of restricted macro tree transducers. Master's thesis, Dresden University of Technology, 2001.

[Voi02a] J. Voigtländer. Conditions for efficiency improvement by tree transducer composition. In *Rewriting Techniques and Applications, Proceedings*, volume 2378 of *LNCS*, pages 222–236. Springer-Verlag, 2002.

[Voi02b] J. Voigtländer. Using circular programs to deforest in accumulating parameters. In *Asian Symposium on Partial Evaluation and Semantics-Based Program Manipulation, Proceedings*, pages 126–137. ACM Press, 2002.

[Voi04] J. Voigtländer. Using circular programs to deforest in accumulating parameters. *Higher-Order and Symb. Comp.*, 17:129–163, 2004.

[vvS02] D. van Arkel, J. van Groningen, and S. Smetsers. Fusion in practice. In *Implementation of Functional Languages, Selected Papers*, volume 2670 of *LNCS*, pages 51–67. Springer-Verlag, 2002.

[Wad88] P. Wadler. Strictness analysis aids time analysis. In *Principles of Programming Languages, Proceedings*, pages 119–132. ACM Press, 1988.

[Wad90] P. Wadler. Deforestation: Transforming programs to eliminate trees. *Theoret. Comput. Sci.*, 73:231–248, 1990.

A Proof appendix

The predominant proof principle to be used in this appendix—except for in the proofs of Lemmata 4.5 and 4.11—works as follows.

Proof principle (simultaneous induction; cf. e.g. [EV85, KV94, FV98, VK04b]) Let Σ be a ranked alphabet. For a tree $t \in T_\Sigma$, a subtree $sub(t, \pi)$ with $\pi \in paths(t)$ is called *proper* if $\pi \neq \varepsilon$ and *direct* if $\pi \in \mathbb{N}_+$. Let (a) and (b) be statements, called *induction hypotheses*, where (a) has a free variable $t \in T_\Sigma$ and (b) has free variables $p \in \mathbb{N}$ and $t_1, \ldots, t_p \in T_\Sigma$. If

(a) $\Leftarrow$ (b) : for every $t \in T_\Sigma$ such that (a) holds for every proper subtree of t and (b) holds for every $p \in \mathbb{N}$ and direct subtrees $t_1, \ldots, t_p$ of t, we can prove that (a) holds, and

(b) $\Leftarrow$ (a) : for every $p \in \mathbb{N}$ and $t_1, \ldots, t_p \in T_\Sigma$ such that (a) holds for each of the $t_1, \ldots, t_p$, we can prove that (b) holds,

then we have proved that (a) holds for every $t \in T_\Sigma$ and that (b) holds for every $p \in \mathbb{N}$ and $t_1, \ldots, t_p \in T_\Sigma$. $\square$

Note the somewhat unusual reference to (a) from within the induction step **(a) $\Leftarrow$ (b)**, which gives more—and needed—freedom in later proofs. Since it refers only to proper subtrees of the one under consideration, the soundness of the proof principle can still be argued along similar lines as in Section 2.5 of [EV85]. Further note that ordinary structural induction over trees is a special case of simultaneous induction, with (b) being trivially true. Simultaneous and structural induction will also be used for trees with a more refined structure, in particular for elements of $RHS(\cdots)$-sets (cf. Definition 2.1) and elements of $WFT_{M_1;M_2}$ (cf. Definition 4.2).

Proof of Lemma 2.7
We prove the following two statements:

(a) $\underline{\text{for every } t \in T_\Sigma:}$
For every $f \in F^{(r'+1)}$, $r \in \mathbb{N}$, $k \in [r]$, and $\theta_1, \ldots, \theta_{r'} \in T_{F \cup \Sigma \cup \Delta}(Y_r)$:
$|nf(\Rightarrow_R, f\ t\ \theta_1 \cdots \theta_{r'})|_{y_k} = \sum\limits_{k' \in [r']} |nf(\Rightarrow_R, f\ t\ y_1 \cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$.[34]

(b) $\underline{\text{for every } p \in \mathbb{N} \text{ and } t_1, \ldots, t_p \in T_\Sigma:}$
For every $r', r \in \mathbb{N}$, $k \in [r]$, $\theta_1, \ldots, \theta_{r'} \in T_{F \cup \Sigma \cup \Delta}(Y_r)$, and $\phi \in RHS(F, \Delta, U_p, Y_{r'})$:

$$\begin{aligned} & |nf(\Rightarrow_R, \phi[u_1, \ldots, u_p, y_1, \ldots, y_{r'} \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_{r'}])|_{y_k} \\ = & \sum_{k' \in [r']} |nf(\Rightarrow_R, \phi[u_i \leftarrow t_i \mid i \in [p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}. \end{aligned}$$

by simultaneous induction. Lemma 2.7 coincides with statement (a).

(a) $\Leftarrow$ (b) : Assume $t = \sigma\ t_1 \cdots t_p$ for $\sigma \in \Sigma^{(p)}$ and $t_1, \ldots, t_p \in T_\Sigma$. We calculate:

[34] To minimize the need for brackets when writing sum formulae, we will use the conventions that $*$ binds stronger than a $\sum$ to the left of it, while $\sum$ binds stronger than $+$, so that, e.g., an expression of the form $\sum a * b + c * \sum d * e$ is parsed as $\Big(\sum(a * b)\Big) + c * \Big(\sum(d * e)\Big)$.

$$|nf(\Rightarrow_R, f\ (\sigma\ t_1 \cdots t_p)\ \theta_1 \cdots \theta_{r'})|_{y_k}$$
$$= \text{(by } \Rightarrow_R\text{)}$$
$$|nf(\Rightarrow_R, rhs_{M,f,\sigma}[u_1, \ldots, u_p, y_1, \ldots, y_{r'} \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_{r'}])|_{y_k}$$
$$= \text{(by induction hypothesis (b) with } \phi = rhs_{M,f,\sigma}\text{)}$$
$$\sum_{k' \in [r']} |nf(\Rightarrow_R, rhs_{M,f,\sigma}[u_i \leftarrow t_i \mid i \in [p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$
$$= \text{(by } \Rightarrow_R\text{)}$$
$$\sum_{k' \in [r']} |nf(\Rightarrow_R, f\ (\sigma\ t_1 \cdots t_p)\ y_1 \cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$

(b) $\Leftarrow$ (a) : for fixed $r', r \in \mathbb{N}$, $k \in [r]$, and $\theta_1, \ldots, \theta_{r'} \in T_{F \cup \Sigma \cup \Delta}(Y_r)$ by structural induction on $\phi \in RHS(F, \Delta, U_p, Y_{r'})$:

$\underline{\phi = y_{k''} \in Y_{r'}}$:

$$|nf(\Rightarrow_R, y_{k''}[u_1, \ldots, u_p, y_1, \ldots, y_{r'} \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_{r'}])|_{y_k}$$
$$= \text{(by substitution)}$$
$$|nf(\Rightarrow_R, \theta_{k''})|_{y_k}$$
$$= \text{(by } |y_{k''}|_{y_{k''}} = 1 \text{ and } |y_{k''}|_{y_{k'}} = 0 \text{ for } k' \neq k''\text{)}$$
$$\sum_{k' \in [r']} |y_{k''}|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$
$$= \text{(by substitution and normal form)}$$
$$\sum_{k' \in [r']} |nf(\Rightarrow_R, y_{k''}[u_i \leftarrow t_i \mid i \in [p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$

$\underline{\phi = \delta\ \phi_1 \cdots \phi_n}$ for some $\delta \in \Delta^{(n)}$ and $\phi_1, \ldots, \phi_n \in RHS(F, \Delta, U_p, Y_{r'})$:

$$|nf(\Rightarrow_R, (\delta\ \phi_1 \cdots \phi_n)[u_1, \ldots, u_p, y_1, \ldots, y_{r'} \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_{r'}])|_{y_k}$$
$$= \text{(by substitution, normal form, and definition of } |\,.\,|_{y_k}\text{)}$$
$$\sum_{b \in [n]} |nf(\Rightarrow_R, \phi_b[u_1, \ldots, u_p, y_1, \ldots, y_{r'} \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_{r'}])|_{y_k}$$
$$= \text{(by the induction hypotheses for the } \phi_1, \ldots, \phi_n\text{)}$$
$$\sum_{b \in [n]} \sum_{k' \in [r']} |nf(\Rightarrow_R, \phi_b[u_i \leftarrow t_i \mid i \in [p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$
$$= \text{(by sum manipulation)}$$
$$\sum_{k' \in [r']} \Big(\sum_{b \in [n]} |nf(\Rightarrow_R, \phi_b[u_i \leftarrow t_i \mid i \in [p]])|_{y_{k'}} \Big) * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$
$$= \text{(by substitution, normal form, and definition of } |\,.\,|_{y_{k'}}\text{)}$$
$$\sum_{k' \in [r']} |nf(\Rightarrow_R, (\delta\ \phi_1 \cdots \phi_n)[u_i \leftarrow t_i \mid i \in [p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$

$\underline{\phi = f\ u_{i'}\ \phi_1 \cdots \phi_{r''}}$ for some $f \in F^{(r''+1)}$, $u_{i'} \in U_p$, and $\phi_1, \ldots, \phi_{r''} \in RHS(F, \Delta, U_p, Y_{r'})$:

$$|nf(\Rightarrow_R, (f\ u_{i'}\ \phi_1 \cdots \phi_{r''})[u_1, \ldots, u_p, y_1, \ldots, y_{r'} \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_{r'}])|_{y_k}$$
$$= \text{(by substitution and induction hypothesis (a) for } t_{i'}\text{)}$$
$$\sum_{k'' \in [r'']} |nf(\Rightarrow_R, f\ t_{i'}\ y_1 \cdots y_{r''})|_{y_{k''}} * |nf(\Rightarrow_R, \phi_{k''}[u_i \leftarrow t_i \mid i \in [p], y_{k'} \leftarrow \theta_{k'} \mid k' \in [r']])|_{y_k}$$
$$= \text{(by the induction hypotheses for the } \phi_1, \ldots, \phi_{r''}\text{)}$$

$$\sum_{k''\in[r'']} |nf(\Rightarrow_R, f\ t_{i'}\ y_1\cdots y_{r''})|_{y_{k''}} * \sum_{k'\in[r']} |nf(\Rightarrow_R, \phi_{k''}[u_i \leftarrow t_i \mid i\in[p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$

$$= \text{(by sum manipulation)}$$

$$\sum_{k'\in[r']} \Big(\sum_{k''\in[r'']} |nf(\Rightarrow_R, f\ t_{i'}\ y_1\cdots y_{r''})|_{y_{k''}} * |nf(\Rightarrow_R, \phi_{k''}[u_i \leftarrow t_i \mid i\in[p]])|_{y_{k'}} \Big) * |nf(\Rightarrow_R, \theta_{k'})|_{y_k}$$

$$= \text{(by substitution and induction hypothesis (a) for } t_{i'})$$

$$\sum_{k'\in[r']} |nf(\Rightarrow_R, (f\ u_{i'}\ \phi_1\cdots\phi_{r''})[u_i \leftarrow t_i \mid i\in[p]])|_{y_{k'}} * |nf(\Rightarrow_R, \theta_{k'})|_{y_k} \quad \square$$

Proof of Lemma 2.8

For a context-linear mtt M we prove the following two statements:

(a) for every $t \in T_\Sigma$:
For every $f \in F^{(r+1)}$ and $k \in [r]$: $|nf(\Rightarrow_R, f\ t\ y_1\cdots y_r)|_{y_k} \leq 1$.

(b) for every $p \in \mathbb{N}$ and $t_1,\ldots,t_p \in T_\Sigma$:
For every $r \in \mathbb{N}$, $k \in [r]$, and $\phi \in RHS(F,\Delta,U_p,Y_r)$:
$|nf(\Rightarrow_R, \phi[u_i \leftarrow t_i \mid i\in[p]])|_{y_k} \leq |\phi|_{y_k}$.

by simultaneous induction. The first statement of Lemma 2.8 then coincides with statement (a). The second statement of Lemma 2.8, for context-nondeleting M, is obtained by replacing all occurrences of $\leq$ and "context-linear" by $\geq$ and "context-nondeleting", respectively, in the above statements and the following proof.

(a) $\Leftarrow$ (b) : Assume $t = \sigma\ t_1\cdots t_p$ for $\sigma \in \Sigma^{(p)}$ and $t_1,\ldots,t_p \in T_\Sigma$. We calculate:

$$|nf(\Rightarrow_R, f\ (\sigma\ t_1\cdots t_p)\ y_1\cdots y_r)|_{y_k}$$
$$= \text{(by } \Rightarrow_R)$$
$$|nf(\Rightarrow_R, rhs_{M,f,\sigma}[u_i \leftarrow t_i \mid i\in[p]])|_{y_k}$$
$$\leq \text{(by induction hypothesis (b) with } \phi = rhs_{M,f,\sigma})$$
$$|rhs_{M,f,\sigma}|_{y_k}$$
$$\leq \text{(since } M \text{ is context-linear)}$$
$$1$$

(b) $\Leftarrow$ (a) : for fixed $r \in \mathbb{N}$ and $k \in [r]$ by structural induction on $\phi \in RHS(F,\Delta,U_p,Y_r)$:

$\phi = y_{k'} \in Y_r$:

$$|nf(\Rightarrow_R, y_{k'}[u_i \leftarrow t_i \mid i\in[p]])|_{y_k}$$
$$= \text{(by substitution and normal form)}$$
$$|y_{k'}|_{y_k}$$

$\phi = \delta\ \phi_1\cdots\phi_n$ for some $\delta \in \Delta^{(n)}$ and $\phi_1,\ldots,\phi_n \in RHS(F,\Delta,U_p,Y_r)$:

$$|nf(\Rightarrow_R, (\delta\ \phi_1\cdots\phi_n)[u_i \leftarrow t_i \mid i\in[p]])|_{y_k}$$
$$= \text{(by substitution, normal form, and definition of } |\,.\,|_{y_k})$$
$$\sum_{b\in[n]} |nf(\Rightarrow_R, \phi_b[u_i \leftarrow t_i \mid i\in[p]])|_{y_k}$$

$$\leq \text{ (by the induction hypotheses for the } \phi_1, \dots, \phi_n)$$
$$\sum_{b \in [n]} |\phi_b|_{y_k}$$
$$= \text{ (by definition of } |\,.\,|_{y_k})$$
$$|\delta\ \phi_1 \cdots \phi_n|_{y_k}$$

$\underline{\phi = f\ u_{i'}\ \phi_1 \cdots \phi_{r'}}$ for some $f \in F^{(r'+1)}$, $u_{i'} \in U_p$, and $\phi_1, \dots, \phi_{r'} \in RHS(F, \Delta, U_p, Y_r)$:

$$|nf(\Rightarrow_R, (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p]])|_{y_k}$$
$$= \text{ (by substitution and Lemma 2.7)}$$
$$\sum_{k' \in [r']} |nf(\Rightarrow_R, f\ t_{i'}\ y_1 \cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_R, \phi_{k'}[u_i \leftarrow t_i \mid i \in [p]])|_{y_k}$$
$$\leq \text{ (by induction hypothesis (a) for } t_{i'},$$
$$\text{using that the factors } |nf(\Rightarrow_R, \phi_{k'}[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} \geq 0)$$
$$\sum_{k' \in [r']} |nf(\Rightarrow_R, \phi_{k'}[u_i \leftarrow t_i \mid i \in [p]])|_{y_k}$$
$$\leq \text{ (by the induction hypotheses for the } \phi_1, \dots, \phi_{r'})$$
$$\sum_{k' \in [r']} |\phi_{k'}|_{y_k}$$
$$= \text{ (by definition of } |\,.\,|_{y_k})$$
$$|f\ u_{i'}\ \phi_1 \cdots \phi_{r'}|_{y_k} \quad \square$$

Proof of Lemma 4.5
We prove (below) by structural induction on $\varrho \in WFT_{M_1;M_2}$ that for every $\overline{\varrho} \in WFT_{M_1;M_2}$ with $\varrho \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho}$:

$$|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \varrho)|_\bullet = 1 + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \overline{\varrho})|_\bullet.$$

Taking into account that for every $\varrho \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^0 \varrho'$ it holds $\varrho = \varrho'$ and that for every $c \in \mathbb{N}$ and $\varrho \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^{c+1} \varrho'$ there is a $\overline{\varrho} \in WFT_{M_1;M_2}$ such that $\varrho \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho}$ and $\overline{\varrho} \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^c \varrho'$, this implies by a straightforward induction on $c \in \mathbb{N}$ that for every $\varrho \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^c \varrho'$ the following holds:

$$|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \varrho)|_\bullet = c + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \varrho')|_\bullet.$$

Lemma 4.5 follows because for every $t \in T_\Sigma$ we can then from

$$e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]] \left(\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}\right)^{cbn_{M_1;M_2}(t)} \tau_{M_2}(\tau_{M_1}(t)) \in T_\Omega$$

conclude that

$$\begin{aligned} &|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]])|_\bullet \\ =\ &cbn_{M_1;M_2}(t) + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \tau_{M_2}(\tau_{M_1}(t)))|_\bullet \end{aligned}$$

and because $|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \tau_{M_2}(\tau_{M_1}(t)))|_\bullet = |\tau_{M_2}(\tau_{M_1}(t))|_\bullet = 0$, while $nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, e_{M_2}[v_1 \leftarrow e_{M_1}[u_1 \leftarrow t]])$ is equal to $\tau_{M_2^{\diamond\to\bullet}}(\tau_{M_1^{\to\diamond}}(t))$.

$\underline{\varrho = g\ (f\ (\sigma\ t_1 \cdots t_p)\ \phi_1 \cdots \phi_r)\ \varrho_1 \cdots \varrho_s}$ for some $g \in G^{(s+1)}$, $f \in F^{(r+1)}$, $\sigma \in \Sigma^{(p)}$, $t_1, \ldots,$ $t_p \in T_\Sigma$, $\phi_1, \ldots, \phi_r \in WFT_{M_1}$, and $\varrho_1, \ldots, \varrho_s \in WFT_{M_1;M_2}$:
By the precondition $g\ (f\ (\sigma\ t_1 \cdots t_p)\ \phi_1 \cdots \phi_r)\ \varrho_1 \cdots \varrho_s \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho}$ and the definition of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ in Definition 4.2, it holds $\overline{\varrho} = g\ rhs_{M_1,f,\sigma}[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \phi_1, \ldots, \phi_r]\ \varrho_1 \cdots \varrho_s$. We calculate:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, g\ (f\ (\sigma\ t_1 \cdots t_p)\ \phi_1 \cdots \phi_r)\ \varrho_1 \cdots \varrho_s)|_\bullet \\
&\quad = \text{(by } \Rightarrow_{R_1^{\to\diamond}}\text{, using that } rhs_{M_1^{\to\diamond},f,\sigma} = \diamond\ rhs_{M_1,f,\sigma} \text{ by Definition 4.4,} \\
&\qquad\quad \text{and by substitution)} \\
&|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, g\ (\diamond\ rhs_{M_1,f,\sigma}[u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \phi_k \mid k \in [r]])\ \varrho_1 \cdots \varrho_s)|_\bullet \\
&\quad = \text{(by } \Rightarrow_{R_2^{\diamond\to\bullet}}\text{, using that } rhs_{M_2^{\diamond\to\bullet},g,\diamond} = \bullet\ (g\ v_1\ z_1 \cdots z_s) \text{ by Definition 4.4,} \\
&\qquad\quad \text{and by normal form and definition of } |\,.\,|_\bullet) \\
&1 + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, g\ rhs_{M_1,f,\sigma}[u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \phi_k \mid k \in [r]]\ \varrho_1 \cdots \varrho_s)|_\bullet
\end{aligned}
$$

$\underline{\varrho = g\ (\delta\ \phi_1 \cdots \phi_q)\ \varrho_1 \cdots \varrho_s}$ for some $g \in G^{(s+1)}$, $\delta \in \Delta^{(q)}$, $\phi_1, \ldots, \phi_q \in WFT_{M_1}$, and $\varrho_1, \ldots, \varrho_s \in WFT_{M_1;M_2}$:
By the precondition $g\ (\delta\ \phi_1 \cdots \phi_q)\ \varrho_1 \cdots \varrho_s \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho}$ and the definition of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ in Definition 4.2, it holds $\overline{\varrho} = rhs_{M_2,g,\delta}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \phi_1, \ldots, \phi_q, \varrho_1, \ldots, \varrho_s]$. We calculate:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, g\ (\delta\ \phi_1 \cdots \phi_q)\ \varrho_1 \cdots \varrho_s)|_\bullet \\
&\quad = \text{(by } \Rightarrow_{R_2^{\diamond\to\bullet}}\text{, using that } rhs_{M_2^{\diamond\to\bullet},g,\delta} = \bullet\ rhs_{M_2,g,\delta} \text{ by Definition 4.4)} \\
&|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, (\bullet\ rhs_{M_2,g,\delta})[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \phi_1, \ldots, \phi_q, \varrho_1, \ldots, \varrho_s])|_\bullet \\
&\quad = \text{(by substitution, normal form, and definition of } |\,.\,|_\bullet) \\
&1 + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, rhs_{M_2,g,\delta}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \phi_1, \ldots, \phi_q, \varrho_1, \ldots, \varrho_s])|_\bullet
\end{aligned}
$$

$\underline{\varrho = \omega\ \varrho_1 \cdots \varrho_n}$ for some $\omega \in \Omega^{(n)}$ and $\varrho_1, \ldots, \varrho_n \in WFT_{M_1;M_2}$:
By the precondition $\omega\ \varrho_1 \cdots \varrho_n \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho}$ and the definition of $\overset{cbn}{\Rightarrow}_{R_1 \cup R_2}$ in Definition 4.2 there exist $b \in [n]$ and $\overline{\varrho_b}$ with $\varrho_b \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho_b}$, such that $\overline{\varrho} = \omega\ \varrho_1 \cdots \varrho_{b-1}\ \overline{\varrho_b}\ \varrho_{b+1} \cdots \varrho_n$. We calculate:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \omega\ \varrho_1 \cdots \varrho_n)|_\bullet \\
&\quad = \text{(by normal form, definition of } |\,.\,|_\bullet\text{, and sum manipulation)} \\
&|nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \varrho_b)|_\bullet + \sum_{d \in [n] \setminus \{b\}} |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \varrho_d)|_\bullet \\
&\quad = \text{(by the induction hypothesis for } \varrho_b \overset{cbn}{\Rightarrow}_{R_1 \cup R_2} \overline{\varrho_b}) \\
&1 + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \overline{\varrho_b})|_\bullet + \sum_{d \in [n] \setminus \{b\}} |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \varrho_d)|_\bullet \\
&\quad = \text{(by normal form, definition of } |\,.\,|_\bullet\text{, and sum manipulation)} \\
&1 + |nf(\Rightarrow_{R_1^{\to\diamond} \cup R_2^{\diamond\to\bullet}}, \omega\ \varrho_1 \cdots \varrho_{b-1}\ \overline{\varrho_b}\ \varrho_{b+1} \cdots \varrho_n)|_\bullet \quad \square
\end{aligned}
$$

Proof of Lemma 4.10
For fixed $p, r \in \mathbb{N}$ we prove the following two statements:

(a) <u>for every $\phi \in RHS(F,\Delta,U_p,Y_r)$:</u>
For every $g \in G^{(s+1)}$ and $\eta_1,\dots,\eta_s,\eta_1^\star,\dots,\eta_s^\star \in T_{F\cup G\cup H\cup\Delta\cup\Omega\cup\{\diamond,\star,\circ,nil\}}(U_p \cup Y_r \cup Z \cup Z_G)$ with $nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \eta_l^\star) = nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \eta_l)$ for every $l \in [s]$:

$$\begin{aligned}&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ \phi[y_k \leftarrow \star\ y_k \mid k \in [r]]\ \eta_1^\star \cdots \eta_s^\star)\\ &= nf(\Rightarrow_{R_2\cup Pre\cup Pair}, g\ \phi\ \eta_1 \cdots \eta_s).\end{aligned}$$

(b) <u>for every $r' \in \mathbb{N}$ and $\phi_1,\dots,\phi_{r'} \in RHS(F,\Delta,U_p,Y_r)$:</u>
For every $f' \in F^{(r'+1)}$, $u_i \in U_p$, and $\eta_{g_1,1},\dots,\eta_{g_\mu,s_\mu},\eta_{g_1,1}^\star,\dots,\eta_{g_\mu,s_\mu}^\star \in T_{F\cup G\cup H\cup\Delta\cup\Omega\cup\{\diamond,\star,\circ,nil\}}(U_p \cup Y_r \cup Z \cup Z_G)$ with $nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \eta_{g',l'}^\star) = nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \eta_{g',l'})$ for every $g' \in G^{(s'+1)}$ and $l' \in [s']$, for every $\mathcal{C} \subseteq [r'] \times G$, $k' \in [r']$, and $g'' \in G$:

$$\begin{aligned}&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, nest_{f'}(k',g'',\mathcal{C})[u \leftarrow u_i,\\ &\qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid b \in [r'],\\ &\qquad z_{g',l'} \leftarrow \eta_{g',l'}^\star \mid g' \in G^{(s'+1)}, l' \in [s']])\\ &= nf(\Rightarrow_{R_2\cup Pre\cup Pair}, nest_{f'}(k',g'',\mathcal{C})[u \leftarrow u_i,\\ &\qquad y'_b \leftarrow \phi_b \mid b \in [r'],\\ &\qquad z_{g',l'} \leftarrow \eta_{g',l'} \mid g' \in G^{(s'+1)}, l' \in [s']]).\end{aligned}$$

by simultaneous induction. Lemma 4.10 then follows from statement (a) with $\phi = rhs_{M_1,f,\sigma}$ and $\eta_l = \eta_l^\star = z_l$ for every $l \in [s]$.

(a) $\Leftarrow$ (b) : by case analysis on $\phi \in RHS(F,\Delta,U_p,Y_r)$:

<u>$\phi = y_{k'} \in Y_r$:</u>

$$\begin{aligned}&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ y_{k'}[y_k \leftarrow \star\ y_k \mid k \in [r]]\ \eta_1^\star \cdots \eta_s^\star)\\ &\quad= \text{(by substitution)}\\ &nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ (\star\ y_{k'})\ \eta_1^\star \cdots \eta_s^\star)\\ &\quad= \text{(by } \Rightarrow_{R_2^{\diamond\star\to\circ}}\text{, using that } rhs_{M_2^{\diamond\star\to\circ},g,\star} = g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s) \text{ by Definition 4.9)}\\ &nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ y_{k'}\ (\circ\ \eta_1^\star)\ \cdots\ (\circ\ \eta_s^\star))\\ &\quad= \text{(by } \Rightarrow_{Pre} \text{ (cf. Construction 3.1) and normal form)}\\ &y_{k',g}\\ &\quad= \text{(by } \Rightarrow_{Pre} \text{ (cf. Construction 3.1) and normal form)}\\ &nf(\Rightarrow_{R_2\cup Pre\cup Pair}, g\ y_{k'}\ \eta_1 \cdots \eta_s)\end{aligned}$$

<u>$\phi = \delta\ \phi_1 \cdots \phi_q$</u> for some $\delta \in \Delta^{(q)}$ and $\phi_1,\dots,\phi_q \in RHS(F,\Delta,U_p,Y_r)$:

$$\begin{aligned}&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ (\delta\ \phi_1 \cdots \phi_q)[y_k \leftarrow \star\ y_k \mid k \in [r]]\ \eta_1^\star \cdots \eta_s^\star)\\ &\quad= \text{(by substitution and } \Rightarrow_{R_2^{\diamond\star\to\circ}},\\ &\qquad\quad \text{using that } rhs_{M_2^{\diamond\star\to\circ},g,\delta} = rhs_{M_2,g,\delta} \text{ by Definition 4.9)}\\ &nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, rhs_{M_2,g,\delta}[v_j \leftarrow \phi_j[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid j \in [q],\\ &\qquad z_l \leftarrow \eta_l^\star \mid l \in [s]])\\ &\quad= \text{(see below)}\\ &nf(\Rightarrow_{R_2\cup Pre\cup Pair}, rhs_{M_2,g,\delta}[v_1,\dots,v_q,z_1,\dots,z_s \leftarrow \phi_1,\dots,\phi_q,\eta_1,\dots,\eta_s])\\ &\quad= \text{(by } \Rightarrow_{R_2}\text{)}\\ &nf(\Rightarrow_{R_2\cup Pre\cup Pair}, g\ (\delta\ \phi_1 \cdots \phi_q)\ \eta_1 \cdots \eta_s)\end{aligned}$$

Above we used that for every $\psi \in RHS(G, \Omega, V_q, Z_s)$:

$$\begin{aligned}
& nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \psi[v_j \leftarrow \phi_j[y_k \leftarrow \star\, y_k \mid k \in [r]] \mid j \in [q], \\
& \qquad\qquad z_l \leftarrow \eta_l^\star \mid l \in [s]]) \\
&= nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \psi[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \phi_1, \ldots, \phi_q, \eta_1, \ldots, \eta_s])\,,
\end{aligned}$$

which can be proved by structural induction. The validity in the base case $\psi \in Z_s$ follows from the precondition that for every $l \in [s]$, $nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \eta_l^\star) = nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \eta_l)$. The one recursive case, $\psi = \omega\ \psi_1 \cdots \psi_n$, is straightforward and in the other one, $\psi = g'\ v_j\ \psi_1 \cdots \psi_{s'}$, the induction hypotheses for the $\psi_1, \ldots, \psi_{s'}$ establish the preconditions necessary to apply induction hypothesis (a) for ϕ_j, which suffices.

$\underline{\phi = f'\ u_i\ \phi_1 \cdots \phi_{r'}}$ for some $f' \in F^{(r'+1)}$, $u_i \in U_p$, and $\phi_1, \ldots, \phi_{r'} \in RHS(F, \Delta, U_p, Y_r)$:

$$\begin{aligned}
& nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, g\ (f'\ u_i\ \phi_1 \cdots \phi_{r'})[y_k \leftarrow \star\, y_k \mid k \in [r]]\ \eta_1^\star \cdots \eta_s^\star) \\
&\quad = \text{(by substitution and } \Rightarrow_{Pair} \text{ (cf. Construction 3.1))} \\
& nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \overline{f'g}\ u_i\ nest_{f'}(1, g_1, \emptyset)[u \leftarrow u_i, \\
& \qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\, y_k \mid k \in [r]] \mid b \in [r'], \\
& \qquad z_{g,l} \leftarrow \eta_l^\star \mid l \in [s], \\
& \qquad z_{g',l'} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l' \in [s']]] \\
& \qquad \cdots \\
& \qquad nest_{f'}(r', g_\mu, \emptyset)[u \leftarrow u_i, \\
& \qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\, y_k \mid k \in [r]] \mid b \in [r'], \\
& \qquad z_{g,l} \leftarrow \eta_l^\star \mid l \in [s], \\
& \qquad z_{g',l'} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l' \in [s']]] \\
& \qquad \eta_1^\star \cdots \eta_s^\star) \\
&\quad = \text{(see below)} \\
& nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \overline{f'g}\ u_i\ nest_{f'}(1, g_1, \emptyset)[u \leftarrow u_i, \\
& \qquad y'_b \leftarrow \phi_b \mid b \in [r'], \\
& \qquad z_{g,l} \leftarrow \eta_l \mid l \in [s], \\
& \qquad z_{g',l'} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l' \in [s']] \\
& \qquad \cdots \\
& \qquad nest_{f'}(r', g_\mu, \emptyset)[u \leftarrow u_i, \\
& \qquad y'_b \leftarrow \phi_b \mid b \in [r'], \\
& \qquad z_{g,l} \leftarrow \eta_l \mid l \in [s], \\
& \qquad z_{g',l'} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l' \in [s']] \\
& \qquad \eta_1 \cdots \eta_s) \\
&\quad = \text{(by } \Rightarrow_{Pair} \text{ (cf. Construction 3.1))} \\
& nf(\Rightarrow_{R_2\cup Pre\cup Pair}, g\ (f'\ u_i\ \phi_1 \cdots \phi_{r'})\ \eta_1 \cdots \eta_s)
\end{aligned}$$

Given the precondition that $nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \eta_l^\star) = nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \eta_l)$ for every $l \in [s]$, the above gap can be closed (by normal forms) if we establish for every $k' \in [r']$

and $g'' \in G$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, nest_{f'}(k',g'',\emptyset)[u \leftarrow u_i,\\
&\qquad\qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid b \in [r'],\\
&\qquad\qquad z_{g,l} \leftarrow \eta^\star_l \mid l \in [s],\\
&\qquad\qquad z_{g',l'} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l' \in [s']]])\\
= {} &nf(\Rightarrow_{R_2\cup Pre\cup Pair}, nest_{f'}(k',g'',\emptyset)[u \leftarrow u_i,\\
&\qquad\qquad y'_b \leftarrow \phi_b \mid b \in [r'],\\
&\qquad\qquad z_{g,l} \leftarrow \eta_l \mid l \in [s],\\
&\qquad\qquad z_{g',l'} \leftarrow nil \mid g' \in G^{(s'+1)} \setminus \{g\}, l' \in [s']]).
\end{aligned}
$$

But this is an instance of induction hypothesis (b), given that $nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, nil)$ $= nf(\Rightarrow_{R_2\cup Pre\cup Pair}, nil)$ and the preconditions on the $\eta_1,\ldots,\eta_s,\eta^\star_1,\ldots,\eta^\star_s$.

(b) $\Leftarrow$ (a) : for fixed $f' \in F^{(r'+1)}$, $u_i \in U_p$, and $\eta_{g_1,1},\ldots,\eta_{g_\mu,s_\mu},\eta^\star_{g_1,1},\ldots,\eta^\star_{g_\mu,s_\mu} \in T_{F\cup G\cup H\cup\Delta\cup\Omega\cup\{\diamond,\star,\circ,nil\}}(U_p \cup Y_r \cup Z \cup Z_G)$ with $nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \eta^\star_{g',l'}) = nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \eta_{g',l'})$ for every $g' \in G^{(s'+1)}$ and $l' \in [s']$, by induction over the reversed subset-order on $\mathcal{P}([r'] \times G)$:

$\underline{\mathcal{C} = [r'] \times G}$: In this base case we have for every $k' \in [r']$ and $g'' \in G$ that $(k',g'') \in \mathcal{C}$ and thus $nest_{f'}(k',g'',\mathcal{C}) = nil$, which makes the desired equation trivially true.

$\underline{\mathcal{C} \subset [r'] \times G}$: For every $k' \in [r']$ and $g'' \in G$ with $(k',g'') \notin \mathcal{C}$—the other case is again trivial—we have (setting $s'' = rank_G(g'') - 1$):

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, nest_{f'}(k',g'',\mathcal{C})[u \leftarrow u_i,\\
&\qquad\qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid b \in [r'],\\
&\qquad\qquad z_{g',l'} \leftarrow \eta^\star_{g',l'} \mid g' \in G^{(s'+1)}, l' \in [s']])\\
&= \text{(by definition of } nest_{f'} \text{ (cf. Construction 3.1) and substitution)}\\
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair},\\
&\quad g''\ \phi_{k'}[y_k \leftarrow \star\ y_k \mid k \in [r]]\\
&\qquad (\overline{k'{}_{f'}1_{g''}}\ u\ nest_{f'}(1,g_1,\mathcal{C}\cup\{(k',g'')\}) \cdots nest_{f'}(r',g_\mu,\mathcal{C}\cup\{(k',g'')\})\\
&\qquad\qquad z_{g_1,1}\cdots z_{g_\mu,s_\mu})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid b \in [r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta^\star_{g',l'} \mid g' \in G^{(s'+1)}, l' \in [s']]\\
&\qquad \cdots\\
&\qquad (\overline{k'{}_{f'}s''_{g''}}\ u\ nest_{f'}(1,g_1,\mathcal{C}\cup\{(k',g'')\}) \cdots nest_{f'}(r',g_\mu,\mathcal{C}\cup\{(k',g'')\})\\
&\qquad\qquad z_{g_1,1}\cdots z_{g_\mu,s_\mu})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid b \in [r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta^\star_{g',l'} \mid g' \in G^{(s'+1)}, l' \in [s']])\\
&= \text{(see below)}
\end{aligned}
$$

$$
\begin{aligned}
&nf(\Rightarrow_{R_2\cup Pre\cup Pair},\\
&\quad g''\ \phi_{k'}\\
&\qquad(\overline{k'{}_{f'}1_{g''}}\ u\ nest_{f'}(1,g_1,\mathcal{C}\cup\{(k',g'')\})\ \cdots\ nest_{f'}(r',g_\mu,\mathcal{C}\cup\{(k',g'')\})\\
&\qquad\qquad z_{g_1,1}\cdots z_{g_\mu,s_\mu})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b \mid b\in[r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta_{g',l'} \mid g'\in G^{(s'+1)}, l'\in[s']]\\
&\qquad\cdots\\
&\qquad(\overline{k'{}_{f'}s''{}_{g''}}\ u\ nest_{f'}(1,g_1,\mathcal{C}\cup\{(k',g'')\})\ \cdots\ nest_{f'}(r',g_\mu,\mathcal{C}\cup\{(k',g'')\})\\
&\qquad\qquad z_{g_1,1}\cdots z_{g_\mu,s_\mu})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b \mid b\in[r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta_{g',l'} \mid g'\in G^{(s'+1)}, l'\in[s']])\\
&\quad= \text{(by definition of } nest_{f'} \text{ and substitution)}\\
&nf(\Rightarrow_{R_2\cup Pre\cup Pair}, nest_{f'}(k',g'',\mathcal{C})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b \mid b\in[r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta_{g',l'} \mid g'\in G^{(s'+1)}, l'\in[s']])
\end{aligned}
$$

The above gap can be closed by using induction hypothesis (a) for $\phi_{k'}$ if we establish for every $l\in[s'']$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair},\\
&\qquad(\overline{k'{}_{f'}l_{g''}}\ u\ nest_{f'}(1,g_1,\mathcal{C}\cup\{(k',g'')\})\ \cdots\ nest_{f'}(r',g_\mu,\mathcal{C}\cup\{(k',g'')\})\\
&\qquad\qquad z_{g_1,1}\cdots z_{g_\mu,s_\mu})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b[y_k\leftarrow \star\ y_k \mid k\in[r]] \mid b\in[r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta^\star_{g',l'} \mid g'\in G^{(s'+1)}, l'\in[s']])\\
&= nf(\Rightarrow_{R_2\cup Pre\cup Pair},\\
&\qquad(\overline{k'{}_{f'}l_{g''}}\ u\ nest_{f'}(1,g_1,\mathcal{C}\cup\{(k',g'')\})\ \cdots\ nest_{f'}(r',g_\mu,\mathcal{C}\cup\{(k',g'')\})\\
&\qquad\qquad z_{g_1,1}\cdots z_{g_\mu,s_\mu})[u \leftarrow u_i,\\
&\qquad\qquad\qquad y'_b \leftarrow \phi_b \mid b\in[r'],\\
&\qquad\qquad\qquad z_{g',l'} \leftarrow \eta_{g',l'} \mid g'\in G^{(s'+1)}, l'\in[s']]).
\end{aligned}
$$

But this follows directly (by substitution and normal forms) from the preconditions on the $\eta_{g_1,1},\ldots,\eta_{g_\mu,s_\mu},\eta^\star_{g_1,1},\ldots,\eta^\star_{g_\mu,s_\mu}$ and applications of the induction hypothesis for $\mathcal{C}\cup\{(k',g'')\}$. $\square$

Proof of Lemma 4.11

For fixed $\sigma\in\Sigma^{(p)}$ and $f\in F^{(r+1)}$ with $r\in\mathbb{N}_+$ we prove by induction over the prefix-order of paths $\pi\in paths(rhs_{M_1,f,\sigma})$ with $lab(rhs_{M_1,f,\sigma},\pi)\notin U_p$ that for every $g\in G^{(s+1)}$ and $l\in[s]$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k\leftarrow\star\, y_k\,\mid\, k\in[r]]}(1\pi,g,l))\\
&= nf(\Rightarrow_{R_2\cup Pre\cup Pair}, par_{M_2,rhs_{M_1,f,\sigma}}(\pi,g,l)).
\end{aligned}
$$

Lemma 4.11 then follows by $Y_r\cap U_p=\emptyset$ and $rhs_{M_1^{\to\diamond\star},f,\sigma} = \diamond\ rhs_{M_1,f,\sigma}[y_k\leftarrow\star\ y_k \mid k\in[r]]$ (cf. Definition 4.9). In the base case ε we have:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k\leftarrow\star\, y_k\,\mid\, k\in[r]]}(1,g,l))\\
&\quad= \text{(by definition of } par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k\leftarrow\star\, y_k\,\mid\, k\in[r]]} \text{ (cf. Construction 3.1),}\\
&\qquad \text{using that } rhs_{M_2^{\diamond\star\to\circ},g,\diamond} = \circ\ (g\ v_1\ z_1\cdots z_s) \text{ by Definition 4.9)}\\
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k\leftarrow\star\, y_k\,\mid\, k\in[r]]}(\varepsilon,g,l))
\end{aligned}
$$

$$
\begin{aligned}
&= \text{(by definition of } par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]} \text{ (cf. Construction 3.1))}\\
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, z_{g,l})\\
&= \text{(by normal forms)}\\
&nf(\Rightarrow_{R_2\cup Pre\cup Pair}, z_{g,l})\\
&= \text{(by definition of } par_{M_2,rhs_{M_1,f,\sigma}} \text{ (cf. Construction 3.1))}\\
&nf(\Rightarrow_{R_2\cup Pre\cup Pair}, par_{M_2,rhs_{M_1,f,\sigma}}(\varepsilon, g, l))
\end{aligned}
$$

The inductive case $\pi j \in paths(rhs_{M_1,f,\sigma})$ with $j \in \mathbb{N}_+$ and $lab(rhs_{M_1,f,\sigma}, \pi j) \notin U_p$ is shown by case distinction on $lab(rhs_{M_1,f,\sigma}, \pi) = lab(\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]], 1\pi)$:

$\underline{lab(rhs_{M_1,f,\sigma}, \pi) = \delta}$ for some $\delta \in \Delta^{(q)}$ and $j \in [q]$:

If there is no $(g\ v_j\ \cdots)$-call in the δ-rules of M_2 and hence also no such call in the δ-rules of $M_2^{\diamond\star\to\circ}$ (cf. Definition 4.9), then both sides of the equation are equal to nil by the definitions of $par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]}$ and $par_{M_2,rhs_{M_1,f,\sigma}}$ (cf. Construction 3.1).

Otherwise, if the unique such call—in both the δ-rules of M_2 and of $M_2^{\diamond\star\to\circ}$—looks, with $g' \in G^{(s'+1)}$ and $\psi_1, \ldots, \psi_s \in RHS(G, \Omega, V_q, Z_{s'})$, as follows:

$$g'\ (\delta\ v_1 \cdots v_q)\ z_1 \cdots z_{s'} \to \cdots (g\ v_j\ \psi_1 \cdots \psi_s) \cdots ,$$

then:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]}(1\pi j, g, l))\\
&= \text{(by definition of } par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]} \text{ (cf. Construction 3.1))}\\
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \psi_l[v_{j'} \leftarrow sub(\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]], 1\pi j') \mid j' \in [q],\\
&\qquad z_{l'} \leftarrow par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]}(1\pi, g', l') \mid l' \in [s']])\\
&= \text{(by definition and properties of } sub)\\
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair}, \psi_l[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j')[y_k \leftarrow \star\, y_k \mid k \in [r]] \mid j' \in [q],\\
&\qquad z_{l'} \leftarrow par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]}(1\pi, g', l') \mid l' \in [s']])\\
&= \text{(see below)}\\
&nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \psi_l[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j') \mid j' \in [q],\\
&\qquad z_{l'} \leftarrow par_{M_2,rhs_{M_1,f,\sigma}}(\pi, g', l') \mid l' \in [s']])\\
&= \text{(by definition of } par_{M_2,rhs_{M_1,f,\sigma}} \text{ (cf. Construction 3.1))}\\
&nf(\Rightarrow_{R_2\cup Pre\cup Pair}, par_{M_2,rhs_{M_1,f,\sigma}}(\pi j, g, l))
\end{aligned}
$$

Above we used that for every $\psi \in RHS(G, \Omega, V_q, Z_{s'})$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ}\cup Pre\cup Pair},\\
&\quad \psi[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j')[y_k \leftarrow \star\, y_k \mid k \in [r]] \mid j' \in [q],\\
&\quad z_{l'} \leftarrow par_{M_2^{\diamond\star\to\circ},\diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k\in[r]]}(1\pi, g', l') \mid l' \in [s']])\\
&= nf(\Rightarrow_{R_2\cup Pre\cup Pair}, \psi[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j') \mid j' \in [q],\\
&\qquad z_{l'} \leftarrow par_{M_2,rhs_{M_1,f,\sigma}}(\pi, g', l') \mid l' \in [s']]),
\end{aligned}
$$

which can be proved by structural induction. The validity in the base case $\psi \in Z_{s'}$ follows from the induction hypothesis for π. One recursive case, $\psi = \omega\ \psi'_1 \cdots \psi'_n$, is straightforward. The validity in the remaining case is proved as follows.

$\underline{\psi = g''\ v_d\ \psi'_1 \cdots \psi'_{s''}}$ for some $g'' \in G^{(s''+1)}$, $v_d \in V_q$, and $\psi'_1, \ldots, \psi'_{s''} \in RHS(G, \Omega,$ $\underline{V_q, Z_{s'})}$:

$$
\begin{array}{l}
nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, \\
\quad (g''\ v_d\ \psi'_1 \cdots \psi'_{s''})[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j')[y_k \leftarrow \star\ y_k \mid k \in [r]] \quad \mid j' \in [q], \\
\qquad\qquad\qquad z_{l'} \leftarrow par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l') \mid l' \in [s']]) \\
\quad = \ \text{(by substitution)} \\
nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, \\
\quad g''\ sub(rhs_{M_1,f,\sigma}, \pi d)[y_k \leftarrow \star\ y_k \mid k \in [r]] \\
\qquad \psi'_1[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j')[y_k \leftarrow \star\ y_k \mid k \in [r]] \quad \mid j' \in [q], \\
\qquad\quad z_{l'} \leftarrow par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l') \mid l' \in [s']] \\
\qquad \ldots \\
\qquad \psi'_{s''}[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j')[y_k \leftarrow \star\ y_k \mid k \in [r]] \quad \mid j' \in [q], \\
\qquad\quad z_{l'} \leftarrow par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l') \mid l' \in [s']]) \\
\quad = \ \text{(by statement (a) from the proof of Lemma 4.10 for} \\
\qquad sub(rhs_{M_1,f,\sigma}, \pi d) \in RHS(F, \Delta, U_p, Y_r)\text{, using the induction hypotheses for} \\
\qquad \text{the } \psi'_1, \ldots, \psi'_{s''} \text{ to establish the necessary preconditions)} \\
nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, g''\ sub(rhs_{M_1,f,\sigma}, \pi d) \\
\qquad\qquad\qquad \psi'_1[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j') \quad \mid j' \in [q], \\
\qquad\qquad\qquad\quad z_{l'} \leftarrow par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g', l') \mid l' \in [s']] \\
\qquad\qquad\qquad \ldots \\
\qquad\qquad\qquad \psi'_{s''}[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j') \quad \mid j' \in [q], \\
\qquad\qquad\qquad\quad z_{l'} \leftarrow par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g', l') \mid l' \in [s']]) \\
\quad = \ \text{(by substitution)} \\
nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, (g''\ v_d\ \psi'_1 \cdots \psi'_{s''})[v_{j'} \leftarrow sub(rhs_{M_1,f,\sigma}, \pi j') \quad \mid j' \in [q], \\
\qquad\qquad\qquad\qquad z_{l'} \leftarrow par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g', l') \mid l' \in [s']])
\end{array}
$$

$\underline{lab(rhs_{M_1,f,\sigma}, \pi) = f'}$ for some $f' \in F^{(r'+1)}$, $j - 1 \in [r']$,
$\underline{\text{and } lab(rhs_{M_1,f,\sigma}, \pi 1)} = lab(\diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]], 1\pi 1) = u_i \in U_p$:

$$
\begin{array}{l}
nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi j, g, l)) \\
\quad = \ \text{(by definition of } par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]} \text{ (cf. Construction 3.1))} \\
nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, \\
\quad (\overline{(j-1)_{f'} l_g}\ u\ nest_{f'}(1, g_1, \{(j-1, g)\})\ \cdots\ nest_{f'}(r', g_\mu, \{(j-1, g)\}) \\
\qquad\qquad z_{g_1,1} \cdots z_{g_\mu, s_\mu}) \\
\quad [u \quad \leftarrow u_i, \\
\quad\ y'_b \quad \leftarrow sub(\diamond\ rhs_{M_1,f,\sigma}[y_k \leftarrow \star\ y_k \mid k \in [r]], 1\pi(b+1)) \mid b \in [r'], \\
\quad\ z_{g',l'} \leftarrow par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l') \mid g' \in G^{(s'+1)}, l' \in [s']]) \\
\quad = \ \text{(by definition and properties of } sub\text{)} \\
nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, \\
\quad (\overline{(j-1)_{f'} l_g}\ u\ nest_{f'}(1, g_1, \{(j-1, g)\})\ \cdots\ nest_{f'}(r', g_\mu, \{(j-1, g)\}) \\
\qquad\qquad z_{g_1,1} \cdots z_{g_\mu, s_\mu}) \\
\quad [u \quad \leftarrow u_i, \\
\quad\ y'_b \quad \leftarrow sub(rhs_{M_1,f,\sigma}, \pi(b+1))[y_k \leftarrow \star\ y_k \mid k \in [r]] \mid b \in [r'], \\
\quad\ z_{g',l'} \leftarrow par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l') \quad \mid g' \in G^{(s'+1)}, l' \in [s']]) \\
\quad = \ \text{(see below)}
\end{array}
$$

$$
\begin{aligned}
&nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, (\overline{(j-1)_{f'} l_g}\ u\ nest_{f'}(1, g_1, \{(j-1, g)\})\ \cdots\ nest_{f'}(r', g_\mu, \{(j-1, g)\}) \\
&\qquad\qquad z_{g_1,1} \cdots z_{g_\mu, s_\mu}) \\
&\qquad [u \leftarrow u_i, \\
&\qquad\ y'_b \leftarrow sub(rhs_{M_1,f,\sigma}, \pi(b+1)) \mid b \in [r'], \\
&\qquad\ z_{g',l'} \leftarrow par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g', l') \mid g' \in G^{(s'+1)}, l' \in [s']]) \\
&= \text{(by definition of } par_{M_2, rhs_{M_1,f,\sigma}} \text{ (cf. Construction 3.1))} \\
&nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, par_{M_2, rhs_{M_1,f,\sigma}}(\pi j, g, l))
\end{aligned}
$$

Since for every $g' \in G^{(s'+1)}$ and $l' \in [s']$,

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l')) \\
&= nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g', l'))
\end{aligned}
$$

by the induction hypothesis for π, the above gap can be closed (by substitution and normal forms) if we establish for every $k' \in [r']$ and $g'' \in G$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_2^{\diamond\star\to\circ} \cup Pre \cup Pair}, \\
&\quad nest_{f'}(k', g'', \{(j-1, g)\}) \\
&\quad [u \leftarrow u_i, \\
&\quad\ y'_b \leftarrow sub(rhs_{M_1,f,\sigma}, \pi(b+1))[y_k \leftarrow \star\, y_k \mid k \in [r]] \mid b \in [r'], \\
&\quad\ z_{g',l'} \leftarrow par_{M_2^{\diamond\star\to\circ}, \diamond\, rhs_{M_1,f,\sigma}[y_k \leftarrow \star\, y_k \mid k \in [r]]}(1\pi, g', l') \mid g' \in G^{(s'+1)}, l' \in [s']]) \\
&= nf(\Rightarrow_{R_2 \cup Pre \cup Pair}, nest_{f'}(k', g'', \{(j-1, g)\}) \\
&\qquad [u \leftarrow u_i, \\
&\qquad\ y'_b \leftarrow sub(rhs_{M_1,f,\sigma}, \pi(b+1)) \mid b \in [r'], \\
&\qquad\ z_{g',l'} \leftarrow par_{M_2, rhs_{M_1,f,\sigma}}(\pi, g', l') \mid g' \in G^{(s'+1)}, l' \in [s']]).
\end{aligned}
$$

But this is an instance of statement (b) from the proof of Lemma 4.10 for $sub(rhs_{M_1,f,\sigma}, \pi 2), \ldots, sub(rhs_{M_1,f,\sigma}, \pi(r'+1)) \in RHS(F, \Delta, U_p, Y_r)$, where the statements obtained by the induction hypothesis for π establish the necessary preconditions on the values substituted for the $z_{g',l'}$. $\square$

Proof of Lemma 4.16

We prove the following two statements:

(a) <u>for every $t' \in T_{\Delta \cup \{\diamond,\star\}}$:</u>
For every $g \in G^{(s+1)}$ and $\eta_1^\circ, \ldots, \eta_s^\circ, \eta_1^\bullet, \ldots, \eta_s^\bullet, \eta_1^{\circ-\bullet}, \ldots, \eta_s^{\circ-\bullet} \in T_{G \cup \Delta \cup \Omega \cup \{\diamond,\star,\circ,\bullet\}}$ with $|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, \eta_l^\circ)|_\circ - |nf(\Rightarrow_{R_2^{\diamond}\to\bullet}, \eta_l^\bullet)|_\bullet = |nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^{\circ-\bullet})|_{\circ-\bullet}$ for every $l \in [s]$:
$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, g\ t'\ \eta_1^\circ \cdots \eta_s^\circ)|_\circ - |nf(\Rightarrow_{R_2^{\diamond}\to\bullet}, g\ t'[\star \twoheadleftarrow id]\ \eta_1^\bullet \cdots \eta_s^\bullet)|_\bullet \\
&= |nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, g\ t'[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet \cdot \delta \mid \delta \in \Delta]\ \eta_1^{\circ-\bullet} \cdots \eta_s^{\circ-\bullet})|_{\circ-\bullet}.
\end{aligned}
$$

(b) <u>for every $q \in \mathbb{N}$ and $t'_1, \ldots, t'_q \in T_{\Delta \cup \{\diamond,\star\}}$:</u>
For every $s \in \mathbb{N}$ and $\eta_1^\circ, \ldots, \eta_s^\circ, \eta_1^\bullet, \ldots, \eta_s^\bullet, \eta_1^{\circ-\bullet}, \ldots, \eta_s^{\circ-\bullet} \in T_{G \cup \Delta \cup \Omega \cup \{\diamond,\star,\circ,\bullet\}}$ with $|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, \eta_l^\circ)|_\circ - |nf(\Rightarrow_{R_2^{\diamond}\to\bullet}, \eta_l^\bullet)|_\bullet = |nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^{\circ-\bullet})|_{\circ-\bullet}$ for every $l \in [s]$, for

every $\psi \in RHS(G,\Omega,V_q,Z_s)$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, \psi[v_j \leftarrow t'_j \mid j\in[q],\\
&\qquad\qquad z_l \leftarrow \eta^\circ_l \mid l\in[s]])|_\circ\\
-\ &|nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, \psi[v_j \leftarrow t'_j[\star \twoheadleftarrow id] \mid j\in[q],\\
&\qquad\qquad z_l \leftarrow \eta^\bullet_l \qquad\qquad \mid l\in[s]])|_\bullet\\
=\ &|nf(\Rightarrow_{R_2^{\star\bullet\to\circ\bullet}}, \psi[v_j \leftarrow t'_j[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta] \mid j\in[q],\\
&\qquad\qquad z_l \leftarrow \eta^{\circ-\bullet}_l \qquad\qquad\qquad\qquad\qquad\qquad \mid l\in[s]])|_{\circ-\bullet}.
\end{aligned}$$

by simultaneous induction. Lemma 4.16 then follows from statement (b) with $q=1$, $t'_1 = t'$, $s=0$, and $\psi = e_{M_2}$.

(a) $\Leftarrow$ (b) : by case analysis on $t' \in T_{\Delta\cup\{\diamond,\star\}}$:

$\underline{t' = \diamond\ t'_1}$ for some $t'_1 \in T_{\Delta\cup\{\diamond,\star\}}$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, g\ (\diamond\ t'_1)\ \eta^\circ_1\cdots\eta^\circ_s)|_\circ - |nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, g\ (\diamond\ t'_1)[\star\twoheadleftarrow id]\ \eta^\bullet_1\cdots\eta^\bullet_s)|_\bullet\\
&\quad= \text{(by substitution, } \Rightarrow_{R_2^{\diamond\star\to\circ}} \text{ and } \Rightarrow_{R_2^{\diamond\to\bullet}}\text{, using that}\\
&\qquad\quad rhs_{M_2^{\diamond\star\to\circ},g,\diamond} = \circ\ (g\ v_1\ z_1\cdots z_s) \text{ and } rhs_{M_2^{\diamond\to\bullet},g,\diamond} = \bullet\ (g\ v_1\ z_1\cdots z_s) \text{ by}\\
&\qquad\quad \text{Definitions 4.9 and 4.4, by normal forms and definitions of } |\,.\,|_\circ \text{ and } |\,.\,|_\bullet)\\
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, g\ t'_1\ \eta^\circ_1\cdots\eta^\circ_s)|_\circ - |nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, g\ t'_1[\star\twoheadleftarrow id]\ \eta^\bullet_1\cdots\eta^\bullet_s)|_\bullet\\
&\quad= \text{(by induction hypothesis (a) for } t'_1)\\
&|nf(\Rightarrow_{R_2^{\star\bullet\to\circ\bullet}}, g\ t'_1[\diamond\twoheadleftarrow id][\delta\twoheadleftarrow\bullet\cdot\delta \mid \delta\in\Delta]\ \eta^{\circ-\bullet}_1\cdots\eta^{\circ-\bullet}_s)|_{\circ-\bullet}\\
&\quad= \text{(by substitution)}\\
&|nf(\Rightarrow_{R_2^{\star\bullet\to\circ\bullet}}, g\ (\diamond\ t'_1)[\diamond\twoheadleftarrow id][\delta\twoheadleftarrow\bullet\cdot\delta \mid \delta\in\Delta]\ \eta^{\circ-\bullet}_1\cdots\eta^{\circ-\bullet}_s)|_{\circ-\bullet}
\end{aligned}$$

$\underline{t' = \star\ t'_1}$ for some $t'_1 \in T_{\Delta\cup\{\diamond,\star\}}$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, g\ (\star\ t'_1)\ \eta^\circ_1\cdots\eta^\circ_s)|_\circ - |nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, g\ (\star\ t'_1)[\star\twoheadleftarrow id]\ \eta^\bullet_1\cdots\eta^\bullet_s)|_\bullet\\
&\quad= \text{(by } \Rightarrow_{R_2^{\diamond\star\to\circ}}\text{, using that } rhs_{M_2^{\diamond\star\to\circ},g,\star} = g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s) \text{ by Definition 4.9,}\\
&\qquad\quad \text{and by substitution)}\\
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, g\ t'_1\ (\circ\ \eta^\circ_1)\ \cdots\ (\circ\ \eta^\circ_s))|_\circ - |nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, g\ t'_1[\star\twoheadleftarrow id]\ \eta^\bullet_1\cdots\eta^\bullet_s)|_\bullet\\
&\quad= \text{(by induction hypothesis (a) for } t'_1\text{, using that for every } l\in[s],\\
&\qquad\quad |nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, \circ\ \eta^\circ_l)|_\circ - |nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, \eta^\bullet_l)|_\bullet = |nf(\Rightarrow_{R_2^{\star\bullet\to\circ\bullet}}, \circ\ \eta^{\circ-\bullet}_l)|_{\circ-\bullet}\text{, which}\\
&\qquad\quad \text{follows from the preconditions on the } \eta^\circ_1,\ldots,\eta^\circ_s,\eta^\bullet_1,\ldots,\eta^\bullet_s,\eta^{\circ-\bullet}_1,\ldots,\eta^{\circ-\bullet}_s \text{ by}\\
&\qquad\quad \text{the definitions of } |\,.\,|_\circ \text{ and } |\,.\,|_{\circ-\bullet})\\
&|nf(\Rightarrow_{R_2^{\star\bullet\to\circ\bullet}}, g\ t'_1[\diamond\twoheadleftarrow id][\delta\twoheadleftarrow\bullet\cdot\delta \mid \delta\in\Delta]\ (\circ\ \eta^{\circ-\bullet}_1)\ \cdots\ (\circ\ \eta^{\circ-\bullet}_s))|_{\circ-\bullet}\\
&\quad= \text{(by substitution and } \Rightarrow_{R_2^{\star\bullet\to\circ\bullet}},\\
&\qquad\quad \text{using that } rhs_{M_2^{\star\bullet\to\circ\bullet},g,\star} = g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s) \text{ by Definition 4.14)}\\
&|nf(\Rightarrow_{R_2^{\star\bullet\to\circ\bullet}}, g\ (\star\ t'_1)[\diamond\twoheadleftarrow id][\delta\twoheadleftarrow\bullet\cdot\delta \mid \delta\in\Delta]\ \eta^{\circ-\bullet}_1\cdots\eta^{\circ-\bullet}_s)|_{\circ-\bullet}
\end{aligned}$$

$\underline{t' = \delta'\ t'_1\cdots t'_q}$ for some $\delta' \in \Delta^{(q)}$ and $t'_1,\ldots,t'_q \in T_{\Delta\cup\{\diamond,\star\}}$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, g\ (\delta'\ t'_1\cdots t'_q)\ \eta^\circ_1\cdots\eta^\circ_s)|_\circ - |nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, g\ (\delta'\ t'_1\cdots t'_q)[\star\twoheadleftarrow id]\ \eta^\bullet_1\cdots\eta^\bullet_s)|_\bullet\\
&\quad= \text{(by substitution, } \Rightarrow_{R_2^{\diamond\star\to\circ}} \text{ and } \Rightarrow_{R_2^{\diamond\to\bullet}}\text{, using that}\\
&\qquad\quad rhs_{M_2^{\diamond\star\to\circ},g,\delta'} = rhs_{M_2,g,\delta'} \text{ and } rhs_{M_2^{\diamond\to\bullet},g,\delta'} = \bullet\ rhs_{M_2,g,\delta'} \text{ by Definitions 4.9}\\
&\qquad\quad \text{and 4.4, by substitution, normal form, and definition of } |\,.\,|_\bullet)\\
&-1 + |nf(\Rightarrow_{R_2^{\diamond\star\to\circ}}, rhs_{M_2,g,\delta'}[v_j \leftarrow t'_j \mid j\in[q],\\
&\qquad\qquad\qquad\qquad z_l \leftarrow \eta^\circ_l \mid l\in[s]])|_\circ\\
&\quad -|nf(\Rightarrow_{R_2^{\diamond\to\bullet}}, rhs_{M_2,g,\delta'}[v_j \leftarrow t'_j[\star\twoheadleftarrow id] \mid j\in[q],\\
&\qquad\qquad\qquad\qquad z_l \leftarrow \eta^\bullet_l \mid l\in[s]])|_\bullet
\end{aligned}$$

$$
\begin{aligned}
&= \text{(by induction hypothesis (b) with } \psi = rhs_{M_2,g,\delta'}) \\
&-1 + |nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, rhs_{M_2,g,\delta'}[v_j \leftarrow t'_j[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta] \mid j\in[q], \\
&\qquad\qquad z_l \leftarrow \eta_l^{\circ-\bullet} \mid l \in [s]])|_{\circ-\bullet} \\
&= \text{(by } \Rightarrow^2_{R_2^{\star}\bullet\to\circ\bullet}\text{, using that } rhs_{M_2^{\star}\bullet\to\circ\bullet,g,\bullet} = \bullet\ (g\ v_1\ z_1\cdots z_s) \text{ and} \\
&\qquad rhs_{M_2^{\star}\bullet\to\circ\bullet,g,\delta'} = rhs_{M_2,g,\delta'} \text{ by Definition 4.14,} \\
&\qquad \text{by normal form and definition of } |\,.\,|_{\circ-\bullet}) \\
&|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, g\ (\bullet\ (\delta'\ t'_1[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta] \\
&\qquad\qquad \cdots \\
&\qquad\qquad t'_q[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]))\ \eta_1^{\circ-\bullet}\cdots\eta_s^{\circ-\bullet})|_{\circ-\bullet} \\
&= \text{(by substitution)} \\
&|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, g\ (\delta'\ t'_1\cdots t'_q)[\diamond \twoheadleftarrow id][\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]\ \eta_1^{\circ-\bullet}\cdots\eta_s^{\circ-\bullet})|_{\circ-\bullet}
\end{aligned}
$$

(b) $\Leftarrow$ (a) : for fixed $s \in \mathbb{N}$ and $\eta_1^{\circ},\ldots,\eta_s^{\circ},\eta_1^{\bullet},\ldots,\eta_s^{\bullet},\eta_1^{\circ-\bullet},\ldots,\eta_s^{\circ-\bullet} \in T_{G\cup\Delta\cup\Omega\cup\{\diamond,\star,\circ,\bullet\}}$ with $|nf(\Rightarrow_{R_2^{\diamond\star\to\circ}},\eta_l^{\circ})|_{\circ} - |nf(\Rightarrow_{R_2^{\diamond}\to\bullet},\eta_l^{\bullet})|_{\bullet} = |nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet},\eta_l^{\circ-\bullet})|_{\circ-\bullet}$ for every $l \in [s]$, by structural induction on $\psi \in RHS(G,\Omega,V_q,Z_s)$. The validity in the base case $\psi \in Z_s$ follows from the preconditions on the $\eta_1^{\circ},\ldots,\eta_s^{\circ},\eta_1^{\bullet},\ldots,\eta_s^{\bullet},\eta_1^{\circ-\bullet},\ldots,\eta_s^{\circ-\bullet}$. The one recursive case, $\psi = \omega\ \psi_1\cdots\psi_n$, is straightforward and in the other one, $\psi = g\ v_j\ \psi_1\cdots\psi_{s'}$, the induction hypotheses for the $\psi_1,\ldots,\psi_{s'}$ establish the preconditions necessary to apply induction hypothesis (a) for t'_j, which suffices. $\square$

Proof of Lemma 4.22

For fixed $\kappa : \{(f,k) \mid f \in F^{(r+1)}, k \in [r]\} \longrightarrow \{0,\ldots,s_{max}\}$ we prove the following two statements:

(a) for every $t \in T_\Sigma$:
For every $f \in F^{(r+1)}$ and $\theta_1^{\circ},\ldots,\theta_r^{\circ},\theta_1^{\star},\ldots,\theta_r^{\star} \in T_{F\cup\Sigma\cup\Delta\cup\{\bullet,\circ,\star\}}$ with $nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}},\theta_k^{\circ}) = nf(\Rightarrow_{R_1^{\to\star\bullet}},\theta_k^{\star})[\star \twoheadleftarrow \circ^{s_{max}}]$ for every $k \in [r]$:
$nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, f\ t\ (\circ^{\kappa_{f,1}}\ \theta_1^{\circ})\ \cdots\ (\circ^{\kappa_{f,r}}\ \theta_r^{\circ})) = nf(\Rightarrow_{R_1^{\to\star\bullet}}, f\ t\ \theta_1^{\star}\cdots\theta_r^{\star})[\star \twoheadleftarrow \circ^{s_{max}}]$.

(b) for every $p \in \mathbb{N}$ and $t_1,\ldots,t_p \in T_\Sigma$:
For every $r \in \mathbb{N}$ and $\theta_1^{\circ},\ldots,\theta_r^{\circ},\theta_1^{\star},\ldots,\theta_r^{\star} \in T_{F\cup\Sigma\cup\Delta\cup\{\bullet,\circ,\star\}}$ with $nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}},\theta_k^{\circ}) = nf(\Rightarrow_{R_1^{\to\star\bullet}},\theta_k^{\star})[\star \twoheadleftarrow \circ^{s_{max}}]$ for every $k \in [r]$, for every $\phi \in RHS(F,\Delta\cup\{\bullet\},U_p,Y_r)$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, \phi[f' \twoheadleftarrow f'\cdot(id \times \circ^{\kappa_{f',1}}\cdots\times\circ^{\kappa_{f',r'}}) \mid f' \in F^{(r'+1)}] \\
&\qquad [u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^{\circ},\ldots,\theta_r^{\circ}]) \\
&= nf(\Rightarrow_{R_1^{\to\star\bullet}}, \phi[u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^{\star},\ldots,\theta_r^{\star}])[\star \twoheadleftarrow \circ^{s_{max}}].
\end{aligned}
$$

by simultaneous induction. Lemma 4.22 then follows from statement (b) with $p = 1$, $t_1 = t$, $r = 0$, and $\phi = e_{M_1}[\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]$, taking into account the definitions of $e_{M_1^{\to\circ\bullet,\kappa}}$ and $e_{M_1^{\to\star\bullet}}$ (according to Definitions 4.20 and 4.14, respectively).

(a) $\Leftarrow$ (b) : Assume $t = \sigma\ t_1\cdots t_p$ for $\sigma \in \Sigma^{(p)}$ and $t_1,\ldots,t_p \in T_\Sigma$. We calculate:

$$
\begin{aligned}
&nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, f\ (\sigma\ t_1\cdots t_p)\ (\circ^{\kappa_{f,1}}\ \theta_1^{\circ})\ \cdots\ (\circ^{\kappa_{f,r}}\ \theta_r^{\circ})) \\
&= \text{(by } \Rightarrow_{R_1^{\to\circ\bullet,\kappa}}\text{, using the definition of } rhs_{M_1^{\to\circ\bullet,\kappa},f,\sigma} \text{ according to Definition 4.20)} \\
&nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, rhs_{M_1,f,\sigma}[\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta] \\
&\qquad [f' \twoheadleftarrow f'\cdot(id \times \circ^{\kappa_{f',1}}\cdots\times\circ^{\kappa_{f',r'}}) \mid f' \in F^{(r'+1)}] \\
&\qquad [y_k \leftarrow \circ^{\overline{\kappa}_{f,k}}\ y_k \mid k \in [r]][u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad y_k \leftarrow \circ^{\kappa_{f,k}}\ \theta_k^{\circ} \mid k \in [r]])
\end{aligned}
$$

$$
\begin{aligned}
&= \text{(by composition of substitutions)}\\
&nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, rhs_{M_1,f,\sigma}[\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]\\
&\qquad [f' \twoheadleftarrow f'\cdot(id\times \circ^{\kappa_{f',1}}\cdots\times\circ^{\kappa_{f',r'}}) \mid f'\in F^{(r'+1)}]\\
&\qquad [u_i \leftarrow t_i \mid i\in[p],\\
&\qquad\ y_k \leftarrow \circ^{\overline{\kappa}_{f,k}}\ (\circ^{\kappa_{f,k}}\ \theta_k^\circ) \mid k\in[r]])\\
&= \text{(see below)}\\
&nf(\Rightarrow_{R_1^{\to\star\bullet}}, rhs_{M_1,f,\sigma}[\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]\\
&\qquad [u_i \leftarrow t_i \mid i\in[p],\\
&\qquad\ y_k \leftarrow \star\ \theta_k^\star \mid k\in[r]])[\star \twoheadleftarrow \circ^{s_{max}}]\\
&= \text{(by composition of substitutions)}\\
&nf(\Rightarrow_{R_1^{\to\star\bullet}}, rhs_{M_1,f,\sigma}[\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]\\
&\qquad [y_k \leftarrow \star\ y_k \mid k\in[r]][u_i \leftarrow t_i \mid i\in[p],\\
&\qquad\qquad y_k \leftarrow \theta_k^\star \mid k\in[r]])[\star \twoheadleftarrow \circ^{s_{max}}]\\
&= \text{(by } \Rightarrow_{R_1^{\to\star\bullet}}\text{, using the definition of } rhs_{M_1^{\to\star\bullet},f,\sigma} \text{ according to Definition 4.14)}\\
&nf(\Rightarrow_{R_1^{\to\star\bullet}}, f\ (\sigma\ t_1\cdots t_p)\ \theta_1^\star\cdots\theta_r^\star)[\star \twoheadleftarrow \circ^{s_{max}}]
\end{aligned}
$$

The above gap can be closed by using induction hypothesis (b) with $\phi = rhs_{M_1,f,\sigma}[\delta \twoheadleftarrow \bullet\cdot\delta \mid \delta\in\Delta]$ if we establish for every $k\in[r]$:

$$nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, \circ^{\overline{\kappa}_{f,k}}\ (\circ^{\kappa_{f,k}}\ \theta_k^\circ)) = nf(\Rightarrow_{R_1^{\to\star\bullet}}, \star\ \theta_k^\star)[\star \twoheadleftarrow \circ^{s_{max}}].$$

But this follows directly (by normal forms and substitution) from the preconditions on the $\theta_1^\circ,\ldots,\theta_r^\circ,\theta_1^\star,\ldots,\theta_r^\star$ and the fact that $\overline{\kappa}_{f,k}+\kappa_{f,k} = s_{max}$ for every $k\in[r]$.

(b) $\Leftarrow$ (a) : for fixed $r\in\mathbb{N}$ and $\theta_1^\circ,\ldots,\theta_r^\circ,\theta_1^\star,\ldots,\theta_r^\star \in T_{F\cup\Sigma\cup\Delta\cup\{\bullet,\circ,\star\}}$ with $nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, \theta_k^\circ) = nf(\Rightarrow_{R_1^{\to\star\bullet}}, \theta_k^\star)[\star \twoheadleftarrow \circ^{s_{max}}]$ for every $k\in[r]$, by structural induction on $\phi\in RHS(F,\Delta\cup\{\bullet\},U_p,Y_r)$. The validity in the base case $\phi\in Y_r$ follows from the preconditions on the $\theta_1^\circ,\ldots,\theta_r^\circ,\theta_1^\star,\ldots,\theta_r^\star$. Two recursive cases, $\phi=\delta\ \phi_1\cdots\phi_n$ and $\phi=\bullet\ \phi_1$, are straightforward. The validity in the remaining case is proved as follows.

$\underline{\phi = f''\ u_i\ \phi_1\cdots\phi_{r''}}$ for some $f''\in F^{(r''+1)}$, $u_i\in U_p$, and $\phi_1,\ldots,\phi_{r''}\in RHS(F,\Delta\cup\{\bullet\},U_p,Y_r)$:

$$
\begin{aligned}
&nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, (f''\ u_i\ \phi_1\cdots\phi_{r''})[f' \twoheadleftarrow f'\cdot(id\times\circ^{\kappa_{f',1}}\cdots\times\circ^{\kappa_{f',r'}}) \mid f'\in F^{(r'+1)}]\\
&\qquad [u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^\circ,\ldots,\theta_r^\circ])\\
&= \text{(by substitution)}\\
&nf(\Rightarrow_{R_1^{\to\circ\bullet,\kappa}}, f''\ t_i\ (\circ^{\kappa_{f'',1}}\ \phi_1[f' \twoheadleftarrow f'\cdot(id\times\circ^{\kappa_{f',1}}\cdots\times\circ^{\kappa_{f',r'}}) \mid f'\in F^{(r'+1)}]\\
&\qquad [u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^\circ,\ldots,\theta_r^\circ])\\
&\qquad \cdots\\
&\qquad (\circ^{\kappa_{f'',r''}}\ \phi_{r''}[f' \twoheadleftarrow f'\cdot(id\times\circ^{\kappa_{f',1}}\cdots\times\circ^{\kappa_{f',r'}}) \mid f'\in F^{(r'+1)}]\\
&\qquad [u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^\circ,\ldots,\theta_r^\circ]))\\
&= \text{(by induction hypothesis (a) for } t_i\text{, using the induction hypotheses for the}\\
&\qquad \phi_1,\ldots,\phi_{r''} \text{ to establish the necessary preconditions)}\\
&nf(\Rightarrow_{R_1^{\to\star\bullet}}, f''\ t_i\ \phi_1[u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^\star,\ldots,\theta_r^\star]\\
&\qquad \cdots\\
&\qquad \phi_{r''}[u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^\star,\ldots,\theta_r^\star])[\star \twoheadleftarrow \circ^{s_{max}}]\\
&= \text{(by substitution)}
\end{aligned}
$$

$$nf(\Rightarrow_{R_1^{\to\star\bullet}}, \\ (f''\ u_i\ \phi_1 \cdots \phi_{r''})[u_1,\ldots,u_p,y_1,\ldots,y_r \leftarrow t_1,\ldots,t_p,\theta_1^\star,\ldots,\theta_r^\star])[\star \twoheadleftarrow \circ^{s_{max}}] \quad \square$$

Proof of Lemma 4.23

For a context-linear mtt M_2 we prove the following two statements:

(a) for every $t' \in T_{\Delta\cup\{\star,\bullet\}}$:
For every $g \in G^{(s+1)}$ and $\eta_1^\circ,\ldots,\eta_s^\circ,\eta_1,\ldots,\eta_s \in T_{G\cup\Delta\cup\Omega\cup\{\star,\bullet,\circ\}}$:

$$|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, g\ t'\ \eta_1^\circ \cdots \eta_s^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, g\ t'[\star \twoheadleftarrow \circ^{s_{max}}]\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$
$$\leq \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t'[\star,\bullet \twoheadleftarrow id, id]\ z_1\cdots z_s)|_{z_l} * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}).$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then equality holds.

(b) for every $q \in \mathbb{N}$ and $t_1',\ldots,t_q' \in T_{\Delta\cup\{\star,\bullet\}}$:
For every $s \in \mathbb{N}$, $\eta_1^\circ,\ldots,\eta_s^\circ,\eta_1,\ldots,\eta_s \in T_{G\cup\Delta\cup\Omega\cup\{\star,\bullet,\circ\}}$, and $\psi \in RHS(G,\Omega,V_q,Z_s)$:

$$|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \psi[v_j \leftarrow t_j' \mid j \in [q],\ z_l \leftarrow \eta_l^\circ \mid l \in [s]])|_{\circ-\bullet}$$
$$- |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \psi[v_j \leftarrow t_j'[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q],\ z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ-\bullet}$$
$$\leq \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, \psi[v_j \leftarrow t_j'[\star,\bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l} * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}).$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then equality holds.

by simultaneous induction. Lemma 4.23 then follows from statement (b) with $q = 1$, $t_1' = t'$, $s = 0$, and $\psi = e_{M_2}$, taking into account that an empty sum is 0.

(a) $\Leftarrow$ (b) : by case analysis on $t' \in T_{\Delta\cup\{\star,\bullet\}}$:

$t' = \star\ t_1'$ for some $t_1' \in T_{\Delta\cup\{\star,\bullet\}}$:

$$|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, g\ (\star\ t_1')\ \eta_1^\circ \cdots \eta_s^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, g\ (\star\ t_1')[\star \twoheadleftarrow \circ^{s_{max}}]\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$

$=$ (by substitution, $\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}$ and $\Rightarrow^{s_{max}}_{R_2^{\circ}\bullet\to\circ\bullet}$, using that $rhs_{M_2^{\star}\bullet\to\circ\bullet, g,\star} = g\ v_1\ (\circ\ z_1)\ \cdots\ (\circ\ z_s)$ and $rhs_{M_2^{\circ}\bullet\to\circ\bullet, g,\circ} = \circ\ (g\ v_1\ z_1 \cdots z_s)$ by Definitions 4.14 and 4.20, by normal form and definition of $|\,.\,|_{\circ-\bullet}$)

$$-s_{max} + |nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, g\ t_1'\ (\circ\ \eta_1^\circ)\ \cdots\ (\circ\ \eta_s^\circ))|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, g\ t_1'[\star \twoheadleftarrow \circ^{s_{max}}]\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$

$\leq$ (by induction hypothesis (a) for t_1')

$$-s_{max} + \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t_1'[\star,\bullet \twoheadleftarrow id, id]\ z_1\cdots z_s)|_{z_l} * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \circ\ \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})$$

$=$ (by normal form, definition of $|\,.\,|_{\circ-\bullet}$, and sum manipulation)

$$-s_{max} + \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t_1'[\star,\bullet \twoheadleftarrow id, id]\ z_1\cdots z_s)|_{z_l} * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})$$
$$+ \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t_1'[\star,\bullet \twoheadleftarrow id, id]\ z_1\cdots z_s)|_{z_l}$$

$\leq$ (by Lemma 2.8 for the context-linear mtt M_2)

$$-s_{max} + \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t_1'[\star,\bullet \twoheadleftarrow id, id]\ z_1\cdots z_s)|_{z_l} * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})$$
$$+ \sum_{l\in[s]} 1$$

$$
\begin{aligned}
&\quad \leq \ (\text{by } \sum_{l\in[s]} 1 = s \leq s_{max}) \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t_1'[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_s)|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}) \\
&\quad = \ (\text{by substitution}) \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ (\star\ t_1')[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_s)|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})
\end{aligned}
$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then induction hypothesis (a) and Lemma 2.8 give equalities instead of inequalities. Moreover, we then also have $s = s_{max}$; hence all inequalities in the above calculation are turned into equalities.

$\underline{t' = \bullet\ t_1'}$ for some $t_1' \in T_{\Delta\cup\{\star,\bullet\}}$:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, g\ (\bullet\ t_1')\ \eta_1^\circ \cdots \eta_s^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, g\ (\bullet\ t_1')[\star \twoheadleftarrow \circ^{s_{max}}]\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \\
&\quad = \ (\text{by substitution, } \Rightarrow_{R_2^\star \bullet\to\circ\bullet} \text{ and } \Rightarrow_{R_2^\circ \bullet\to\circ\bullet}\text{, using that} \\
&\qquad\quad rhs_{M_2^\star \bullet\to\circ\bullet, g, \bullet} = rhs_{M_2^\circ \bullet\to\circ\bullet, g, \bullet} = \bullet\ (g\ v_1\ z_1 \cdots z_s) \text{ by Definitions 4.14 and 4.20,} \\
&\qquad\quad \text{by normal forms and definition of } |\,.\,|_{\circ-\bullet}) \\
&|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, g\ t_1'\ \eta_1^\circ \cdots \eta_s^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, g\ t_1'[\star \twoheadleftarrow \circ^{s_{max}}]\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \\
&\quad \leq \ (\text{by induction hypothesis (a) for } t_1') \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ t_1'[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_s)|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}) \\
&\quad = \ (\text{by substitution}) \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ (\bullet\ t_1')[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_s)|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})
\end{aligned}
$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then induction hypothesis (a) gives equality instead of inequality in the above calculation.

$\underline{t' = \delta\ t_1' \cdots t_q'}$ for some $\delta \in \Delta^{(q)}$ and $t_1', \ldots, t_q' \in T_{\Delta\cup\{\star,\bullet\}}$:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, g\ (\delta\ t_1' \cdots t_q')\ \eta_1^\circ \cdots \eta_s^\circ)|_{\circ-\bullet} \\
&-|nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, g\ (\delta\ t_1' \cdots t_q')[\star \twoheadleftarrow \circ^{s_{max}}]\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \\
&\quad = \ (\text{by substitution, } \Rightarrow_{R_2^\star \bullet\to\circ\bullet} \text{ and } \Rightarrow_{R_2^\circ \bullet\to\circ\bullet}\text{, using that} \\
&\qquad\quad rhs_{M_2^\star \bullet\to\circ\bullet, g, \delta} = rhs_{M_2^\circ \bullet\to\circ\bullet, g, \delta} = rhs_{M_2, g, \delta} \text{ by Definitions 4.14 and 4.20)} \\
&|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, rhs_{M_2,g,\delta}[v_j \leftarrow t_j' \mid j \in [q], \\
&\qquad\qquad\qquad\qquad\qquad z_l \leftarrow \eta_l^\circ \mid l \in [s]])|_{\circ-\bullet} \\
&-|nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, rhs_{M_2,g,\delta}[v_j \leftarrow t_j'[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q], \\
&\qquad\qquad\qquad\qquad\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ-\bullet} \\
&\quad \leq \ (\text{by induction hypothesis (b) with } \psi = rhs_{M_2,g,\delta}) \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, rhs_{M_2,g,\delta}[v_j \leftarrow t_j'[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}) \\
&\quad = \ (\text{by substitution and } \Rightarrow_{R_2}) \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, g\ (\delta\ t_1' \cdots t_q')[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_s)|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^\star \bullet\to\circ\bullet}, \eta_l^\circ)|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^\circ \bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})
\end{aligned}
$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then induction hypothesis (b) gives equality instead of inequality in the above calculation.

(b) $\Leftarrow$ (a) : for fixed $s \in \mathbb{N}$ and $\eta_1^\circ, \ldots, \eta_s^\circ, \eta_1, \ldots, \eta_s \in T_{G \cup \Delta \cup \Omega \cup \{\star, \bullet, \circ\}}$ by structural induction on $\psi \in RHS(G, \Omega, V_q, Z_s)$:

$\underline{\psi = z_{l'} \in Z_s}$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, z_{l'}[v_j \leftarrow t'_j \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l^\circ \mid l \in [s]])|_{\circ - \bullet}\\
&-|nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, z_{l'}[v_j \leftarrow t'_j[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ - \bullet}\\
&\quad = \text{(by substitution)}\\
&|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \eta_{l'}^\circ)|_{\circ - \bullet} - |nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \eta_{l'})|_{\circ - \bullet}\\
&\quad = \text{(by } |z_{l'}|_{z_{l'}} = 1 \text{ and } |z_{l'}|_{z_l} = 0 \text{ for } l \neq l')\\
&\sum_{l \in [s]} |z_{l'}|_{z_l} * (|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \eta_l^\circ)|_{\circ - \bullet} - |nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \eta_l)|_{\circ - \bullet})\\
&\quad = \text{(by substitution and normal form)}\\
&\sum_{l \in [s]} |nf(\Rightarrow_{R_2}, z_{l'}[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l}\\
&\qquad * (|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \eta_l^\circ)|_{\circ - \bullet} - |nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \eta_l)|_{\circ - \bullet})
\end{aligned}$$

$\underline{\psi = \omega\ \psi_1 \cdots \psi_n}$ for some $\omega \in \Omega^{(n)}$ and $\psi_1, \ldots, \psi_n \in RHS(G, \Omega, V_q, Z_s)$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, (\omega\ \psi_1 \cdots \psi_n)[v_j \leftarrow t'_j \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l^\circ \mid l \in [s]])|_{\circ - \bullet}\\
&-|nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, (\omega\ \psi_1 \cdots \psi_n)[v_j \leftarrow t'_j[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ - \bullet}\\
&\quad = \text{(by substitution, normal forms, and definition of } |\,.\,|_{\circ - \bullet})\\
&\sum_{b \in [n]} (|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \psi_b[v_j \leftarrow t'_j \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l^\circ \mid l \in [s]])|_{\circ - \bullet}\\
&\qquad -|nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \psi_b[v_j \leftarrow t'_j[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ - \bullet})\\
&\quad \leq \text{(by the induction hypotheses for the } \psi_1, \ldots, \psi_n)\\
&\sum_{b \in [n]} \sum_{l \in [s]} |nf(\Rightarrow_{R_2}, \psi_b[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l}\\
&\qquad * (|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \eta_l^\circ)|_{\circ - \bullet} - |nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \eta_l)|_{\circ - \bullet})\\
&\quad = \text{(by sum manipulation)}\\
&\sum_{l \in [s]} \Big(\sum_{b \in [n]} |nf(\Rightarrow_{R_2}, \psi_b[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l} \Big)\\
&\qquad * (|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \eta_l^\circ)|_{\circ - \bullet} - |nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \eta_l)|_{\circ - \bullet})\\
&\quad = \text{(by substitution, normal form, and definition of } |\,.\,|_{z_l})\\
&\sum_{l \in [s]} |nf(\Rightarrow_{R_2}, (\omega\ \psi_1 \cdots \psi_n)[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l}\\
&\qquad * (|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, \eta_l^\circ)|_{\circ - \bullet} - |nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, \eta_l)|_{\circ - \bullet})
\end{aligned}$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then the induction hypotheses for the $\psi_1, \ldots, \psi_n$ give equality instead of inequality in the above calculation.

$\underline{\psi = g\ v_{j'}\ \psi_1 \cdots \psi_{s'}}$ for some $g \in G^{(s'+1)}$, $v_{j'} \in V_q$, and $\psi_1, \ldots, \psi_{s'} \in RHS(G, \Omega, V_q, Z_s)$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^\star \bullet \to \circ \bullet}, (g\ v_{j'}\ \psi_1 \cdots \psi_{s'})[v_j \leftarrow t'_j \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l^\circ \mid l \in [s]])|_{\circ - \bullet}\\
&-|nf(\Rightarrow_{R_2^\circ \bullet \to \circ \bullet}, (g\ v_{j'}\ \psi_1 \cdots \psi_{s'})[v_j \leftarrow t'_j[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q],\\
&\qquad\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ - \bullet}
\end{aligned}$$

$$
\begin{aligned}
&\leq \text{ (by substitution and induction hypothesis (a) for } t'_{j'}) \\
&\sum_{l'\in[s']} |nf(\Rightarrow_{R_2}, g\ t'_{j'}[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_{s'})|_{z_{l'}} \\
&\quad * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \psi_{l'}[v_j \leftarrow t'_j \mid j \in [q], \\
&\qquad\qquad z_l \leftarrow \eta_l^{\circ} \mid l \in [s]])|_{\circ-\bullet} \\
&\quad\ -|nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \psi_{l'}[v_j \leftarrow t'_j[\star \twoheadleftarrow \circ^{s_{max}}] \mid j \in [q], \\
&\qquad\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ-\bullet}) \\
&\leq \text{ (by the induction hypotheses for the } \psi_1, \ldots, \psi_{s'}, \\
&\qquad \text{using that the factors } |nf(\Rightarrow_{R_2}, g\ t'_{j'}[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_{s'})|_{z_{l'}} \geq 0) \\
&\sum_{l'\in[s']} |nf(\Rightarrow_{R_2}, g\ t'_{j'}[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_{s'})|_{z_{l'}} \\
&\quad * \sum_{l\in[s]} |nf(\Rightarrow_{R_2}, \psi_{l'}[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l} \\
&\qquad * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^{\circ})|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}) \\
&= \text{ (by sum manipulation)} \\
&\sum_{l\in[s]} \Big(\sum_{l'\in[s']} |nf(\Rightarrow_{R_2}, g\ t'_{j'}[\star, \bullet \twoheadleftarrow id, id]\ z_1 \cdots z_{s'})|_{z_{l'}} \\
&\qquad * |nf(\Rightarrow_{R_2}, \psi_{l'}[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l} \Big) \\
&\quad * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^{\circ})|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet}) \\
&= \text{ (by substitution and Lemma 2.7 for the mtt } M_2) \\
&\sum_{l\in[s]} |nf(\Rightarrow_{R_2}, (g\ v_{j'}\ \psi_1 \cdots \psi_{s'})[v_j \leftarrow t'_j[\star, \bullet \twoheadleftarrow id, id] \mid j \in [q]])|_{z_l} \\
&\quad * (|nf(\Rightarrow_{R_2^{\star}\bullet\to\circ\bullet}, \eta_l^{\circ})|_{\circ-\bullet} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\circ-\bullet})
\end{aligned}
$$

If additionally M_2 is context-nondeleting and $G = G^{(s_{max}+1)}$, then induction hypothesis (a) and the induction hypotheses for the $\psi_1, \ldots, \psi_{s'}$ give equalities instead of inequalities in the above calculation. $\square$

Proof of Lemma 4.31

For an atmost mtt M_2 we prove the following two statements:

(a) $\underline{\text{for every } \theta \in T_{\Delta\cup\{\circ,\bullet\}}:}$

For every $\theta' \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'$, for every $g \in G^{(s+1)}$ and $\eta_1, \ldots, \eta_s$, $\eta'_1, \ldots, \eta'_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, g\ \theta\ \eta_1 \cdots \eta_s)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, g\ \theta'\ \eta'_1 \cdots \eta'_s)|_{\bullet-\circ} \\
&\leq |\theta|_\bullet - |\theta'|_\bullet + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, g\ \theta\ z_1 \cdots z_s)|_{z_l} \\
&\qquad * (|nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta'_l)|_{\bullet-\circ}).
\end{aligned}
$$

(b) $\underline{\text{for every } q \in \mathbb{N} \text{ and } \theta_1, \ldots, \theta_q \in T_{\Delta\cup\{\circ,\bullet\}}:}$

For every $\theta'_1, \ldots, \theta'_q \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta_j \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_j$ for every $j \in [q]$, for every $s \in \mathbb{N}$, $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$, and $\psi \in RHS(G, \Omega, V_q, Z_s)$, if M_2 is context-linear or ψ contains no nested calls, then:

$$
\begin{aligned}
&\quad |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \psi[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta_1, \ldots, \theta_q, \eta_1, \ldots, \eta_s])|_{\bullet-\circ} \\
&- |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \psi[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta'_1, \ldots, \theta'_q, \eta'_1, \ldots, \eta'_s])|_{\bullet-\circ} \\
&\leq \sum_{j\in[q]} |\psi|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \psi[v_j \leftarrow \theta_j \mid j \in [q]])|_{z_l} \\
&\qquad * (|nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ}\bullet\to\circ\bullet}, \eta'_l)|_{\bullet-\circ}).
\end{aligned}
$$

by simultaneous induction. The first statement of Lemma 4.31 then follows from statement (a) with $\eta'_l = \eta_l$ for every $l \in [s]$, taking into account that a sum of multiples of 0 is 0, and that $|\varrho|_{\bullet-\circ} = -|\varrho|_{\circ-\bullet}$ for every $\varrho \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$. The second statement of Lemma 4.31, for atleast M_2, is obtained correspondingly after replacing all occurrences of

$\leq$ and "-linear" by $\geq$ and "-nondeleting", respectively, in the above statements and the following proof.

(a) $\Leftarrow$ (b) : by case analysis on $\theta \in T_{\Delta\cup\{\circ,\bullet\}}$ and $\theta \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'$, distinguishing four cases:

$\underline{\theta = \bullet\ \theta_1,\ \theta' = \theta'_1}$ for some $\theta_1, \theta'_1 \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta_1 \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_1$:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\bullet\ \theta_1)\ \eta_1 \cdots \eta_s)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta'_1\ \eta'_1 \cdots \eta'_s)|_{\bullet-\circ}\\
&\quad = \text{(by } \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}} \text{ (cf. Definition 4.20), normal form, and definition of } |\,.\,|_{\bullet-\circ})\\
&1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta_1\ \eta_1 \cdots \eta_s)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta'_1\ \eta'_1 \cdots \eta'_s)|_{\bullet-\circ}\\
&\quad \leq \text{(by induction hypothesis (a) for } \theta_1 \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_1)\\
&1 + |\theta_1|_\bullet - |\theta'_1|_\bullet + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta_1\ z_1 \cdots z_s)|_{z_l}\\
&\qquad\qquad * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})\\
&\quad = \text{(by definition of } |\,.\,|_\bullet, \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}} \text{ (cf. Definition 4.20), normal form,}\\
&\qquad\quad \text{and definition of } |\,.\,|_{z_l})\\
&|\bullet\ \theta_1|_\bullet - |\theta'_1|_\bullet + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\bullet\ \theta_1)\ z_1 \cdots z_s)|_{z_l}\\
&\qquad\qquad * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})
\end{aligned}
$$

$\underline{\theta = \circ\ \theta_1,\ \theta' = \circ\ \theta'_1}$ for some $\theta_1, \theta'_1 \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta_1 \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_1$:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\circ\ \theta_1)\ \eta_1 \cdots \eta_s)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\circ\ \theta'_1)\ \eta'_1 \cdots \eta'_s)|_{\bullet-\circ}\\
&\quad = \text{(by (twice) } \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\\
&\qquad\quad \text{using that } rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\circ} = \circ\ (g\ v_1\ z_1 \cdots z_s) \text{ by Definition 4.20,}\\
&\qquad\quad \text{by normal forms and definition of } |\,.\,|_{\bullet-\circ})\\
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta_1\ \eta_1 \cdots \eta_s)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta'_1\ \eta'_1 \cdots \eta'_s)|_{\bullet-\circ}\\
&\quad \leq \text{(by induction hypothesis (a) for } \theta_1 \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_1)\\
&|\theta_1|_\bullet - |\theta'_1|_\bullet + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta_1\ z_1 \cdots z_s)|_{z_l}\\
&\qquad\qquad * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})\\
&\quad = \text{(by definition of } |\,.\,|_\bullet, \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \text{ using that } rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\circ} = \circ\ (g\ v_1\ z_1 \cdots z_s) \text{ by}\\
&\qquad\quad \text{Definition 4.20, by normal form and definition of } |\,.\,|_{z_l})\\
&|\circ\ \theta_1|_\bullet - |\circ\ \theta'_1|_\bullet + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\circ\ \theta_1)\ z_1 \cdots z_s)|_{z_l}\\
&\qquad\qquad * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})
\end{aligned}
$$

$\underline{\theta = \bullet\ \theta_1,\ \theta' = \bullet\ \theta'_1}$ for some $\theta_1, \theta'_1 \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta_1 \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_1$:
analogous to the previous case.

$\underline{\theta = \delta\ \theta_1 \cdots \theta_q,\ \theta' = \delta\ \theta'_1 \cdots \theta'_q}$ for some $\delta \in \Delta^{(q)}$ and $\theta_1, \ldots, \theta_q, \theta'_1, \ldots, \theta'_q \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta_j \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_j$ for every $j \in [q]$:

$$
\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta_1 \cdots \theta_q)\ \eta_1 \cdots \eta_s)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta'_1 \cdots \theta'_q)\ \eta'_1 \cdots \eta'_s)|_{\bullet-\circ}\\
&\quad = \text{(by (twice) } \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \text{ using that } rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\delta} = rhs_{M_2,g,\delta} \text{ by Definition 4.20)}\\
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, rhs_{M_2,g,\delta}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta_1, \ldots, \theta_q, \eta_1, \ldots, \eta_s])|_{\bullet-\circ}\\
&-|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, rhs_{M_2,g,\delta}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta'_1, \ldots, \theta'_q, \eta'_1, \ldots, \eta'_s])|_{\bullet-\circ}\\
&\quad \leq \text{(by induction hypothesis (b) with } \psi = rhs_{M_2,g,\delta},\\
&\qquad\quad \text{using that } M_2 \text{ is context-linear or } M_2 \text{ is basic,}\\
&\qquad\quad \text{in which case } rhs_{M_2,g,\delta} \text{ contains no nested calls)}\\
&\sum_{j\in[q]} |rhs_{M_2,g,\delta}|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, rhs_{M_2,g,\delta}[v_j \leftarrow \theta_j \mid j \in [q]])|_{z_l}\\
&\qquad\qquad * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})
\end{aligned}
$$

$\leq$ (by $|rhs_{M_2,g,\delta}|_{v_j} \leq 1$ for every $j \in [q]$ (since M_2 is recursion-linear), using that the factors $|\theta_j|_\bullet - |\theta'_j|_\bullet \geq 0$ due to $\theta_j \Rightarrow^*_{\{\bullet\ v \to v\}} \theta'_j$)

$$\sum_{j\in[q]} (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, rhs_{M_2,g,\delta}[v_j \leftarrow \theta_j \mid j \in [q]])|_{z_l} * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})$$

$=$ (by definition of $|\,.\,|_\bullet$ and $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$, using that $rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\delta} = rhs_{M_2,g,\delta}$ by Definition 4.20)

$$|\delta\ \theta_1 \cdots \theta_q|_\bullet - |\delta\ \theta'_1 \cdots \theta'_q|_\bullet + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ \theta_1 \cdots \theta_q)\ z_1 \cdots z_s)|_{z_l} * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})$$

(b) $\Leftarrow$ (a) : for fixed $\theta'_1, \ldots, \theta'_q \in T_{\Delta\cup\{\circ,\bullet\}}$ with $\theta_j \Rightarrow^*_{\{\bullet\ v\to v\}} \theta'_j$ for every $j \in [q]$, for fixed $s \in \mathbb{N}$ and $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$, by structural induction on $\psi \in RHS(G, \Omega, V_q, Z_s)$:

$\underline{\psi = z_{l'} \in Z_s}$:

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, z_{l'}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta_1, \ldots, \theta_q, \eta_1, \ldots, \eta_s])|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, z_{l'}[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta'_1, \ldots, \theta'_q, \eta'_1, \ldots, \eta'_s])|_{\bullet-\circ}$$

$=$ (by substitution)

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_{l'})|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_{l'})|_{\bullet-\circ}$$

$=$ (by $|z_{l'}|_{z_{l'}} = 1$ and $|z_{l'}|_{z_l} = 0$ for $l \neq l'$)

$$\sum_{l\in[s]} |z_{l'}|_{z_l} * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})$$

$=$ (by $|z_{l'}|_{v_j} = 0$, substitution, and normal form)

$$\sum_{j\in[q]} |z_{l'}|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, z_{l'}[v_j \leftarrow \theta_j \mid j \in [q]])|_{z_l} * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})$$

$\underline{\psi = \omega\ \psi_1 \cdots \psi_n}$ for some $\omega \in \Omega^{(n)}$ and $\psi_1, \ldots, \psi_n \in RHS(G, \Omega, V_q, Z_s)$:

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, (\omega\ \psi_1 \cdots \psi_n)[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta_1, \ldots, \theta_q, \eta_1, \ldots, \eta_s])|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, (\omega\ \psi_1 \cdots \psi_n)[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta'_1, \ldots, \theta'_q, \eta'_1, \ldots, \eta'_s])|_{\bullet-\circ}$$

$=$ (by substitution, normal forms, and definition of $|\,.\,|_{\bullet-\circ}$)

$$\sum_{b\in[n]} (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \psi_b[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta_1, \ldots, \theta_q, \eta_1, \ldots, \eta_s])|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \psi_b[v_1, \ldots, v_q, z_1, \ldots, z_s \leftarrow \theta'_1, \ldots, \theta'_q, \eta'_1, \ldots, \eta'_s])|_{\bullet-\circ})$$

$\leq$ (by the induction hypotheses for the $\psi_1, \ldots, \psi_n$, using that if $\psi = \omega\ \psi_1 \cdots \psi_n$ contains no nested calls, then neither does any of the $\psi_1, \ldots, \psi_n$)

$$\sum_{b\in[n]} \Big(\sum_{j\in[q]} |\psi_b|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \psi_b[v_j \leftarrow \theta_j \mid j \in [q]])|_{z_l} * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ}) \Big)$$

$=$ (by sum manipulation)

$$\sum_{j\in[q]} \Big(\sum_{b\in[n]} |\psi_b|_{v_j} \Big) * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} \Big(\sum_{b\in[n]} |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \psi_b[v_j \leftarrow \theta_j \mid j \in [q]])|_{z_l} \Big) * (|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\bullet-\circ})$$

$=$ (by definition of $|\,.\,|_{v_j}$, substitution, normal form, and definition of $|\,.\,|_{z_l}$)

$$\sum_{j\in[q]} |\omega\ \psi_1\cdots\psi_n|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet)$$
$$+ \sum_{l\in[s]} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, (\omega\ \psi_1\cdots\psi_n)[v_j \leftarrow \theta_j \mid j\in[q]])|_{z_l} * (|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta'_l)|_{\bullet-\circ})$$

$\underline{\psi = g\ v_{j'}\ \psi_1\cdots\psi_{s'}}$ for some $g\in G^{(s'+1)}$, $v_{j'}\in V_q$, and $\psi_1,\ldots,\psi_{s'}\in RHS(G,\Omega,V_q,Z_s)$:

$$|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, (g\ v_{j'}\ \psi_1\cdots\psi_{s'})[v_1,\ldots,v_q,z_1,\ldots,z_s \leftarrow \theta_1,\ldots,\theta_q,\eta_1,\ldots,\eta_s])|_{\bullet-\circ}$$
$$-|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, (g\ v_{j'}\ \psi_1\cdots\psi_{s'})[v_1,\ldots,v_q,z_1,\ldots,z_s \leftarrow \theta'_1,\ldots,\theta'_q,\eta'_1,\ldots,\eta'_s])|_{\bullet-\circ}$$
$\leq$ (by substitution and induction hypothesis (a) for $\theta_{j'} \Rightarrow^*_{\{\bullet\ v\to v\}} \theta'_{j'}$)
$$|\theta_{j'}|_\bullet - |\theta'_{j'}|_\bullet + \sum_{l'\in[s']} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * (|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \psi_{l'}[v_j\leftarrow\theta_j \mid j\in[q], z_l \leftarrow \eta_l \mid l\in[s]])|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \psi_{l'}[v_j\leftarrow\theta'_j \mid j\in[q], z_l \leftarrow \eta'_l \mid l\in[s]])|_{\bullet-\circ})$$
$\leq$ (by the induction hypotheses for the $\psi_1,\ldots,\psi_{s'}$,
using that if $\psi = g\ v_{j'}\ \psi_1\cdots\psi_{s'}$ contains no nested calls,
then neither does any of the $\psi_1,\ldots,\psi_{s'}$,
and that the factors $|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}}\geq 0$)
$$|\theta_{j'}|_\bullet - |\theta'_{j'}|_\bullet + \sum_{l'\in[s']} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * \Big(\sum_{j\in[q]} |\psi_{l'}|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \psi_{l'}[v_j\leftarrow\theta_j \mid j\in[q]])|_{z_l} * (|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta'_l)|_{\bullet-\circ})\Big)$$
$=$ (by sum manipulation)
$$|\theta_{j'}|_\bullet - |\theta'_{j'}|_\bullet + \sum_{l'\in[s']} \Big(\sum_{j\in[q]} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * |\psi_{l'}|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \psi_{l'}[v_j\leftarrow\theta_j \mid j\in[q]])|_{z_l} * (|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta'_l)|_{\bullet-\circ})\Big)$$
$\leq$ (by $|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * |\psi_{l'}|_{v_j} \leq |\psi_{l'}|_{v_j}$
for every $l'\in[s']$ and $j\in[q]$ (see below),
using that the factors $|\theta_j|_\bullet - |\theta'_j|_\bullet \geq 0$ due to $\theta_j \Rightarrow^*_{\{\bullet\ v\to v\}} \theta'_j$)
$$|\theta_{j'}|_\bullet - |\theta'_{j'}|_\bullet + \sum_{l'\in[s']} \Big(\sum_{j\in[q]} |\psi_{l'}|_{v_j} * (|\theta_j|_\bullet - |\theta'_j|_\bullet) + \sum_{l\in[s]} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \psi_{l'}[v_j\leftarrow\theta_j \mid j\in[q]])|_{z_l} * (|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta'_l)|_{\bullet-\circ})\Big)$$
$=$ (by sum manipulation)
$$|\theta_{j'}|_\bullet - |\theta'_{j'}|_\bullet + \sum_{j\in[q]} \Big(\sum_{l'\in[s']} |\psi_{l'}|_{v_j}\Big) * (|\theta_j|_\bullet - |\theta'_j|_\bullet)$$
$$+ \sum_{l\in[s]} \Big(\sum_{l'\in[s']} |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \psi_{l'}[v_j\leftarrow\theta_j \mid j\in[q]])|_{z_l}\Big) * (|nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta_l)|_{\bullet-\circ} - |nf(\Rightarrow_{R_2^\circ}\bullet\!\to\!\circ\bullet, \eta'_l)|_{\bullet-\circ})$$
$=$ (by $|v_{j'}|_{v_{j'}} = 1$ and $|v_{j'}|_{v_j} = 0$ for $j\neq j'$)

$$\sum_{j\in[q]}\Big(|v_{j'}|_{v_j}+\sum_{l'\in[s']}|\psi_{l'}|_{v_j}\Big)*(|\theta_j|_\bullet-|\theta'_j|_\bullet)$$
$$+\sum_{l\in[s]}\Big(\sum_{l'\in[s']}|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}} * |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\psi_{l'}[v_j\leftarrow\theta_j\mid j\in[q]])|_{z_l}\Big)$$
$$*(|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta_l)|_{\bullet-\circ}-|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta'_l)|_{\bullet-\circ})$$
$$=\ \text{(by definition of } |\,.\,|_{v_j}\text{, substitution, and Lemma 2.7 for the mtt } M_2^{\circ\bullet\to\circ\bullet})$$
$$\sum_{j\in[q]}|g\ v_{j'}\ \psi_1\cdots\psi_{s'}|_{v_j}*(|\theta_j|_\bullet-|\theta'_j|_\bullet)$$
$$+\sum_{l\in[s]}|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},(g\ v_{j'}\ \psi_1\cdots\psi_{s'})[v_j\leftarrow\theta_j\mid j\in[q]])|_{z_l} *(|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta_l)|_{\bullet-\circ}-|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta'_l)|_{\bullet-\circ})$$

The statement, used above, that for every $l'\in[s']$ and $j\in[q]$,

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}}*|\psi_{l'}|_{v_j}\le|\psi_{l'}|_{v_j}\ ,$$

follows from the precondition that M_2, and hence also $M_2^{\circ\bullet\to\circ\bullet}$, is context-linear or $\psi=g\ v_{j'}\ \psi_1\cdots\psi_{s'}$ contains no nested calls, since in the former case,

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta_{j'}\ z_1\cdots z_{s'})|_{z_{l'}}\le 1$$

by Lemma 2.8 for $M_2^{\circ\bullet\to\circ\bullet}$ (and obviously $|\psi_{l'}|_{v_j}\ge 0$), while in the latter case $|\psi_{l'}|_{v_j}=0$ holds. □

Proof of Lemma 4.33
Let $M_1^{\to\circ\bullet}=(F,\Sigma,\Delta\cup\{\circ,\bullet\},e_{M_1^{\to\circ\bullet}},R_1^{\to\circ\bullet})$ and $M_1^{\to\circ\bullet\prime}=(F,\Sigma,\Delta\cup\{\circ,\bullet\},e_{M_1^{\to\circ\bullet\prime}},R_1^{\to\circ\bullet\prime})$, such that $M_1^{\to\circ\bullet}\rightsquigarrow_{\mathcal{E}_{M_1,M_2}}M_1^{\to\circ\bullet\prime}$. We prove the following two statements (a) and (b):

(a) <u>for every $t\in T_\Sigma$:</u>
For every $f\in F^{(r+1)}$ and $\theta_1,\dots,\theta_r,\theta'_1,\dots,\theta'_r\in T_{\Delta\cup\{\circ,\bullet\}}$, if for every $k\in[r]$, $g\in G^{(s+1)}$, and $\eta_1,\dots,\eta_s,\eta'_1,\dots,\eta'_s\in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$ from $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta_l)|_{\circ-\bullet}\le|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta'_l)|_{\circ-\bullet}$ for every $l\in[s]$ follows that

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta_k\ \eta_1\cdots\eta_s)|_{\circ-\bullet}\le|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta'_k\ \eta'_1\cdots\eta'_s)|_{\circ-\bullet}\ ,$$

then for every $g\in G^{(s+1)}$ and $\eta_1,\dots,\eta_s,\eta'_1,\dots,\eta'_s\in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$ with $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta_l)|_{\circ-\bullet}\le|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta'_l)|_{\circ-\bullet}$ for every $l\in[s]$:

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}},f\ t\ \theta_1\cdots\theta_r)\ \eta_1\cdots\eta_s)|_{\circ-\bullet}$$
$$\le|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}},f\ t\ \theta'_1\cdots\theta'_r)\ \eta'_1\cdots\eta'_s)|_{\circ-\bullet}.$$

(b) <u>for every $p\in\mathbb{N}$ and $t_1,\dots,t_p\in T_\Sigma$:</u>
For every $r\in\mathbb{N}$ and $\theta_1,\dots,\theta_r,\theta'_1,\dots,\theta'_r\in T_{\Delta\cup\{\circ,\bullet\}}$, if for every $k\in[r]$, $g\in G^{(s+1)}$, and $\eta_1,\dots,\eta_s,\eta'_1,\dots,\eta'_s\in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$ from $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta_l)|_{\circ-\bullet}\le|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\eta'_l)|_{\circ-\bullet}$ for every $l\in[s]$ follows that

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta_k\ \eta_1\cdots\eta_s)|_{\circ-\bullet}\le|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},g\ \theta'_k\ \eta'_1\cdots\eta'_s)|_{\circ-\bullet}\ ,$$

then the following statements (i) and (ii) hold:

(i) <u>for every $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:</u>
For every $\phi' \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$ with $\phi \Rightarrow^{?}_{\mathcal{E}_{M_1,M_2}} \phi'$, for every $g \in G^{(s+1)}$ and $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$ with $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\circ-\bullet} \leq |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\circ-\bullet}$ for every $l \in [s]$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi[u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \\
\leq\ &|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'[u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}.
\end{aligned}$$

(ii) <u>for every $q \in \mathbb{N}$ and $\phi_1, \ldots, \phi_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:</u>
For every $\phi'_1, \ldots, \phi'_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$ with $\phi_j \Rightarrow^{?}_{\mathcal{E}_{M_1,M_2}} \phi'_j$ for every $j \in [q]$, for every $s \in \mathbb{N}$ and $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s \in T_{G\cup\Omega\cup\Delta\cup\{\circ,\bullet\}}$ with $|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\circ-\bullet} \leq |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\circ-\bullet}$ for every $l \in [s]$, for every $\psi \in RHS(G, \Omega, V_q, Z_s)$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \psi[v_j \leftarrow nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_j[u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad\qquad y_k \leftarrow \theta_k \mid k \in [r]]) \mid j \in [q], \\
&\qquad z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ-\bullet} \\
\leq\ &|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \psi[v_j \leftarrow nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'_j[u_i \leftarrow t_i \mid i \in [p], \\
&\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \mid j \in [q], \\
&\qquad z_l \leftarrow \eta'_l \mid l \in [s]])|_{\circ-\bullet}.
\end{aligned}$$

by simultaneous induction, where the two statements (i) and (ii) nested inside (b) are also proved by simultaneous induction. The first statement of Lemma 4.33,

$$\tau_{M_1^{\to\circ\bullet}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet} \leq \tau_{M_1^{\to\circ\bullet'}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet}\,,$$

then follows from statement (b)(ii) with $p = 1$, $r = 0$, $q = 1$, $\phi_1 = e_{M_1^{\to\circ\bullet}}$, $\phi'_1 = e_{M_1^{\to\circ\bullet'}}$, $s = 0$, and $\psi = e_{M_2}$.

If for every $\downarrow$- or $\uparrow$-rule used for rewriting the initial expression or right-hand sides of rules of $M_1^{\to\circ\bullet}$, $\mathcal{E}_{M_1,M_2}$ also contains the symmetric rule, then there exists an $M_1^{\to\circ\bullet''} \in \mathcal{M}_1^{\to\circ\bullet}$ such that $M_1^{\to\circ\bullet'} \rightsquigarrow_{\mathcal{E}_{M_1,M_2}} M_1^{\to\circ\bullet''}$ and $M_1^{\to\circ\bullet''}$ can be obtained from $M_1^{\to\circ\bullet}$ by rewriting only with the rules $\bullet\ (\circ\ v) \to v$ and $\circ\ (\bullet\ v) \to v$.[35] Since under the restriction of $\mathcal{E}_{M_1,M_2}$ to $\{\bullet\ (\circ\ v) \to v, \circ\ (\bullet\ v) \to v\}$ all inequalities in the above statements and the following proof—except for the last four cases of **(i)** $\Leftarrow$ **(ii)**, which however need not be considered under the assumed restriction of $\mathcal{E}_{M_1,M_2}$—can be strengthened to equalities, the latter implies $\tau_{M_1^{\to\circ\bullet}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet} = \tau_{M_1^{\to\circ\bullet''}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet}$. Together with $\tau_{M_1^{\to\circ\bullet'}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet} \leq \tau_{M_1^{\to\circ\bullet''}} ; \tau_{M_2^{\circ\bullet\to\circ\bullet}} ; |\,.\,|_{\circ-\bullet}$ (which follows from $M_1^{\to\circ\bullet'} \rightsquigarrow_{\mathcal{E}_{M_1,M_2}} M_1^{\to\circ\bullet''}$) we then get equality instead of the above inequality.

(a) $\Leftarrow$ (b) : Assume $t = \sigma\ t_1 \cdots t_p$ for $\sigma \in \Sigma^{(p)}$ and $t_1, \ldots, t_p \in T_\Sigma$. We calculate:

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ (\sigma\ t_1 \cdots t_p)\ \theta_1 \cdots \theta_r)\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$

[35] This $M_1^{\to\circ\bullet''}$ is obtained from $M_1^{\to\circ\bullet'}$ by "undoing" all the rewrites with $\downarrow$- or $\uparrow$-rules performed in $M_1^{\to\circ\bullet} \rightsquigarrow_{\mathcal{E}_{M_1,M_2}} M_1^{\to\circ\bullet'}$, while nothing is done about the rules $\bullet\ (\circ\ v) \to v$ and $\circ\ (\bullet\ v) \to v$.

$= $ (by $\Rightarrow_{R_1^{\to\circ\bullet}}$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, rhs_{M_1^{\to\circ\bullet},f,\sigma}[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$\leq$ (by induction hypothesis (b), statement (i), with $\phi = rhs_{M_1^{\to\circ\bullet},f,\sigma} \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} rhs_{M_1^{\to\circ\bullet'},f,\sigma} = \phi'$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, rhs_{M_1^{\to\circ\bullet'},f,\sigma}[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$= $ (by $\Rightarrow_{R_1^{\to\circ\bullet'}}$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, f\ (\sigma\ t_1 \cdots t_p)\ \theta'_1 \cdots \theta'_r)\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

(b) $\Leftarrow$ **(a) :** for fixed r and $\theta_1, \ldots, \theta_r, \theta'_1, \ldots, \theta'_r \in T_{\Delta \cup \{\circ,\bullet\}}$ with the associated preconditions, by simultaneous induction of (i) and (ii):

(i) $\Leftarrow$ **(ii) :** by case analysis on $\phi \in RHS(F, \Delta \cup \{\circ,\bullet\}, U_p, Y_r)$ and $\phi \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'$, distinguishing eleven cases:

$\underline{\phi = \phi' = y_k \in Y_r}$:

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, y_k[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$= $ (by substitution and since $\theta_k \in T_{\Delta\cup\{\circ,\bullet\}}$ is a normal form w.r.t. $\Rightarrow_{R_1^{\to\circ\bullet}}$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta_k\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$\leq$ (by the preconditions on θ_k, θ'_k and the $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ \theta'_k\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$= $ (by substitution and since $\theta'_k \in T_{\Delta\cup\{\circ,\bullet\}}$ is a normal form w.r.t. $\Rightarrow_{R_1^{\to\circ\bullet'}}$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, y_k[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$\underline{\phi = \circ\ \phi_1,\ \phi' = \circ\ \phi'_1}$ for some $\phi_1, \phi'_1 \in RHS(F, \Delta \cup \{\circ,\bullet\}, U_p, Y_r)$ with $\phi_1 \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_1$:

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\circ\ \phi_1)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$= $ (by substitution, normal form, $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$ (cf. Definition 4.20), normal form, and definition of $|\,.\,|_{\circ-\bullet}$)

$1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$\leq$ (by induction hypothesis (i) for $\phi_1 \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_1$)

$1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$= $ (by substitution, normal form, $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$ (cf. Definition 4.20), normal form, and definition of $|\,.\,|_{\circ-\bullet}$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (\circ\ \phi'_1)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$\underline{\phi = \bullet\ \phi_1,\ \phi' = \bullet\ \phi'_1}$ for some $\phi_1, \phi'_1 \in RHS(F, \Delta \cup \{\circ,\bullet\}, U_p, Y_r)$ with $\phi_1 \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_1$: analogous to the previous case, but with "−1+" instead of "1+".

$\underline{\phi = \delta\ \phi_1 \cdots \phi_q,\ \phi' = \delta\ \phi'_1 \cdots \phi'_q}$ for some $\delta \in \Delta^{(q)}$ and $\phi_1, \ldots, \phi_q, \phi'_1, \ldots, \phi'_q \in RHS(F, \Delta \cup \{\circ,\bullet\}, U_p, Y_r)$ with $\phi_j \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_j$ for every $j \in [q]$:

$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\delta\ \phi_1 \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$
$= $ (by substitution, normal form, and $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$, using that $rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\delta} = rhs_{M_2,g,\delta}$ by Definition 4.20)
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, rhs_{M_2,g,\delta}[v_j \leftarrow nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_j[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]]) \mid j \in [q], z_l \leftarrow \eta_l \mid l \in [s]])|_{\circ-\bullet}$$
$\leq$ (by induction hypothesis (ii) with $\psi = rhs_{M_2,g,\delta}$)
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, rhs_{M_2,g,\delta}[v_j \leftarrow nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'_j[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]) \mid j \in [q], z_l \leftarrow \eta'_l \mid l \in [s]])|_{\circ-\bullet}$$
$= $ (by substitution, normal form, and $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$, using that $rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\delta} = rhs_{M_2,g,\delta}$ by Definition 4.20)
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (\delta\ \phi'_1 \cdots \phi'_q)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$$

$\underline{\phi = f\ u_{i'}\ \phi_1 \cdots \phi_{r'},\ \phi' = f\ u_{i'}\ \phi'_1 \cdots \phi'_{r'}}$ for some $f \in F^{(r'+1)}$, $u_{i'} \in U_p$, and $\phi_1, \ldots, \phi_{r'}, \phi'_1, \ldots, \phi'_{r'} \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$ with $\phi_k \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_k$ for every $k \in [r']$:
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$
$= $ (by substitution and normal form)
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]]) \cdots nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_{r'}[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]]))\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$
$\leq$ (by induction hypothesis (a) for $t_{i'}$,[36] using the induction hypotheses (i) for $\phi_1 \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_1, \ldots, \phi_{r'} \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_{r'}$ to establish the necessary precondition)
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, f\ t_{i'}\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]) \cdots nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'_{r'}[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]))\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$$
$= $ (by substitution and normal form)
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (f\ u_{i'}\ \phi'_1 \cdots \phi'_{r'})[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$$

$\underline{\phi = \bullet\ (\circ\ \phi_1),\ \phi' = \phi_1}$ for some $\phi_1 \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:
$$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\bullet\ (\circ\ \phi_1))[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$$

[36] To make sure that the "typing" agrees here, we have to establish that for every $k' \in [r']$, $nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_{k'}[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])$ and $nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi'_{k'}[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta'_1, \ldots, \theta'_r])$ are in $T_{\Delta\cup\{\circ,\bullet\}}$. But this follows easily from $\phi_{k'}, \phi'_{k'} \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$, $t_1, \ldots, t_p \in T_\Sigma$, $\theta_1, \ldots, \theta_r, \theta'_1, \ldots, \theta'_r \in T_{\Delta\cup\{\circ,\bullet\}}$, and the state, input, and output alphabets of $M_1^{\to\circ\bullet}$ and $M_1^{\to\circ\bullet'}$ being F, Σ, and $\Delta \cup \{\circ, \bullet\}$, respectively. Similar issues arise when applying Lemma 4.32 in other cases below, and are resolved likewise.

$= $ (by substitution, normal form, $\Rightarrow^2_{R_2^{\circ\bullet\to\circ\bullet}}$, using that $rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\bullet} = \bullet\ (g\ v_1\ z_1 \cdots z_s)$ and $rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\circ} = \circ\ (g\ v_1\ z_1 \cdots z_s)$ by Definition 4.20, by normal form and definition of $|\,.\,|_{\circ-\bullet}$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$\leq$ (by induction hypothesis (i) for $\phi_1 \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi_1$)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$\underline{\phi = \circ\ (\bullet\ \phi_1),\ \phi' = \phi_1}$ for some $\phi_1 \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$: analogous to the previous case.

$\underline{\phi = \bullet\ (\delta\ \phi_1 \cdots \phi_q),\ \phi' = \delta\ \phi_1 \cdots (\bullet\ \phi_j) \cdots \phi_q}$ for some $\delta \in \Delta^{(q)}$, $\phi_1, \ldots, \phi_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$, and $j \in [q]$ with $\downarrow_{\delta,j} \in \mathcal{E}_{M_1,M_2}$:

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\bullet\ (\delta\ \phi_1 \cdots \phi_q))[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$=$ (by substitution, normal form, $\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}$, using that $rhs_{M_2^{\circ\bullet\to\circ\bullet},g,\bullet} = \bullet\ (g\ v_1\ z_1 \cdots z_s)$ by Definition 4.20, by normal form and definition of $|\,.\,|_{\circ-\bullet}$)

$-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\delta\ \phi_1 \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$\leq$ (by induction hypothesis (i) for $\delta\ \phi_1 \cdots \phi_q \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \delta\ \phi_1 \cdots \phi_q$)

$-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, (\delta\ \phi_1 \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$=$ (by substitution and normal form)

$$\begin{aligned}
-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ & nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]) \\
& \cdots \\
& nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, \phi_q[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]))\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}
\end{aligned}$$

$\leq$ (by Lemma 4.32(1a))

$$\begin{aligned}
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ & nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]) \\
& \cdots \\
& (\bullet\ nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, \phi_j[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])) \\
& \cdots \\
& nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, \phi_q[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]]))\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}
\end{aligned}$$

$=$ (by substitution and normal form)

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet\prime}}, (\delta\ \phi_1 \cdots (\bullet\ \phi_j) \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}$

$\underline{\phi = \delta\ \phi_1 \cdots (\bullet\ \phi_j) \cdots \phi_q,\ \phi' = \bullet\ (\delta\ \phi_1 \cdots \phi_q)}$ for some $\delta \in \Delta^{(q)}$, $\phi_1, \ldots, \phi_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$, and $j \in [q]$ with $\uparrow_{\delta,j} \in \mathcal{E}_{M_1,M_2}$:

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\delta\ \phi_1 \cdots (\bullet\ \phi_j) \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p], y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}$

$$\begin{aligned}
&\le \text{(analogous to the case } \phi = \delta\ \phi_1 \cdots \phi_q \text{ above,}\\
&\qquad \text{using that } \phi_1 \Rightarrow^{?}_{\mathcal{E}_{M_1,M_2}} \phi_1, \ldots, \bullet\ \phi_j \Rightarrow^{?}_{\mathcal{E}_{M_1,M_2}} \bullet\ \phi_j, \ldots, \phi_q \Rightarrow^{?}_{\mathcal{E}_{M_1,M_2}} \phi_q)\\
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (\delta\ \phi_1 \cdots (\bullet\ \phi_j) \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}\\
&= \text{(by substitution and normal form)}\\
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_1[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\\
&\qquad\qquad \cdots\\
&\qquad\qquad (\bullet\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_j[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]))\\
&\qquad\qquad \cdots\\
&\qquad\qquad nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_q[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]))\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}\\
&\le \text{(by Lemma 4.32(1b))}\\
&-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ (\delta\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_1[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\\
&\qquad\qquad \cdots\\
&\qquad\qquad nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_q[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]))\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}\\
&= \text{(by substitution, normal form, } \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}\text{, using that}\\
&\qquad rhs_{M_2^{\circ\bullet\to\circ\bullet}, g, \bullet} = \bullet\ (g\ v_1\ z_1 \cdots z_s) \text{ by Definition 4.20,}\\
&\qquad \text{by normal form and definition of } |\,.\,|_{\circ-\bullet})\\
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (\bullet\ (\delta\ \phi_1 \cdots \phi_q))[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}
\end{aligned}$$

$\phi = \bullet\ (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})$, $\phi' = f\ u_{i'}\ \phi_1 \cdots (\bullet\ \phi_{k'}) \cdots \phi_{r'}$ for some $f \in F^{(r'+1)}$, $u_{i'} \in U_p$, $\phi_1, \ldots, \phi_{r'} \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$, and $k' \in [r']$ with $\downarrow_{f,k'} \in \mathcal{E}_{M_1,M_2}$:

$$\begin{aligned}
&|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\bullet\ (f\ u_{i'}\ \phi_1 \cdots \phi_{r'}))[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}\\
&= \text{(by substitution, normal form, } \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}\text{, using that}\\
&\qquad rhs_{M_2^{\circ\bullet\to\circ\bullet}, g, \bullet} = \bullet\ (g\ v_1\ z_1 \cdots z_s) \text{ by Definition 4.20,}\\
&\qquad \text{by normal form and definition of } |\,.\,|_{\circ-\bullet})\\
&-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta_k \mid k \in [r]])\ \eta_1 \cdots \eta_s)|_{\circ-\bullet}\\
&\le \text{(by induction hypothesis (i) for } f\ u_{i'}\ \phi_1 \cdots \phi_{r'} \Rightarrow^{?}_{\mathcal{E}_{M_1,M_2}} f\ u_{i'}\ \phi_1 \cdots \phi_{r'})\\
&-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}\\
&= \text{(by substitution and normal form)}\\
&-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}},\\
&\qquad g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, f\ t_{i'}\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_1[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])\\
&\qquad\qquad \cdots\\
&\qquad\qquad nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_{r'}[u_i \leftarrow t_i \mid i \in [p],\\
&\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]))\ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}\\
&\le \text{(by Lemma 4.32(2a) for the mtt } M_1^{\to\circ\bullet'})
\end{aligned}$$

$$
\begin{array}{l}
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, f\ t_{i'}\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\bullet\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_{k'}[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad y_k \leftarrow \theta'_k \mid k \in [r]])) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_{r'}[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])) \ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet} \\
\quad = \text{(by substitution and normal form)} \\
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \\
\qquad g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (f\ u_{i'}\ \phi_1 \cdots (\bullet\ \phi_{k'}) \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}
\end{array}
$$

$\underline{\phi = f\ u_{i'}\ \phi_1 \cdots (\bullet\ \phi_{k'}) \cdots \phi_{r'},\ \phi' = \bullet\ (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})}$ for some $f \in F^{(r'+1)}$, $u_{i'} \in U_p$, $\phi_1, \ldots, \phi_{r'} \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$, and $k' \in [r']$ with $\uparrow_{f,k'} \in \mathcal{E}_{M_1,M_2}$:

$$
\begin{array}{l}
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \\
\qquad g\ nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (f\ u_{i'}\ \phi_1 \cdots (\bullet\ \phi_{k'}) \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta_k \mid k \in [r]]) \ \eta_1 \cdots \eta_s)|_{\circ-\bullet} \\
\quad \leq \text{(analogous to the case } \phi = f\ u_{i'}\ \phi_1 \cdots \phi_{r'} \text{ above,} \\
\qquad\ \text{using that } \phi_1 \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi_1, \ldots, \bullet\ \phi_{k'} \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \bullet\ \phi_{k'}, \ldots, \phi_{r'} \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi_{r'}) \\
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \\
\qquad g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (f\ u_{i'}\ \phi_1 \cdots (\bullet\ \phi_{k'}) \cdots \phi_{r'})[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet} \\
\quad = \text{(by substitution and normal form)} \\
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, f\ t_{i'}\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\bullet\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_{k'}[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad y_k \leftarrow \theta'_k \mid k \in [r]])) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_{r'}[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])) \ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet} \\
\quad \leq \text{(by Lemma 4.32(2b) for the mtt } M_1^{\to\circ\bullet'}) \\
-1 + |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \\
\qquad\quad g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, f\ t_{i'}\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_1[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \\
\qquad\qquad\qquad\qquad\qquad\qquad \cdots \\
\qquad\qquad\qquad\qquad\qquad\qquad nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, \phi_{r'}[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]])) \ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet} \\
\quad = \text{(by substitution, normal form, } \Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}\text{, using that} \\
\qquad\ rhs_{M_2^{\circ\bullet\to\circ\bullet}, g, \bullet} = \bullet\ (g\ v_1\ z_1 \cdots z_s) \text{ by Definition 4.20,} \\
\qquad\ \text{by normal form and definition of } |\,.\,|_{\circ-\bullet}) \\
|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, g\ nf(\Rightarrow_{R_1^{\to\circ\bullet'}}, (\bullet\ (f\ u_{i'}\ \phi_1 \cdots \phi_{r'}))[u_i \leftarrow t_i \mid i \in [p], \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_k \leftarrow \theta'_k \mid k \in [r]]) \ \eta'_1 \cdots \eta'_s)|_{\circ-\bullet}
\end{array}
$$

(ii) $\Leftarrow$ (i) : for fixed $\phi'_1, \ldots, \phi'_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$ with $\phi_j \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_j$ for every $j \in [q]$, for fixed $s \in \mathbb{N}$ and $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s \in T_{G \cup \Omega \cup \Delta \cup \{\circ, \bullet\}}$ with

$|nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta_l)|_{\circ-\bullet} \leq |nf(\Rightarrow_{R_2^{\circ\bullet\to\circ\bullet}}, \eta'_l)|_{\circ-\bullet}$ for every $l \in [s]$, by structural induction on $\psi \in RHS(G, \Omega, V_q, Z_s)$. The validity in the base case $\psi \in Z_s$ follows from the preconditions on the $\eta_1, \ldots, \eta_s, \eta'_1, \ldots, \eta'_s$. The one recursive case, $\psi = \omega\ \psi_1 \cdots \psi_n$, is straightforward and in the other one, $\psi = g\ v_j\ \psi_1 \cdots \psi_{s'}$, the induction hypotheses for the $\psi_1, \ldots, \psi_{s'}$ establish the preconditions necessary to apply induction hypothesis (i) for $\phi_j \Rightarrow^?_{\mathcal{E}_{M_1,M_2}} \phi'_j$, which suffices. $\square$

Proof of Lemma 4.35

Let $M_1^{\to\circ\bullet} = (F, \Sigma, \Delta \cup \{\circ, \bullet\}, e_{M_1^{\to\circ\bullet}}, R_1^{\to\circ\bullet}) \in \mathcal{M}_1^{\to\circ\bullet}$ be such that the right-hand sides of the rules in $R_1^{\to\circ\bullet}$ contain no $\circ$-symbols. We prove the following two statements:

(a) for every $t \in T_\Sigma$:
For every $f \in F^{(r+1)}$ and $\theta_1, \ldots, \theta_r \in T_{F\cup\Sigma\cup\Delta\cup\{\circ,\bullet\}}$:
$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ \theta_1 \cdots \theta_r)|_\circ = \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t\ y_1 \cdots y_r)|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$.

(b) for every $p \in \mathbb{N}$ and $t_1, \ldots, t_p \in T_\Sigma$:
For every $r \in \mathbb{N}$, $\theta_1, \ldots, \theta_r \in T_{F\cup\Sigma\cup\Delta\cup\{\circ,\bullet\}}$, and $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$, if $|\phi|_\circ = 0$ or M_1 is context-linear, then:
$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ$
$\leq |\phi|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$.
If $|\phi|_\circ = 0$ or additionally M_1 is context-nondeleting, then equality holds.

by simultaneous induction. Lemma 4.35 then follows from statement (b) with $p = 1$, $t_1 = t$, $r = 0$, and $\phi = e_{M_1^{\to\circ\bullet}}$, taking into account that an empty sum is 0.

(a) $\Leftarrow$ (b) : Assume $t = \sigma\ t_1 \cdots t_p$ for $\sigma \in \Sigma^{(p)}$ and $t_1, \ldots, t_p \in T_\Sigma$. We calculate:

$$
\begin{aligned}
& |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ (\sigma\ t_1 \cdots t_p)\ \theta_1 \cdots \theta_r)|_\circ \\
& \quad = \text{(by } \Rightarrow_{R_1^{\to\circ\bullet}}\text{)} \\
& |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, rhs_{M_1^{\to\circ\bullet},f,\sigma}[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ \\
& \quad = \text{(by induction hypothesis (b) with } \phi = rhs_{M_1^{\to\circ\bullet},f,\sigma}, \\
& \qquad\quad \text{using that } |rhs_{M_1^{\to\circ\bullet},f,\sigma}|_\circ = 0 \text{ by the precondition of the lemma)} \\
& \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, rhs_{M_1^{\to\circ\bullet},f,\sigma}[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ \\
& \quad = \text{(by } \Rightarrow_{R_1^{\to\circ\bullet}}\text{)} \\
& \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ (\sigma\ t_1 \cdots t_p)\ y_1 \cdots y_r)|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ
\end{aligned}
$$

(b) $\Leftarrow$ (a) : for fixed $r \in \mathbb{N}$ and $\theta_1, \ldots, \theta_r \in T_{F\cup\Sigma\cup\Delta\cup\{\circ,\bullet\}}$ by structural induction on $\phi \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:

$\phi = y_{k'} \in Y_r$:

$$
\begin{aligned}
& |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, y_{k'}[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ \\
& \quad = \text{(by substitution)} \\
& |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_{k'})|_\circ \\
& \quad = \text{(by } |y_{k'}|_{y_{k'}} = 1 \text{ and } |y_{k'}|_{y_k} = 0 \text{ for } k \neq k'\text{)}
\end{aligned}
$$

$\sum_{k\in[r]} |y_{k'}|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$
$= $ (by definition of $|\,.\,|_\circ$, substitution, and normal form)
$|y_{k'}|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, y_{k'}[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$

$\phi = \bullet\ \phi_1$ for some $\phi_1 \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:

$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\bullet\ \phi_1)[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ$
$=$ (by substitution, normal form, and definition of $|\,.\,|_\circ$)
$0 + |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_1[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ$
$\leq$ (by the induction hypothesis for ϕ_1,
using that if $|\phi|_\circ = |\bullet\ \phi_1|_\circ = 0$, then also $|\phi_1|_\circ = 0$)
$0 + |\phi_1|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_1[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$
$=$ (by definition of $|\,.\,|_\circ$, substitution, normal form, and definition of $|\,.\,|_{y_k}$)
$|\bullet\ \phi_1|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\bullet\ \phi_1)[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$

If $|\phi|_\circ = |\bullet\ \phi_1|_\circ = 0$ (and thus $|\phi_1|_\circ = 0$) or additionally M_1 is context-nondeleting, then the induction hypothesis for ϕ_1 gives equality instead of inequality in the above calculation.

$\phi = \circ\ \phi_1$ for some $\phi_1 \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:
analogous to the previous case, but with "1+" instead of "0+".

$\phi = \delta\ \phi_1 \cdots \phi_q$ for some $\delta \in \Delta^{(q)}$ and $\phi_1, \ldots, \phi_q \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:

$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\delta\ \phi_1 \cdots \phi_q)[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ$
$=$ (by substitution, normal form, and definition of $|\,.\,|_\circ$)
$\sum_{j\in[q]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_j[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ$
$\leq$ (by the induction hypotheses for the $\phi_1, \ldots, \phi_q$,
using that if $|\phi|_\circ = |\delta\ \phi_1 \cdots \phi_q|_\circ = 0$, then also $|\phi_j|_\circ = 0$ for every $j \in [q]$)
$\sum_{j\in[q]} \Big(|\phi_j|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_j[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ \Big)$
$=$ (by sum manipulation)
$\sum_{j\in[q]} |\phi_j|_\circ + \sum_{k\in[r]} \Big(\sum_{j\in[q]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_j[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} \Big) * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$
$=$ (by definition of $|\,.\,|_\circ$, substitution, normal form, and definition of $|\,.\,|_{y_k}$)
$|\delta\ \phi_1 \cdots \phi_q|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (\delta\ \phi_1 \cdots \phi_q)[u_i \leftarrow t_i \mid i \in [p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$

If $|\phi|_\circ = |\delta\ \phi_1 \cdots \phi_q|_\circ = 0$ (and thus $|\phi_j|_\circ = 0$ for every $j \in [q]$) or additionally M_1 is context-nondeleting, then the induction hypotheses for the $\phi_1, \ldots, \phi_q$ give equalities instead of inequalities in the above calculation.

$\phi = f\ u_{i'}\ \phi_1 \cdots \phi_{r'}$ for some $f \in F^{(r'+1)}$, $u_{i'} \in U_p$, and $\phi_1, \ldots, \phi_{r'} \in RHS(F, \Delta \cup \{\circ, \bullet\}, U_p, Y_r)$:

$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (f\ u_{i'}\ \phi_1 \cdots \phi_{r'})[u_1, \ldots, u_p, y_1, \ldots, y_r \leftarrow t_1, \ldots, t_p, \theta_1, \ldots, \theta_r])|_\circ$
$=$ (by substitution and induction hypothesis (a) for $t_{i'}$)

$$\sum_{k'\in[r']} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_{k'}[u_i \leftarrow t_i \mid i\in[p], y_k \leftarrow \theta_k \mid k\in[r]])|_\circ$$

$\leq$ (by the induction hypotheses for the $\phi_1,\ldots,\phi_{r'}$, using that if $|\phi|_\circ = |f\ u_{i'}\ \phi_1\cdots\phi_{r'}|_\circ = 0$, then also $|\phi_{k'}|_\circ = 0$ for every $k'\in[r']$, and that the factors $|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} \geq 0$)

$$\sum_{k'\in[r']} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * \Big(|\phi_{k'}|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_{k'}[u_i\leftarrow t_i \mid i\in[p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ\Big)$$

$=$ (by sum manipulation)

$$\sum_{k'\in[r']} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * |\phi_{k'}|_\circ$$
$$+ \sum_{k\in[r]}\Big(\sum_{k'\in[r']} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_{k'}[u_i\leftarrow t_i \mid i\in[p]])|_{y_k}\Big) * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$$

$\leq$ (by $|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * |\phi_{k'}|_\circ \leq |\phi_{k'}|_\circ$ for every $k'\in[r']$, see below)

$$\sum_{k'\in[r']} |\phi_{k'}|_\circ + \sum_{k\in[r]}\Big(\sum_{k'\in[r']} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \phi_{k'}[u_i\leftarrow t_i \mid i\in[p]])|_{y_k}\Big) * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$$

$=$ (by definition of $|\,.\,|_\circ$, substitution, and Lemma 2.7 for the mtt $M_1^{\to\circ\bullet}$)

$$|f\ u_{i'}\ \phi_1\cdots\phi_{r'}|_\circ + \sum_{k\in[r]} |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, (f\ u_{i'}\ \phi_1\cdots\phi_{r'})[u_i\leftarrow t_i \mid i\in[p]])|_{y_k} * |nf(\Rightarrow_{R_1^{\to\circ\bullet}}, \theta_k)|_\circ$$

The statement, used above, that for every $k'\in[r']$,

$$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} * |\phi_{k'}|_\circ \leq |\phi_{k'}|_\circ\,,$$

follows from the precondition that $|\phi|_\circ = |f\ u_{i'}\ \phi_1\cdots\phi_{r'}|_\circ = 0$ or M_1, and hence also $M_1^{\to\circ\bullet}$, is context-linear, since in the former case $|\phi_{k'}|_\circ = 0$, while in the latter case,

$$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} \leq 1$$

by Lemma 2.8(1) for $M_1^{\to\circ\bullet}$ (and obviously $|\phi_{k'}|_\circ \geq 0$).
If $|\phi|_\circ = |f\ u_{i'}\ \phi_1\cdots\phi_{r'}|_\circ = 0$ (and thus $|\phi_{k'}|_\circ = 0$ for every $k'\in[r']$) or additionally M_1 (and hence also $M_1^{\to\circ\bullet}$) is context-nondeleting, then the induction hypotheses for the $\phi_1,\ldots,\phi_{r'}$ give equalities instead of inequalities, and also the other occurrences of $\leq$ above can be replaced by $=$ because for context-nondeleting $M_1^{\to\circ\bullet}$,

$$|nf(\Rightarrow_{R_1^{\to\circ\bullet}}, f\ t_{i'}\ y_1\cdots y_{r'})|_{y_{k'}} \geq 1$$

holds by Lemma 2.8(2). □

Proof of Lemma 4.37

For atmost M_2, we take the statements (a) and (b) and the simultaneous induction establishing their validity in the proof of Lemma 4.31. We replace all occurrences of $\Rightarrow^*_{\{\bullet\ v\to v\}}$, $|\,.\,|_\bullet$, and $|\,.\,|_{\bullet-\circ}$ by $\Rightarrow^*_{\{\circ\ v\to v\}}$, $|\,.\,|_\circ$, and $|\,.\,|_\circ$, respectively, and all occurrences of $\bullet\ \theta_1$ by $\circ\ \theta_1$ in the first case of **(a)** $\Leftarrow$ **(b)**. This yields again a valid proof. We obtain

$$|\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\theta)|_\circ - |\tau_{M_2^{\circ\bullet\to\circ\bullet}}(\theta[\circ \twoheadleftarrow id])|_\circ \leq |e_{M_2}|_{v_1} * |\theta|_\circ$$

from the thus adjusted statement (b) with $q = 1$, $\theta_1 = \theta$, $\theta'_1 = \theta[\circ \twoheadleftarrow id]$, $s = 0$, and $\psi = e_{M_2}$, taking into account that M_2 is context-linear or M_2 is basic, in which case e_{M_2} contains no nested calls, and that an empty sum is 0, and so is $|\theta[\circ \twoheadleftarrow id]|_\circ$. The first statement of Lemma 4.37 then follows because $|\tau_{M_2^\circ\bullet\to\circ\bullet}(\theta[\circ \twoheadleftarrow id])|_\circ = 0$ by Lemma 4.36. For atleast M_2, we can replace all occurrences of $\leq$ and "-linear" by $\geq$ and "-nondeleting", respectively, in the adjusted statements and proof, again without compromising validity, and correspondingly (using that an atleast mtt is context-nondeleting or basic) obtain $|\tau_{M_2^\circ\bullet\to\circ\bullet}(\theta)|_\circ \geq |e_{M_2}|_{v_1} * |\theta|_\circ$. $\square$