INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

On Computational Interpretations of the Modal Logic S4 IIIb. Confluence, Termination of the $\lambda \mathrm{ev} Q_H$-Calculus.

Jean Goubault-Larrecq

N° 3164
Mai 1997

THÈME 2

Rapport de recherche

ISSN 0249-6399

On Computational Interpretations of the Modal Logic S4
IIIb. Confluence, Termination of the $\lambda \mathtt{ev} Q_H$-Calculus.

Jean Goubault-Larrecq *

Thème 2 — Génie logiciel
et calcul symbolique

Projet Coq

Rapport de recherche n° 3164 — Mai 1997 — 48 pages

Abstract: A language of proof terms for minimal logic is the λ-calculus, where cut-elimination is encoded as β-reduction. We examine corresponding languages for the minimal version of the modal logic S4, with notions of reduction that encodes cut-elimination for the corresponding sequent system. It turns out that a natural interpretation of the latter constructions is a λ-calculus extended by an idealized version of Lisp's `eval` and `quote` constructs.

In this Part IIIb, we complete the results of Part IIIa. We show that the typed $\lambda \mathtt{ev} Q_H$-calculus is confluent. It follows that the typed $\lambda \mathtt{ev} Q_H$-calculus is a conservative extension of the typed $\lambda_{\mathrm{S4}H}$-calculus. We also prove that the typed $\lambda \mathtt{ev} Q_H$-calculus is weakly terminating.

Some problems remain open. In particular, we still don't know whether the typed $\lambda \mathtt{ev} Q$-calculus terminates weakly, or whether the untyped $\lambda \mathtt{ev} Q$-calculus is confluent.

Finally, we correct a few mistakes and inaccuracies that occurred in previous parts of this work.

Key-words: modal logic, S4, $\lambda\sigma$-calculus, explicit substitutions, termination, confluence, λ-calculus, simple types

(Résumé : tsvp)

*`Jean.Goubault@inria.fr`

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

À propos des interprétations calculatoires de la logique modale S4
IIIb. Confluence, terminaison du $\lambda\mathtt{ev}Q_H$-calcul.

Résumé : Un langage de termes de preuve pour la logique minimale est le λ-calcul, où l'élimination des coupures est codée par la β-reduction. Nous examinons des langages correspondants pour la version minimale de la logique modale S4, avec des notions de réductions qui codent l'élimination des coupures dans le système de séquents correspondant. Il s'avère qu'une interprétation naturelle de ces constructions est un λ-calcul étendu avec une version idéalisée des constructions `eval` et `quote` de Lisp.

Dans cette partie IIIb, nous complétons les résultats de la partie IIIa. Nous montrons que le $\lambda\mathtt{ev}Q_H$-calcul typé est confluent. Il s'ensuit que le $\lambda\mathtt{ev}Q_H$-calcul typé est une extension conservative du $\lambda_{\mathrm{S4}H}$-calcul typé. Nous prouvons aussi que le $\lambda\mathtt{ev}Q_H$-calcul est faiblement terminant.

Il reste quelques problèmes ouverts. En particulier, nous ne savons toujours pas si le $\lambda\mathtt{ev}Q$-calcul typé termine faiblement, ou si le $\lambda\mathtt{ev}Q$-calcul non typé est confluent.

Finalement, nous corrigeons quelques erreurs et inexactitudes des parties précédentes de ce travail.

Mots-clé : logique modale, S4, $\lambda\sigma$-calcul, substitutions explicites, terminaison, confluence, λ-calcul, types simples

Plan

We start by correcting a few mistakes or inaccuracies of previous parts in Section 1, then establish that the typed $\lambda \mathtt{ev} Q_H$-calculus is confluent in Section 2 —and although the untyped calculus is not confluent, it is confluent in the typed case even in the previous of fixed point operators—, and that the typed $\lambda \mathtt{ev} Q_H$-calculus terminates weakly in Section 3. We examine the case of the typed $\lambda \mathtt{ev} Q$-calculus, and explain why the techniques we have used are not able to show that it terminates weakly: the difficulty is mainly due to the complicated form of (typed) Σ-normal terms.

For definitions and previously-known results, see the reports [GL96a], hereafter denoted as Part I, [GL96b] (Part II) and [GL96c] (Part IIIa).

1 Corrigendum

1.1 Termination of Typed Σ_H

Termination of the typed version Σ_H was proved in Part IIIa by reducing it to a variant of typed λ-calculus with first-order operators enjoying specific reduction rules, the $\lambda\oplus$-calculus. The latter was then proved terminating by using Jouannaud and Rubio's higher-order recursive path ordering (horpo), as defined in the preliminary paper [JR96a]. Unfortunately, this proposal was wrong, in the sense that it manages to prove terminating even non-terminating rewrite systems. (For instance, the untyped λ-calculus: define the Scott translation from untyped λ-terms to simply-typed ones by: $(\lambda x \cdot u)^* = f(\lambda x : o \cdot u^*)$, $(uv)^* = g(u^*)v^*$, where f maps terms of type $o \to o$ to terms of type o, and g maps terms of type o to type $o \to o$, and obey the reduction rule $g(f(u)) \to u$.)

The corrected version of the horpo is in the final paper [JR96b]. Unfortunately, the result presented there is too weak for our purposes, even though we actually only need reductions on η-long β-normal forms, not the full reductions of the typed $\lambda\oplus$-calculus.

We therefore present a direct proof of the termination of the typed $\lambda\oplus$-calculus, based on reducibility arguments à la Tait-Girard [GLT89] (called strong computability in [HS88]). The definition of reducibility and of neutrality is standard, as the properties (CR1), (CR2) and (CR3) and their proofs. The only difficulty lies in the proof of the fact that all terms are reducible. We adopt the presentation of [GLT89], Chapter 6. Since the notions of negative and positive types does not matter, we drop the $+$ and $-$ superscripts on θ-types in most cases.

Definition 1.1 *We define the sets* RED^θ *of* reducible $\lambda\oplus$-terms of type θ *by induction on the θ-type θ by:*

- RED^o *is the set of strongly normalizing $\lambda\oplus$-terms of type o;*
- $\mathrm{RED}^{\theta_1 \times \theta_2} = \{s : \theta_1 \times \theta_2 \mid \pi_1 s \in \mathrm{RED}^{\theta_1}, \pi_2 s \in \mathrm{RED}^{\theta_2}\}$*;*
- $\mathrm{RED}^{\theta_1 \to \theta_2} = \{s : \theta_1 \to \theta_2 \mid \forall t \in \mathrm{RED}^{\theta_1} \cdot st \in \mathrm{RED}^{\theta_2}\}$*.*

Definition 1.2 *We say that a $\lambda\oplus$-term is* neutral *if and only if it is neither a lambda-abstraction $\lambda x \cdot s$ nor a pair $\langle s, t\rangle$.*

The conditions of reducibility are:

(CR1) *If $s \in \mathrm{RED}^\theta$, then s is strongly normalizing.*

(CR2) *If $s \in \mathrm{RED}^\theta$ and $s \longrightarrow t$, then $t \in \mathrm{RED}^\theta$.*

(CR3) *If s is neutral of type θ, and for every t such that $s \longrightarrow t$, $t \in \mathrm{RED}^\theta$, then $s \in \mathrm{RED}^\theta$.*

Lemma 1.1 *Properties (CR1), (CR2) and (CR3) are valid.*

Proof: By simultaneous induction on the type θ:

- If $\theta = o$:

- (CR1) is a tautology.
- (CR2): If $s \in \mathrm{RED}^o$, then by definition s is strongly normalizing, so whenever $s \longrightarrow t$, t is strongly normalizing as well; by subject reduction, t has type o, so by Definition 1.1, $s \in \mathrm{RED}^o$.
- (CR3): If s is neutral of type o, and every t such that $s \longrightarrow t$ is in RED^o, then every such t is strongly normalizing. As any maximal reduction sequence starting from s must pass through one of these t, any such maximal sequence is finite, hence s is strongly normalizing. By Definition 1.1, $s \in \mathrm{RED}^o$.

- If θ is a product type $\theta_1 \times \theta_2$:
  - (CR1): Let $s \in \mathrm{RED}^\theta$. Then by Definition 1.1, $\pi_1 s \in \mathrm{RED}^{\theta^1}$, so by induction hypothesis $\pi_1 s$ is strongly normalizing. But s, as a subterm of π_1, must be strongly normalizing as well. (Any infinite sequence in s would induce one in $\pi_1 s$.)
  - (CR2): Let $s \in \mathrm{RED}^\theta$ and $s \longrightarrow t$. Then $\pi_1 s \longrightarrow \pi_1 t$, and since $\pi_1 s \in \mathrm{RED}^{\theta^1}$, by induction hypothesis $\pi_1 t \in \mathrm{RED}^{\theta_1}$. Similarly, $\pi_2 t \in \mathrm{RED}^{\theta_2}$. So by Definition 1.1, $t \in \mathrm{RED}^\theta$.
  - (CR3): Let s be neutral of type θ and assume that for every t such that $s \longrightarrow t$, $t \in \mathrm{RED}^\theta$. Now any maximal reduction sequence starting from $\pi_1 s$ must first reduce inside s because s, as a neutral term, is not a pair; so the first rewriting step must be of the form $\pi_1 s \longrightarrow \pi_1 t$ for some t as above. In particular, $t \in \mathrm{RED}^\theta$ by assumption, so $\pi_1 t \in \mathrm{RED}^{\theta_1}$ by Definition 1.1. Since $\pi_1 t$ ranges over all one-step reducts of $\pi_1 s$, we may apply the induction hypothesis and conclude that $\pi_1 s \in \mathrm{RED}^{\theta_1}$. Similarly, $\pi_2 s \in \mathrm{RED}^{\theta_2}$. By Definition 1.1, then, $s \in \mathrm{RED}^\theta$.

- If θ is an arrow type $\theta_1 \to \theta_2$:
  - (CR1): Let s be in RED^θ, and x be a variable of type θ_1. By induction hypothesis, condition (CR3), on θ_1, and since x has no one-step reduct (x is normal), $x \in \mathrm{RED}^{\theta_1}$. So by Definition 1.1, $sx \in \mathrm{RED}^{\theta_2}$. By induction hypothesis, condition (CR1), sx is then strongly normalizing. But then s, as a subterm of sx, is also strongly normalizing.
  - (CR2): Let $s \in \mathrm{RED}^\theta$ and $s \longrightarrow t$. For any $u \in \mathrm{RED}^{\theta_1}$, $su \in \mathrm{RED}^{\theta_2}$ and $su \longrightarrow tu$, so by induction hypothesis $tu \in \mathrm{RED}^{\theta_2}$. Since u is arbitrary, by Definition 1.1 $t \in \mathrm{RED}^\theta$.
  - (CR3): Let s be neutral of type θ, and assume that for every t such that $s \longrightarrow t$, $t \in \mathrm{RED}^\theta$. Let u be an arbitrary term in RED^{θ_1}. Then su may only reduce to: (1) tu, for some term t such that $s \longrightarrow t$, or to: (2) sv, for some term v such that $u \longrightarrow v$; indeed, s is neutral, and in particular is not a λ-abstraction, so no redex can be reduced at the top. In case (1), by assumption $t \in \mathrm{RED}^\theta$, so by Definition 1.1 $tu \in \mathrm{RED}^{\theta_2}$. In case (2), by induction hypothesis on θ_2, condition (CR2), $v \in \mathrm{RED}^{\theta_2}$, so by Definition 1.1 $sv \in \mathrm{RED}^{\theta_2}$. In both cases, su rewrites (in one-step) to terms in RED^{θ_2} only. Since su is neutral, by induction hypothesis (CR3), $su \in \mathrm{RED}^{\theta_2}$.

□

For any strongly normalizing $\lambda\oplus$-term s, let $\nu(s)$ denote the length of the longest reduction sequence starting from s. Such a sequence exists by König's Lemma, since any term can only rewrite in one step to finitely many different terms. Let also $|s|$ denote the size of s. We define a strict ordering $\succ_2$ on terms as follows:

Definition 1.3 *For every strongly normalizing terms s and t, let $s \succ_1 t$ if and only if $\nu(s) > \nu(t)$, or $\nu(s) \geq \nu(t)$ and $|s| > |t|$.*

Map each term s to a multiset of terms $D(s)$ by:

- $D(s_1 \oplus s_2) = D(s_1) \uplus D(s_2)$;
- $D(s) = \{\!|s|\!\}$ *if s is not of the form $s_1 \oplus s_2$.*

Let $s \succ_2 t$ if and only if $D(s) \succ_1^{mul} D(t)$, where $\succ_1^{mul}$ is the multiset extension of $\succ_1$. Similarly, $s \succeq_2 t$ if and only if $s \succ_2 t$ or $D(s) = D(t)$.

Finally, let $s \succ_3 t$ if and only if $s \succ_2 t$, or $s \succeq_2 t$ and $\nu(s) > \nu(t)$.

Observe that $\succ_1$, $\succ_2$ and $\succ_3$ are well-founded orderings on strongly normalizing $\lambda\oplus$-terms. Observe also that $\succ_2$ is an ordering included in $\succ_3$.

Lemma 1.2 *For every strongly normalizing term s, $\nu(\pi_1 s) = \nu(\pi_2 s)$.*

Proof: Consider any longest reduction R starting from $\pi_i s$, $i \in \{1, 2\}$ (possibly infinite, although we shall see that it is in fact impossible). Since s is strongly normalizing, there is a longest initial sequence of rewrite steps consisting entirely of rewrites inside s. Let $n(R)$ ($\leq \nu(s)$) be the length of this initial sequence and $s'(R)$ be the obtained reduct of s. Then, either R stops after $n(R)$ steps, or the $n(R) + 1$st step contracts at the top (i.e., $s'(R) = \langle s_1, s_2\rangle$ and this step contracts $\pi_i s'(R)$ to s_i). Define $\nu(R)$ as 0 in the first case, and as $\nu(s'(R))$ in the second case. (Observe that $\nu(s'(R)) \leq \nu(s) - n(R)$.)

Amongst all such R's, there is one that minimizes $\nu(R)$, since this ordering is well-founded. So, choose some R that is minimal in this sense, amongst the longest reductions starting from $\pi_i s$.

Case 1: $s'(R)$ is not a pair. We then claim that $s'(R)$ is normal. Indeed, otherwise, we would be able to get a longer reduction than R by continuing to rewrite inside $s'(R)$, contradicting the fact that R is longest. (Observe that R is indeed finite in this case.)

Case 2: $s'(R)$ is a pair. the $n(R) + 1$st rewrite step in R occurs in s_j (with $s_j \longrightarrow s'_j$). If $i = j$ (say $i = j = 1$), then we transform R into the following R':

$$s \underbrace{\longrightarrow \ldots \longrightarrow}_{n(R) \text{ steps}} \pi_1\langle s_1, s_2\rangle \longrightarrow \pi_1\langle s'_1, s_2\rangle \longrightarrow s'_1 \longrightarrow \ldots$$

which has the same length as R (so it is longest as well), but $\nu(R') = \nu(\langle s'_1, s_2\rangle) < \nu(R)$, contradicting the fact that R was minimal. And if $i \neq j$ (say $i = 1$, $j = 2$), then we transform R into the following R':

$$s \underbrace{\longrightarrow \ldots \longrightarrow}_{n(R) \text{ steps}} \pi_1\langle s_1, s_2\rangle \longrightarrow \pi_1\langle s_1, s'_2\rangle \longrightarrow s_1 \longrightarrow \ldots$$

which is strictly longer than R. (The case where R has infinite length is excluded because s, hence s_1 is strongly normalizing.) But this contradicts the fact that R was longest.

It follows that the length of R is exactly $\nu(s) + 1$ if s has a longest reduction ending on a pair $\langle s_1, s_2\rangle$ (which is then in normal form), and is $\nu(s)$ otherwise. Moreover, this holds whether $i = 1$ or $i = 2$. So this quantity is both equal to $\nu(\pi_1 s)$ and to $\nu(\pi_2 s)$. □

Lemma 1.3 *For every terms s, t:*

(*i*) *If $s \oplus t$ is strongly normalizing, then $s \oplus t \succ_3 s$, $s \oplus t \succ_3 t$ (in fact also with $\succ_2$).*

(*ii*) *If ϵs is strongly normalizing, then $\epsilon s \succ_3 s$ (in fact also with $\succ_2$).*

(*iii*) *If ιs is strongly normalizing, then $\iota s \succ_3 s$ (in fact also with $\succ_2$).*

(*iv*) *If $\iota\langle\pi_1 s, t\rangle$ is strongly normalizing, then $\iota\langle\pi_1 s, t\rangle \succ_3 \epsilon\pi_2 s$ (also with $\succ_1$ or $\succ_2$ instead of $\succ_3$).*

(*v*) *If $\iota\langle\pi_1 s, t\rangle$ is strongly normalizing, then $\iota\langle\pi_1 s, t\rangle \succ_3 s \oplus \epsilon\pi_2 s$ (in fact also with $\succ_2$).*

(*vi*) *If $s_1 \oplus \epsilon\iota\langle\pi_1 s, t\rangle$ is strongly normalizing, then $s_1 \oplus \epsilon\iota\langle\pi_1 s, t\rangle \succ_3 s_1 \oplus \epsilon\pi_2 s$ (in fact also with $\succ_2$).*

(*vii*) *If s is strongly normalizing, and $s \longrightarrow t$, then $s \succeq_2 t$ and $s \succ_3 t$.*

Proof:

(*i*) If $s \oplus t$ is strongly normalizing, then s and t are as well. Then, $D(s \oplus t) = D(s) \uplus D(t)$ by definition; the result follows by the fact that $D(s)$ and $D(t)$ are not empty.

(*ii*) Assume ϵs to be strongly normalizing, and let $\{\!|s_1, \ldots, s_n|\!\}$ be $D(s)$. Observe that $D(\epsilon s) = \{\!|\epsilon s|\!\}$. For each $1 \leq i \leq n$, $\nu(s_i) \leq \nu(s)$. Moreover, $\nu(s) \leq \nu(\epsilon s) - 1$: take a reduction of length $\nu(s)$ starting from s, then the reduction starting from ϵs that reduces for $\nu(s)$ steps inside s, then applies (ϵ), has length $\nu(s) + 1$. So $\nu(s_i) < \nu(\epsilon s)$ for every $1 \leq i \leq n$. In particular, $\epsilon s \succ_1 s_i$ for each i, so $\epsilon s \succ_2 s$.

(*iii*) Similarly.

(*iv*) Assume $\iota\langle\pi_1 s, t\rangle$ strongly normalizing. We claim that $\nu(\epsilon\pi_2 s) \leq \nu(\iota\langle\pi_1 s, t\rangle)$. First, $\nu(\epsilon\pi_2 s) \leq \nu(\pi_2 s) + 1 = \nu(\pi_1 s) + 1$ (by Lemma 1.2) $\leq \nu(\langle\pi_1 s, t\rangle) + 1 \leq \nu(\iota\langle\pi_1 s, t\rangle)$. Then, $|\epsilon\pi_2 s| = 2 + |s| < 3 + |s| + |t| = |\iota\langle\pi_1 s, t\rangle|$. So $\iota\langle\pi_1 s, t\rangle \succ_1 \epsilon\pi_2 s$. Since $D(\iota\langle\pi_1 s, t\rangle|) = \{\!|\iota\langle\pi_1 s, t\rangle|\!\}$ and $D(\epsilon\pi_2 s) = \{\!|\epsilon\pi_2 s|\!\}$, $\iota\langle\pi_1 s, t\rangle \succ_2 \epsilon\pi_2 s$.

(*v*) Assume $\iota\langle\pi_1 s, t\rangle$ strongly normalizing. In particular, s is strongly normalizing and $\nu(s) \leq \nu(\iota\langle\pi_1 s, t\rangle) - 1$ (we can again complete any reduction in s by rule (ι)). Let $\{\!|s_1, \ldots, s_n|\!\}$ be $D(s)$; then $\nu(s_i) \leq \nu(s) < \nu(\iota\langle\pi_1 s, t\rangle)$ for each i, $1 \leq i \leq n$, so $\iota\langle\pi_1 s, t\rangle \succ_1 s_i$. Since $\iota\langle\pi_1 s, t\rangle \succ_1 \epsilon\pi_2 s$ by (*iv*), it follows that $D(\iota\langle\pi_1 s, t\rangle) = \{\!|\iota\langle\pi_1 s, t\rangle|\!\} \succ_1^{mul} \{\!|s_1, \ldots, s_n, \epsilon\pi_2 s|\!\} = D(s \oplus \epsilon\pi_2 s)$.

(*vi*) Assume $s_1 \oplus \epsilon\iota\langle\pi_1 s, t\rangle$ strongly normalizing. As in (*iv*), $\nu(\epsilon\pi_2 s) \leq \nu(\iota\langle\pi_1 s, t\rangle)$, so $\nu(\epsilon\pi_2 s) < \nu(\epsilon\iota\langle\pi_1 s, t\rangle)$. It follows that $D(s_1 \oplus \epsilon\iota\langle\pi_1 s, t\rangle) = D(s_1) \uplus \{\!|\epsilon\iota\langle\pi_1 s, t\rangle|\!\} \succ_1^{mul} D(s_1) \uplus \{\!|\epsilon\pi_2 s|\!\} = D(s_1 \oplus \epsilon\pi_2 s)$, i.e. $s_1 \oplus \epsilon\iota\langle\pi_1 s, t\rangle \succ_2 s_1 \oplus \epsilon\pi_2 s$.

(*vii*) Assume s strongly normalizing, and $s \longrightarrow t$. More precisely, let s be $\mathcal{C}[s_1]$ for some context $\mathcal{C}$, where s_1 is a redex that contracts to t_1, and $t = \mathcal{C}[t_1]$.

- We first examine the case where $\mathcal{C}$ is the empty context, that is, where s is itself the contracted redex. If the applied rule is (β), then s and t have positive types, in particular their head symbol is not $\oplus$; so $D(s) = \{\!|s|\!\}$, $D(t) = \{\!|t|\!\}$, and since by construction $\nu(s) > \nu(t)$, $s \succ_1 t$, hence $s \succ_2 t$.
  If the rule is (π_1), then $s = \pi_1\langle u, v\rangle$ and $t = u$. Let $D(t)$ be $\{\!|t_1, \ldots, t_n|\!\}$; $D(s)$ equals $\{\!|s|\!\}$. Now, $\nu(s) \geq \nu(t_i)$ and $|s| > |t_i|$ for every i, so $s \succ_1 t_i$ for every i, and therefore $s \succ_2 t$. Similarly if the rule is (π_2) or $(\eta\pi)$.
  If the rule is $(\oplus-)$ (resp. (ϵ), (ι)), then $s \succ_2 t$ by (*i*) (resp. (*ii*), (*iii*)).
  If the rule is $(\oplus)$, then $D(s) = D(t)$, so $s \succeq_2 t$. However, $\nu(s) > \nu(t)$ by assumption, so $s \succ_3 t$ as well.
  If the rule is $(\iota\pi_1)$, then s is of the form $\iota\langle\pi_1 u, v\rangle \oplus w$, $t = (u \oplus \epsilon\pi_2 u) \oplus w$, so $D(s) = D(\iota\langle\pi_1 u, v\rangle) \uplus D(w)$, $D(t) = D(u \oplus \epsilon\pi_2 u) \uplus D(w)$, so by (*v*), $s \succ_2 t$. Similarly with $(\iota\pi_2)$, using (*iv*).
  Finally, in the case of the rules (L), $(L\oplus)$, (LC) and $(L\epsilon)$, $D(s) = \{\!|s|\!\}$ and $D(t) = \{\!|t|\!\}$, so $s \succ_2 t$ since $\nu(s) > \nu(t)$.
- We now examine the general case, where $\mathcal{C}$ is any context. We prove it by structural induction on $\mathcal{C}$; we have just proved the base case.
  If $\mathcal{C} = \mathcal{C}_1 \oplus u$ for some term u, then $D(s) = D(\mathcal{C}_1[s_1]) \uplus D(u) \succeq_1^{mul} D(\mathcal{C}_1[t_1]) \uplus D(u)$ (by induction hypothesis) $= D(t)$. So $s \succeq_2 t$; since $\nu(s) > \nu(t)$, $s \succ_3 t$ as well. Similarly if $\mathcal{C}$ has the form $u \oplus \mathcal{C}_1$.
  Finally, if $\mathcal{C}$ is any other context, then $D(s) = \{\!|s|\!\}$, $D(t) = \{\!|t|\!\}$, so because $\nu(s) > \nu(t)$, $s \succ_2 t$.

□

Lemma 1.4 *If s is strongly normalizing, and $t \in \mathrm{RED}^{\theta^-}$, then $s \oplus t \in \mathrm{RED}^{\theta^-}$.*

Proof: By well-founded induction on $(s, \nu(t))$, ordered by the lexicographic product of $\succ_3$ and $>$, using (CR3). (By (CR1), t is strongly normalizing, hence $\nu(t)$ is meaningful.) Consider any one-step reduct u of $s \oplus t$:

- If u was obtained by reducing $s \oplus t$ at the top, then:
  - by $(\oplus-)$: then $u = t \in \mathrm{RED}^{\theta^-}$;
  - by $(\oplus)$: then s has the form $s_1 \oplus s_2$, and $u = s_1 \oplus (s_2 \oplus t)$. By Lemma 1.3 (*i*), $s \succ_3 s_2$, so we can apply the induction hypothesis: $s_2 \oplus t \in \mathrm{RED}^{\theta^-}$. Again by (*i*), $s \succ_3 s_1$, so by induction hypothesis again, $s_1 \oplus (s_2 \oplus t) \in \mathrm{RED}^{\theta^-}$, i.e. $u \in \mathrm{RED}^{\theta^-}$;
  - by $(\iota\pi_1)$: then s has the form $\iota\langle\pi_1 s_1, s_2\rangle$, and $u = (s_1 \oplus \epsilon\pi_2 s_1) \oplus t$. By Lemma 1.3 (*v*), we can apply the induction hypothesis, so $u \in \mathrm{RED}^{\theta^-}$;

- by $(\iota\pi_2)$: then s has the form $s_1 \oplus \epsilon\iota\langle\pi_1 s_2, s_3\rangle$, and $u = (s_1 \oplus \epsilon\pi_2 s_2) \oplus t$. By Lemma 1.3 (vi), $s_1 \oplus \epsilon\iota\langle\pi_1 s_2, s_3\rangle \succ_3 s_1 \oplus \epsilon\pi_2 s_2$, so by induction hypothesis $u \in \mathrm{RED}^{\theta^-}$;

- If u was obtained by contracting a redex in s, then $u = s' \oplus t$, with $s \longrightarrow s'$. By Lemma 1.3 (vii), $s \succ_3 s'$, so by induction hypothesis $u \in \mathrm{RED}^{\theta^-}$;
- If u was obtained by contracting a redex in t, then $u = s \oplus t'$, with $t \longrightarrow t'$. But $\nu(t) > \nu(t')$, so by induction hypothesis $u \in \mathrm{RED}^{\theta^-}$.

In any case $u \in \mathrm{RED}^{\theta^-}$. By (CR3), and since $s \oplus t$ is neutral, it follows that $s \oplus t \in \mathrm{RED}^{\theta^-}$. □

Lemma 1.5 *For any strongly normalizing $\lambda\oplus$-terms $s_1, \ldots, s_n$ and s, if $L(s_1, \ldots, s_n; s)$ is of type θ, then $L(s_1, \ldots, s_n; s) \in \mathrm{RED}^\theta$.*

Proof: By induction on $(\{\!|s_1, \ldots, s_n|\!\}, \nu(s))$ ordered by the lexicographic product of $\succ_1^{mul}$ and $>$, using (CR3). Consider any u such that $L(s_1, \ldots, s_n; s) \longrightarrow u$:

- If the contraction occurs at the top:
  - by rule (L): then for some $1 \leq i < n$, $s_i = t[t'_1/x'_1, \ldots, t'_m/x'_m]$, where $x'_{1\,\theta'^-_1}, \ldots, x'_{m\,\theta'^-_m}$ occur free in t, and the x'_j are pairwise distinct and different from t; and $u = L(s_1, \ldots, s_{i-1}, t'_1, \ldots, t'_m, s_{i+1}, \ldots, s_n; \lambda x_1 \ldots x_{i-1} x'_1 \ldots x'_m x_{i+1} \ldots x_n \cdot s x_1 \ldots x_{i-1} t x_{i+1} \ldots x_n)$. As strict subterms of s_i, $t'_1, \ldots, t'_m$ are strictly less than s_i in the $\succ_1$ ordering. So the induction hypothesis applies, hence $u \in \mathrm{RED}^\theta$;
  - by rule $(L\oplus)$: then $s_n = t \oplus t'$, and $u = L(s_1, \ldots, s_{n-1}, t, t'; \lambda x_1 \ldots x_{n-1} x x' \cdot s x_1 \ldots x_{n-1}(x \oplus x'))$. As strict subterms of s_i, t and t' are strictly less than s_n in the $\succ_1$ ordering. So the induction hypothesis applies, hence $u \in \mathrm{RED}^\theta$;
  - by (LC): then $s_i = s_j$, for some $1 \leq i < j < n$, and u equals:

    $$L(s_1, \ldots, s_i, \ldots, s_{j-1}, s_{j+1}, \ldots, s_n; \lambda x_1 \ldots x_i \ldots x_{j-1} x_{j+1} \ldots x_n \cdot s x_1 \ldots x_i \ldots x_{j-1} x_i x_{j+1} \ldots x_n)$$

    Clearly $\{\!|s_1, \ldots, s_i, \ldots, s_{j-1}, s_j, s_{j+1}, \ldots, s_n|\!\} \succ_1^{mul} \{\!|s_1, \ldots, s_i, \ldots, s_{j-1}, s_{j+1}, \ldots, s_n|\!\}$, so by induction hypothesis $u \in \mathrm{RED}^\theta$;
  - by $(L\epsilon)$: then $s_n = \epsilon t[t_1/x'_1, \ldots, t_n/x'_n]$, and $u = L(s_1, \ldots, s_{n-1}; v)$ for some (rather large) term v. However, $\{\!|s_1, \ldots, s_{n-1}, s_n|\!\} \succ_1^{mul} \{\!|s_1, \ldots, s_{n-1}|\!\}$, so by induction hypothesis $u \in \mathrm{RED}^\theta$;
- If the contraction occurs inside some s_i, $1 \leq i \leq n$, then $\{\!|s_1, \ldots, s_n|\!\}$ decreases in the $\succ_1$ ordering, so by induction hypothesis $u \in \mathrm{RED}^\theta$.
- And if the contraction occurs inside t, then $\{\!|s_1, \ldots, s_n|\!\}$ does not increase, but $\nu(t)$ decreases: by induction hypothesis again, $u \in \mathrm{RED}^\theta$.

Since u is arbitrary and $L(s_1, \ldots, s_n; s)$ is neutral, by (CR3) $L(s_1, \ldots, s_n; s) \in \mathrm{RED}^\theta$. □

Lemma 1.6 *For every $u \in \mathrm{RED}^{\theta_1}$, $v \in \mathrm{RED}^{\theta_2}$, $A(u, v) \in \mathrm{RED}^{\theta_1}$.*

Proof: By induction on $\nu(u) + \nu(v)$. $A(u, v)$ can only reduce to $A(u', v)$ with $u \longrightarrow u'$ or to $A(u, v')$ with $v \longrightarrow v'$, which are reducible by induction hypothesis. Since $A(u, v)$ is neutral, by (CR3) $A(u, v)$ is reducible. □

Lemma 1.7 *For every $u \in \mathrm{RED}^{\theta^-}$, ϵu and ιu are in RED^{θ^-}.*

Proof: By induction on $\nu(u)$ (since by (CR1) u is strongly normalizing). ϵu may reduce either to u (by (ϵ)) or to $\epsilon u'$ with $u \longrightarrow u'$. In the first case, $u \in \mathrm{RED}^{\theta^-}$ by assumption; in the second case $\epsilon u' \in \mathrm{RED}^{\theta^-}$ by induction hypothesis. Since ϵu is neutral, by (CR3) $\epsilon u \in \mathrm{RED}^{\theta^-}$. Similarly for ιu. □

The rest of the proof is standard:

Lemma 1.8 *If* $u \in \mathrm{RED}^{\theta^+}$*,* $v \in \mathrm{RED}^{\theta^-}$*, then* $\langle u, v\rangle \in \mathrm{RED}^{\theta^+\times\theta^-}$*.*

Proof: We first show that $\pi_1\langle u, v\rangle \in \mathrm{RED}^{\theta^+}$ by induction on $\nu(u) + \nu(v)$. $\pi_1\langle u, v\rangle$ may contract either:

- to u by (π_1), which is in RED^{θ^+} by assumption;
- to $\pi_1 w$, where $\langle u, v\rangle \longrightarrow w$ by rule $(\eta\pi)$. Then $u = \pi_1 w$, $v = \pi_2 w$: since by assumption $u \in \mathrm{RED}^{\theta^+}$, the result $\pi_1 w$ of the contraction is in RED^{θ^+};
- to $\pi_1\langle u', v\rangle$, with $u \longrightarrow u'$, then by induction hypothesis $\pi_1\langle u', v\rangle \in \mathrm{RED}^{\theta^+}$;
- or to $\pi_1\langle u, v'\rangle$, with $v \longrightarrow v'$, then by induction hypothesis $\pi_1\langle u, v'\rangle \in \mathrm{RED}^{\theta^+}$.

In any case, $\pi_1\langle u, v\rangle$ contracts to reducible terms only; since it is neutral, by (CR3) $\pi_1\langle u, v\rangle \in \mathrm{RED}^{\theta^+}$.

Similarly, $\pi_2\langle u, v\rangle \in \mathrm{RED}^{\theta^+}$. By Definition 1.1, $\langle u, v\rangle \in \mathrm{RED}^{\theta^+\times\theta^-}$. □

Lemma 1.9 *If for every* $u \in \mathrm{RED}^{\theta_2}$*,* $v[u/x_{\theta_2}] \in \mathrm{RED}^{\theta_1}$*, then* $\lambda x_{\theta_2} \cdot v \in \mathrm{RED}^{\theta_2\to\theta_1}$*.*

Proof: The claim follows from the fact that $(\lambda x \cdot v)u \in \mathrm{RED}^{\theta_1}$ for all reducible u. We prove the latter by induction on $\nu(u)+\nu(v)$. The term $(\lambda x \cdot v)u$ contracts either to $v[u/x]$, which is in RED^{θ_1} by assumption; or to $(\lambda x \cdot u')v$ with $u \longrightarrow u'$, or to $(\lambda x \cdot u)v'$ with $v \longrightarrow v'$, in which cases the induction hypothesis applies, so that the contractum is in RED^{θ_1}. Since $(\lambda x \cdot v)u$ is neutral, by (CR3) it follows that $(\lambda x \cdot v)u \in \mathrm{RED}^{\theta_1}$. □

Theorem 1.10 *Every typed* $\lambda\oplus$*-term is reducible, hence strongly normalizing.*

Proof: We show that: (1) $u[v_1/x_1, \ldots, v_n/x_n]$ is reducible for every typed term u and every reducible terms $v_1, \ldots, v_n$ of the right types, by structural induction on u.

If u is a variable x_i, $1 \le i \le n$, then $u[v_1/x_1, \ldots, v_n/x_n] = v_i$ is reducible by assumption. If u is another variable, then $u = u[v_1/x_1, \ldots, v_n/x_n]$ is normal, hence reducible by (CR3).

If u is of the form $\pi_1 w$, then by induction hypothesis $w[v_1/x_1, \ldots, v_n/x_n]$ is reducible, hence by definition $u[v_1/x_1, \ldots, v_n/x_n]$ is reducible. Similarly if u is of the form $\pi_2 w$ or ww'.

If u is of the form $\langle w, w'\rangle$, then the conclusion follows from Lemma 1.8.

If u is of the form $\lambda x \cdot w$, then for every reducible v, $w[v_1/x_1, \ldots, v_n/x_n, v/x]$ is reducible by induction hypothesis. By Lemma 1.9, $u[v_1/x_1, \ldots, v_n/x_n]$ is reducible.

If u is of the form $w \oplus w'$, then by induction hypothesis $w[v_1/x_1, \ldots, v_n/x_n]$ is reducible, hence strongly normalizing by (CR1), and $w'[v_1/x_1, \ldots, v_n/x_n]$ is reducible. By Lemma 1.4, $u[v_1/x_1, \ldots, v_n/x_n]$ is reducible.

Similarly, we conclude by using Lemma 1.5 if u is of the form $L(s_1, \ldots, s_n; s)$, by Lemma 1.6 if $u = A(w, w')$, and by Lemma 1.7 if u has the form ϵw or ιw.

Claim (1) is now proved. Now, every variable is reducible, so taking $v_1, \ldots, v_n$ as $x_1, \ldots, x_n$ respectively, we conclude that u is reducible. By (CR1), u is therefore strongly normalizing. □

1.2 Confluence of Typed $\lambda\mathsf{ev}Q$

In Part IIIa, Section 3.1, we concluded that the typed $\lambda\mathsf{ev}Q$-calculus was confluent from local confluence properties of the untyped $\lambda\mathsf{ev}Q$-calculus. Local confluence of the typed calculus was inferred from the subject reduction property. However, this argument is only correct in type assignment systems, where types are viewed as predicates on untyped terms. This does not apply to Church-style typed terms, where variables, and subterms of the form 1^ℓ, $\uparrow^\ell$, id^ℓ, $\Uparrow^\ell u$ and $Q^\ell u$ are annotated with types (see Definition 3.7, Part II): see for instance [MS95] for the case of the simply-typed λ-calculus.

We therefore have to change our proof slightly — in fact, to check all critical pairs as *typed* critical pairs. Then, a difficulty awaits us: we compute critical pairs of rule *schemas* depending on the values of integer variables ℓ, $\mathcal{L}$ subject to certain sets of linear constraint; then, although it is easy to infer types for terms, it is very difficult to infer types for term *schemas*. We avoid the difficulty by using a simple criterion for deciding when the critical pair (or rather, critical pair schema) is joinable in the typed case.

Consider for instance a generic (untyped) critical pair:

$$\begin{tikzcd}
 & u \arrow[dl] \arrow[dr] & \\
v \arrow[dr, dashed, "*"'] & & v' \arrow[dl, dashed, "*"] \\
 & w &
\end{tikzcd}$$

Our criterion $C(w)$ is that if we know the types of free (meta-)variables of w, and we know the type of w, then we know all type annotations of all subterms of w. If, for each untyped critical pair as above, $C(w)$ holds, then we claim that the corresponding typed critical pair is joined. Indeed, the typed reductions in Church-style are such that the (unique) type of all reducts of some term u is exactly the type of u; so the type of w is determined. Then, all the (meta-)variables occurring in w are place-holders for real terms whose types we actually know. So by $C(w)$, there is a unique way of decorating all subterms of w with types so that w has the given type, and therefore the typed critical pair joins.

It is almost impossible however to compute these types, so we design a new approximate criterion for detecting when $C(w)$ holds without computing any types. The idea is to abstract types as sets of components of these types that we know. We first represent types:

$$\varsigma_1 \stackrel{\Box}{\Rightarrow} \ldots \stackrel{\Box}{\Rightarrow} \varsigma_\ell \stackrel{\Box}{\Rightarrow} \Phi$$

where Φ is either a base type b or a stack type ς, as infinite lists $[\varsigma_1, \ldots, \varsigma_\ell, \Phi, 0, 0, \ldots]$, where 0 is a new distinguished object. We now abstract our knowledge of the type viewed as a list $[\varsigma_1, \ldots, \varsigma_i, \ldots]$, by a set of integer indexes i where we know ς_i. For instance, if we know the type τ of u, then this set will be $\{1, 2, \ldots\}$; and $Q_T^\ell u$ will then be (included in) $\{1, 2, \ldots, \ell-1, \ell+1, \ldots\}$.

This means that we design an abstract interpretation of the type inferencing problem, where only the fact that some component of a type is (in principle) deducible is computed, and where the information is given as a lower approximant, i.e. a subset of the intended set of integer indices. We use the following functions *bottom_up*, which takes a term u and returns (a lower approximant of) the set of indices of components of its type that we know for sure, knowing the types of its free (meta-)variables; and *top_down*, which takes a term u and returns true whenever u and all of its subterms have (for sure) unique types determined by the knowledge of the type of u and of all its free (meta-)variables. We write $[a, b]$ for the set $\{i \in \mathbb{N} \mid a \leq i \wedge i \leq b\}$, and $[a, +\infty[$ for the set $\{i \in \mathbb{N} \mid a \leq i\}$. The tests of the form $i \in S$ are to be read as "it is sure (provable) that i is in S"; similarly for inclusions $\subseteq$.

From this description, it is clear that we can use finite unions of disjoint intervals of integers as the data-structure representing sets of integers. A nice way of representing them is as lists of integer expressions (which will be, as for the confluence tests in the untyped case, integer constants or linear functions of one integer variable ℓ or $\mathcal{L}$), of the form $[i_1, i_2, \ldots, i_n]$, where it is provable that $i_1 \leq i_2 \leq \ldots \leq i_n$, but none of $i_1 = i_2$, $\ldots$, $i_{n-1} = i_n$ is provable from the set of linear constraints on integer variables defined by the critical pair. Then such a list denotes $\emptyset$ if the list is empty ($n = 0$), or $[i_1, +\infty[\backslash S$, where S denotes the rest of the list, that is if the list if of the form $i_1 :: S$ (:: denotes cons, i.e. the operator that tacks i_1 in front of S and returns the new list). In other words, the list $[i_1, i_2, \ldots, i_n]$ denotes the set $[i_1, i_2[\cup[i_3, i_4[\cup \ldots \begin{cases} [i_{n-1}, i_n[& \text{if } n \text{ is even} \\ [i_n, +\infty[& \text{if } n \text{ is odd} \end{cases}$.

The basic set constructions are defined by (where $\leq$ denotes provable equality from the set of integer constraints considered):

$$\begin{array}{ll}
 & (*\ \mathit{int_minus}(i, l) \text{ computes } [i, +\infty[\backslash l\ *) \\
\mathit{int_minus}(i, []) & = [i] \\
\mathit{int_minus}(i, i' :: l) & = \text{if } i' \leq i \text{ then } \mathit{int_inter}(i, l) \\
 & \quad \text{else if } i \leq i' \text{ then } i :: i' :: l \\
 & \quad \text{else error}
\end{array}$$

$$
\begin{array}{ll}
bottom\_up(x) & = [1,+\infty[\\
bottom\_up(()) & = [1,+\infty[\\
bottom\_up(\lambda^\ell u) & = \begin{cases} (bottom\_up(u)\cap[1,\ell])\cup[\ell+1,+\infty[& \text{if } \ell\in bottom\_up(u) \text{ and } \ell+1\in bottom\_up(u) \\ (bottom\_up(u)\cap[1,\ell])\cup[\ell+2,+\infty[& \text{otherwise} \end{cases} \\
bottom\_up(\lambda x\cdot u) & = \begin{cases} [1,+\infty[& \text{if } bottom\_up(u)=[1,+\infty[\\ [2,+\infty[& \text{otherwise} \end{cases} \\
bottom\_up(u\star^\ell v) & = \begin{cases} ((bottom\_up(u)\cup bottom\_up(v))\cap[1,\ell]) & \\ \qquad \cup[\ell+1,+\infty[& \text{if } \ell+1\in bottom\_up(u) \quad (\ell\geq 0) \\ (bottom\_up(u)\cup bottom\_up(v))\cap[1,\ell] & \text{otherwise} \end{cases} \\
bottom\_up(u\circ^\ell v) & = (bottom\_up(u)\setminus\{\ell\})\cup(bottom\_up(v)\cap[1,\ell]) \\
bottom\_up(u\bullet^\ell v) & = ((bottom\_up(u)\cup bottom\_up(v))\cap[1,\ell]) \\
 & \quad \cup(bottom\_up(u)\cap bottom\_up(v)\cap[1,\ell+1]) \\
 & \quad \cup[\ell+2,+\infty[\quad (\ell\geq 0) \\
bottom\_up(\Uparrow^\ell u) & = bottom\_up(u)\setminus[\ell,\ell+1] \\
bottom\_up(\uparrow^\ell) & = [1,\ell-1]\cup[\ell+2,+\infty[\\
bottom\_up(1^\ell) & = [1,\ell-1] \\
bottom\_up(id^\ell) & = [1,\ell-1]\cup[\ell+2,+\infty[\\
bottom\_up(\mathtt{ev}^\ell uv) & = ((bottom\_up(u)\cup bottom\_up(v))\cap[1,\ell-1]) \\
 & \quad \cup\{i-1 \mid i\in bottom\_up(u)\cap[\ell+1,+\infty[\} \\
bottom\_up(Q^\ell u) & = (bottom\_up(u)\cap[1,\ell-1]) \\
 & \quad \cup\{i+1 \mid i\in bottom\_up(u)\cap[\ell,+\infty[\} \\
bottom\_up(\uparrow u) & = bottom\_up(u) \\
bottom\_up(1u) & = \begin{cases} [1,+\infty[& \text{if } 1\in bottom\_up(u) \\ \emptyset & \text{otherwise} \end{cases}
\end{array}
$$

Figure 1: Function *bottom_up*

$$
\begin{array}{ll}
top\_down(x) & = true \\
top\_down(()) & = true \\
top\_down(\lambda^\ell u) & = top\_down(u) \\
top\_down(\lambda x\cdot u) & = top\_down(u) \\
top\_down(u\star^\ell v) & = (\ell+1\in bottom\_up(u)\vee[\ell+1,+\infty[\subseteq bottom\_(v)) \\
 & \quad \wedge top\_down(u)\wedge top\_down(v) \quad (\ell\geq 0) \\
top\_down(u\circ^\ell v) & = (\ell\in bottom\_up(u)\vee \ell+1\in bottom\_up(v)) \\
 & \quad \wedge top\_down(u)\wedge top\_down(v) \\
top\_down(u\bullet^\ell v) & = top\_down(u)\wedge top\_down(v) \quad (\ell\geq 0) \\
top\_down(\Uparrow^\ell u) & = top\_down(u) \\
top\_down(\uparrow^\ell) & = true \\
top\_down(1^\ell) & = true \\
top\_down(id^\ell) & = true \\
top\_down(\mathtt{ev}^\ell uv) & = (\ell\in bottom\_up(u)\vee \ell\in bottom\_up(v)) \\
 & \quad \wedge top\_down(u)\wedge top\_down(v) \\
top\_down(Q^\ell u) & = top\_down(u) \\
top\_down(\uparrow u) & = 1\in bottom\_up(u)\wedge top\_down(u) \\
top\_down(1u) & = 1\in bottom\_up(u)\wedge top\_down(u)
\end{array}
$$

Figure 2: Function *top_down*

$$
\begin{array}{ll}
 & (*\ \mathit{int\_inter}(i,l) \text{ computes } [i,+\infty[\cap l\ *) \\
\mathit{int\_inter}(i,[]) & = [] \\
\mathit{int\_inter}(i,i'::l) & = \text{if } i \leq i' \text{ then } i'::l \\
 & \text{else if } i' \leq i \text{ then } \mathit{int\_minus}(i,l) \\
 & \text{else error}
\end{array}
$$

$$
\begin{array}{ll}
 & (*\ \mathit{inter}(l_1,l_2) \text{ computes } l_1 \cap l_2\ *) \\
\mathit{inter}([],l_2) & = [] \\
\mathit{inter}(l_1,[]) & = [] \\
\mathit{inter}(i_1::l_1, i_2::l_2) & = \text{if } i_1 \leq i_2 \text{ then } \mathit{int\_minus}(i_2, \mathit{union}(l_1,l_2)) \\
 & \text{else if } i_2 \leq i_1 \text{ then } \mathit{int\_minus}(i_1, \mathit{union}(l_1,l_2)) \\
 & \text{else error}
\end{array}
$$

$$
\begin{array}{ll}
 & (*\ \mathit{union}(l_1,l_2) \text{ computes } l_1 \cup l_2\ *) \\
\mathit{inter}([],l_2) & = l_2 \\
\mathit{inter}(l_1,[]) & = l_1 \\
\mathit{inter}(i_1::l_1, i_2::l_2) & = \text{if } i_1 \leq i_2 \text{ then } \mathit{int\_minus}(i_1, \mathit{diff}(l_1, i_2::l_2)) \\
 & \text{else if } i_2 \leq i_1 \text{ then } \mathit{int\_minus}(i_2, \mathit{diff}(l_2, i_1::l_1)) \\
 & \text{else error}
\end{array}
$$

$$
\begin{array}{ll}
 & (*\ \mathit{diff}(l_1,l_2) \text{ computes } l_1 \setminus l_2\ *) \\
\mathit{diff}([],l_2) & = [] \\
\mathit{diff}(l_1,[]) & = l_1 \\
\mathit{diff}(i_1::l_1, i_2::l_2) & = \text{if } i_1 \leq i_2 \text{ then } i_1 :: \mathit{union}(l_1, i_2::l_2) \\
 & \text{else if } i_2 \leq i_1 \text{ then } \mathit{inter}(i_1::l_1, l_2) \\
 & \text{else error}
\end{array}
$$

$$
\begin{array}{ll}
 & (*\ \mathit{must\_be\_in}(i,l) \text{ is true iff } i \in l \text{ is sure (provable) } *) \\
\mathit{must\_be\_in}(i,[]) & = \mathit{false} \\
\mathit{must\_be\_in}(i,i'::l) & = i' \leq i \wedge \mathit{must\_not\_be\_in}(i,l)
\end{array}
$$

$$
\begin{array}{ll}
 & (*\ \mathit{must\_not\_be\_in}(i,l) \text{ is true iff } i \notin l \text{ is provable } *) \\
\mathit{must\_not\_be\_in}(i,[]) & = \mathit{true} \\
\mathit{must\_not\_be\_in}(i,i'::l) & = i < i' \vee \mathit{must\_be\_in}(i,l)
\end{array}
$$

Finally, it turns out that for all critical pairs in $\lambda\mathsf{ev}Q$ and $\lambda\mathsf{ev}Q_H$, $C(w)$ is computed without ever leading to "error" and returns true, so the typed versions of these calculi, in Church-style, are again locally confluent. By the same proofs as in Part IIIa for $\lambda\mathsf{ev}Q$, and as in Section 2 of this Part IIIb, they are confluent.

1.3 Conservativity

Another problem lies in our counterexample to property (2) in Section 4, Part IIIa (more precisely, to the conjecture that $G(u) \longrightarrow^* G(v)$ in $\lambda\mathsf{ev}Q$, resp. $\lambda\mathsf{ev}Q_H$, implies $u \longrightarrow^* v$ in λ_{S4}, resp. $\lambda_{\mathrm{S4}H}$). The example we give is indeed not typable, contrarily to what is claimed. Instead, consider:

$$
\begin{array}{ll}
u & = \mathsf{unbox}\ ((\mathsf{unbox}\ x)(\mathsf{unbox}\ y))^{\text{\textquoteleft}} \\
v & = (\mathsf{unbox}\ (\mathsf{unbox}\ x)^{\text{\textquoteleft}})(\mathsf{unbox}\ (\mathsf{unbox}\ y)^{\text{\textquoteleft}})
\end{array}
$$

where $x : \Box(\Phi_1 \Rightarrow \Box\Phi_2)$, $y : \Box\Phi_1$. Then u and v have type $\Box\Phi_2$, and:

$$
\begin{array}{ll}
G(u) & = \mathsf{ev}_T^1((\mathsf{ev}_T^2(Q_T^1 x)id^1) \star^1 (\mathsf{ev}_T^2(Q_T^1 y)id^1))() \\
G(v) & = (\mathsf{ev}_T^1(\mathsf{ev}_T^2(Q_T^1 x)id^1)())(\mathsf{ev}_T^1(\mathsf{ev}_T^2(Q_T^1 y)id^1)())
\end{array}
$$

Now $G(u)$ rewrites to $G(v)$ in one application of rule $(\mathsf{ev}\star^1)$, but u does not rewrite to v. In fact, u can only reduce to $(\mathsf{unbox}\ x)(\mathsf{unbox}\ y)$, which is normal.

1.4 Other Minor Corrections

There are several slight technical problems with our treatment of the λ_{S4}-calculus. The first is the fact that rule (box) should actually be:

$$\begin{array}{l}\mathsf{box}\ u\ \mathsf{with}\ v_1,\ldots,v_n\ \mathsf{for}\ x_1,\ldots,x_n \\ \to \mathsf{box}\ u[(\mathsf{box}\ v'_i\ \mathsf{with}\ z_1,\ldots,z_m\ \mathsf{for}\ y_1,\ldots,y_m)/x_i] \\ \qquad \mathsf{with}\ v_1,\ldots,v_{i-1},w_1,\ldots,w_m,v_{i+1},\ldots,v_n \\ \qquad \mathsf{for}\ x_1,\ldots,x_{i-1},z_1,\ldots,z_m,x_{i+1},\ldots,x_n\end{array}$$

provided that $v_i = \mathsf{box}\ v'_i\ \mathsf{with}\ w_1,\ldots,w_m\ \mathsf{for}\ y_1,\ldots,y_m$, and where $z_1, \ldots, z_m$ are fresh variables.

The difference is that we use $\mathsf{box}\ v'_i\ \mathsf{with}\ z_1,\ldots,z_m\ \mathsf{for}\ y_1,\ldots,y_m$ instead of (some variant of) $v'^{\text{‘}}_i$. The latter, or rather $v'^{\text{‘}}_i[z_1/y_1,\ldots,z_m/y_m]$, is $\mathsf{box}\ v'_i\ \mathsf{with}\ z_{i_1},\ldots,z_{i_k}\ \mathsf{for}\ y_{i_1},\ldots,y_{i_k}$, where $1 \leq i_1 < i_2 < \ldots < i_k \leq m$. That is, it is not in general equal to the former, rather it rewrites in finitely many (gc) steps to the latter.

Changing (box) to the new rule above is the right thing to do: all the arguments that we have developed until now, except in Part II, Section 5.3 and a few other places, actually use this form of the rule. The argument of Part II, Section 5.3 is easy to adapt (nothing really changes). The other occurrences are:

- In Part II, Section 4.2, Lemma 4.4, p.34. Between (box) and $(\eta\mathsf{box})$ (first case), when $i = j$:

  $$\begin{array}{ll}\mathsf{box} & \mathsf{unbox}\ x_i \\ & \mathsf{with}\ v_1,\ldots,v_{i-1},(\mathsf{box}\ u\ \mathsf{with}\ w_1,\ldots,w_m\ \mathsf{for}\ y_1,\ldots,y_m),v_{i+1},\ldots,v_n \\ & \mathsf{for}\ x_1,\ldots,x_i,\ldots,x_n\end{array}$$

  reduces to $\mathsf{box}\ u\ \mathsf{with}\ w_1,\ldots,w_m\ \mathsf{for}\ y_1,\ldots,y_m$ by $(\eta\mathsf{box})$, and to:

  $$\begin{array}{ll}\mathsf{box} & \mathsf{unbox}\ (\mathsf{box}\ u\ \mathsf{with}\ z_1,\ldots,z_m\ \mathsf{for}\ y_1,\ldots,y_m) \\ & \mathsf{with}\ v_1,\ldots,v_{i-1},w_1,\ldots,w_m,v_{i+1},\ldots,v_n \\ & \mathsf{for}\ x_1,\ldots,x_{i-1},z_1,\ldots,z_m,x_{i+1},\ldots,x_n\end{array}$$

  by (box), which rewrites to:

  $$\begin{array}{ll}\mathsf{box} & u[z_1/y_1,\ldots,z_m/y_m] \\ & \mathsf{with}\ v_1,\ldots,v_{i-1},w_1,\ldots,w_m,v_{i+1},\ldots,v_n \\ & \mathsf{for}\ x_1,\ldots,x_{i-1},z_1,\ldots,z_m,x_{i+1},\ldots,x_n\end{array}$$

  by (unbox), and this is α-equivalent to:

  $$\begin{array}{ll}\mathsf{box} & u \\ & \mathsf{with}\ v_1,\ldots,v_{i-1},w_1,\ldots,w_m,v_{i+1},\ldots,v_n \\ & \mathsf{for}\ x_1,\ldots,x_{i-1},y_1,\ldots,y_m,x_{i+1},\ldots,x_n\end{array}$$

  which rewrites to $\mathsf{box}\ u\ \mathsf{with}\ w_1,\ldots,w_m\ \mathsf{for}\ y_1,\ldots,y_m$ by (gc).

- In Part II, Section 4.2, Lemma 4.4, p.34. Between (box) and $(\eta\mathsf{box})$ (second case):

  $$\begin{array}{ll}\mathsf{box}\ u & \mathsf{with}\ v_1,\ldots,v_{i-1},(\mathsf{box}\ \mathsf{unbox}\ y_j\ \mathsf{with}\ w_1,\ldots,w_m\ \mathsf{for}\ y_1,\ldots,y_m),v_{i+1},\ldots,v_n \\ & \mathsf{for}\ x_1,\ldots,x_i,\ldots,x_n\end{array}$$

  reduces in one step to:

  $$\mathsf{box}\ u\ \mathsf{with}\ v_1,\ldots,v_{i-1},w_j,v_{i+1},\ldots,v_n\ \mathsf{for}\ x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n$$

  by $(\eta\mathsf{box})$, or to:

  $$\begin{array}{ll}\mathsf{box}\ u[\mathsf{box}\ \mathsf{unbox}\ y_j\ \mathsf{with}\ z_1,\ldots,z_m\ \mathsf{for}\ y_1,\ldots,y_m/x_i] & \mathsf{with}\ v_1,\ldots,v_{i-1},w_1,\ldots,w_m,v_{i+1},\ldots,v_n \\ & \mathsf{for}\ x_1,\ldots,x_{i-1},z_1,\ldots,z_m,x_{i+1},\ldots,x_n\end{array}$$

by (box). This rewrites to:

$$\begin{array}{ll} \mathsf{box}\ u[z_j/x_i] & \mathsf{with}\ v_1, \ldots, v_{i-1}, w_1, \ldots, w_m, v_{i+1}, \ldots, v_n \\ & \mathsf{for}\ x_1, \ldots, x_{i-1}, z_1, \ldots, z_m, x_{i+1}, \ldots, x_n \end{array}$$

by $(\eta\mathsf{box})$, then to:

$$\begin{array}{ll} \mathsf{box}\ u[z_j/x_i] & \mathsf{with}\ v_1, \ldots, v_{i-1}, w_j, v_{i+1}, \ldots, v_n \\ & \mathsf{for}\ x_1, \ldots, x_{i-1}, z_j, x_{i+1}, \ldots, x_n \end{array}$$

by (gc), and this is α-equivalent to:

$$\begin{array}{ll} \mathsf{box}\ u & \mathsf{with}\ v_1, \ldots, v_{i-1}, w_j, v_{i+1}, \ldots, v_n \\ & \mathsf{for}\ x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n \end{array}$$

This confusion between both (box) rules was actually not very serious, because we can postpone (gc) and (ctract):

Lemma 1.11 *For every rule R in λ_{S4H}, if $u \overset{(\mathsf{ctract})}{\longrightarrow} v \overset{R}{\longrightarrow} w$, then $u \overset{R}{\longrightarrow}{}^{+} v' \overset{(\mathsf{ctract})}{\longrightarrow}{}^{*} w$ for some v'.*
More precisely, either: (i) $u \overset{R}{\longrightarrow}{}^{+} v' \overset{(\mathsf{ctract})}{\longrightarrow} w$ or: (ii) $u \overset{R}{\longrightarrow} v' \overset{(\mathsf{ctract})}{\longrightarrow}{}^{*} w$.

Proof: If R is (ctract) itself, the lemma is obvious (we are both in cases (i) and (ii)). Otherwise, observe that R is left-linear. There are several cases, according to whether the redex in u is above that in v, below, or neither. In the last case, the two contractions simply permute (i.e., both (i) and (ii) hold; here, the contractions could be done in parallel).

If the redex in u is above that in v, then $u = \mathcal{C}[\mathsf{box}\ u'\ \mathsf{with}\ \ldots, v_i, \ldots, v_j, \ldots\ \mathsf{for}\ \ldots, x_i, \ldots, x_j, \ldots]$ with $i \neq j$, $v_i = v_j$ and $v = \mathcal{C}[\mathsf{box}\ u'[x_i/x_j]\ \mathsf{with}\ \ldots, v_i, \ldots, \ldots\ \mathsf{for}\ \ldots, x_i, \ldots, \ldots]$. If the R-rewrite from v to w occurs in $u'[x_i/x_j]$, then because R is linear, we could have already done it in u', and the two contractions simply permute. If the R-rewrite from v to w occurs in some v_k, $k \neq i$, $k \neq j$, then the two rewrites permute again. Finally, if the R-rewrite from v to w occurs in v_i, rewriting it to v'_i, then u rewrites in two R steps to $v' = \mathcal{C}[\mathsf{box}\ u'\ \mathsf{with}\ \ldots, v'_i, \ldots, v'_i, \ldots\ \mathsf{for}\ \ldots, x_i, \ldots, x_j, \ldots]$, then to w by (ctract), so (i) holds.

If the redex in u is below that in v, then because R is left-linear, we may first contract the R-redex in u, getting v', and then contract all the residuals of the (ctract)-redex in u inside v': so (ii) holds. □

Lemma 1.12 *For every rule R in λ_{S4H} except (ctract), if $u \overset{(\mathsf{gc})}{\longrightarrow} v \overset{R}{\longrightarrow} w$, then $u \overset{R}{\longrightarrow} v' \overset{(\mathsf{gc})}{\longrightarrow}{}^{*} w$ for some v'.*

Proof: If R is (gc) as well, the lemma is obvious. Otherwise, there are several cases, according to whether the redex in u is above that in v, below, or neither. In the last case, the result is obvious (the contractions could be done in parallel).

If the redex in u is above that in v, then $u = \mathcal{C}[\mathsf{box}\ u'\ \mathsf{with}\ \ldots, v_i, \ldots, \ldots\ \mathsf{for}\ \ldots, x_i, \ldots]$ with $x_i \notin \mathrm{fv}\, u'$ and $v = \mathcal{C}[\mathsf{box}\ u'\ \mathsf{with}\ \ldots, \ldots\ \mathsf{for}\ \ldots, \ldots]$. If the R-rewrite from v to w occurs in u' or in any v_k, $k \neq i$, then because R is linear, the two contractions permute. If the R-rewrite from v to w occurs in v_i, rewriting it to v'_i, then u rewrites in no R step to $v' = u$, which rewrites to w by (gc).

If the redex in u is below that in v, then because R is left-linear, we may first contract the R-redex in u, getting v', and then contract all the residuals of the (gc)-redex in u inside v'. □

It follows that:

Lemma 1.13 *If $u \longrightarrow^{*} v$ in λ_{S4} (resp. λ_{S4H}), then $u \longrightarrow^{*} v_1 \overset{(\mathsf{gc})}{\longrightarrow}{}^{*} v_2 \overset{(\mathsf{ctract})}{\longrightarrow}{}^{*} v$ for some v_1, v_2, and where the rewrites from u to v_1 are λ_{S4}-rewrites (resp. λ_{S4H}-rewrites) not using (gc) or (ctract).*

Proof: By Lemma 1.11, we can define a transformation on rewrites r from u to v as follows. Let r be a rewrite from u to v, i.e. a sequence of contractions of the form:

$$u = u_0 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_n = v$$

This can be written in a unique way as:

$$u = u_0 \overset{R}{\longrightarrow}{}^{*} u_{i_1} \overset{(\mathsf{ctract})}{\longrightarrow}{}^{+} u_{j_1} \overset{R}{\longrightarrow}{}^{+} u_{i_2} \overset{(\mathsf{ctract})}{\longrightarrow}{}^{+} u_{j_2} \ldots \overset{(\mathsf{ctract})}{\longrightarrow}{}^{+} u_{j_{k-1}} \overset{R}{\longrightarrow}{}^{+} u_{i_k} \overset{(\mathsf{ctract})}{\longrightarrow}{}^{*} u_n = v$$

where $0 \leq i_1 < j_1 < i_2 < j_2 < \ldots < j_{k-1} < i_k \leq n$, $k \geq 1$ and R denotes any rule except (ctract). If $k \geq 2$, then we apply Lemma 1.11 on the subsequence $u_{j_1-1} \xrightarrow{(\mathsf{ctract})} u_{j_1} \xrightarrow{R} u_{j_1+1}$. Note that, if $j_1 = i_1 + 1$, then this decreases k, otherwise it decreases $j_i - i_1$ while leaving k unchanged. So, if we order rewrites r by k first, then $j_1 - i_1$ (lexicographically), then the transformations above make r decrease in this ordering, which is clearly well-founded. Any rewrite that cannot be transformed any longer then has $k = 1$, i.e. it is of the form $= u_0 \xrightarrow{R}{}^* u_{i_1} \xrightarrow{(\mathsf{ctract})}{}^* u_n = v$. Let v_2 be u_{i1}.

Similarly, we transform the rewrite from u to v_2 by repeatedly using Lemma 1.12, and the result follows. □

Lemma 1.14 *The rewrite system* $(\mathsf{gc}), (\mathsf{ctract})$ *is terminating and confluent.*

Proof: Local confluence follows from the arguments in Lemma 5.6, Part I for the λ_{S4}-calculus, and in addition from those in Lemma 4.4, Part II for the λ_{S4H}-calculus. Termination is obvious, since each rule decreases the size of the term strictly. □

Lemma 1.15 *For any rule R other than* $(\mathsf{gc}), (\mathsf{ctract})$*:*

$$\begin{tikzcd}
 & u \arrow[dl, "R"'] \arrow[dr, "{(\mathsf{gc}), (\mathsf{ctract})}"] & \\
u' \arrow[ddr, dashed, "{(\mathsf{gc}), (\mathsf{ctract})}"', "*"] & & u'' \arrow[d, dashed, "R", "*"'] \\
 & & {} \arrow[dl, dashed, "{(\mathsf{gc}), (\mathsf{ctract})}", "*"'] \\
 & v &
\end{tikzcd}$$

Proof: See the proofs of Lemma 5.6, Part I for the λ_{S4}-calculus, and of Lemma 4.4, Part II for the λ_{S4H}-calculus. □

Lemma 1.16 *For any family R of rules other than* $(\mathsf{gc}), (\mathsf{ctract})$*:*

$$\begin{tikzcd}
 & u \arrow[dl, "R"', "*"] \arrow[dr, "{(\mathsf{gc}), (\mathsf{ctract})}", "*"'] & \\
u' \arrow[ddr, dashed, "{(\mathsf{gc}), (\mathsf{ctract})}"', "*"] & & u'' \arrow[d, dashed, "R", "*"'] \\
 & & {} \arrow[dl, dashed, "{(\mathsf{gc}), (\mathsf{ctract})}", "*"'] \\
 & v &
\end{tikzcd}$$

Proof: Let C denote $(\mathsf{gc}), (\mathsf{ctract})$. Since C terminates, we can define the length $\nu(u)$ of the longest C-reduction starting from u. The result is by induction on $\nu(u)$. If $u'' = u$ and the result is obvious. So assume that $u'' \neq u$ (in particular $\nu(u) \neq 0$), in particular:

$$u \xrightarrow{C} u_2 \xrightarrow{C}{}^* u''$$

for some u_2.

Now, we claim that: (1) there is a term v_2 such that $u' \xrightarrow{C}{}^* v_2$ and $u_2 \xrightarrow{R+C}{}^* v_2$. This is by induction on the length of the R-rewrite from u to u'. This is obvious when this length is 0, otherwise we use the following diagram:

$$\begin{tikzcd}
 & & u \arrow[dl, "R"'] \arrow[dr, "{(\mathsf{gc}),(\mathsf{ctract})}"] & & \\
 & u_1 \arrow[dl, "R"', "*"] \arrow[dr, dashed, "{(\mathsf{gc}),(\mathsf{ctract})}", "*"'] & & u_2 \arrow[dl, dashed, "{R\ (\mathsf{gc}),(\mathsf{ctract})}", "*"'] & \\
u' \arrow[dr, dashed, "{(\mathsf{gc}),(\mathsf{ctract})}"', "*"] & & v_1 \arrow[dl, dashed, "{R\ (\mathsf{gc}),(\mathsf{ctract})}", "*"'] & & \\
 & v_2 & & &
\end{tikzcd}$$

where the upper diagram comes from Lemma 1.15 and the lower diagram is by induction hypothesis (the one for claim (1)). Thus we have proved (1).

Now from (1) and Lemma 1.13, $u_2 \xrightarrow{R}{}^* v_3 \xrightarrow{C}{}^* v_2$ for some term v_3. And since $\nu(u_2) < \nu(u)$, we can apply the induction hypothesis and draw the following diagram:

$$\begin{tikzcd}
 & & u_2 \arrow[dl, "R"', "*"] \arrow[dr, "{(\mathsf{gc}),(\mathsf{ctract})}", "*"'] & & \\
 & v_3 \arrow[dl, "{(\mathsf{gc}),(\mathsf{ctract})}"', "*"] \arrow[dr, dashed, "{(\mathsf{gc}),(\mathsf{ctract})}", "*"'] & & u'' \arrow[dl, dashed, "{R\ (\mathsf{gc}),(\mathsf{ctract})}", "*"'] & \\
v_2 \arrow[dr, dashed, "{(\mathsf{gc}),(\mathsf{ctract})}"', "*"] & & u_3 \arrow[dl, dashed, "{(\mathsf{gc}),(\mathsf{ctract})}", "*"'] & & \\
 & v & & &
\end{tikzcd}$$

where the upper diagram comes from induction hypothesis and the lower diagram is by confluence of (gc), (ctract) (Lemma 1.14). □

Therefore:

Lemma 1.17 *If $u \longrightarrow^* v$ in $\lambda^{\approx}_{S4}$ (resp. $\lambda^{\approx}_{S4H}$), then $u \longrightarrow^* v'$ in λ_{S4} (resp. λ_{S4H}) for some term v' such that v rewrites to v' by* (gc), (ctract).

Proof: Because (gc), (ctract) is confluent, we may assume that:

$$u = u_0 \xrightarrow{R}{}^* v_0 \xrightarrow{C}{}^* w_0 {}^* \xleftarrow{C} u_1 \xrightarrow{R}{}^* v_1 \xrightarrow{C}{}^* w_1 {}^* \xleftarrow{C} u_2 \xrightarrow{R}{}^* \ldots \xrightarrow{C}{}^* w_{n-1} {}^* \xleftarrow{C} u_n = v$$

where $C = (\mathsf{gc}), (\mathsf{ctract})$ and R is the set of rules in the calculus minus C. The result is by induction on n. This is clear if $n = 1$. Otherwise, by Lemma 1.16, there are terms v_1' and u_1' such that:

$$w_0 \xrightarrow{R}{}^* v_1' \xrightarrow{C}{}^* u_1' {}^* \xleftarrow{C} v_1$$

So:

$$w_0 \xrightarrow{R}{}^* v_1' \xrightarrow{C}{}^* u_1' {}^* \xleftarrow{C} v_1 \xrightarrow{C}{}^* w_1 {}^* \xleftarrow{C} u_2 \xrightarrow{R}{}^* \ldots \xrightarrow{C}{}^* w_{n-1} {}^* \xleftarrow{C} u_n = v$$

By Lemma 1.14, u_1' and w_1 have a common C-reduct w_0', so:

$$w_0 \xrightarrow{R}{}^* v_1' \xrightarrow{C}{}^* u_1' \xrightarrow{C}{}^* w_0' {}^* \xleftarrow{C} w_1 {}^* \xleftarrow{C} u_2 \xrightarrow{R}{}^* \ldots \xrightarrow{C}{}^* w_{n-1} {}^* \xleftarrow{C} u_n = v$$

or, to make things simpler:

$$w_0 \xrightarrow{R}{}^* v_1' \xrightarrow{C}{}^* w_0' {}^* \xleftarrow{C} u_2 \xrightarrow{R}{}^* \ldots \xrightarrow{C}{}^* w_{n-1} {}^* \xleftarrow{C} u_n = v$$

To show the whole reduction:

$$u = u_0 \xrightarrow{R}{}^* v_0 \xrightarrow{C}{}^* w_0 \xrightarrow{R}{}^* v'_1 \xrightarrow{C}{}^* w'_0 {}^* \xleftarrow{C} u_2 \xrightarrow{R}{}^* \ldots \xrightarrow{C}{}^* w_{n-1} {}^* \xleftarrow{C} u_n = v$$

Now by Lemma 1.13, for some term v'_0:

$$u = u_0 \xrightarrow{R}{}^* v'_0 \xrightarrow{C}{}^* w'_0 {}^* \xleftarrow{C} u_2 \xrightarrow{R}{}^* \ldots \xrightarrow{C}{}^* w_{n-1} {}^* \xleftarrow{C} u_n = v$$

and we apply the induction hypothesis. □

Therefore, we can reason up to (gc), (ctract)-equivalence without losing generality.

A few other details must be modified:

- In Part I, in the proof of Lemma 5.6 (p.22), in the cases of the $(\mathsf{box})/(\mathsf{gc})$ and $(\mathsf{box})/(\mathsf{ctract})$ critical pairs, v_i should be $\mathsf{box}\ v\ \mathsf{with}\ w_1, \ldots, w_m\ \mathsf{for}\ y_1, \ldots, y_m$, not $\mathsf{box}\ v\ \mathsf{with}\ w_1, \ldots, w_m\ \mathsf{for}\ x_1, \ldots, x_m$, and appropriate occurrences of x_k should be changed to y_k, $1 \leq k \leq m$.

- In Part I, in the proof of Lemma 5.6 (p.22), the subcase where $i = j$ was forgotten in the case of $(\mathsf{box})/(\mathsf{gc})$ critical pairs. This case is when:

  $$v_i = \mathsf{box}\ v\ \mathsf{with}\ w_1, \ldots, w_m\ \mathsf{for}\ y_1, \ldots, y_m$$

  and x_i is not free in u, then:

  $$\mathsf{box}\ u\ \mathsf{with}\ \ldots, v_i, \ldots\ \mathsf{for}\ \ldots, x_i, \ldots$$

  reduces in one (box) step either to:

  $$\begin{array}{l} \mathsf{box}\ u[(\mathsf{box}\ v\ \mathsf{with}\ z_1, \ldots, z_m\ \mathsf{for}\ y_1, \ldots, y_m)/x_i] \\ \mathsf{with}\ \ldots, w_1, \ldots, w_m, \ldots \\ \mathsf{for}\ \ldots, z_1, \ldots, z_m, \ldots \end{array}$$

  or to:

  $$\mathsf{box}\ u\ \mathsf{with}\ \ldots, \ldots\ \mathsf{for}\ \ldots, \ldots$$

  by (gc). Since x_i is not free in u, the former is actually:

  $$\begin{array}{l} \mathsf{box}\ u \\ \mathsf{with}\ \ldots, w_1, \ldots, w_m, \ldots \\ \mathsf{for}\ \ldots, z_1, \ldots, z_m, \ldots \end{array}$$

  which reduces to the latter, closing the confluence diagram, in m (gc) steps, since $z_1, \ldots, z_m$, being fresh, cannot be free in u.

- In Part II, Section 3.3, top of p.31, the typse of u and w were badly typeset, and should be $\overline{\varsigma^{\ell-1} \overset{\Box}{\Rightarrow}} \tau_1 \times \varsigma' \overset{\Box}{\Rightarrow} \tau_2$, and that of w should be $\overline{\varsigma^{\ell-1} \overset{\Box}{\Rightarrow}} \varsigma'$.

2 Confluence

2.1 The Base $\lambda\mathsf{ev}Q_H$-Calculus

The main problem in proving the confluence of the typed $\lambda\mathsf{ev}Q_H$-calculus is due to the non-left-linear rules $(\eta\ \bullet)$ and $(\eta \bullet \circ^\ell)$. These are the rules that we have used to show in Part IIIa that the untyped calculus was not confluent. The situation is much like the one encountered in the $\lambda\sigma$-calculus, for which the only proof method that we know of today is based on Thérèse Hardin's interpretation technique.

Consider for example the proof method of Part IIIa, section 3.1. We started by showing that, whenever $u \xrightarrow{\beta_{||}} w$ and $u \xrightarrow{\Sigma} v$, then there was a term t such that $w \xrightarrow{\Sigma}{}^* t$ and v rewrote to t by $\Sigma^* \beta_{||} \Sigma^*$. It is tempting to just substitute Σ_H for Σ above, but it does not work. Indeed, let $u = (1^1 \circ^1 v) \bullet^1 (\uparrow^1 \circ^1 v)$; check that

$u \xrightarrow{\Sigma_H} v$. Imagine that v contains a (β^1)-redex, and let v' be the result of contracting this redex in v. Then $u \xrightarrow{\beta_{||}} w = (1^1 \circ^1 v') \bullet^1 (\uparrow^1 \circ^1 v)$ for instance. To close the confluence diagram, the obvious solution is to reduce the other occurrence of v in w to get $(1^1 \circ^1 v') \bullet^1 (\uparrow^1 \circ^1 v')$, then v' $(= t)$, but in doing this we have used (β^1) to go from w to t, which was not allowed.

There is another solution to close the confluence diagram. The types tell us that v and v' should be non-empty stacks. So, intuitively, v should Σ_H-rewrite to some stack of the form $v_1 \bullet^1 v_2$, and the (β^1)-redexes should be in v_1 or in v_2; say $v' = v'_1 \bullet^1 v'_2$ with $v_1 \xrightarrow{\beta_{||}} v'_1$ and $v_2 \xrightarrow{\beta_{||}} v'_2$. Then we can close the confluence diagram as follows: $w = (1^1 \circ^1 (v'_1 \bullet^1 v'_2)) \bullet^1 (\uparrow^1 \circ^1 (v_1 \bullet^1 v_2)) \longrightarrow v'_1 \bullet^1 (\uparrow^1 \circ^1 (v_1 \bullet^1 v_2))$ (by (1^1)) $\longrightarrow v'_1 \bullet^1 v_2$ (by $(\uparrow^1)$) $= t$ (notice that we did not use (β^1)), while $v = v_1 \bullet^1 v_2 \xrightarrow{\Sigma_H^* \beta_{||} \Sigma_H^*} v'_1 \bullet^1 v_2$ by construction.

The details are a bit more complicated, and use the fact that if there is really a contracted (β^1)-redex in v, then the redex must have sort T, i.e. v contains a subterm of sort T. We formalize it by identifying such terms, which we call *not T-free*. For reasons that will become clear in Section 3, we consider that terms of the form $\Uparrow^\ell u$ are not T-free either, even when u is T-free; this can be justified by the intuition that $\Uparrow^\ell u$ is more or less an abbreviation for $1^\ell \bullet^\ell (u \circ^\ell_S \uparrow^\ell)$, which has 1^ℓ, a term of sort T, as subterm.

Definition 2.1 *We say that a $\lambda\mathsf{ev}Q$-term is* T-free *if and only if none of it subterms (including itself) has sort T or has the form $\Uparrow^\ell u$ for some $\ell \geq 1$ and some term u.*

We also say that a term is Σ_H-normal if and only if it is reducible by no rule in Σ_H. We denote by $\Sigma_H(u)$ the unique normal form of u by Σ_H. (Because of Theorem 2.71 and Lemma 3.1 of Part IIIa, Σ_H is convergent in the typed case, hence this normal form exists and is unique.)

Lemma 2.1 *Let u be a typed $\lambda\mathsf{ev}Q$-term such that:*

(1) *u has type $\overline{\varsigma^\ell \overset{\Box}{\Rightarrow}} \varsigma_{\ell+1}$ for some $\ell \geq 0$, and some stack types $\varsigma_1, \ldots, \varsigma_\ell, \varsigma_{\ell+1}$;*

(2) *u is Σ_H-normal;*

(3) *and u is T-free.*

Then:

(4) *either $\ell = 0$, $u = ()$ (and $\varsigma = \top$);*

(5) *or $\ell \geq 1$ and $u = \mathsf{pop}^\ell_k$ for some $k \geq 0$.*

Proof: Recall that for $k \geq 1$, pop^ℓ_k is defined as $\underbrace{\uparrow^\ell \circ^\ell \ldots \circ^\ell \uparrow^\ell}_{k \text{ times}}$, where $\circ^\ell$ associates to the right; and that pop^ℓ_0 is defined as id^ℓ. When there is a risk of confusion, we write $(1)[u]$ to say that property (1) applies to the term u, and similarly for properties (2) through (5).

The lemma is proved by structural induction on u. Because u is a stack, we have the following cases:

- $u = ()$: then (4) holds;
- $u = s \bullet t$: s has sort T, so this is impossible by $(3)[u]$;
- $u = \uparrow s$: s then obeys (1) (with type $\overline{\varsigma^\ell \overset{\Box}{\Rightarrow}} \tau \times \varsigma_{\ell+1}$ for some τ), (2) and (3); by induction hypothesis, then either $(4)[s]$ or $(5)[s]$ holds; if the former holds, then $s : \top$ and u would not be typable; and if the latter holds, then $\ell \geq 1$, and again u would not be typable, a contradiction;
- $u = s \circ^n_S t$: by $(1)[u]$, we must have (i) $n \leq \ell$. Then, s also obeys (1) with some type $\varsigma_1 \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{n-1} \overset{\Box}{\Rightarrow} \varsigma'_n \overset{\Box}{\Rightarrow} \varsigma_{n+1} \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{\ell+1}$, and obeys (2) and (3). By induction hypothesis, then, either $(4)[s]$ or $(5)[s]$ holds. The former is impossible by typing (property $(1)[u]$), so s equals some term of the form pop^ℓ_i, and $i \geq 0$. If $n < \ell$, then u would be reducible by rules in group (D), so by $(2)[u]$ and (i), $n = \ell$. To recap, $u = \mathsf{pop}^\ell_i \circ^\ell t$. The latter is not Σ_H-normal if $i \geq 2$ (by rule $(\circ^\ell)$) or if $i = 0$ (by rule $(id\circ^\ell)$). So by $(2)[u]$, $i = 1$, i.e. $u = \uparrow^\ell \circ^\ell t$. Then t has type $\overline{\varsigma^\ell \overset{\Box}{\Rightarrow}} \tau \times \varsigma_{\ell+1}$ for some τ, and obeys (1), (2) and (3): by induction hypothesis, either $(4)[t]$ or $(5)[t]$ holds, and by typing in fact $t = \mathsf{pop}^\ell_j$ for some $j \geq 1$, so $u = \mathsf{pop}^\ell_k$ with $k = j + 1 \geq 1$.

- $u = id^n$: by (1), $n = \ell$, then (5) holds with $k = 0$;
- $u = s \bullet^n t$: impossible by (3);
- $u = \uparrow^n$: by (1), $n = \ell$, so (5) holds with $k = 1$;
- $u = \Uparrow^n s$: impossible by (2) (rule $(\eta \Uparrow^n)$ is applicable); or alternatively by (3) (by convention u is not T-free);
- $u = \mathtt{ev}^n_S st$: by (1)[u], we must have (i) $n \leq \ell + 1$. Moreover, s must have a type of the form $\varsigma_1 \overset{\square}{\Rightarrow} \ldots \overset{\square}{\Rightarrow} \varsigma_{n-1} \overset{\square}{\Rightarrow} \varsigma' \overset{\square}{\Rightarrow} \varsigma_n \overset{\square}{\Rightarrow} \ldots \overset{\square}{\Rightarrow} \varsigma_{\ell+1}$. Since s obeys (1), (2) and (3), by induction hypothesis either (4)[s] or (5)[s] holds. By typing only the latter can hold, so $s = \mathsf{pop}^{\ell+1}_i$, with $i \geq 0$. If $n < \ell + 1$, then $u = \mathtt{ev}^n_S st$ would be reducible by some rule in group (E), contradicting (2)[u]; and if $n = \ell + 1$, then u would be reducible by $(\mathtt{ev}id^{\ell+1})$ if $i = 0$, by $(\mathtt{ev} \uparrow^{\ell+1})$ if $i = 1$ and by $(\mathtt{ev}\circ^{\ell+1})$ if $i \geq 2$. By (i), there are no other cases. So the case $u = \mathtt{ev}^n_S st$ is actually impossible.
- $u = Q^n_S t$: by (1)[u], we must have (i) $n \leq \ell - 1$. Moreover, t must have a type of the form $\varsigma_1 \overset{\square}{\Rightarrow} \ldots \overset{\square}{\Rightarrow} \varsigma_{n-1} \overset{\square}{\Rightarrow} \varsigma_n \overset{\square}{\Rightarrow} \ldots \varsigma_\ell \overset{\square}{\Rightarrow} \varsigma_{\ell+1}$. Since t obeys (1), (2) and (3), by induction hypothesis either (4)[t] or (5)[t] holds. The former is impossible by (i), so $t = \mathsf{pop}^{\ell-1}_i$ for some $i \geq 0$ (by (i) $\ell - 1 \geq n \geq 1$). If $i = 0$, then u is reducible by $(Q^n id^\ell)$ because of (i); if $i = 1$, by $(Q^n \uparrow^\ell)$; if $i \geq 2$, by $(Q^n \circ^\ell)$. In any case, this contradicts (2)[u], so this case is impossible as well.

□

Lemma 2.2 *Let u be a typed $\lambda\mathtt{ev}Q$-term such that:*

(1) *u has type $\overline{\varsigma^\ell \overset{\square}{\Rightarrow}}\varsigma_{\ell+1}$ for some $\ell \geq 0$, and some stack types $\varsigma_1, \ldots, \varsigma_\ell, \varsigma_{\ell+1}$;*

(2) *u is Σ_H-normal;*

(3) *and u is* not *T-free.*

Then u is of the form $u_1 \bullet^\ell u_2$.

Proof: As for Lemma 2.1, by structural induction on u. Because u is a stack, we have the following cases:

- $u = ()$: impossible by (3)[u];
- $u = s \bullet t$: claim proved if $\ell = 0$, impossible otherwise;
- $u = \uparrow s$: by (1)[u], $\ell = 0$; also, s obeys (1), (2) and (3) so $s = s_1 \bullet s_2$ for some s_1 and s_2, but this contradicts (2)[u], because u is then reducible by rule $(\uparrow)$;
- $u = s \circ^n_S t$: by (1)[u], we must have (i) $n \leq \ell$. Then, s also obeys (1) with some type $\varsigma_1 \overset{\square}{\Rightarrow} \ldots \overset{\square}{\Rightarrow} \varsigma_{n-1} \overset{\square}{\Rightarrow} \varsigma'_n \overset{\square}{\Rightarrow} \varsigma_{n+1} \overset{\square}{\Rightarrow} \ldots \overset{\square}{\Rightarrow} \varsigma_{\ell+1}$, and obeys (2). We then have two cases, according to whether s is or is not T-free.

  If s is T-free, by Lemma 2.1, either $s = ()$ or $s = \mathsf{pop}^\ell_k$ for some $k \geq 0$. By (i) $\ell \geq n \geq 1$, so $s = \mathsf{pop}^\ell_k$. If $n < \ell$, then u is reducible by $(id^\ell \circ^n)$ (if $k = 0$), $(\uparrow^\ell \circ^n)$ (if $k = 1$) or $(\circ^\ell \circ^n)$ (if $k \geq 2$); so by (2)[u] $n \geq \ell$, and by (i) in fact $n = \ell$. By (3)[u], t cannot be T-free, so t obeys (1), (2) and (3), and by induction hypothesis $t = t_1 \bullet^\ell t_2$ for some t_1 and t_2. Then, the case $k = 0$ is impossible by (1)[u] (typing); if $k = 1$, then u is reducible by rule $(\uparrow^\ell)$, and if $k \geq 2$, u is reducible by rule $(\circ^\ell)$; so in any case we get a contradiction with (2)[u].

  It follows that s is not T-free, i.e. (3)[s] holds: by induction hypothesis, s has the form $s_1 \bullet^\ell s_2$. But then u is reducible by $(\bullet^\ell \circ^n)$ if $n < \ell$, and by $(\bullet^\ell)$ if $n = \ell$; by (i) there is no other case, so u cannot have the form $s \circ^n_S t$.
- $u = id^n$: impossible by (3);
- $u = s \bullet^n t$: by (1)[u], $n = \ell$, and the claim is proved in this case;

- $u = \uparrow^n$: impossible by (3);
- $u = \Uparrow^n s$: impossible by (2) (rule $(\eta \Uparrow^n)$ is applicable);
- $u = \mathsf{ev}^n_S st$: by (1)[u], we must have (i) $n \leq \ell + 1$. Moreover, s must have a type of the form $\varsigma_1 \stackrel{\Box}{\Rightarrow} \ldots \stackrel{\Box}{\Rightarrow} \varsigma_{n-1} \stackrel{\Box}{\Rightarrow} \varsigma' \stackrel{\Box}{\Rightarrow} \varsigma_n \stackrel{\Box}{\Rightarrow} \ldots \stackrel{\Box}{\Rightarrow} \varsigma_{\ell+1}$.

  If s if T-free, then by Lemma 2.1 and typing considerations, $s = \mathsf{pop}^{\ell+1}_k$ for some $k \geq 0$. If $n < \ell + 1$, then u is reducible by $(\mathsf{ev}^n id^{\ell+1})$ if $k = 0$, by $(\mathsf{ev}^n \uparrow^{\ell+1})$ if $k = 1$, and by $(\mathsf{ev}^n \circ^{\ell+1})$ if $k \geq 2$. If $n = \ell + 1$, then u is reducible by $(\mathsf{ev} id^{\ell+1})$ if $k = 0$, by $(\mathsf{ev} \uparrow^{\ell+1})$, and by $(\mathsf{ev} \circ^{\ell+1})$ if $k \geq 2$. By (i), there is no other case; by (2)[u], all these cases are impossible.

  So s is not T-free, i.e. obeys (3). Since s also obeys (1) and (2), by induction hypothesis $s = s_1 \bullet^{\ell+1} s_2$ for some terms s_1 and s_2. If $n < \ell + 1$, then u is reducible by rule $(\mathsf{ev}^n \bullet^{\ell+1})$, and if $n = \ell + 1$, then u is reducible by rule $(\mathsf{ev} \bullet^{\ell+1})$. By (i), there is no other case, so u cannot have the form $\mathsf{ev}^n_S st$.
- $u = Q^n_S t$: by (1)[u], we must have (i) $n \leq \ell - 1$. Moreover, t must have a type of the form $\varsigma_1 \stackrel{\Box}{\Rightarrow} \ldots \stackrel{\Box}{\Rightarrow} \varsigma_{n-1} \stackrel{\Box}{\Rightarrow} \varsigma_n \stackrel{\Box}{\Rightarrow} \ldots \varsigma_\ell \stackrel{\Box}{\Rightarrow} \varsigma_{\ell+1}$. Since t obeys (1), (2) and (3), by induction hypothesis $t = t_1 \bullet^{\ell-1} t_2$ for some terms t_1 and t_2; but then u is reducible by rule $(Q^n \bullet^\ell)$, which contradicts (2)[u].

$\Box$

It follows that typed Σ_H-normal stacks are either () or stacks of the form $u_1 \bullet^\ell u_2 \bullet^\ell \ldots \bullet^\ell u_m \bullet^\ell \mathsf{pop}^\ell_n$, where $m \geq 0$, $n \geq 0$, u_i is a Σ_H-normal term of sort T for every $1 \leq i \leq m$, and if $m \geq 0$ and $n \neq 0$, then $u_m \neq \mathsf{get}^\ell_{n-1}$.

Lemma 2.3 *We say that a reduction step (resp. sequence) is* restricted *if and only it does not use rule $(\eta \bullet)$ and every instance of rule $(\eta \bullet \circ^\ell)$ is of the form:*

$$(\eta \bullet' \circ^\ell) \quad (1^\ell \circ^\ell u) \bullet^\ell (\uparrow^\ell \circ^\ell u) \to u$$

where u is T-free and Σ_H-normal.

For any typed term u, there is a restricted reduction from u to $\Sigma_H(u)$.

Proof: Because Σ_H is convergent in the typed case, we may choose any reduction strategy: let's choose some innermost reduction strategy.

Rule $(\eta \bullet)$ can then only be used on redexes of the form $1u \bullet\!\uparrow u$, where u is Σ_H-normal: if u is T-free, then by Lemma 2.1, either $u = ()$ or $u = \mathsf{pop}^\ell_k$ for some $k \geq 0$, $\ell \geq 1$; both cases are impossible by typing. And if u is not T-free, then by Lemma 2.2 and by typing $u = u_1 \bullet u_2$ for some u_1 and u_2, so we can replace the single reduction step $1u \bullet\!\uparrow u \to u$ (by $(\eta \bullet)$) by the length two-reduction $1u \bullet\!\uparrow u \to u_1 \bullet\!\uparrow u \to u_1 \bullet u_2 = u$ (by (1) followed by $(\uparrow)$).

Similarly, consider reduction steps $(1^\ell \circ^\ell u) \bullet^\ell (\uparrow^\ell \circ^\ell u)$, by $(\eta \bullet \circ^\ell)$, where again u is Σ_H-normal. If u is not T-free, then by Lemma 2.2 $u = u_1 \bullet^\ell u_2$ for some u_1, u_2, so we can replace the latter reduction step by the following two: $(1^\ell \circ^\ell u) \bullet^\ell (\uparrow^\ell \circ^\ell u) \to u_1 \bullet^\ell (\uparrow^\ell \circ^\ell u)$ (by (1^ℓ)) $\to u_1 \bullet^\ell u_2$ (by $(\uparrow^\ell)$) $= u$.

By induction on the length of any innermost reduction sequence from u to $\Sigma_H(u)$ using the above transformations, we get another (innermost) reduction sequence from u to $\Sigma_H(u)$, which is clearly restricted. (Observe that we this technique shows that we can in fact restrict derivations even more, so that u in rule $(\eta \bullet' \circ^\ell)$ is in fact of the form pop^ℓ_k for some $k \geq 0$.) $\Box$

This observation enables us to mostly replay the proof of confluence of the typed $\lambda\mathsf{ev}Q$-calculus (see Part IIIa).

Recall that we have defined $\stackrel{\beta_{||}}{\longrightarrow}$ as the parallel reduction of (β) and (β^ℓ) steps, for all $\ell \geq 1$. In the sequel, we won't mention again that all the terms that we use are well-typed, as this will always be the case.

Lemma 2.4 *Let Σ' denote the one-step restricted Σ_H reduction relation, ${\Sigma'}^*$ denote its reflexive transitive closure, and ${\Sigma'}^*\beta_{||}{\Sigma'}^*$ denote the composition of ${\Sigma'}^*$, $\stackrel{\beta_{||}}{\longrightarrow}$ and ${\Sigma'}^*$. Then:*

$$\begin{tikzcd} u \arrow[r, "\beta_{||}"] \arrow[d, "\Sigma'"'] & w \arrow[d, dashed, "\Sigma'^*"] \\ v \arrow[r, dashed, "\Sigma'^*\beta_{||}\Sigma'^*"'] & t \end{tikzcd}$$

Proof: Observe that Σ' is terminating (as a subrelation of Σ_H) and confluent: indeed, if u rewrites by Σ'^* to s and t, then s and t both reduce to $\Sigma_H(u) = \Sigma_H(s) = \Sigma_H(t)$, and we apply Lemma 2.3.

If u rewrites to v by some rule other than $(\eta \bullet' \circ^\ell)$, then this rule is left-linear, so we just have to consider the critical pairs between this rule and $\xrightarrow{\beta_{||}}$. Compared with Lemma 3.2, there are no additional critical pairs (there were five), so the claim is proved in this case.

If u rewrites to v by $(\eta \bullet' \circ^\ell)$, then $u = \mathcal{C}[u_1]$ for some context $\mathcal{C}$ and with $u_1 = (1^\ell \circ^\ell u_2) \bullet^\ell (\uparrow^\ell \circ^\ell u_2)$, and $v = \mathcal{C}[u_2]$, where u_2 is T-free: in particular, no $\xrightarrow{\beta_{||}}$-contraction occurs inside u_1. We then prove the claim by structural induction on $\mathcal{C}$.

If $\mathcal{C} = (\lambda x \cdot \mathcal{C}_1[])u'$, and $\mathcal{C}_1[u_1] \xrightarrow{\beta_{||}} w_1$, $u' \xrightarrow{\beta_{||}} w_2$, $w = w_1[w_2/x]$, then observe that w_1 must be of the form $\mathcal{C}'_1[u_1]$, since no $\xrightarrow{\beta_{||}}$-reduction can occur inside u_1; since u_1 is T-free, in particular x does not occur in u_1, so $(\eta \bullet' \circ^\ell)$ and $\xrightarrow{\beta_{||}}$ actually commute.

If $\mathcal{C} = (\lambda x \cdot u')\mathcal{C}_2[]$, and $u' \xrightarrow{\beta_{||}} w_1$, $\mathcal{C}_2[u_1] \xrightarrow{\beta_{||}} w_2$, $w = w_1[w_2/x]$, then we go from w to t by reducing all residuals of u_1 in w by $(\eta \bullet' \circ^\ell)$.

All other cases are trivial appeals to the induction hypothesis. □

Lemma 2.5 *In the typed case, we have:*

$$\begin{tikzcd} u \arrow[r, "\beta_{||}"] \arrow[d, "\Sigma'^*"'] & w \arrow[d, dashed, "\Sigma'^*"] \\ v \arrow[r, dashed, "\Sigma'^*\beta_{||}\Sigma'^*"'] & t \end{tikzcd}$$

Proof: As for Lemma 3.4 of Part IIIa, replacing Σ by Σ' (which is also confluent and terminating), and using Lemma 2.4 instead of Lemma 3.2 of Part IIIa. □

Lemma 2.6 *In the typed case, we have:*

$$\begin{tikzcd} u \arrow[r, "\Sigma'^*\beta_{||}\Sigma'^*"] \arrow[d, "\Sigma'^*\beta_{||}\Sigma'^*"'] & w \arrow[d, dashed, "\Sigma'^*\beta_{||}\Sigma'^*"] \\ v \arrow[r, dashed, "\Sigma'^*\beta_{||}\Sigma'^*"'] & t \end{tikzcd}$$

that is, $\Sigma'^*\beta_{||}\Sigma'^*$ is strongly confluent.

Proof: As for Lemma 3.5 of Part IIIa, replacing Σ by Σ', and using Lemma 2.5 instead of Lemma 3.4 of Part IIIa. □

At this point, we must however change our strategy to prove confluence. Indeed, we cannot conclude directly by noticing that $\Sigma'^*\beta_{||}\Sigma'^*$ has the same reflexive transitive closure as $\lambda\mathtt{ev}Q_H$, because it is just wrong: there are too few reductions in $\Sigma'^*\beta_{||}\Sigma'^*$. We have to use Hardin's interpretation technique, whose core is Lemma 2.8 below.

Lemma 2.7 *Let $B_{||}$ be the rewrite relation defined on Σ_H-normal forms by: $u \xrightarrow{B_{||}} v$ if and only if $u \xrightarrow{\beta_{||}} w$ for some w such that $\Sigma_H(w) = v$.*

Then $B_{||}$ is strongly confluent.

Proof: $B_{||}$ is a subrelation of $\Sigma'^*\beta_{||}\Sigma'^*$; indeed, if $u \xrightarrow{B_{||}} v$, then $u \xrightarrow{\Sigma'}{}^* u$ (trivially), $u \xrightarrow{\beta_{||}} w$ for some w such that $\Sigma_H(w) = v$ (by definition), hence $w \xrightarrow{\Sigma'}{}^* v$ (by Lemma 2.3).

Assume that $u \xrightarrow{B_{||}} v_1$ and $u \xrightarrow{B_{||}} v_2$. Then u rewrites to v_1 and to v_2 by $\Sigma'^*\beta_{||}\Sigma'^*$. By Lemma 2.6, there is a term w such that v_i rewrites to w by $\Sigma'^*\beta_{||}\Sigma'^*$, $i = 1, 2$. Without loss of generality, we may assume that w is Σ_H-normal, otherwise we normalize it by Σ'. It follows that there are terms w'_i and w''_i ($i = 1, 2$) such that $v_i \xrightarrow{\Sigma'}{}^* w'_i \xrightarrow{\beta_{||}} w''_i$ and $w = \Sigma_H(w''_i)$; since v_i is Σ_H-normal, $w'_i = v_i$, so in fact $v_i \xrightarrow{B_{||}} w$, $i = 1, 2$. □

Lemma 2.8 *If $u \xrightarrow{\beta_{||}} v$, then $\Sigma_H(u) \xrightarrow{B_{||}} \Sigma_H(v)$.*

Proof: Let $u \xrightarrow{\beta_{||}} v$. By Lemma 2.3, $u \xrightarrow{\Sigma'}{}^* \Sigma_H(u)$ and $v \xrightarrow{\Sigma'}{}^* \Sigma_H(v)$. By Lemma 2.5, it follows that $\Sigma_H(u)$ rewrites to $\Sigma_H(v)$ by $\Sigma'^*\beta_{||}\Sigma'^*$. Since $\Sigma_H(u)$ and $\Sigma_H(v)$ are both normal, $\Sigma_H(u) \xrightarrow{B_{||}} \Sigma_H(v)$. □

Lemma 2.9 *If $u \longrightarrow^* v$ in $\lambda\mathtt{ev}Q_H$, then $\Sigma_H(u) \xrightarrow{B_{||}}{}^* \Sigma_H(v)$.*

Proof: By induction on the length of the reduction from u to v. Each (β) or (β^ℓ) step is also a $\beta_{||}$ step, and we then apply Lemma 2.8; every other step is translated to a zero-length rewrite step. □

Theorem 2.10 *The typed $\lambda\mathtt{ev}Q_H$-calculus is confluent.*

Proof: Assume that $u \longrightarrow^* v_1$ and $u \longrightarrow^* v_2$. Then we have:

$$\begin{tikzcd}
 & u \arrow[dl, "*"'] \arrow[dr, "*"] \arrow[dd, dashed, "\Sigma'^*"] & \\
v_1 \arrow[dd, dashed, "\Sigma'^*"'] \arrow[r, phantom, "(1)"] & & v_2 \arrow[dd, dashed, "\Sigma'^*"] \arrow[l, phantom, "(2)"] \\
 & \Sigma_H(u) \arrow[dl, dashed, "B_{||}^*"'] \arrow[dr, dashed, "B_{||}^*"] \arrow[d, phantom, "(3)"] & \\
\Sigma_H(v_1) \arrow[dr, dashed, "B_{||}^*"'] & \, & \Sigma_H(v_2) \arrow[dl, dashed, "B_{||}^*"] \\
 & w &
\end{tikzcd}$$

for some w, where (1) and (2) follow from Lemma 2.9, and (3) follows from Lemma 2.7. □

Corollary 2.11 (Conservativity) *The typed $\lambda\mathtt{ev}Q_H$-calculus is a conservative extension of the typed $\lambda^{\approx}_{\mathrm{S4}H}$-calculus, i.e. for every typed λ_{S4}-terms u and v, u and v are interconvertible modulo the rules of $\lambda_{\mathrm{S4}H}$ if and only if $G(u)$ and $G(v)$ are interconvertible modulo the rules of $\lambda\mathtt{ev}Q_H$.*

Proof: By Theorem 4.7, Part IIIa using Theorem 2.10. □

2.2 Adding Fixpoints

Theorems like Theorem 2.10 or Corollary 2.11 look weak because they only apply to the typed systems. We have seen (Theorem 3.7, Part IIIa) that there was no hope of generalizing the confluence results to the untyped cases, at least for $\lambda\mathtt{ev}Q_H$, and the situation for the untyped $\lambda\mathtt{ev}Q$-calculus remains unclear, because Σ does not terminate in the untyped case. (The confluence of the untyped $\lambda\mathtt{ev}Q$-calculus is open.)

We shall see that all these typed systems are weakly normalizing. A natural question is then: how much are these confluence and conservativity results tied to the question of termination? Our aim in this subsection is to illustrate by a case study our opinion that the link is weak or even non-existent.

Consider indeed the following enrichment $\lambda_{\mathrm{S4}} + Y$ (resp. $\lambda_{\mathrm{S4}H} + Y$) of the λ_{S4} (resp. $\lambda_{\mathrm{S4}H}$) calculus. We add a new unary operator Y, with the typing rule:

$$\frac{\Gamma \vdash u : \tau \Rightarrow \tau}{\Gamma \vdash Yu : \tau}$$

and the reduction rule:

$$(Y) \quad Yu \rightarrow u(Yu)$$

Y is simply a fixpoint combinator. The typed $\lambda_{S4} + Y$ and $\lambda_{S4H} + Y$-calculi are then clearly non-weakly-normalizing, but they are still confluent:

Theorem 2.12 *The $\lambda_{S4} + Y$ and $\lambda_{S4H} + Y$-calculi (even in their untyped versions) are confluent.*

Proof: We first prove a theorem of finite developments, as in Part I, Section 5.1. We extend the $\lambda_{S4}{}'$-calculus (where (β) is replaced by (β') $(\lambda' x \cdot u)v \rightarrow u[v/x]$) by replacing (Y) by the following rule:

$$(Y') \; Y'u \rightarrow u(Yu)$$

where Y' is a new unary operator. Call $(\lambda_{S4}{}' + Y')$ the resulting calculus.

Observe that, in the untyped case, we may just define Y and Y' by:

$$\begin{array}{ll} Yu & = PPu \\ P & = \lambda x \cdot \lambda y \cdot y((xx)y) \\ Y'u & = (\lambda' y \cdot y((PP)y))u \end{array}$$

which mimics the definition of Turing's fixpoint combinator. Then (Y') is just a special case of rule (β'). Since $(\lambda_{S4}{}')$ terminates (finiteness of λ_{S4} developments), it follows that $(\lambda_{S4}{}' + Y')$ terminates, hence that $(\lambda_{S4} + Y)$ only has finite developments.

On the other hand, $(\lambda_{S4}{}' + Y')$ is clearly locally confluent, as (Y') introduces no new critical pair. So $(\lambda_{S4}{}' + Y')$ is confluent. Since the rewrite relations of $(\lambda_{S4}{}' + Y')$ and $(\lambda_{S4} + Y)$ have the same reflexive transitive closure, it follows that $(\lambda_{S4} + Y)$ is confluent.

Finally, the typed $(\lambda_{S4} + Y)$ calculus obeys the subject reduction property. From this and the fact that the untyped calculus is confluent, we conclude that the typed calculus is, too. $\Box$

To mirror this on the $\lambda \mathbf{ev} Q$-side, we add a combinator Y and infinitely many combinators Y^ℓ, $\ell \geq 1$, (for brevity, let Y^0 denote Y), with the following typing rules:

$$\frac{\Gamma \vdash u : \overline{\varsigma^\ell \stackrel{\Box}{\Rightarrow}} \tau \Rightarrow \tau}{\Gamma \vdash Y^\ell u : \overline{\varsigma^\ell \stackrel{\Box}{\Rightarrow}} \tau}$$

for every $\ell \geq 0$, and the following reduction rules:

$$(Y^\ell) \quad Y^\ell u \rightarrow u \star^\ell (Y^\ell u)$$

Extend the G-translation by:

$$\begin{array}{ll} G(Yu) & = Y^0(G(u)) \\ (Y^\ell u)^{\text{‘}} \rho & = Y^{\ell+1}(u^{\text{‘}} \rho) \end{array}$$

It is then clear that the resulting calculi $(\lambda \mathbf{ev} Q + Y)$ and $(\lambda \mathbf{ev} Q_H + Y)$ are respectively extensions of $(\lambda_{S4} + Y)$ and $(\lambda_{S4H} + Y)$.

Then:

Theorem 2.13 *The typed $(\lambda \mathbf{ev} Q + Y)$ and $(\lambda \mathbf{ev} Q_H + Y)$-calculi are confluent.*

Proof: The additional rules incur no new critical pair. Let $(\beta + Y)_{||}$ be the parallel reduction defined above (β), (β^ℓ) $(\ell \geq 1)$ and (Y^ℓ) $(\ell \geq 0)$. Since $(\beta + Y)_{||}$ is left-linear and has no critical pair, it is strongly confluent.

The typed $(\lambda \mathbf{ev} Q + Y)$-calculus is then proved confluent by replaying the proofs of Lemma 3.3 through Theorem 3.6 of Part IIIa, replacing $\beta_{||}$ by $(\beta + Y)_{||}$.

Similarly, the typed $\lambda \mathbf{ev} Q_H + Y$-calculus is then proved confluent by replaying the proofs of Lemma 2.7 through Theorem 2.10, replacing $\beta_{||}$ by $(B + Y)_{||}$, where the latter is the parallel reduction defined above B (namely, (β) union (β^ℓ) followed by Σ_H-normalization, $\ell \geq 1$) and (Y^ℓ), $\ell \geq 0$, on Σ_H-normal terms. The arguments are unchanged. $\Box$

Conservativity is preserved, too, although we cannot depend on $(\lambda_{S4} + Y)$ or $(\lambda_{S4H} + Y)$ being terminating:

Theorem 2.14 *The typed* $(\lambda\mathtt{ev}Q + Y)$*, resp.* $(\lambda\mathtt{ev}Q_H + Y)$*-calculus is a conservative extension of the typed* $(\lambda^{\approx}_{\mathrm{S4}} + Y)$*, resp.* $(\lambda^{\approx}_{\mathrm{S4}H} + Y)$*-calculus, i.e. for every typed* λ_{S4}*-terms* u *and* v*,* u *and* v *are interconvertible modulo the rules of* $(\lambda_{\mathrm{S4}} + Y)$*, resp.* $(\lambda_{\mathrm{S4}H} + Y)$ *if and only if* $G(u)$ *and* $G(v)$ *are interconvertible modulo the rules of* $(\lambda\mathtt{ev}Q + Y)$*, resp.* $(\lambda\mathtt{ev}Q_H + Y)$*.*

Proof: Since the typed $(\lambda_{\mathrm{S4}} + Y)$ calculus does not terminate, we use another. Let $(\lambda_{\mathrm{S4}} + Y_*)$ be $(\lambda_{\mathrm{S4}} + Y)$ with the (Y) rule replaced by:

$$(Y_n)\ Y_n u \to u(Y_{n-1} u)$$

where $n \geq 1$, and Y_i, $i \geq 0$, are new unary operators. Similarly, let $(\lambda\mathtt{ev}Q + Y_*)$ be $(\lambda\mathtt{ev}Q + Y)$ with the (Y^ℓ) rules, $\ell \geq 0$, replaced by:

$$(Y^\ell_n) \quad Y^\ell_n u \to u \star^\ell (Y^\ell_{n_1} u)$$

where $n \geq 1$. Extend G so that:

$$\begin{aligned} G(Y_n u) &= Y^0_n(G(u)) \\ (Y^\ell_n u)\text{`}\rho &= Y^{\ell+1}_n(u\text{`}\rho) \end{aligned}$$

Again, G makes $(\lambda\mathtt{ev}Q + Y_*)$ an extension of $(\lambda_{\mathrm{S4}} + Y_*)$.

Moreover, the typed $(\lambda_{\mathrm{S4}} + Y_*)$ calculus terminates. Indeed, Y_n is definable by:

$$\begin{aligned} Y_n u &= (\lambda x \cdot x(Y_{n-1} x))u \quad \text{if } n \geq 1 \\ Y_0 u &= Y u \end{aligned}$$

which is well-typed, provided that Y is a new free variable of type $(\tau \Rightarrow \tau) \Rightarrow \tau$. Termination then follows from the termination of the typed (λ_{S4})-calculus. Observe also that $(\lambda_{\mathrm{S4}} + Y_*)$ is confluent: it is locally confluent, because the critical pairs are exactly those of the λ_{S4}-calculus. Hence every typed $(\lambda_{\mathrm{S4}} + Y_*)$-term has a unique normal form.

We define the typed $(\lambda\mathtt{ev}Q + Y_*)$ calculus similarly, letting:

$$(Y^\ell_n) \quad Y^\ell_n u \to u \star^\ell (Y^\ell_{n-1} u)$$

for every $n \geq 1$, $\ell \geq 0$, and extending the G-translation by:

$$\begin{aligned} G(Y_n u) &= Y^0_n(G(u)) \\ (Y^\ell_n u)\text{`}\rho &= Y^{\ell+1}_n(u\text{`}\rho) \end{aligned}$$

By similar arguments as in Theorem 2.13, the typed $(\lambda\mathtt{ev}Q + Y_*)$ calculus is confluent as well, although it is not terminating.

Finally, observe that Lemma 4.5 of Part IIIa still holds in the extended calculi; namely, if u is $(\lambda_{\mathrm{S4}} + Y_*)$-normal, then $G(u)$ is $(\lambda\mathtt{ev}Q + Y_*)$-normal. Indeed, if u is $(\lambda_{\mathrm{S4}} + Y_*)$-normal, then it is λ_{S4}-normal and contains no occurrence of Y_n for any $n \geq 1$; so by Lemma 4.5 of Part IIIa, $G(u)$ is $\lambda\mathtt{ev}Q$-normal, and $G(u)$ contains no occurrence of any Y^ℓ_n, $\ell \geq 0$, $n \geq 1$. Therefore $G(u)$ is $(\lambda\mathtt{ev}Q + Y_*)$-normal.

Now prove the theorem. If u and v are interconvertible in $(\lambda_{\mathrm{S4}} + Y)$, it is clear that $G(u)$ and $G(v)$ are interconvertible in $(\lambda\mathtt{ev}Q + Y)$ as well.

Conversely, assume that $G(u)$ and $G(v)$ are interconvertible in $(\lambda\mathtt{ev}Q + Y)$. We lift u, v to $(\lambda_{\mathrm{S4}} + Y_*)$-terms (hence $G(u)$ and $G(v)$ to $(\lambda\mathtt{ev}Q + Y_*)$-terms) by putting high enough indices on each occurrence of Y, so that each reduction step between $G(u)$ and $G(v)$ is mapped to a corresponding reduction step in $(\lambda_{\mathrm{S4}} + Y_*)$. By abuse of language, consider now that u and v really are $(\lambda_{\mathrm{S4}} + Y_*)$-terms, so that $G(u)$ and $G(v)$ are interconvertible in $(\lambda\mathtt{ev}Q + Y_*)$. Let u' and v' be the respective unique normal forms of u and v in $(\lambda_{\mathrm{S4}} + Y_*)$. Then $G(u')$ and $G(v')$ are interconvertible in $(\lambda\mathtt{ev}Q + Y_*)$. Since $(\lambda\mathtt{ev}Q + Y_*)$ is confluent, there is a $(\lambda\mathtt{ev}Q + Y_*)$-term t such that $G(u')$ and $G(v')$ both reduce to t in $(\lambda\mathtt{ev}Q + Y_*)$. Now by the analogue of Lemma 4.5, Part IIIa, the fact that u' and v' are $(\lambda_{\mathrm{S4}} + Y_*)$-normal entails that $G(u')$ and $G(v')$ are both $(\lambda\mathtt{ev}Q + Y_*)$-normal, so $G(u') = G(v')$. The analogue of Lemma 4.3, Part IIIa clearly holds, namely G is injective from $(\lambda^{\approx}_{\mathrm{S4}} + Y_*)$ to $(\lambda\mathtt{ev}Q + Y_*)$; so $u' \approx v'$. In particular, u and v are interconvertible modulo the rules of $(\lambda_{\mathrm{S4}} + Y_*)$. Erase all indices on Y: then u and v are interconvertible modulo the rules of $(\lambda_{\mathrm{S4}} + Y)$.

The arguments are similar for $(\lambda_{\mathrm{S4}H} + Y)$ and $(\lambda\mathtt{ev}Q_H + Y)$. $\square$

3 Weak Termination

Our next goal is to show that the typed $\lambda\mathtt{ev}Q$ and $\lambda\mathtt{ev}Q_H$-calculi are weakly terminating, that is, that every typed term has a normal form. As we have already seen (Theorem 2.1, Part IIIa), strong termination (that every reduction strategy leads to a normal form) does not hold. But we shall be able to show in Section 3.3 that the typed $\lambda\mathtt{ev}Q_H$-calculus terminates weakly.

The same technique will allow us to find back the strong normalization result for the typed λ_{S4}-calculus in Section 3.4, because G injects the latter inside a subcalculus of $\lambda\mathtt{ev}Q$ that terminates — namely one where all stacks are empty. We won't be able to show that the typed $\lambda\mathtt{ev}Q$-calculus terminates weakly using these techniques, however, and we shall explain why this seems difficult in Section 3.5. The question of the weak termination of the typed $\lambda\mathtt{ev}Q$-calculus is therefore still open.

Let's embark on proving that the typed $\lambda\mathtt{ev}Q_H$-calculus terminates weakly. We shall prove a bit more than weak termination, actually: namely, that there is a non-empty class $\mathcal{R}$ of reduction strategies that all lead to normal forms, and such that every reduction in λ_{S4} (resp. λ_{S4H}) is interpreted as some reduction obeying some strategy in $\mathcal{R}$. This is a kind of relative strong termination result.

We achieve this by defining yet another interpretation of typed $\lambda\mathtt{ev}Q$-terms such that the (β) and (β^ℓ) rules, $\ell \geq 1$, are strictly decreasing in some well-founded ordering $\succ_{\lambda\mathtt{ev}Q}$, and such that the other rules are decreasing for $\succeq_{\lambda\mathtt{ev}Q}$. (The translation is similar to that used in Part IIIa; actually, this is the other way around: the translation of Part IIIa was an elaboration on this one.) The non-strict part of the ordering will be stable by context application ($u \succeq_{\lambda\mathtt{ev}Q} v$ implies $\mathcal{C}[u] \succeq_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$), but the strict part won't (we will only have $u \succ_{\lambda\mathtt{ev}Q} v$ implies $\mathcal{C}[u] \succeq_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$); this is mainly why we don't get strong termination. However, we shall get termination as soon as every (β) and (β^ℓ) redex u is reduced under a context $\mathcal{C}$ such that $u \succ_{\lambda\mathtt{ev}Q} v$ implies $\mathcal{C}[u] \succ_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$; we shall call such contexts *safe contexts*.

We shall then show that, provided we normalize terms by Σ (resp. Σ_H) before reducing any (β) or (β^ℓ)-redex, all contexts under which such a redex can be found are safe, hence that the typed $\lambda\mathtt{ev}Q$-calculus (resp. $\lambda\mathtt{ev}Q_H$) terminates provided that we apply Σ (resp. Σ_H) rules eagerly:

Definition 3.1 *We say that a term u is* relatively strongly normalizing, *or* RSN, *with respect to a subsystem R if and only if every rewrite sequence:*

$$u \xrightarrow{R}{}^* u_1 \longrightarrow u'_1 \xrightarrow{R}{}^* u_2 \longrightarrow u'_2 \xrightarrow{R}{}^* \ldots \xrightarrow{R}{}^* u_k \longrightarrow u'_k \xrightarrow{R}{}^* \ldots$$

*where $u_1, u_2, \ldots, u_k, \ldots$ are R-normal, is finite. We also say for short that u is R-*SN.

*We call such sequences R-*eager rewrites.

*A rewrite system S is R-*SN, *where $R \subseteq S$, if and only if every term is R-SN for rewrites in S.*

3.1 The $[\![\_]\!]'$ Translation

Our normalization proof needs a translation into another typed calculus to handle the case of (β) and (β^ℓ) steps, this time the typed λ-calculus with pairs. We define $\mathrm{fst}\langle s,t\rangle = s$, and $\mathrm{fst}\ s = \pi_1 s$ if s is not a pair, and similarly $\mathrm{snd}\langle s,t\rangle = t$, and $\mathrm{snd}\ s = \pi_2 s$ if s is not a pair.

$$\frac{}{\vdash x_\theta : \theta}$$

$$\frac{\vdash s : \theta^- \to \theta^+ \quad \vdash t : \theta^-}{\vdash st : \theta^+} \qquad \frac{\vdash s : \theta^+}{\vdash \lambda x_\theta \cdot s : \theta \to \theta^+}$$

$$\frac{\vdash s : \theta^+ \times \theta^-}{\vdash \pi_1 s : \theta^+} \qquad \frac{\vdash s : \theta^+ \times \theta^-}{\vdash \pi_2 s : \theta^-} \qquad \frac{\vdash s : \theta^+ \quad \vdash t : \theta^-}{\vdash \langle s,t\rangle : \theta^+ \times \theta^-}$$

Figure 3: Typing the λ-terms

Definition 3.2 *For every λ-term s of type θ^-, we say that s is* stack-like*, and we define its* components *recursively, by:*

- $\theta^- = \top$*; then s has no component;*
- *or θ^- is a pair type $\theta^+ \times \theta'^-$, and s is of the form $\langle t, s'\rangle$, where $t : \theta^+$ and $s' : \theta'^-$ is stack-like; then the components of s are t and the components of s'.*

We write $s = (t_j)_{1\leq j\leq m}$ to say that s is stack-like of components $t_1, \ldots, t_m$, and that:

$$s = \langle t_1, \langle t_2, \ldots, \langle t_m, s'\rangle \ldots\rangle\rangle$$

where $s' : \top$.

The translation rules are almost the same than in Figure 4, Part IIIa. We drop most symbols (ϵ, ι, $\oplus$, L, A), and the translations of applications $\star^\ell$ and abstractions λ^ℓ change; see Figure 4. For any stack type $\theta_1^+ \times \ldots \times \theta_n^+ \times \top$, and for every term u of this type, we write $\hat{u}$ for the stack-like term defined as u if $n = 0$, and as $\langle \pi_1 u, \widehat{\pi_2 u}\rangle$ otherwise.

Moreover, we translate types to θ-types in a slightly different way: instead of forgetting $\Rightarrow$ and converting $\overset{\Box}{\Rightarrow}$ to $\rightarrow$ (see Definition 2.9, Part IIIa), we now convert both to $\rightarrow$. The definition of θ-types needs to be changed a bit as well.

Definition 3.3 *The* positive θ-types θ^+ *and the* negative θ-types θ^- *are defined by the following grammar:*

$$\begin{array}{ll} \theta^+ & ::= o \mid \theta^- \rightarrow \theta^+ \mid \theta^+ \rightarrow \theta^+ \\ \theta^- & ::= \top \mid \theta^+ \times \theta^- \end{array}$$

where o is a distinguished base type.

Call a signature Σ *any expression of the form $\theta_1^-, \ldots, \theta_n^- \rightsquigarrow \theta$, where $n \geq 0$. Its* arity *is n. It is a* term signature *if θ is positive, and a* stack signature *otherwise.*

Given a term signature $\Sigma = \theta_1^-, \ldots, \theta_n^- \rightsquigarrow \theta^+$, let $\Sigma^\bullet$ be the positive type $\theta_1^- \rightarrow \ldots \rightarrow \theta_n^- \rightarrow \theta^+$. Also, write $\theta^- \rightsquigarrow \Sigma$ for $\theta^-, \theta_1^-, \ldots, \theta_n^- \rightsquigarrow \theta^+$.

Define the following translation from $\lambda\mathtt{ev}Q$ types to signatures:

$$\begin{array}{ll} [\![b]\!]' = (\rightsquigarrow o) \text{ for any base type } b & \\ [\![\tau_1 \Rightarrow \tau_2]\!]' = (\rightsquigarrow [\![\tau_1]\!]'^\bullet \rightarrow [\![\tau_2]\!]'^\bullet) & [\![\varsigma]\!]' = (\rightsquigarrow [\![\varsigma]\!]'_1) \\ [\![\top]\!]'_1 = \top & [\![\tau \times \varsigma]\!]'_1 = [\![\tau]\!]'^\bullet \times [\![\varsigma]\!]'_1 \\ [\![\varsigma \overset{\Box}{\Rightarrow} \tau]\!]' = [\![\varsigma]\!]'_1 \rightsquigarrow [\![\tau]\!]' & \end{array}$$

Lemma 3.1 *The $[\![\_]\!]'$ translation on types is well-defined. Moreover, if τ is a term type, then $[\![\tau]\!]$ is a term signature; if ς is a stack type, then $[\![\varsigma]\!]_1$ is a negative type; and if μ is a metastack type, then $[\![\mu]\!]$ is a stack signature.*

Proof: By structural induction on the argument Φ of the translation. If Φ is a base type b, then it is a term type and o is a positive type, so $[\![\Phi]\!]'$ is a term signature. If Φ is a function type $\tau_1 \Rightarrow \tau_2$, then it is a term type; moreover, tau_1 and τ_2 are term types, so by induction hypothesis $[\![\tau_1]\!]'$ and $[\![\tau_2]\!]'$ are term signatures, hence $[\![\Phi]\!]'$ is a term signature (this is where we need positive types on the left of arrows $\rightarrow$). If Φ is $\top$, then it is a stack type and a negative type; so $[\![\Phi]\!]'_1$ is a negative type and $[\![\Phi]\!]'$ is a stack signature. If Φ is $\tau \times \varsigma$, then it is a stack type, and by induction hypothesis $[\![\tau]\!]'$ is a term signature and $[\![\varsigma]\!]'_1$ is a negative type; so $[\![\tau]\!]'^\bullet$ is a positive type, and $[\![\Phi]\!]'_1$ is indeed a negative type; it also follows that $[\![\Phi]\!]'$ is a stack signature.

If Φ is $\varsigma \overset{\Box}{\Rightarrow} \tau$, then it is a term type; by induction hypothesis $[\![\varsigma]\!]'_1$ is a negative type, $[\![\tau]\!]'$ is a term signature, so $[\![\Phi]\!]'$ is a term signature. And if Φ is $\varsigma \overset{\Box}{\Rightarrow} \mu$, then it is a metastack type and by a similar argument $[\![\Phi]\!]'$ is a stack signature. □

The corresponding notion of arity for types Φ, defined as the arity of $[\![\Phi]\!]'$, is exactly the number of consecutive $\overset{\Box}{\Rightarrow}$'s in front of Φ. Again, we only reason on $\lambda\mathtt{ev}Q^+$-terms, not general $\lambda\mathtt{ev}Q$-terms; recall that basically $\lambda\mathtt{ev}Q^+$-terms are quotations of $\lambda\mathtt{ev}Q$-terms and live only at levels 1 or higher.

$$
\begin{array}{l}
[\![\mathtt{ev}^{\ell+1} uv]\!]' s_1 \ldots s_n = [\![u]\!]' s_1 \ldots s_\ell ([\![v]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\
[\![Q^\ell u]\!]' s_1 \ldots s_n = [\![u]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} s_{\ell+2} \ldots s_n \\
[\![u \circ^\ell v]\!]' s_1 \ldots s_n = [\![u]\!]' s_1 \ldots s_{\ell-1} ([\![v]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\
[\![\lambda^\ell u]\!]' s_1 \ldots s_\ell = \lambda x_{[\![\tau_1]\!]'\bullet} \cdot \lambda y_{\ell+1\,\theta^-_{\ell+1}} \cdot \ldots \cdot \lambda y_{n\,\theta^-_n} \cdot [\![u]\!]' s_1 \ldots s_{\ell-1} \langle x, s_\ell \rangle \hat{y}_{\ell+1} \ldots \hat{y}_n \\
\qquad \text{where } u : \overline{\varsigma^{\ell-1} \overset{\Box}{\Rightarrow}} \tau_1 \times \varsigma_\ell \overset{\Box}{\Rightarrow} \tau_2,\ [\![\tau_2]\!]' = \theta^-_{\ell+1}, \ldots, \theta^-_n \rightsquigarrow \theta^+ \\
\qquad \text{with } n \geq \ell \\
[\![u \star^\ell v]\!]' s_1 \ldots s_n = ([\![u]\!]' s_1 \ldots s_\ell)(\lambda z_{\ell+1\,\theta'^-_{\ell+1}} \cdot \ldots \cdot \lambda z_{m\,\theta'^-_m} \cdot [\![v]\!]' s_1 \ldots s_\ell \hat{z}_{\ell+1} \ldots \hat{z}_m) s_{\ell+1} \ldots s_n \\
\qquad \text{where } v : \tau',\ [\![\tau']\!] = \theta^-_1, \ldots, \theta^-_\ell, \theta'^-_{\ell+1}, \ldots, \theta'^-_m \rightsquigarrow \theta^+ \\
\qquad \text{with } m \geq \ell \\
[\![u \bullet^\ell v]\!]' s_1 \ldots s_\ell = \langle \lambda y_{\ell+1\,\theta'^-_{\ell+1}} \cdot \ldots \cdot \lambda y_{m\,\theta'^-_m} \cdot [\![u]\!]' s_1 \ldots s_\ell \hat{y}_{\ell+1} \ldots \hat{y}_m, [\![v]\!]' s_1 \ldots s_\ell \rangle \\
\qquad \text{where } u : \tau',\ [\![\tau']\!] = \theta^-_1, \ldots, \theta^-_\ell, \theta'^-_{\ell+1}, \ldots, \theta'^-_m \rightsquigarrow \theta^+ \\
\qquad \text{with } m \geq \ell \\
[\![id^\ell]\!]' s_1 \ldots s_\ell = s_\ell \\
[\![1^\ell]\!]' s_1 \ldots s_n = (\text{fst } s_\ell) s_{\ell+1} \ldots s_n \\
[\![\uparrow^\ell]\!]' s_1 \ldots s_\ell = \text{snd } s_\ell \\
[\![\Uparrow^\ell u]\!]' s_1 \ldots s_\ell = \langle \text{fst } s_\ell, [\![u]\!]' s_1 \ldots s_{\ell-1} (\text{snd } s_\ell) \rangle
\end{array}
$$

Figure 4: The $[\![\_]\!]'_\_$ interpretation ($\ell \geq 1$).

Definition 3.4 *We define the λ-term $[\![t]\!]' s_1 s_2 \ldots s_n$, for every ground typed $\lambda\mathtt{ev}Q^+$-term t of type Φ of arity n, and for every sequence of n λ-terms s_1 of type θ^-_1, ..., s_n of type θ^-_n, where $[\![\Phi]\!]' = \theta^-_1, \ldots, \theta^-_n \rightsquigarrow \theta$, as shown in Figure 4, where all λ-bound variables are assumed to be fresh.*

Check that the definition is well-formed, i.e. that all the type constraints are verified. A side-effect of this definition is that $[\![t]\!]' s_1 s_2 \ldots s_n$ has a positive θ-type if t is of sort T, and that it has a negative type if t is a stack.

$$
\begin{array}{ll}
(\beta) & (\lambda x \cdot s) t \to s[t/x] \\
(\pi_1) & \pi_1 \langle s, t \rangle \to s \\
(\pi_2) & \pi_2 \langle s, t \rangle \to t
\end{array}
$$

Figure 5: Reduction rules of the λ-calculus

It is well-known that the typed λ-calculus has the subject reduction property. We first check that reduction by any of the rules in Σ_H leaves the interpretation $[\![\_]\!]'$ of a term invariant, or that it decreases in the $\longrightarrow^*$ ordering on typed λ-terms (Lemma 3.5). We first need to establish a few useful facts.

Lemma 3.2 *For every ground typed $\lambda\mathtt{ev}Q^+$-term u, for every stack-like λ-terms $s_1 = (t_{1j})_{1 \leq j \leq m_1}$, ..., $s_n = (t_{1j})_{1 \leq j \leq m_n}$ of the right types:*

$$[\![u]\!]' s_1 \ldots s_n = ([\![u]\!]' (x_{1j})_{1 \leq j \leq m_1} \ldots (x_{nj})_{1 \leq j \leq m_n})[t_{ij}/x_{ij}, 1 \leq i \leq n, 1 \leq j \leq m_i]$$

for every $\sum_{i=1}^n m_i$ distinct variables x_{ij}, $1 \leq i \leq n$, $1 \leq j \leq m_i$. Moreover, if u has sort S, then $[\![u]\!]' s_1 \ldots s_n$ is stack-like.

Proof: Compared to Lemma 2.28, Part IIIa, it may now happen that some t_{ij}'s are not subterms of $[\![u]\!]' s_1 \ldots s_n$, because some cases in the definition of $[\![\_]\!]'$ decompose and forget part of the s_i, i.e. forget some t_{ij}'s. This happens in the translations of $Q^\ell u$ (where s_ℓ is entirely forgotten), of id^ℓ (where s_1, ..., $s_{\ell-1}$ are forgotten), as well as 1^ℓ and $\uparrow^\ell$.

The proof is by structural induction on u.

If $u = \mathtt{ev}^{\ell+1} u' v$, then:

$$[\![v]\!]' s_1 \dots s_\ell = ([\![u]\!]' (x_{1j})_{1\le j\le m_1} \dots (x_{\ell j})_{1\le j\le m_\ell})[t_{ij}/x_{ij}, 1 \le i \le \ell, 1 \le j \le m_i]$$

by induction hypothesis on v, and this quantity is stack-like. We can therefore apply the induction hypothesis on u', from which the claimed identity follows. Moreover, if u' (equivalently, u) is a stack, then the translation of u', hence that of u, is stack-like.

The case $u = u' \circ^\ell v$ is similar. All other cases are trivial. □

It follows a result analogous to Lemma 2.29, Part IIIa, but weaker, since it does not hold with $\longrightarrow^+$ instead of $\longrightarrow^*$:

Lemma 3.3 *If $t_{i_0 j_0} \longrightarrow^* t'_{i_0 j_0}$ for some $1 \le i_0 \le \ell$, $1 \le j \le m_{i_0}$, then:*

$$[\![v]\!]' s_1 \dots s_n \longrightarrow^* [\![v]\!]' s_1 \dots s_{i_0-1} s'_{i_0} s_{i_0+1} \dots s_n$$

where $s_i = (t_{ij})_{1\le j\le m_i}$ for all i, and $s'_{i_0} = (t'_{i_0 j})_{1\le j\le m_{i_0}}$, with $t'_{i_0 j} = t_{i_0 j}$ for $j \ne j_0$ by convention.

Lemma 3.4 *For every u, and stack-like λ-terms $s_1, \dots, s_n$, of the right types:*

$$([\![u]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_n)[s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n] \longrightarrow^* [\![u]\!]' s_1 \dots s_\ell s_{\ell+1} \dots s_n$$

where $y_{\ell+1}, \dots, y_n$ are not free in $s_1, \dots, s_\ell$.

Proof: We first claim that: (1) for every stack-like term s, $\hat{s} \longrightarrow^* s$. This is by induction on m, where s has type $\theta_1^+ \times \dots \times \theta_m^+ \times \top$. If $m = 0$, then $\hat{s} = s$, so this is clear. Otherwise, $\hat{s} = \langle \pi_1 s, \widehat{\pi_2 s} \rangle$, and since s is stack-like, s is of the form $\langle s_1, s_2 \rangle$ where s_2 is again stack-like; so $\hat{s} = \langle \pi_1 \langle s_1, s_2 \rangle, \pi_2 \widehat{\langle s_1, s_2 \rangle} \rangle \longrightarrow \langle s_1, \pi_2 \widehat{\langle s_1, s_2 \rangle} \rangle \longrightarrow \langle s_1, \widehat{s_2} \rangle \longrightarrow^* \langle s_1, s_2 \rangle$ (by induction hypothesis) $= s$.

We then claim that: (2) If $s = (t_i)_{1\le i\le k}$ is stack-like and $s \longrightarrow^* s'$, then s' is stack-like of the form $(t'_i)_{1\le i\le k}$, with $t_i \longrightarrow^* t'_i$ for each i, $1 \le i \le k$. This is a straightforward induction on k, using the fact that terms of the form $\langle s, t \rangle$ only rewrite to terms of the form $\langle s', t' \rangle$, where $s \longrightarrow^* s'$ and $t \longrightarrow t'$.

By Lemma 3.2:

$$\begin{aligned}&[\![u]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_n \\ &= ([\![u]\!]' (x_{1j})_{1\le j\le m_1} \dots (x_{nj})_{1\le j\le m_n})[t_{ij}/x_{ij}, 1 \le i \le n, 1 \le j \le m_i]\end{aligned}$$

where $s_i = (t_{ij})_{1\le j\le m_i}$ for every $1 \le i \le \ell$, and $\hat{y}_i = (t_{ij})_{1\le j\le m_i}$ for every $\ell+1 \le i \le n$. So:

$$\begin{aligned}&([\![u]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_n)[s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n] \\ &= ([\![u]\!]' (x_{1j})_{1\le j\le m_1} \dots (x_{nj})_{1\le j\le m_n})[t_{ij}/x_{ij}, 1 \le i \le n, 1 \le j \le m_i][s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n] \\ &= ([\![u]\!]' (x_{1j})_{1\le j\le m_1} \dots (x_{nj})_{1\le j\le m_n})[t_{ij}[s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n]/x_{ij}, 1 \le i \le n, 1 \le j \le m_i] \\ &= ([\![u]\!]' (x_{1j})_{1\le j\le m_1} \dots (x_{nj})_{1\le j\le m_n})[\left\{\begin{array}{l} t_{ij}/x_{ij}, 1 \le i \le \ell, 1 \le j \le m_i \\ t_{ij}[s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n]/x_{ij}, \ell+1 \le i \le n, 1 \le j \le m_i \end{array}\right. \quad] \\ &\qquad \text{because } y_{\ell+1}, \dots, y_n \text{ are not free in } s_1, \dots, s_\ell \\ &= [\![u]\!]' s_1 \dots s_\ell (\hat{y}_{\ell+1}[s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n]) \dots (\hat{y}_n[s_{\ell+1}/y_{\ell+1}, \dots, s_n/y_n]) \\ &\qquad \text{by Lemma 3.2} \\ &= [\![u]\!]' s_1 \dots s_\ell \hat{s}_{\ell+1} \dots \hat{s}_n \\ &\longrightarrow^* [\![u]\!]' s_1 \dots s_\ell s_{\ell+1} \dots s_n\end{aligned}$$

by claim (1) (saying that $\hat{s}_i \longrightarrow^* s_i$ for every $\ell+1 \le i \le n$), claim (2) (saying that if $\hat{s}_i = {t'_{ij}}_{1\le j\le m_i}$ and $s_i = {t''_{ij}}_{1\le j\le m_i}$, then $t'_{ij} \longrightarrow t''_{ij}$ for every i, j) and Lemma 3.3 (which allows us to conclude). □

Lemma 3.5 *For any rule $l \to r$ in Σ_H, where l and r are ground $\lambda\mathtt{ev}Q^+$-terms, for any stack-like λ-terms $s_1, \dots, s_n$ of the right types,*

$$[\![l]\!]' s_1 \dots s_n \longrightarrow^* [\![r]\!]' s_1 \dots s_n$$

Proof: Group (A): no rule to examine, since l will never be a $\lambda \mathtt{ev} Q^+$-term.

Group (H):

- Rule $(\eta\mathtt{ev}^\ell)$, $l = \mathtt{ev}^{\ell+1}(Q^\ell u)v$, $r = u \circ^\ell v$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_n = [\![Q^\ell u]\!]' s_1 \dots s_\ell([\![v]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\ &= [\![u]\!]' s_1 \dots s_{\ell-1}([\![v]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\ &= [\![u \circ^\ell v]\!]' s_1 \dots s_n\end{aligned}$$

- Rule $(\eta \bullet^\ell)$, $l = 1^\ell \bullet^\ell \uparrow^\ell$, $r = id^\ell$:

$$[\![l]\!]' s_1 \dots s_\ell = \langle [\![1^\ell]\!]' s_1 \dots s_\ell, [\![\uparrow^\ell]\!]' s_1 \dots s_\ell \rangle = \langle \text{fst } s_\ell, \text{snd } s_\ell \rangle = s_\ell = [\![id^\ell]\!]' s_1 \dots s_\ell = [\![r]\!]' s_1 \dots s_\ell$$

  because s_ℓ is stack-like. (We don't need any surjective pairing rule for this.)

- Rule $(\eta \bullet \circ^\ell)$, $l = (1^\ell \circ^\ell u) \bullet^\ell (\uparrow^\ell \circ^\ell u)$, $r = u$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_\ell = \langle [\![1^\ell \circ^\ell u]\!]' s_1 \dots s_\ell, [\![\uparrow^\ell \circ^\ell u]\!]' s_1 \dots s_\ell \rangle \\ &= \langle [\![1^\ell]\!]' s_1 \dots s_{\ell-1}([\![u]\!]' s_1 \dots s_\ell), [\![\uparrow^\ell]\!]' s_1 \dots s_{\ell-1}([\![u]\!]' s_1 \dots s_\ell) \rangle \\ &= \langle \text{fst}[\![u]\!]' s_1 \dots s_\ell, \text{snd}[\![u]\!]' s_1 \dots s_\ell \rangle \\ &= [\![u]\!]' s_1 \dots s_\ell \qquad \text{because } [\![u]\!]' s_1 \dots s_\ell \text{ is stack-like, by Lemma 3.2} \\ &= [\![id^\ell]\!]' s_1 \dots s_{\ell-1}([\![u]\!]' s_1 \dots s_\ell) = [\![r]\!]' s_1 \dots s_\ell\end{aligned}$$

Group (B):

- Rule $(\circ id^\ell)$, $l = u \circ^\ell id^\ell$, $r = u$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_\ell = [\![u]\!]' s_1 \dots s_{\ell-1}([\![id^\ell]\!]' s_1 \dots s_\ell) \\ &= [\![u]\!]' s_1 \dots s_{\ell-1}(s_\ell) = [\![r]\!]' s_1 \dots s_\ell\end{aligned}$$

- Rule $(id\circ^\ell)$, $l = id^\ell \circ^\ell u$, $r = u$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_\ell = [\![id^\ell]\!]' s_1 \dots s_{\ell-1}([\![u]\!]' s_1 \dots s_\ell) \\ &= [\![u]\!]' s_1 \dots s_\ell = [\![r]\!]' s_1 \dots s_\ell\end{aligned}$$

- Rule $(\circ^\ell)$, $l = (u \circ^\ell v) \circ^\ell w$, $r = u \circ^\ell (v \circ^\ell w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_n = [\![u \circ^\ell v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\ &= [\![u]\!]' s_1 \dots s_{\ell-1}([\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell)) s_{\ell+1} \dots s_n\end{aligned}$$

  while:

$$\begin{aligned}&[\![r]\!]' s_1 \dots s_n = [\![u]\!]' s_1 \dots s_{\ell-1}([\![v \circ^\ell w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\ &= [\![u]\!]' s_1 \dots s_{\ell-1}([\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell)) s_{\ell+1} \dots s_n\end{aligned}$$

- Rule $(\uparrow^\ell)$, $l = \uparrow^\ell \circ^\ell (u \bullet^\ell v)$, $r = v$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_\ell = [\![\uparrow^\ell]\!]' s_1 \dots s_{\ell-1}([\![u \bullet^\ell v]\!]' s_1 \dots s_\ell) \\ &= \text{snd}([\![u \bullet^\ell v]\!]' s_1 \dots s_\ell) \\ &= \text{snd}\langle \lambda \overline{y} \cdot [\![u]\!]' s_1 \dots s_\ell \overline{\hat{y}}, [\![v]\!]' s_1 \dots s_\ell \rangle \\ &= [\![v]\!]' s_1 \dots s_\ell = [\![r]\!]' s_1 \dots s_\ell\end{aligned}$$

  where $\overline{y}$ denotes a sequence of fresh variables of the right types and in the right number, and $\overline{\hat{y}}$ is the corresponding sequence of stack-like terms obtained from them.

- Rule (1^ℓ), $l = 1^\ell \circ^\ell (u \bullet^\ell v)$, $r = u$:

$$\begin{aligned}
&[\![l]\!]' s_1 \dots s_n = [\![1^\ell]\!]' s_1 \dots s_{\ell-1}([\![u \bullet^\ell v]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\
&= \mathrm{fst}([\![u \bullet^\ell v]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\
&= \mathrm{fst}\langle \lambda y_{\ell+1} \cdot \dots \cdot y_n \cdot [\![u]\!]' s_1 \dots s_\ell \overline{\hat{y}}, [\![v]\!]' s_1 \dots s_\ell \rangle s_{\ell+1} \dots s_n \\
&= (\lambda y_{\ell+1} \cdot \dots \cdot y_n \cdot [\![u]\!]' s_1 \dots s_\ell \overline{\hat{y}}) s_{\ell+1} \dots s_n \\
&\longrightarrow^* [\![u]\!]' s_1 \dots s_n = [\![r]\!]' s_1 \dots s_n
\end{aligned}$$

by rule (β) $n - \ell$ times and Lemma 3.4.

- Rule $(\bullet^\ell)$, $l = (u \bullet^\ell v) \circ^\ell w$, $r = (u \circ^\ell w) \bullet^\ell (v \circ^\ell w)$:

$$\begin{aligned}
&[\![l]\!]' s_1 \dots s_\ell = [\![u \bullet^\ell v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) \\
&= \langle \lambda \overline{y} \cdot [\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) \overline{\hat{y}}, [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) \rangle \\
&= \langle \lambda \overline{y} \cdot [\![u \circ^\ell w]\!]' s_1 \dots s_\ell \overline{\hat{y}}, [\![v \circ^\ell w]\!]' s_1 \dots s_\ell \rangle \\
&= [\![(u \circ^\ell w) \bullet^\ell (v \circ^\ell w)]\!]' s_1 \dots s_\ell = [\![r]\!]' s_1 \dots s_\ell
\end{aligned}$$

- Rule (λ^ℓ), $l = (\lambda^\ell u) \circ^\ell w$, $r = \lambda^\ell (u \circ^\ell \Uparrow^\ell w)$:

$$\begin{aligned}
&[\![l]\!]' s_1 \dots s_\ell = [\![\lambda^\ell u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) \\
&= \lambda x \cdot \lambda \overline{y} \cdot [\![u]\!]' s_1 \dots s_{\ell-1} \langle x, [\![w]\!]' s_1 \dots s_\ell \rangle \overline{\hat{y}}
\end{aligned}$$

while:

$$\begin{aligned}
&[\![r]\!]' s_1 \dots s_\ell = \lambda x \cdot \overline{y} \cdot [\![u \circ^\ell \Uparrow^\ell w]\!]' s_1 \dots s_{\ell-1} \langle x, s_\ell \rangle \overline{\hat{y}} \\
&= \lambda x \cdot \overline{y} \cdot [\![u]\!]' s_1 \dots s_{\ell-1}([\![\Uparrow^\ell w]\!]' s_1 \dots s_{\ell-1} \langle x, s_\ell \rangle) \overline{\hat{y}} \\
&= \lambda x \cdot \overline{y} \cdot [\![u]\!]' s_1 \dots s_{\ell-1} \langle \mathrm{fst}\langle x, s_\ell \rangle, [\![w]\!]' s_1 \dots s_{\ell-1}(\mathrm{snd}\langle x, s_\ell \rangle) \rangle \overline{\hat{y}} \\
&= \lambda x \cdot \overline{y} \cdot [\![u]\!]' s_1 \dots s_{\ell-1} \langle x, [\![w]\!]' s_1 \dots s_{\ell-1} s_\ell \rangle \overline{\hat{y}}
\end{aligned}$$

- Rule $(\star^\ell)$, $l = (u \star^\ell v) \circ^\ell w$, $r = (u \circ^\ell w) \star^\ell (v \circ^\ell w)$:

$$\begin{aligned}
&[\![l]\!]' s_1 \dots s_n = [\![u \star^\ell v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\
&= ([\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell))(\lambda \overline{z} \cdot [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) \overline{\hat{z}}) s_{\ell+1} \dots s_n \\
&= ([\![u \circ^\ell w]\!]' s_1 \dots s_{\ell-1} s_\ell)(\lambda \overline{z} \cdot [\![v \circ^\ell w]\!]' s_1 \dots s_{\ell-1} s_\ell \overline{\hat{z}}) s_{\ell+1} \dots s_n \\
&= [\![(u \circ^\ell w) \star^\ell (v \circ^\ell w)]\!]' s_1 \dots s_n = [\![r]\!]' s_1 \dots s_n
\end{aligned}$$

- Rule $(Q\circ^\ell)$, $l = Q^\ell u \circ^\ell v$, $r = Q^\ell u$:

$$\begin{aligned}
&[\![l]\!]' s_1 \dots s_n = [\![Q^\ell u]\!]' s_1 \dots s_{\ell-1}([\![v]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\
&= [\![u]\!]' s_1 \dots s_{\ell-1} s_{\ell+1} \dots s_n \\
&= [\![Q^\ell u]\!]' s_1 \dots s_n = [\![r]\!]' s_1 \dots s_n
\end{aligned}$$

- Rule $(1 \Uparrow^\ell)$: notice that $[\![\Uparrow^\ell u]\!]' s_1 \dots s_\ell = [\![1^\ell \bullet^\ell (u \circ^\ell \uparrow^\ell)]\!]' s_1 \dots s_\ell$, so the result follows from the case of rule (1^ℓ). Similarly for rules $(1 \Uparrow \circ^\ell)$, $(\uparrow\Uparrow^\ell)$, $(\uparrow\Uparrow \circ^\ell)$, $(\Uparrow\Uparrow^\ell)$, $(\Uparrow\Uparrow \circ^\ell)$, $(\Uparrow\bullet^\ell)$, and $(\Uparrow id^\ell)$ (the latter follows from rule $(\eta \bullet^\ell)$).

Group (C):

- Rule $(\mathtt{ev}\lambda^\ell)$, $l = \mathtt{ev}^\ell(\lambda^\ell u) w$, $r = \lambda^{\ell-1}(\mathtt{ev}^\ell(u \circ^{\ell-1} \uparrow^{\ell-1})(1^{\ell-1} \bullet^{\ell-1} (w \circ^{\ell-1} \uparrow^{\ell-1})))$ (since we only consider $\lambda\mathtt{ev}Q^+$-terms, $\ell \geq 2$):

$$\begin{aligned}
&[\![l]\!]' s_1 \dots s_{\ell-1} = [\![\lambda^\ell u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_{\ell-1}) \\
&= \lambda x \cdot \lambda y_{\ell+1} \cdot \dots \cdot \lambda y_n \cdot [\![u]\!]' s_1 \dots s_{\ell-1} \langle x, [\![w]\!]' s_1 \dots s_{\ell-1}) \rangle \hat{y}_{\ell+1} \dots \hat{y}_n
\end{aligned}$$

while:

$$
\begin{aligned}
&[\![r]\!]'s_1 \dots s_{\ell-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot [\![\mathtt{ev}^\ell (u \circ^{\ell-1} \uparrow^{\ell-1})(1^{\ell-1} \bullet^{\ell-1} (w \circ^{\ell-1} \uparrow^{\ell-1}))]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot \\
&\quad [\![u \circ^{\ell-1} \uparrow^{\ell-1}]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle ([\![1^{\ell-1} \bullet^{\ell-1} (w \circ^{\ell-1} \uparrow^{\ell-1})]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle) \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot \\
&\quad [\![u]\!]' s_1 \dots s_{\ell-2} ([\![\uparrow^{\ell-1}]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle) \\
&\qquad ([\![1^{\ell-1} \bullet^{\ell-1} (w \circ^{\ell-1} \uparrow^{\ell-1})]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle) \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot \\
&\quad [\![u]\!]' s_1 \dots s_{\ell-2} s_{\ell-1} ([\![1^{\ell-1} \bullet^{\ell-1} (w \circ^{\ell-1} \uparrow^{\ell-1})]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle) \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot \\
&\quad [\![u]\!]' s_1 \dots s_{\ell-2} s_{\ell-1} \langle [\![1^{\ell-1}]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle, [\![w \circ^{\ell-1} \uparrow^{\ell-1}]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle\rangle \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot [\![u]\!]' s_1 \dots s_{\ell-2} s_{\ell-1} \langle x, [\![w \circ^{\ell-1} \uparrow^{\ell-1}]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle\rangle \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot [\![u]\!]' s_1 \dots s_{\ell-2} s_{\ell-1} \langle x, [\![w]\!]' s_1 \dots s_{\ell-2} ([\![\uparrow^{\ell-1}]\!]' s_1 \dots s_{\ell-2} \langle x, s_{\ell-1}\rangle)\rangle \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_\ell \cdot \dots \cdot \lambda y_{n-1} \cdot [\![u]\!]' s_1 \dots s_{\ell-2} s_{\ell-1} \langle x, [\![w]\!]' s_1 \dots s_{\ell-2} s_{\ell-1}\rangle \hat{y}_\ell \dots \hat{y}_{n-1} \\
&= [\![l]\!]' s_1 \dots s_{\ell-1}
\end{aligned}
$$

- Rule $(\mathtt{ev}\star^\ell)$, $l = \mathtt{ev}^\ell (u \star^\ell v) w$, $r = (\mathtt{ev}^\ell u w) \star^{\ell-1} (\mathtt{ev}^\ell v w)$:

$$
\begin{aligned}
&[\![l]\!]' s_1 \dots s_n = [\![u \star^\ell v]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1}) s_\ell \dots s_n \\
&= ([\![u]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1})) \\
&\qquad (\lambda z_{\ell+1} \cdot \dots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1}) \hat{z}_{\ell+1} \dots \hat{z}_m) s_\ell \dots s_n
\end{aligned}
$$

while:

$$
\begin{aligned}
&[\![r]\!]' s_1 \dots s_n \\
&= ([\![\mathtt{ev}^\ell u w]\!]' s_1 \dots s_{\ell-1}) \\
&\qquad (\lambda z_\ell \cdot \dots \cdot \lambda z_{m-1} \cdot [\![\mathtt{ev}^\ell v w]\!]' s_1 \dots s_{\ell-1} \hat{z}_\ell \dots \hat{z}_{m-1}) s_\ell \dots s_n \\
&= ([\![u]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1})) \\
&\qquad (\lambda z_\ell \cdot \dots \cdot \lambda z_{m-1} \cdot [\![v]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1}) \hat{z}_\ell \dots \hat{z}_{m-1}) s_\ell \dots s_n \\
&= [\![l]\!]' s_1 \dots s_n
\end{aligned}
$$

- Rule $(\mathtt{ev}id^\ell)$, $l = \mathtt{ev}^\ell id^\ell w$, $r = w$ (with $\ell \geq 2$):

$$
\begin{aligned}
&[\![l]\!]' s_1 \dots s_{\ell-1} = [\![id^\ell]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1}) \\
&= [\![w]\!]' s_1 \dots s_{\ell-1} = [\![r]\!]' s_1 \dots s_{\ell-1}
\end{aligned}
$$

- Rule $(\mathtt{ev}\circ^\ell)$, $l = \mathtt{ev}^\ell (u \circ^\ell v) w$, $r = \mathtt{ev}^\ell u (\mathtt{ev}^\ell v w)$ (with $\ell \geq 2$):

$$
\begin{aligned}
&[\![l]\!]' s_1 \dots s_n = [\![u \circ^\ell v]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1}) s_\ell \dots s_n \\
&= [\![u]\!]' s_1 \dots s_{\ell-1} ([\![v]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1})) s_\ell \dots s_n \\
&= [\![u]\!]' s_1 \dots s_{\ell-1} ([\![\mathtt{ev}^\ell v w]\!]' s_1 \dots s_{\ell-1}) s_\ell \dots s_n \\
&= [\![\mathtt{ev}^\ell u (\mathtt{ev}^\ell v w)]\!]' s_1 \dots s_{\ell-1} s_\ell \dots s_n = [\![r]\!]' s_1 \dots s_n
\end{aligned}
$$

- Rule $(\mathtt{ev} \uparrow^\ell)$, $l = \mathtt{ev}^\ell \uparrow^\ell w$, $r = \uparrow^{\ell-1} \circ^{\ell-1} w$ (with $\ell \geq 2$):

$$
\begin{aligned}
&[\![l]\!]' s_1 \dots s_{\ell-1} = [\![\uparrow^\ell]\!]' s_1 \dots s_{\ell-1} ([\![w]\!]' s_1 \dots s_{\ell-1}) \\
&= \mathrm{snd}([\![w]\!]' s_1 \dots s_{\ell-1})
\end{aligned}
$$

while:

$$
\begin{aligned}
&[\![r]\!]' s_1 \dots s_{\ell-1} = [\![\uparrow^{\ell-1}]\!]' s_1 \dots s_{\ell-2} ([\![w]\!]' s_1 \dots s_{\ell-1}) \\
&= \mathrm{snd}([\![w]\!]' s_1 \dots s_{\ell-1})
\end{aligned}
$$

- Rule ($\mathtt{ev}1^\ell$), $l = \mathtt{ev}^\ell 1^\ell w$, $r = 1^{\ell-1} \circ^{\ell-1} w$ (with $\ell \geq 2$):

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_n = [\![1^\ell]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_{\ell-1}) s_\ell \dots s_n\\ &= (\mathrm{fst}([\![w]\!]' s_1 \dots s_{\ell-1})) s_\ell \dots s_n\end{aligned}$$

while:

$$\begin{aligned}&[\![r]\!]' s_1 \dots s_n = [\![1^{\ell-1}]\!]' s_1 \dots s_{\ell-2}([\![w]\!]' s_1 \dots s_{\ell-1}) s_\ell \dots s_n\\ &= (\mathrm{fst}([\![w]\!]' s_1 \dots s_{\ell-1})) s_\ell \dots s_n\end{aligned}$$

- Rule ($\mathtt{ev}\ \bullet^\ell$), $l = \mathtt{ev}^\ell(u \bullet^\ell v)w$, $r = (\mathtt{ev}^\ell uw) \bullet^{\ell-1} (\mathtt{ev}^\ell vw)$ (with $\ell \geq 2$):

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_{\ell-1} = [\![u \bullet^\ell v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_{\ell-1})\\ &= \langle \lambda \overline{y} \cdot [\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_{\ell-1}) \hat{\overline{y}}, [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_{\ell-1}) \rangle\\ &= \langle \lambda \overline{y} \cdot [\![\mathtt{ev}^\ell uw]\!]' s_1 \dots s_{\ell-1} \hat{\overline{y}}, [\![\mathtt{ev}^\ell vw]\!]' s_1 \dots s_{\ell-1} \rangle\\ &= [\![(\mathtt{ev}^\ell uw) \bullet^{\ell-1} (\mathtt{ev}^\ell vw)]\!]' s_1 \dots s_{\ell-1} = [\![r]\!]' s_1 \dots s_{\ell-1}\end{aligned}$$

- Rule ($\mathtt{ev}\ \Uparrow^\ell$): since $\Uparrow^\ell u$ is interpreted exactly as $1^\ell \bullet^\ell (u \circ^\ell \uparrow^\ell)$, the result follows by the same arguments as for rules ($\mathtt{ev}\ \bullet^\ell$), ($\mathtt{ev}1^\ell$), ($1^{\ell-1}$), ($\mathtt{ev}\circ^\ell$), ($\mathtt{ev}\uparrow^\ell$), ($\uparrow^{\ell-1}$). Similarly for rules ($1\mathtt{ev}\Uparrow^\ell$) (using the same arguments as for rules ($\mathtt{ev}\ \bullet^\ell$), ($1^{\ell-1}$), ($\mathtt{ev}1^\ell$)), ($\uparrow\mathtt{ev}\ \Uparrow^\ell$) (using the same arguments as for rules ($\mathtt{ev}\ \bullet^\ell$), ($\uparrow^{\ell-1}$), ($\mathtt{ev}\circ^\ell$), ($\mathtt{ev}\uparrow^\ell$)) and ($\mathtt{ev}\ \Uparrow\Uparrow^\ell$) (using the fact that $\mathtt{ev}^\ell(\Uparrow^\ell u)(\mathtt{ev}^\ell(\Uparrow^\ell v)w)$ is interpreted just the same as $\mathtt{ev}^\ell(\Uparrow^\ell u \circ^\ell \Uparrow^\ell v)w$ by the arguments of rule ($\mathtt{ev}\circ^\ell$), and then reusing the arguments for ($\Uparrow\Uparrow^\ell$)).

- Rule ($\mathtt{ev}Q^\ell$), $l = \mathtt{ev}^\ell(Q^\ell u)w$, $r = u$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_n = [\![Q^\ell u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_{\ell-1}) s_\ell \dots s_n\\ &= [\![u]\!]' s_1 \dots s_{\ell-1} s_\ell \dots s_n = [\![r]\!]' s_1 \dots s_n\end{aligned}$$

Group (D) (we let $2 \leq \ell < \mathcal{L}$ in all these rules):

- Rule ($\lambda^\mathcal{L}\circ^\ell$), $l = (\lambda^\mathcal{L} u) \circ^\ell w$, $r = \lambda^\mathcal{L}(u \circ^\ell w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_\mathcal{L} = [\![\lambda^\mathcal{L} u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_\mathcal{L}\\ &= \lambda x \cdot \lambda y_{\mathcal{L}+1} \cdot \dots \cdot \lambda y_n \cdot [\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_{\mathcal{L}-1} \langle x, s_\mathcal{L} \rangle \hat{y}_{\mathcal{L}+1} \dots \hat{y}_n\end{aligned}$$

while:

$$\begin{aligned}&[\![r]\!]' s_1 \dots s_\mathcal{L} = \lambda x \cdot \lambda y_{\mathcal{L}+1} \cdot \dots \cdot \lambda y_n \cdot [\![u \circ^\ell w]\!]' s_1 \dots s_{\mathcal{L}-1} \langle x, s_\mathcal{L} \rangle \hat{y}_{\mathcal{L}+1} \dots \hat{y}_n\\ &= \lambda x \cdot \lambda y_{\mathcal{L}+1} \cdot \dots \cdot \lambda y_n \cdot [\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_{\mathcal{L}-1} \langle x, s_\mathcal{L} \rangle \hat{y}_{\mathcal{L}+1} \dots \hat{y}_n\end{aligned}$$

- Rule ($\star^\mathcal{L}\circ^\ell$), $l = (u \star^\mathcal{L} v) \circ^\ell w$, $r = (u \circ^\ell w) \star^\mathcal{L} (v \circ^\ell w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_n = [\![u \star^\mathcal{L} v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n\\ &= ([\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_\mathcal{L})\\ &\qquad (\lambda z_{\mathcal{L}+1} \cdot \dots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_\mathcal{L} \hat{z}_{\mathcal{L}+1} \dots \hat{z}_m) s_{\mathcal{L}+1} \dots s_n\end{aligned}$$

while:

$$\begin{aligned}&[\![r]\!]' s_1 \dots s_n = ([\![u \circ^\ell w]\!]' s_1 \dots s_\mathcal{L})(\lambda z_{\mathcal{L}+1} \cdot \dots \cdot \lambda z_m \cdot [\![v \circ^\ell w]\!]' s_1 \dots s_\mathcal{L} \hat{z}_{\mathcal{L}+1} \dots \hat{z}_m) s_{\mathcal{L}+1} \dots s_n\\ &= ([\![u]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_\mathcal{L})\\ &\qquad (\lambda z_{\mathcal{L}+1} \cdot \dots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_\mathcal{L} \hat{z}_{\mathcal{L}+1} \dots \hat{z}_m) s_{\mathcal{L}+1} \dots s_n\end{aligned}$$

- Rule ($id^\mathcal{L}\circ^\ell$), $l = id^\mathcal{L} \circ^\ell w$, $r = id^\mathcal{L}$:

$$\begin{aligned}&[\![l]\!]' s_1 \dots s_\mathcal{L} = [\![id^\mathcal{L}]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_\mathcal{L}\\ &= s_\mathcal{L} = [\![id^\mathcal{L}]\!]' s_1 \dots s_\mathcal{L} = [\![r]\!]' s_1 \dots s_\mathcal{L}\end{aligned}$$

- Rule $(\circ^{\mathcal{L}}\circ^{\ell})$, $l = (u \circ^{\mathcal{L}} v) \circ^{\ell} w$, $r = (u \circ^{\ell} w) \circ^{\mathcal{L}} (v \circ^{\ell} w)$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_n = [\![u \circ^{\mathcal{L}} v]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_n \\
&= [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}-1}([\![v]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}})s_{\mathcal{L}+1} \dots s_n \\
&= [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}-1}([\![v \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}})s_{\mathcal{L}+1} \dots s_n \\
&= [\![u \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}-1}([\![v \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}})s_{\mathcal{L}+1} \dots s_n \\
&= [\![(u \circ^{\ell} w) \circ^{\mathcal{L}} (v \circ^{\ell} w)]\!]'s_1 \dots s_n = [\![r]\!]'s_1 \dots s_n
\end{aligned}$$

- Rule $(\uparrow^{\mathcal{L}} \circ^{\ell})$, $l = \uparrow^{\mathcal{L}} \circ^{\ell} w$, $r = \uparrow^{\mathcal{L}}$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_{\mathcal{L}} = [\![\uparrow^{\mathcal{L}}]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}} \\
&= \mathrm{snd}\ s_{\mathcal{L}} = [\![\uparrow^{\mathcal{L}}]\!]'s_1 \dots s_{\mathcal{L}} = [\![r]\!]'s_1 \dots s_{\mathcal{L}}
\end{aligned}$$

- Rule $(1^{\mathcal{L}}\circ^{\ell})$, $l = 1^{\mathcal{L}} \circ^{\ell} w$, $r = 1^{\mathcal{L}}$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_n = [\![1^{\mathcal{L}}]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_n \\
&= (\mathrm{fst}\ s_{\mathcal{L}})s_{\mathcal{L}+1} \dots s_n = [\![1^{\mathcal{L}}]\!]'s_1 \dots s_n = [\![r]\!]'s_1 \dots s_n
\end{aligned}$$

- Rule $(\bullet^{\mathcal{L}} \circ^{\ell})$, $l = (u \bullet^{\mathcal{L}} v) \circ^{\ell} w$, $r = (u \circ^{\ell} w) \bullet^{\mathcal{L}} (v \circ^{\ell} w)$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_{\mathcal{L}} = [\![u \bullet^{\mathcal{L}} v]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_n \\
&= \langle \lambda\overline{y} \cdot [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}}\hat{\overline{y}}, [\![v]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}}\rangle \\
&= \langle \lambda\overline{y} \cdot [\![u \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}}\hat{\overline{y}}, [\![v \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}}\rangle \\
&= [\![(u \circ^{\ell} w) \bullet^{\mathcal{L}} (v \circ^{\ell} w)]\!]'s_1 \dots s_{\mathcal{L}} = [\![r]\!]'s_1 \dots s_{\mathcal{L}}
\end{aligned}$$

- Rule $(\Uparrow^{\mathcal{L}} \circ^{\ell})$ follows from the cases of the rules $(\bullet^{\mathcal{L}} \circ^{\ell})$, $(1^{\mathcal{L}}\circ^{\ell})$, $(\circ^{\mathcal{L}}\circ^{\ell})$, $(\uparrow^{\mathcal{L}} \circ^{\ell})$ and the fact that $\Uparrow^{\mathcal{L}} u$ is interpreted just as $1^{\mathcal{L}} \bullet^{\mathcal{L}} (u\circ^{\mathcal{L}} \uparrow^{\mathcal{L}})$.
- Rule $(Q^{\mathcal{L}}\circ^{\ell})$, $l = (Q^{\mathcal{L}}u) \circ^{\ell} w$, $r = Q^{\mathcal{L}}(u \circ^{\ell} w)$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_n = [\![Q^{\mathcal{L}}u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_n \\
&= [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}-1}s_{\mathcal{L}+1} \dots s_n \\
&= [\![u \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}-1}s_{\mathcal{L}+1} \dots s_n \\
&= [\![Q^{\mathcal{L}}(u \circ^{\ell} w)]\!]'s_1 \dots s_n = [\![r]\!]'s_1 \dots s_n
\end{aligned}$$

- Rule $(\mathtt{ev}^{\mathcal{L}}\circ^{\ell})$, $l = (\mathtt{ev}^{\mathcal{L}}uv) \circ^{\ell} w$, $r = \mathtt{ev}^{\mathcal{L}}(u \circ^{\ell} w)(v \circ^{\ell} w)$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_n = [\![\mathtt{ev}^{\mathcal{L}}uv]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_n \\
&= [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}-1}([\![v]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}-1})s_{\mathcal{L}} \dots s_n \\
&= [\![u \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}-1}([\![v]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_\ell)s_{\ell+1} \dots s_{\mathcal{L}-1})s_{\mathcal{L}} \dots s_n \\
&= [\![u \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}-1}([\![v \circ^{\ell} w]\!]'s_1 \dots s_{\mathcal{L}-1})s_{\mathcal{L}} \dots s_n \\
&= [\![\mathtt{ev}^{\mathcal{L}}(u \circ^{\ell} w)(v \circ^{\ell} w)]\!]'s_1 \dots s_n = [\![r]\!]'s_1 \dots s_n
\end{aligned}$$

Group (E) (we let $2 \leq \ell < \mathcal{L}$ in all these rules):

- Rule $(\mathtt{ev}^{\ell}\lambda^{\mathcal{L}})$, $l = \mathtt{ev}^{\ell}(\lambda^{\mathcal{L}}u)w$, $r = \lambda^{\mathcal{L}-1}(\mathtt{ev}^{\ell}uw)$:

$$\begin{aligned}
&[\![l]\!]'s_1 \dots s_{\mathcal{L}-1} = [\![\lambda^{\mathcal{L}}u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_{\ell-1})s_\ell \dots s_{\mathcal{L}-1} \\
&= \lambda x \cdot \lambda y_{\mathcal{L}+1} \cdot \dots \cdot \lambda y_n \cdot [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_{\ell-1})s_\ell \dots s_{\mathcal{L}-2}\langle x, s_{\mathcal{L}-1}\rangle\hat{y}_{\mathcal{L}+1} \dots \hat{y}_n
\end{aligned}$$

while:

$$\begin{aligned}
&[\![r]\!]'s_1 \dots s_{\mathcal{L}-1} = \lambda x \cdot \lambda y_{\mathcal{L}} \cdot \dots \cdot \lambda y_{n-1} \cdot [\![\mathtt{ev}^{\ell}uw]\!]'s_1 \dots s_{\mathcal{L}-2}\langle x, s_{\mathcal{L}-1}\rangle\hat{y}_{\mathcal{L}} \dots \hat{y}_{n-1} \\
&= \lambda x \cdot \lambda y_{\mathcal{L}} \cdot \dots \cdot \lambda y_{n-1} \cdot [\![u]\!]'s_1 \dots s_{\ell-1}([\![w]\!]'s_1 \dots s_{\ell-1})s_\ell \dots s_{\mathcal{L}-2}\langle x, s_{\mathcal{L}-1}\rangle\hat{y}_{\mathcal{L}} \dots \hat{y}_{n-1}
\end{aligned}$$

- Rule $(\mathtt{ev}^\ell \star^{\mathcal{L}})$, $l = \mathtt{ev}^\ell (u \star^{\mathcal{L}} v) w$, $r = (\mathtt{ev}^\ell u w) \star^{\mathcal{L}-1} (\mathtt{ev}^\ell v w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_{n-1} = [\![u \star^{\mathcal{L}} v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{n-1}\\ &= ([\![u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1})\\ &\qquad (\lambda z_{\mathcal{L}+1} \cdot \ldots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1} \hat{z}_{\mathcal{L}+1} \ldots \hat{z}_m) s_{\mathcal{L}} \ldots s_{n-1}\end{aligned}$$

while:

$$\begin{aligned}&[\![r]\!]' s_1 \ldots s_{n-1}\\ &= ([\![\mathtt{ev}^\ell u w]\!]' s_1 \ldots s_{\mathcal{L}-1}) (\lambda z_{\mathcal{L}} \cdot \ldots \cdot \lambda z_{m-1} \cdot [\![\mathtt{ev}^\ell v w]\!]' s_1 \ldots s_{\mathcal{L}-1}) \hat{z}_{\mathcal{L}} \ldots \hat{z}_{m-1}) s_{\mathcal{L}} \ldots s_{n-1}\\ &= ([\![u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1})\\ &\qquad (\lambda z_{\mathcal{L}} \cdot \ldots \cdot \lambda z_{m-1} \cdot [\![v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1} \hat{z}_{\mathcal{L}} \ldots \hat{z}_{m-1}) s_{\mathcal{L}} \ldots s_{n-1}\end{aligned}$$

- Rule $(\mathtt{ev}^\ell id^{\mathcal{L}})$, $l = \mathtt{ev}^\ell id^{\mathcal{L}} w$, $r = id^{\mathcal{L}-1}$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_{\mathcal{L}-1} = [\![id^{\mathcal{L}}]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1}\\ &= s_{\mathcal{L}-1} = [\![id^{\mathcal{L}-1}]\!]' s_1 \ldots s_{\mathcal{L}-1} = [\![r]\!]' s_1 \ldots s_{\mathcal{L}-1}\end{aligned}$$

- Rule $(\mathtt{ev}^\ell \circ^{\mathcal{L}})$, $l = \mathtt{ev}^\ell (u \circ^{\mathcal{L}} v) w$, $r = (\mathtt{ev}^\ell u w) \circ^{\mathcal{L}-1} (\mathtt{ev}^\ell v w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_n = [\![u \circ^{\mathcal{L}} v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_n\\ &= [\![u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-2} ([\![v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1}) s_{\mathcal{L}} \ldots s_n\\ &= [\![u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-2} ([\![\mathtt{ev}^\ell v w]\!]' s_1 \ldots s_{\mathcal{L}-1}) s_{\mathcal{L}} \ldots s_n\\ &= [\![\mathtt{ev}^\ell u w]\!]' s_1 \ldots s_{\mathcal{L}-2} ([\![\mathtt{ev}^\ell v w]\!]' s_1 \ldots s_{\mathcal{L}-1}) s_{\mathcal{L}} \ldots s_n\\ &= [\![(\mathtt{ev}^\ell u w) \circ^{\mathcal{L}-1} (\mathtt{ev}^\ell v w)]\!]' s_1 \ldots s_n = [\![r]\!]' s_1 \ldots s_n\end{aligned}$$

- Rule $(\mathtt{ev}^\ell \uparrow^{\mathcal{L}})$, $l = \mathtt{ev}^\ell \uparrow^{\mathcal{L}} w$, $r = \uparrow^{\mathcal{L}-1}$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_{\mathcal{L}-1} = [\![\uparrow^{\mathcal{L}}]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1}\\ &= \mathrm{snd}\ s_{\mathcal{L}-1} = [\![\uparrow^{\mathcal{L}-1}]\!]' s_1 \ldots s_{\mathcal{L}-1} = [\![r]\!]' s_1 \ldots s_{\mathcal{L}-1}\end{aligned}$$

- Rule $(\mathtt{ev}^\ell 1^{\mathcal{L}})$, $l = \mathtt{ev}^\ell 1^{\mathcal{L}} w$, $r = 1^{\mathcal{L}-1}$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_n = [\![1^{\mathcal{L}}]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_n\\ &= (\mathrm{fst}\ s_{\mathcal{L}-1}) s_{\mathcal{L}} \ldots s_n\\ &= [\![1^{\mathcal{L}-1}]\!]' s_1 \ldots s_n = [\![r]\!]' s_1 \ldots s_n\end{aligned}$$

- Rule $(\mathtt{ev}^\ell \bullet^{\mathcal{L}})$, $l = \mathtt{ev}^\ell (u \bullet^{\mathcal{L}} v) w$, $r = (\mathtt{ev}^\ell u w) \bullet^{\mathcal{L}-1} (\mathtt{ev}^\ell v w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_{\mathcal{L}-1} = [\![u \bullet^{\mathcal{L}} v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1}\\ &= \langle \lambda \overline{y} \cdot [\![u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1} \hat{\overline{y}}, [\![v]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-1} \rangle\\ &= \langle \lambda \overline{y} \cdot [\![\mathtt{ev}^\ell u w]\!]' s_1 \ldots s_{\mathcal{L}-1} \hat{\overline{y}}, [\![\mathtt{ev}^\ell v w]\!]' s_1 \ldots s_{\mathcal{L}-1} \rangle\\ &= [\![(\mathtt{ev}^\ell u w) \bullet^{\mathcal{L}-1} (\mathtt{ev}^\ell v w)]\!]' s_1 \ldots s_{\mathcal{L}-1} = [\![r]\!]' s_1 \ldots s_{\mathcal{L}-1}\end{aligned}$$

- Rule $(\mathtt{ev}^\ell \Uparrow^{\mathcal{L}})$: follows from $(\mathtt{ev}^\ell \bullet^{\mathcal{L}})$, $(\mathtt{ev}^\ell 1^{\mathcal{L}})$, $(\mathtt{ev}^\ell \circ^{\mathcal{L}})$, $(\mathtt{ev}^\ell \uparrow^{\mathcal{L}})$ and the fact that $\Uparrow^{\mathcal{L}} u$ is interpreted just as $1^{\mathcal{L}} \bullet^{\mathcal{L}} (u \circ^{\mathcal{L}} \uparrow^{\mathcal{L}})$.
- Rule $(\mathtt{ev}^\ell Q^{\mathcal{L}})$, $l = \mathtt{ev}^\ell (Q^{\mathcal{L}} u) w$, $r = Q^{\mathcal{L}-1} (\mathtt{ev}^\ell u w)$:

$$\begin{aligned}&[\![l]\!]' s_1 \ldots s_n = [\![Q^{\mathcal{L}} u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_n\\ &= [\![u]\!]' s_1 \ldots s_{\ell-1} ([\![w]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_{\mathcal{L}-2} s_{\mathcal{L}} \ldots s_n\\ &= [\![\mathtt{ev}^\ell u w]\!]' s_1 \ldots s_{\mathcal{L}-2} s_{\mathcal{L}} \ldots s_n\\ &= [\![Q^{\mathcal{L}-1} (\mathtt{ev}^\ell u w)]\!]' s_1 \ldots s_n = [\![r]\!]' s_1 \ldots s_n\end{aligned}$$

- Rule $(\mathtt{ev}^\ell\mathtt{ev}^\mathcal{L})$, $l = \mathtt{ev}^\ell(\mathtt{ev}^\mathcal{L} uv)w$, $r = \mathtt{ev}^{\mathcal{L}-1}(\mathtt{ev}^\ell uw)(\mathtt{ev}^\ell vw)$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_n = [\![\mathtt{ev}^\mathcal{L} uv]\!]'s_1\dots s_{\ell-1}([\![w]\!]'s_1\dots s_{\ell-1})s_\ell\dots s_n\\
&= [\![u]\!]'s_1\dots s_{\ell-1}([\![w]\!]'s_1\dots s_{\ell-1})s_\ell\dots s_{\mathcal{L}-2}([\![v]\!]'s_1\dots s_{\ell-1}([\![w]\!]'s_1\dots s_{\ell-1})s_\ell\dots s_{\mathcal{L}-2})s_{\mathcal{L}-1}\dots s_n\\
&= [\![u]\!]'s_1\dots s_{\ell-1}([\![w]\!]'s_1\dots s_{\ell-1})s_\ell\dots s_{\mathcal{L}-2}([\![\mathtt{ev}^\ell vw]\!]'s_1\dots s_{\mathcal{L}-2})s_{\mathcal{L}-1}\dots s_n\\
&= [\![\mathtt{ev}^\ell uw]\!]'s_1\dots s_{\mathcal{L}-2}([\![\mathtt{ev}^\ell vw]\!]'s_1\dots s_{\mathcal{L}-2})s_{\mathcal{L}-1}\dots s_n\\
&= [\![\mathtt{ev}^{\mathcal{L}-1}(\mathtt{ev}^\ell uw)(\mathtt{ev}^\ell vw)]\!]'s_1\dots s_n = [\![r]\!]'s_1\dots s_n
\end{aligned}$$

Group (F) (with $2 \le \ell < \mathcal{L}$ again):

- Rule $(Q^\ell\lambda^\mathcal{L})$, $l = Q^\ell(\lambda^{\mathcal{L}-1}u)$, $r = \lambda^\mathcal{L}(Q^\ell u)$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_\mathcal{L} = [\![\lambda^{\mathcal{L}-1}u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_{\ calL}\\
&= \lambda x\cdot\lambda y_\mathcal{L}\cdot\ldots\cdot\lambda y_n\cdot[\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_{\mathcal{L}-1}\langle x, s_\mathcal{L}\rangle\hat{y}_\mathcal{L}\dots\hat{y}_n
\end{aligned}$$

while:

$$\begin{aligned}
&[\![r]\!]'s_1\dots s_\mathcal{L}\\
&= \lambda x\cdot\lambda y_{\mathcal{L}+1}\cdot\ldots\cdot\lambda y_{n+1}\cdot[\![Q^\ell u]\!]'s_1\dots s_{\mathcal{L}-1}\langle x, s_\mathcal{L}\rangle\hat{y}_{\mathcal{L}+1}\dots\hat{y}_{n+1}\\
&= \lambda x\cdot\lambda y_{\mathcal{L}+1}\cdot\ldots\cdot\lambda y_{n+1}\cdot[\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_{\mathcal{L}-1}\langle x, s_\mathcal{L}\rangle\hat{y}_{\mathcal{L}+1}\dots\hat{y}_{n+1}
\end{aligned}$$

- Rule $(Q^\ell\star^\mathcal{L})$, $l = Q^\ell(u\star^{\mathcal{L}-1}v)$, $r = Q^\ell u\star^\mathcal{L} Q^\ell v$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_n = [\![u\star^{\mathcal{L}-1}v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_n\\
&= ([\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L})(\lambda z_\mathcal{L}\cdot\ldots\cdot\lambda z_m\cdot[\![v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\hat{z}_\mathcal{L}\dots\hat{z}_m)s_{\mathcal{L}+1}\dots s_n
\end{aligned}$$

while:

$$\begin{aligned}
&[\![r]\!]'s_1\dots s_n = ([\![Q^\ell u]\!]'s_1\dots s_\mathcal{L})(\lambda z_{\mathcal{L}+1}\cdot\ldots\cdot\lambda z_{m+1}\cdot[\![Q^\ell v]\!]'s_1\dots s_\mathcal{L}\hat{z}_{\mathcal{L}+1}\dots\hat{z}_{m+1})s_{\mathcal{L}+1}\dots s_n\\
&= ([\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L})(\lambda z_{\mathcal{L}+1}\cdot\ldots\cdot\lambda z_{m+1}\cdot[\![v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\hat{z}_{\mathcal{L}+1}\dots\hat{z}_{m+1})s_{\mathcal{L}+1}\dots s_n
\end{aligned}$$

- Rule $(Q^\ell id^\mathcal{L})$, $l = Q^\ell id^{\mathcal{L}-1}$, $r = id^\mathcal{L}$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_\mathcal{L} = [\![id^{\mathcal{L}-1}]\!]'s_1\dots s_{\ell-1}s_{\ell+1}s_{\ell+2}\dots s_\mathcal{L}\\
&= s_\mathcal{L} = [\![id^\mathcal{L}]\!]'s_1\dots s_\mathcal{L} = [\![r]\!]'s_1\dots s_\mathcal{L}
\end{aligned}$$

- Rule $(Q^\ell\circ^\mathcal{L})$, $l = Q^\ell(u\circ^{\mathcal{L}-1}v)$, $r = Q^\ell u\circ^\mathcal{L} Q^\ell v$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_n = [\![u\circ^{\mathcal{L}-1}v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_n\\
&= [\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_{\mathcal{L}-1}([\![v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L})s_{\mathcal{L}+1}\dots s_n\\
&= [\![Q^\ell u]\!]'s_1\dots s_{\mathcal{L}-1}([\![Q^\ell v]\!]'s_1\dots s_\mathcal{L})s_{\mathcal{L}+1}\dots s_n\\
&= [\![Q^\ell u\circ^\mathcal{L} Q^\ell v]\!]'s_1\dots s_n = [\![r]\!]'s_1\dots s_n
\end{aligned}$$

- Rule $(Q^\ell\uparrow^\mathcal{L})$, $l = Q^\ell\uparrow^{\mathcal{L}-1}$, $r = \uparrow^\mathcal{L}$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_\mathcal{L} = [\![\uparrow^{\mathcal{L}-1}]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\\
&= \text{snd } s_\mathcal{L} = [\![\uparrow^\mathcal{L}]\!]'s_1\dots s_\mathcal{L} = [\![r]\!]'s_1\dots s_\mathcal{L}
\end{aligned}$$

- Rule $(Q^\ell 1^\mathcal{L})$, $l = Q^\ell 1^{\mathcal{L}-1}$, $r = 1^\mathcal{L}$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_n = [\![1^{\mathcal{L}-1}]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_n\\
&= (\text{fst } s_\mathcal{L})s_{\mathcal{L}+1}\dots s_n = [\![1^\mathcal{L}]\!]'s_1\dots s_n = [\![r]\!]'s_1\dots s_n
\end{aligned}$$

- Rule $(Q^\ell\bullet^\mathcal{L})$, $l = Q^\ell(u\bullet^{\mathcal{L}-1}v)$, $r = Q^\ell u\bullet^\mathcal{L} Q^\ell v$:

$$\begin{aligned}
&[\![l]\!]'s_1\dots s_\mathcal{L} = [\![u\bullet^{\mathcal{L}-1}v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\\
&= \langle\lambda\overline{y}\cdot[\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\overline{\hat{y}}, [\![v]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\rangle\\
&= \langle\lambda\overline{y}\cdot[\![u]\!]'s_1\dots s_{\ell-1}s_{\ell+1}\dots s_\mathcal{L}\overline{\hat{y}}, [\![Q^\ell v]\!]'s_1\dots s_\mathcal{L}\rangle
\end{aligned}$$

- Rule ($Q^\ell \Uparrow^\mathcal{L}$): follows from rules ($Q^\ell \bullet^\mathcal{L}$), ($Q^\ell 1^\mathcal{L}$), ($Q^\ell \circ^\mathcal{L}$) and ($Q^\ell \uparrow^\mathcal{L}$).
- Rule ($Q^\ell Q^\mathcal{L}$), $l = Q^\ell(Q^{\mathcal{L}-1}u)$, $r = Q^\mathcal{L}(Q^\ell u)$:

$$\begin{array}{l}
[\![l]\!]' s_1 \ldots s_n = [\![Q^{\mathcal{L}-1}u]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} \ldots s_n \\
= [\![u]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} \ldots s_{\mathcal{L}-1} s_{\mathcal{L}+1} \ldots s_n \\
= [\![Q^\ell u]\!]' s_1 \ldots s_{\mathcal{L}-1} s_{\mathcal{L}+1} \ldots s_n \\
= [\![Q^\mathcal{L}(Q^\ell u)]\!]' s_1 \ldots s_n = [\![r]\!]' s_1 \ldots s_n
\end{array}$$

- Rule ($Q^\ell \mathtt{ev}^\mathcal{L}$), $l = Q^\ell(\mathtt{ev}^\mathcal{L} uw)$, $r = \mathtt{ev}^{\mathcal{L}+1}(Q^\ell u)(Q^\ell w)$:

$$\begin{array}{l}
[\![l]\!]' s_1 \ldots s_n = [\![\mathtt{ev}^\mathcal{L} uw]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} \ldots s_n \\
= [\![u]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} \ldots s_\mathcal{L} ([\![v]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} \ldots s_\mathcal{L}) s_{\mathcal{L}+1} \ldots s_n \\
= [\![u]\!]' s_1 \ldots s_{\ell-1} s_{\ell+1} \ldots s_\mathcal{L} ([\![Q^\ell v]\!]' s_1 \ldots s_\mathcal{L}) s_{\mathcal{L}+1} \ldots s_n \\
= [\![Q^\ell u]\!]' s_1 \ldots s_\mathcal{L} ([\![Q^\ell v]\!]' s_1 \ldots s_\mathcal{L}) s_{\mathcal{L}+1} \ldots s_n \\
= [\![\mathtt{ev}^{\mathcal{L}+1}(Q^\ell u)(Q^\ell v)]\!]' s_1 \ldots s_n = [\![r]\!]' s_1 \ldots s_n
\end{array}$$

□

Lemma 3.6 *For every λ-terms $s_1, \ldots, s_n$ of the right types:*

$$[\![(\lambda^\ell u) \star^\ell v]\!]' s_1 \ldots s_n \longrightarrow^+ [\![u \circ^\ell (v \bullet^\ell id^\ell)]\!]' s_1 \ldots s_n$$

Proof:

$$\begin{array}{l}
[\![(\lambda^\ell u) \star^\ell v]\!]' s_1 \ldots s_n \\
= ([\![\lambda^\ell u]\!]' s_1 \ldots s_\ell)(\lambda z_{\ell+1} \cdot \ldots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \ldots s_\ell \hat{z}_{\ell+1} \ldots \hat{z}_m) s_{\ell+1} \ldots s_n \\
= (\lambda x \cdot \lambda y_{\ell+1} \cdot \ldots \cdot \lambda y_n \cdot [\![u]\!]' s_1 \ldots s_{\ell-1} \langle x, s_\ell \rangle \hat{y}_{\ell+1} \ldots \hat{y}_n) \\
\qquad (\lambda z_{\ell+1} \cdot \ldots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \ldots s_\ell \hat{z}_{\ell+1} \ldots \hat{z}_m) s_{\ell+1} \ldots s_n \\
\longrightarrow^+ [\![u]\!]' s_1 \ldots s_{\ell-1} \langle \lambda z_{\ell+1} \cdot \ldots \cdot \lambda z_m \cdot [\![v]\!]' s_1 \ldots s_\ell \hat{z}_{\ell+1} \ldots \hat{z}_m, s_\ell \rangle s_{\ell+1} \ldots s_n \\
\qquad \text{by } n - \ell + 1 \geq 1 \text{ applications of } (\beta) \text{ and Lemma 3.4}
\end{array}$$

while:

$$\begin{array}{l}
[\![u \circ^\ell (v \bullet^\ell id^\ell)]\!]' s_1 \ldots s_n \\
= [\![u]\!]' s_1 \ldots s_{\ell-1} ([\![v \bullet^\ell id^\ell]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\
= [\![u]\!]' s_1 \ldots s_{\ell-1} \langle \lambda y_{\ell+1} \ldots y_m \cdot [\![v]\!]' s_1 \ldots s_\ell \hat{y}_{\ell+1} \ldots \hat{y}_m, s_\ell \rangle s_{\ell+1} \ldots s_n
\end{array}$$

□

Definition 3.5 *For any ground typed $\lambda\mathtt{ev}Q^+$-terms u, v, we say that $u \succeq_{\lambda\mathtt{ev}Q} v$ (resp. $u \succ_{\lambda\mathtt{ev}Q} v$) if and only if, for every stack-like λ-terms of the right type:*

$$[\![u]\!]' s_1 \ldots s_n \longrightarrow^* [\![v]\!]' s_1 \ldots s_n$$

(resp. $\longrightarrow^+$).

Definition 3.6 *For any typed $\lambda\mathtt{ev}Q$-terms u, v, we say that $u \succeq'_{\lambda\mathtt{ev}Q} v$ (resp. $u \succ'_{\lambda\mathtt{ev}Q} v$) if and only if $u`\rho \succeq_{\lambda\mathtt{ev}Q} v`\rho$ (resp. $\succ_{\lambda\mathtt{ev}Q}$) for every environment ρ whose domain contains all the free variables of both u and v.*

Observe that, since the typed λ-calculus terminates, $\succ_{\lambda\mathtt{ev}Q}$ and $\succ'_{\lambda\mathtt{ev}Q}$ are well-founded.

We reformulate Lemma 3.5 and Lemma 3.6 as follows:

Lemma 3.7 *For every rule $l \to r$ in $\lambda\mathtt{ev}Q_H$, $l \succeq'_{\lambda\mathtt{ev}Q} r$. Moreover, in the case of rules (β) and (β^ℓ), $l \succ'_{\lambda\mathtt{ev}Q} r$.*

Unfortunately, $\succ_{\lambda\mathtt{ev}Q}$ (or $\succ'_{\lambda\mathtt{ev}Q}$) is not monotonic, but $\succeq_{\lambda\mathtt{ev}Q}$ and $\succeq'_{\lambda\mathtt{ev}Q}$ are:

Lemma 3.8 *If $u \succeq_{\lambda\mathtt{ev}Q} v$, then for every context $\mathcal{C}$ such that $\mathcal{C}[u]$ is well-typed, $\mathcal{C}[u] \succeq_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$, and similarly with $\succeq'_{\lambda\mathtt{ev}Q}$.*

Proof: By structural induction on $\mathcal{C}$, we show that $[\![\mathcal{C}[u]]\!]' s_1 \ldots s_n \longrightarrow^* [\![\mathcal{C}[v]]\!]' s_1 \ldots s_n$ for every wstack-like terms $s_1, \ldots, s_n$ of the right types. This is obvious if $\mathcal{C}$ is the empty context [].

If $\mathcal{C} = \mathcal{C}_1 \circ^\ell w$, then $[\![\mathcal{C}[u]]\!]' s_1 \ldots s_n = [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_{\ell-1}([\![w]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \longrightarrow^* [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_{\ell-1}([\![w]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n$ (by induction hypothesis) $= [\![\mathcal{C}[v]]\!]' s_1 \ldots s_n$.

If $\mathcal{C} = w \circ^\ell \mathcal{C}_1$, then $[\![\mathcal{C}[u]]\!]' s_1 \ldots s_n = [\![w]\!]' s_1 \ldots s_{\ell-1}(\mathcal{C}_1[u] s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n$. Now by induction hypothesis $\mathcal{C}_1[u] s_1 \ldots s_\ell \longrightarrow^* \mathcal{C}_1[v] s_1 \ldots s_\ell$, so by Lemma 3.3, $[\![w]\!]' s_1 \ldots s_{\ell-1}(\mathcal{C}_1[u] s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \longrightarrow^* [\![w]\!]' s_1 \ldots s_{\ell-1}(\mathcal{C}_1[v] s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n = [\![\mathcal{C}[v]]\!]' s_1 \ldots s_n$.

The cases when $\mathcal{C} = \mathtt{ev}^\ell \mathcal{C}_1 w$ and $\mathcal{C} = \mathtt{ev}^\ell w \mathcal{C}_1$ are similar. All other cases follow directly from the induction hypothesis.

The Theorem then follows. □

The only problem is that we cannot conclude that whenever u reduces to v by (β) or (β^ℓ), then $u \succ'_{\lambda\mathtt{ev}Q} v$. Indeed, the best we can conclude is $u \succeq'_{\lambda\mathtt{ev}Q} v$; if we had the stronger $u \succ'_{\lambda\mathtt{ev}Q} v$, then we could conclude that the typed $\lambda\mathtt{ev}Q_H$-calculus were strongly normalizing (by the lexicographic product of $\succ'_{\lambda\mathtt{ev}Q}$ and the well-founded derivation ordering on typed Σ_H), which is wrong (Theorem 2.1, Part IIIa).

3.2 Safe Contexts

We solve the problem by allowing reduction to take place under so-called safe contexts:

Definition 3.7 *A context $\mathcal{C}$ is said to be* safe *if and only if, for every $\lambda\mathtt{ev}Q$-terms u and v of the same type such that $\mathcal{C}[u]$ is typable in $\lambda\mathtt{ev}Q$, $u \succ'_{\lambda\mathtt{ev}Q} v$ implies $\mathcal{C}[u] \succ'_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$.*

We then proceed to identify a few safe contexts. First, passing from $\lambda\mathtt{ev}Q$ to (ground) $\lambda\mathtt{ev}Q^+$ will provide a slightly simplified condition:

Definition 3.8 *A context $\mathcal{C}$ is said to be* safe$^+$ *if and only if, for every $\lambda\mathtt{ev}Q^+$-terms u and v of the same type such that $\mathcal{C}[u]$ is typable, $u \succ_{\lambda\mathtt{ev}Q} v$ implies $\mathcal{C}[u] \succ_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$.*

We shall first identify a class of safe$^+$ contexts, from which it will be easy to infer a corresponding class of safe contexts. The main example of unsafe$^+$ context is $w \circ^\ell_T \mathcal{C}$, where w is a term such that $[\![w]\!]' s_1 \ldots s_n$ does not depend on some or all $t_{\ell j}$, where $s_i = (t_{ij})_{1 \leq j \leq m_i}$, $1 \leq i \leq n$. Indeed, $[\![w \circ^\ell_T \mathcal{C}[u]]\!]' s_1 \ldots s_n = [\![w]\!]' s_1 \ldots s_{\ell-1}([\![\mathcal{C}[u]]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n$ will then actually be independent of $[\![\mathcal{C}[u]]\!]' s_1 \ldots s_\ell$, which means that $w \circ^\ell_T \mathcal{C}[u]$ is the same for all u: such a context $w \circ^\ell_T \mathcal{C}$ is unsafe$^+$. This happens in particular when $w = Q^\ell_T w'$, or when $w = 1^\ell$. We therefore start by defining a class of terms T^k, for each $k \geq 1$, and similarly a class of stacks S^k, such that for every w in these classes, $[\![w]\!]' s_1 \ldots s_n$ really depends on every t_{ij}, $i \geq k$, $1 \leq j \leq m_i$. By Lemma 3.2, this means that whenever the t_{ij} are variables, they must occur free in $[\![w]\!]' s_1 \ldots s_n$ for every $i \geq k$.

Definition 3.9 *For every $k \geq 1$, we define the sets T^k and S^k as the smallest sets of $\lambda\mathtt{ev}Q$-terms of sort T and S respectively such that:*

- $\lambda^\ell u \in T^k$ *iff* $k \geq \ell + 1$ *or* $u \in T^k$*;*
- $u \star^\ell v \in T^k$ *iff* $k \geq \ell + 1$ *or* $u \in T^k$ *or* $v \in T^k$*;*
- $1^\ell \in T^k$ *iff* $k \geq \ell + 1$*;*
- $u \circ^\ell_T v \in T^k$ *iff either* $k \geq \ell + 1$ *and* $u \in T^k$*, or* $k \leq \ell$ *and* $u \in T^\ell$ *and* $v \in S^k$*;*
- $\mathtt{ev}^\ell_T uv \in T^k$ *iff either* $k \geq \ell$ *and* $u \in T^{k+1}$*, or* $k < \ell$ *and* $u \in T^\ell$ *and* $v \in S^k$*;*
- $Q^\ell_T u \in T^k$ *iff* $k \geq \ell + 1$ *and* $u \in T^{k-1}$*;*

and respectively:

- $\uparrow^\ell \in S^k$ *iff* $k \geq \ell + 1$*;*

- $id^\ell \in S^k$ *iff* $k \geq \ell$*;*

- $u \bullet^\ell v \in S^k$ *iff* $k \geq \ell + 1$ *or* $u \in T^k$ *or* $v \in S^k$*;*

- $\Uparrow^\ell u \in S^k$ *iff* $k \geq \ell$*, and if* $k = \ell$ *then* $u \in S^k$*;*

- $u \circ^\ell_S v \in T^k$ *iff either* $k \geq \ell + 1$ *and* $u \in T^k$*, or* $k \leq \ell$ *and* $u \in T^\ell$ *and* $v \in S^k$*;*

- $\mathtt{ev}^\ell_S uv \in S^k$ *iff either* $k \geq \ell$ *and* $u \in S^{k+1}$*, or* $k < \ell$ *and* $u \in S^\ell$ *and* $v \in S^k$*;*

- $Q^\ell_S u \in S^k$ *iff* $k \geq \ell + 1$ *and* $u \in S^{k-1}$*;*

Lemma 3.9 *For every* $u \in T^k$ *(resp.* $u \in S^k$*),* $k \geq 1$*, for every stack-like* λ*-terms* $s_1 = (t_{1j})_{1\leq j\leq m_1}$*, …,* $s_n = (t_{1j})_{1\leq j\leq m_n}$ *of the right types, and where* t_{ij} *are pairwise distinct variables,* t_{ij} *is free in* $[\![u]\!]' s_1 \dots s_n$ *for every* $i \geq k$*,* $i \leq n$*,* $1 \leq j \leq m_i$*.*

Proof: By structural induction on u.

If $u = \lambda^\ell v$, then $u \in T^k$ implies that $k \geq \ell + 1$ or $v \in T^k$. Now $n = \ell$ and $[\![u]\!]' s_1 \dots s_\ell = \lambda x \cdot \lambda y_{\ell+1} \cdot \ldots \cdot \lambda y_n \cdot [\![v]\!]' s_1 \dots s_{\ell-1} \langle x, s_\ell\rangle \hat{y}_{\ell+1} \dots \hat{y}_n$. If $i \geq k$, $i \leq n$, then t_{ij} occurs free in the latter; indeed, either $k \geq \ell + 1$ and there is no such i ($i \geq k$ implies $i \geq \ell + 1$, and $i \leq n$ implies $i \leq \ell$), or $v \in T^k$, so by induction hypothesis t_{ij} is free in $[\![v]\!]' s_1 \dots s_{\ell-1}\langle x, s_\ell\rangle \hat{y}_{\ell+1} \dots \hat{y}_n$ for every $i \geq k$, $i \leq \ell$, $1 \leq j \leq m_i$ (as well as x and every component of $\hat{y}_{\ell+1}$, …, $\hat{y}_n$). Hence t_{ij} is free in $[\![u]\!]' s_1 \dots s_\ell$ as soon as $i \geq k$.

If $u = v \star^\ell w$ is in T^k, then $k \geq \ell + 1$ or $v \in T^k$ or $w \in T^k$. Now $[\![u]\!]' s_1 \dots s_n = ([\![v]\!]' s_1 \dots s_\ell)(\lambda z_{\ell+1} \cdot \ldots \cdot \lambda z_m \cdot [\![w]\!]' s_1 \dots s_\ell \hat{z}_{\ell+1} \dots \hat{z}_m) s_{\ell+1} \dots s_n$. We consider t_{ij} for $i \geq k$, $i \leq n$, $1 \leq j \leq m_i$; we then have three cases. If $i \geq \ell + 1$, then t_{ij} is free in s_i by definition, i.e. in one of $s_{\ell+1}$, …, s_n, hence in $[\![u]\!]' s_1 \dots s_n$; this is in particular the case when $k \geq \ell + 1$. If $i \leq \ell$ and $v \in T^k$, then t_{ij} is free in $[\![v]\!]' s_1 \dots s_\ell$ by induction hypothesis, hence in $[\![u]\!]' s_1 \dots s_n$. If $i \leq \ell$ and $w \in T^k$, then t_{ij} is free in $[\![w]\!]' s_1 \dots s_\ell \hat{z}_{\ell+1} \dots \hat{z}_m$ by induction hypothesis, hence in $[\![u]\!]' s_1 \dots s_n$ (because the $z_{i'j'}$ are fresh, hence different from the t_{ij}).

If $u = 1^\ell$, then $u \in T^k$ implies that $k \geq \ell + 1$. Then $[\![u]\!]' s_1 \dots s_n = (\mathrm{fst}\ s_\ell) s_{\ell+1} \dots s_n$, and as soon as $i \geq k$, since $k \geq \ell + 1$, t_{ij} is free in s_i, i.e. in one of $s_{\ell+1}$, …, s_n, hence in $[\![u]\!]' s_1 \dots s_n$.

If $u = \uparrow^\ell$, then $u \in S^k$ implies that $k \geq \ell + 1$. Then $n = \ell$ and $[\![u]\!]' s_1 \dots s_\ell = \mathrm{snd}\ s_\ell$, so as soon as $i \geq k$ and $i \leq n$, t_{ij} is free in $[\![u]\!]' s_1 \dots s_\ell$, since $i \geq k$ and $i \leq n$ is impossible.

If $u = id^\ell$, then $u \in S^k$ implies that $k \geq \ell$. Then $n = \ell$ and $[\![u]\!]' s_1 \dots s_\ell = s_\ell$, so as soon as $i \geq k$ and $i \leq n$, $t_{ij} = t_{\ell j}$ (indeed, $\ell \leq k \leq i \leq n = \ell$, so all these quantities are equal) is free in $s_\ell = [\![u]\!]' s_1 \dots s_\ell$.

If $u = v \circ^\ell_T w$, then $u \in T^k$ implies that either $k \geq \ell + 1$ and $v \in T^k$, or $k \leq \ell$ and $v \in T^\ell$ and $w \in S^k$. Now, $[\![u]\!]' s_1 \dots s_n = [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n$. If (first case) $k \geq \ell+1$ and $v \in T^k$, then for every $i \geq k$, $i \geq \ell + 1$, so t_{ij} occurs free in $[\![u]\!]' s_1 \dots s_n$ because $v \in T^k$ and by induction hypothesis. If (second case) $k \leq \ell$ and $v \in T^\ell$ and $w \in S^k$, then we have two subcases. If $i \geq \ell + 1$ (in particular $i \geq \ell$), then t_{ij} occurs free in $[\![u]\!]' s_1 \dots s_n$ because $v \in T^\ell$ and by induction hypothesis. If $i \geq k$, $i \leq \ell$, then t_{ij} occurs free in $[\![w]\!]' s_1 \dots s_\ell$, since $w \in T^k$ and by induction hypothesis. Now, by Lemma 3.2:

$$\begin{aligned}
&[\![u]\!]' s_1 \dots s_n = [\![v]\!]' s_1 \dots s_{\ell-1}([\![w]\!]' s_1 \dots s_\ell) s_{\ell+1} \dots s_n \\
&= \left([\![v]\!]' s_1 \dots s_{\ell-1}(y_p)_{1\leq p\leq m_\ell} s_{\ell+1} \dots s_n\right)[t'_1/y_1, \dots, t_{m_\ell}/y_{m_\ell}]
\end{aligned}$$

where $(t'_1, \dots, t'_{m_\ell}) = [\![w]\!]' s_1 \dots s_\ell$ and y_1, …, y_{m_ℓ} are fresh variables. Now, since t_{ij} occurs free in $[\![w]\!]' s_1 \dots s_\ell$, t_{ij} occurs free in some t'_p, $1 \leq p \leq m_\ell$. Since $v \in T^\ell$ and by induction hypothesis, y_p occurs free in $[\![v]\!]' s_1 \dots s_{\ell-1}(y_p)_{1\leq p\leq m_\ell} s_{\ell+1} \dots s_n$, hence t_{ij} occurs free in $[\![u]\!]' s_1 \dots s_n = ([\![v]\!]' s_1 \dots s_{\ell-1}(y_p)_{1\leq p\leq m_\ell} s_{\ell+1} \dots s_n)[t'_1/y_1, \dots, t_{m_\ell}/y_{m_\ell}]$.

The case when $u = v \circ^\ell_S w$ is entirely analogous.

The cases when $u = \mathtt{ev}^\ell_T vw$ or when $u = \mathtt{ev}^\ell_S vw$ follow by similar arguments.

When $u = Q^\ell_T v$, the fact that $u \in T^k$ means that $k \geq \ell + 1$ and $u \in T^{k-1}$. Now $[\![u]\!]' s_1 \dots s_n = [\![v]\!]' s_1 \dots s_{\ell-1} s_{\ell+1} \dots s_n$, so for every $i \geq k$, $i - 1 \geq k - 1 \geq \ell$ and by induction hypothesis t_{ij} occurs free in $[\![v]\!]' s_1 \dots s_{\ell-1} s_{\ell+1} \dots s_n$, hence in $[\![u]\!]' s_1 \dots s_n$.

The case when $u = Q^\ell_S v$ is entirely analogous.

When $u = v \bullet^\ell w$ is in S^k, we have $k \geq \ell+1$ or $v \in T^k$ or $w \in S^k$. Now $n = \ell$ and $[\![u]\!]' s_1 \dots s_\ell = \langle \lambda y_{\ell+1} \cdot \dots \cdot \lambda y_m \cdot [\![v]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_m, [\![w]\!]' s_1 \dots s_\ell \rangle$. If $k \geq \ell+1$, then for every $i \geq k$, $i \leq n$, t_{ij} is free in $[\![u]\!]' s_1 \dots s_\ell$ because there is no such i. If $v \in T^k$, then for every $i \geq k$, $i \leq n$, t_{ij} is free in $[\![v]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_m$, hence in $[\![u]\!]' s_1 \dots s_\ell$. And similarly if $w \in S^k$, t_{ij} is free in $[\![w]\!]' s_1 \dots s_\ell$, hence in $[\![u]\!]' s_1 \dots s_\ell$.

And if $u = \Uparrow^\ell v$ is in S^k, then $k \geq \ell$, and if $k = \ell$ then $u \in S^k$. Now $n = \ell$ and $[\![u]\!]' s_1 \dots s_\ell = \langle \text{fst } s_\ell, [\![v]\!]' s_1 \dots s_{\ell-1} (\text{snd } s_\ell) \rangle$. If $k \geq \ell+1$, then for every $i \geq k$, $i \leq n$, t_{ij} is free in $[\![u]\!]' s_1 \dots s_\ell$ because there is no such i. And if $k = \ell$, then for every $i \geq k$, $i \leq n$, i in fact equals n and ℓ. If $j = 1$, then t_{ij} is free in $[\![u]\!]' s_1 \dots s_\ell$ because $t_{ij} = t_{\ell 1} = \text{fst } s_\ell$. And if $j \geq 1$, then t_{ij} is free in $[\![v]\!]' s_1 \dots s_{\ell-1} (\text{snd } s_\ell) = [\![v]\!]' s_1 \dots s_{\ell-1} (t_{\ell j})_{j \geq 2}$, so it is free in $[\![u]\!]' s_1 \dots s_\ell$. □

Lemma 3.10 *For every $v \in T^k \cup S^k$, if $t_{i_0 j_0} \longrightarrow t'_{i_0 j_0}$ for some $k \leq i_0 \leq \ell$, $1 \leq j \leq m_{i_0}$, then:*

$$[\![v]\!]' s_1 \dots s_n \longrightarrow^+ [\![v]\!]' s_1 \dots s_{i_0 - 1} s'_{i_0} s_{i_0+1} \dots s_n$$

where $s_i = (t_{ij})_{1 \leq j \leq m_i}$ for all i, and $s'_{i_0} = (t'_{i_0 j})_{1 \leq j \leq m_{i_0}}$, with $t'_{i_0 j} = t_{i_0 j}$ for $j \neq j_0$ by convention.

Proof: Same as Lemma 3.3, but this time, $i_0 \geq k$ and $v \in T^k \cup S^k$ implies that $t_{i_0 j_0}$ occurs at least once in $[\![v]\!]' s_1 \dots s_n$ by Lemma 3.9. Replacing these occurrences by $t'_{i_0 j_0}$ yields $[\![v]\!]' s_1 \dots s_{i_0-1} s'_{i_0} s_{i_0+1} \dots s_n$ in at least one step, hence the result. (This is in fact more similar to Lemma 2.29, Part IIIa.) □

Definition 3.10 *We define the class Sf^+ of* syntactically safe$^+$ contexts *as the union of Sf^+_T and Sf^+_S, where:*

$$
\begin{array}{rcll}
Sf^+_T & ::= & [] \mid \lambda^\ell Sf^+_T \mid Sf^+_T \star^\ell T \mid T \star^\ell Sf_T & \\
 & & \mid Sf^+_T \circ^\ell_T S \mid T^\ell \circ^\ell_T Sf^+_S & \\
 & & \mid \mathtt{ev}^\ell_T Sf^+_T S \mid \mathtt{ev}^\ell_T T^\ell Sf^+_S & \\
 & & \mid Q^\ell_T Sf^+_T & \textit{(for all } \ell \geq 1\textit{)} \\
Sf^+_S & ::= & Sf^+_T \bullet^\ell S \mid T \bullet^\ell Sf^+_S \mid \Uparrow^\ell Sf^+_S & \\
 & & \mid Sf^+_S \circ^\ell_S S \mid S^\ell \circ^\ell_S Sf^+_S & \\
 & & \mid \mathtt{ev}^\ell_S Sf^+_S S \mid \mathtt{ev}^\ell_S S^\ell Sf^+_S & \\
 & & \mid Q^\ell_S Sf^+_S & \textit{(for all } \ell \geq 1\textit{)}
\end{array}
$$

Lemma 3.11 *Every syntactically safe$^+$ context is safe$^+$.*

Proof: By structural induction on a syntactically safe$^+$ context $\mathcal{C}$, we show that for any two $\lambda\mathtt{ev}Q^+$-terms u and v of the same type such that $\mathcal{C}[u]$ is typable and $u \succ_{\lambda\mathtt{ev}Q} v$: $\mathcal{C}[u] \succ_{\lambda\mathtt{ev}Q} \mathcal{C}[v]$.

If $\mathcal{C} = []$, then $[\![\mathcal{C}[u]]\!]' s_1 \dots s_n = [\![u]\!]' s_1 \dots s_n \longrightarrow^+ [\![v]\!]' s_1 \dots s_n = [\![\mathcal{C}[v]]\!]' s_1 \dots s_n$ because $u \succ_{\lambda\mathtt{ev}Q} v$.

If $\mathcal{C} = \lambda^\ell \mathcal{C}_1$, with $\mathcal{C}_1$ safe$^+$. Then $[\![\mathcal{C}[u]]\!]' s_1 \dots s_\ell = \lambda x \cdot \lambda y_{\ell+1} \cdot \dots \cdot \lambda y_m \cdot [\![\mathcal{C}_1[u]]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_m \longrightarrow^+ \lambda x \cdot \lambda y_{\ell+1} \cdot \dots \cdot \lambda y_m \cdot [\![\mathcal{C}_1[v]]\!]' s_1 \dots s_\ell \hat{y}_{\ell+1} \dots \hat{y}_m$ (since by induction hypothesis $\mathcal{C}_1[u] \succ_{\lambda\mathtt{ev}Q} \mathcal{C}_1[v]$) $= [\![\mathcal{C}[v]]\!]' s_1 \dots s_\ell$.

If $\mathcal{C} = \mathcal{C}_1 \star^\ell w$ with $\mathcal{C}_1$ safe$^+$, then:

$$
\begin{array}{ll}
[\![\mathcal{C}[u]]\!]' s_1 \dots s_n = & \\
([\![\mathcal{C}_1[u]]\!]' s_1 \dots s_\ell)(\lambda z_{\ell+1} \cdot \dots \cdot \lambda z_m \cdot [\![w]\!]' s_1 \dots s_\ell \hat{z}_{\ell+1} \dots \hat{z}_m) s_{\ell+1} \dots s_n & \\
\longrightarrow^+ ([\![\mathcal{C}_1[v]]\!]' s_1 \dots s_\ell)(\lambda z_{\ell+1} \cdot \dots \cdot \lambda z_m \cdot [\![w]\!]' s_1 \dots s_\ell \hat{z}_{\ell+1} \dots \hat{z}_m) s_{\ell+1} \dots s_n & \text{(by induction hypothesis)} \\
= [\![\mathcal{C}[v]]\!]' s_1 \dots s_n &
\end{array}
$$

If $\mathcal{C} = w \star^\ell \mathcal{C}_1$ with $\mathcal{C}_1$ safe$^+$, then:

$$
\begin{array}{ll}
[\![\mathcal{C}[u]]\!]' s_1 \dots s_n = & \\
([\![w]\!]' s_1 \dots s_\ell)(\lambda z_{\ell+1} \cdot \dots \cdot \lambda z_m \cdot [\![\mathcal{C}_1[u]]\!]' s_1 \dots s_\ell \hat{z}_{\ell+1} \dots \hat{z}_m) s_{\ell+1} \dots s_n & \\
\longrightarrow^+ ([\![w]\!]' s_1 \dots s_\ell)(\lambda z_{\ell+1} \cdot \dots \cdot \lambda z_m \cdot [\![\mathcal{C}_1[v]]\!]' s_1 \dots s_\ell \hat{z}_{\ell+1} \dots \hat{z}_m) s_{\ell+1} \dots s_n & \text{(by induction hypothesis)} \\
= [\![\mathcal{C}[v]]\!]' s_1 \dots s_n &
\end{array}
$$

If $\mathcal{C} = \mathcal{C}_1 \circ_T^\ell w$, with $\mathcal{C}_1$ safe$^+$, then:

$$\begin{array}{l} [\![\mathcal{C}[u]]\!]' s_1 \ldots s_n \\ = [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_{\ell-1}([\![w]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\ \longrightarrow^+ [\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_{\ell-1}([\![w]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\ = [\![\mathcal{C}[v]]\!]' s_1 \ldots s_n \end{array}$$

by induction hypothesis. Similarly when $\mathcal{C} = \mathtt{ev}_T^\ell \mathcal{C}_1 w$, or when $\mathcal{C} = Q_T^\ell \mathcal{C}_1$; similarly also when $\mathcal{C}$ has the form $\mathcal{C}_1 \circ_S^\ell w$, $\mathtt{ev}_S^\ell \mathcal{C}_1 w$ or $Q_S^\ell \mathcal{C}_1$.

If $\mathcal{C} = w \circ_T^\ell \mathcal{C}_1$ with $w \in T^\ell$ and $\mathcal{C}_1$ safe$^+$, then:

$$\begin{array}{l} [\![\mathcal{C}[u]]\!]' s_1 \ldots s_n \\ = [\![w]\!]' s_1 \ldots s_{\ell-1}([\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\ \longrightarrow^+ [\![w]\!]' s_1 \ldots s_{\ell-1}([\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_\ell) s_{\ell+1} \ldots s_n \\ \qquad \text{because } [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_\ell \longrightarrow^+ [\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_\ell \text{ by induction hypothesis} \\ \qquad \text{and by Lemma 3.10 applied to } w, \text{ since } w \in T^\ell \\ = [\![\mathcal{C}[v]]\!]' s_1 \ldots s_n \end{array}$$

Similarly when $\mathcal{C} = w \circ_S^\ell \mathcal{C}_1$.

If $\mathcal{C} = \mathtt{ev}_T^\ell w \mathcal{C}_1$ with $w \in T^\ell$ and $\mathcal{C}_1$ safe$^+$, then:

$$\begin{array}{ll} [\![\mathcal{C}[u]]\!]' s_1 \ldots s_n & \\ = [\![w]\!]' s_1 \ldots s_{\ell-1}([\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_n & \\ \longrightarrow^+ [\![w]\!]' s_1 \ldots s_{\ell-1}([\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_{\ell-1}) s_\ell \ldots s_n & \text{by the same argument as above} \\ = [\![\mathcal{C}[v]]\!]' s_1 \ldots s_n & \end{array}$$

And similarly when $\mathcal{C} = \mathtt{ev}_S^\ell w \mathcal{C}_1$.

If $\mathcal{C} = \mathcal{C}_1 \bullet^\ell w$ with $\mathcal{C}_1$ safe$^+$, then:

$$\begin{array}{l} [\![\mathcal{C}[u]]\!]' s_1 \ldots s_\ell = \\ \langle \lambda y_{\ell+1} \cdot \ldots \cdot \lambda y_m \cdot [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_\ell \hat{y}_{\ell+1} \ldots \hat{y}_m, [\![w]\!]' s_1 \ldots s_\ell \rangle \\ \longrightarrow^+ \langle \lambda y_{\ell+1} \cdot \ldots \cdot \lambda y_m \cdot [\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_\ell \hat{y}_{\ell+1} \ldots \hat{y}_m, [\![w]\!]' s_1 \ldots s_\ell \rangle = \\ [\![\mathcal{C}[v]]\!]' s_1 \ldots s_\ell \end{array}$$

If $\mathcal{C} = w \bullet^\ell \mathcal{C}_1$ with $\mathcal{C}_1$ safe$^+$, then:

$$\begin{array}{l} [\![\mathcal{C}[u]]\!]' s_1 \ldots s_\ell = \\ \langle \lambda y_{\ell+1} \cdot \ldots \cdot \lambda y_m \cdot [\![w]\!]' s_1 \ldots s_\ell \hat{y}_{\ell+1} \ldots \hat{y}_m, [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_\ell \rangle \\ \longrightarrow^+ \langle \lambda y_{\ell+1} \cdot \ldots \cdot \lambda y_m \cdot [\![w]\!]' s_1 \ldots s_\ell \hat{y}_{\ell+1} \ldots \hat{y}_m, [\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_\ell \rangle \\ = [\![\mathcal{C}[v]]\!]' s_1 \ldots s_\ell \end{array}$$

by induction hypothesis.

When $\mathcal{C} = \Uparrow^\ell \mathcal{C}_1$, then:

$$\begin{array}{l} [\![\mathcal{C}[u]]\!]' s_1 \ldots s_\ell \\ = \langle \text{fst } s_\ell, [\![\mathcal{C}_1[u]]\!]' s_1 \ldots s_{\ell-1}(\text{snd } s_\ell) \rangle \\ \longrightarrow^+ \langle \text{fst } s_\ell, [\![\mathcal{C}_1[v]]\!]' s_1 \ldots s_{\ell-1}(\text{snd } s_\ell) \rangle \\ = [\![\mathcal{C}[v]]\!]' s_1 \ldots s_\ell \end{array}$$

by induction hypothesis. □

3.3 Termination of the $\lambda\mathtt{ev}Q_H$-Calculus

We first apply the previous techniques to show that the typed $\lambda\mathtt{ev}Q_H$-calculus is Σ_H-SN. Recall that this means that all Σ_H-eager rewrites are finite. Actually, we do not need all the rules of Σ_H to achieve this goal, but there are several subsystems of Σ_H that would work, and the proof would not change. We let the reader modify the techniques to any system that he wishes.

The case of $\lambda\mathtt{ev}Q_H$ is actually the simplest case (compared to $\lambda\mathtt{ev}Q$ at least, i.e. compared to the system without the rules in group (H)), because Σ_H-normal stacks take on a rather simple form. We only need a subsystem of Σ_H to get this pleasant normal form:

Definition 3.11 *Let S_0 be the following set of rules in $\lambda\mathtt{ev}Q$:*

- *in group (A):* $(\uparrow)$;
- *in group (B):* $(\circ_S id^\ell)$, $(id\circ_S^\ell)$, $(\uparrow^\ell)$, $(\circ_S^\ell)$, $(\bullet^\ell)$;
- *in group (C):* $(\mathtt{ev}_S id^\ell)$, $(\mathtt{ev}_S \circ_S^\ell)$, $(\mathtt{ev}_S \uparrow^\ell)$, $(\mathtt{ev}_S \bullet^\ell)$;
- *in group (D):* $(id^{\mathcal{L}} \circ_S^\ell)$, $(\circ_S^{\mathcal{L}} \circ_S^\ell)$, $(\uparrow^{\mathcal{L}} \circ_S^\ell)$, $(\bullet^{\mathcal{L}} \circ_S^\ell)$;
- *in group (E):* $(\mathtt{ev}_S^\ell id^{\mathcal{L}})$, $(\mathtt{ev}_S^\ell \circ_S^{\mathcal{L}})$, $(\mathtt{ev}_S^\ell \uparrow^{\mathcal{L}})$, $(\mathtt{ev}_S^\ell \bullet_S^{\mathcal{L}})$;
- *in group (F):* $(Q_S^\ell id^{\mathcal{L}})$, $(Q_S^\ell \circ_S^{\mathcal{L}})$, $(Q_S^\ell \uparrow^{\mathcal{L}})$, $(Q_S^\ell \bullet^{\mathcal{L}})$;

for every $1 \leq \ell < \mathcal{L}$.

Let S be S_0 plus the rules:

- *in group (B):* $(\uparrow\Uparrow^\ell)$, $(\uparrow\Uparrow \circ^\ell)$, $(\Uparrow\bullet^\ell)$, $(\Uparrow\Uparrow^\ell)$, $(\Uparrow\Uparrow \circ^\ell)$, $(\Uparrow id^\ell)$;
- *in group (C):* $(\uparrow \mathtt{ev}_S \Uparrow^\ell)$, $(\mathtt{ev}_S \Uparrow^\ell)$, $(\mathtt{ev}_S \Uparrow\Uparrow^\ell)$;
- *in group (D):* $(\Uparrow^{\mathcal{L}} \circ_S^\ell)$, $(\mathtt{ev}_S^{\mathcal{L}} \circ_S^\ell)$;
- *in group (E):* $(\mathtt{ev}_S^\ell \Uparrow^{\mathcal{L}})$, $(\mathtt{ev}_S^\ell \mathtt{ev}_S^{\mathcal{L}})$;
- *in group (F):* $(Q_S^\ell \Uparrow^{\mathcal{L}})$, $(Q_S^\ell \mathtt{ev}^{\mathcal{L}})$;

for every $1 \leq \ell < \mathcal{L}$.

Let S_H be S_0 plus the rules $(\eta \Uparrow^\ell)$, $\ell \geq 1$.

Observe that S and S_H are not confluent, by the way; but they can be completed to a confluent system, namely Σ and Σ_H respectively, in particular.

Lemma 3.12 *Lemma 2.1 still holds if we replace* (2) *by "u is S_0-normal", in particular by "u is S_H-normal" or by "u is S-normal". That is, every typed S_0-normal T-free stack is either* () *or some* pop_k^ℓ *with $\ell \geq 1$ and $k \geq 0$.*

Proof: Same as Lemma 2.1. □

Lemma 3.13 *Lemma 2.2 still holds if we replace* (2) *by "u is S_H-normal". That is, every typed S_H-normal non-T-free stack is of the form $u_1 \bullet^\ell u_2$ for some $\ell \geq 0$, u_1 and u_2.*

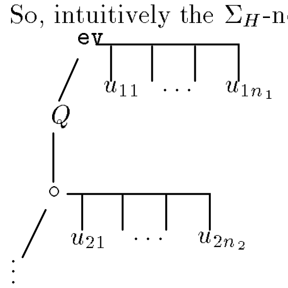
Proof: Same as Lemma 2.2. (Again, we don't need all rules in S_H.) □

Lemma 3.14 *The typed S_H-normal stacks u of types of the form $\overline{\varsigma^\ell \overset{\square}{\Rightarrow} \varsigma}$, $\ell \geq 0$, have the following form:*

- $s_1 \bullet \ldots \bullet s_n \bullet ()$ *for some $n \geq 0$, if $\ell = 0$;*
- $s_1 \bullet^\ell \ldots \bullet^\ell s_n \bullet^\ell \mathsf{pop}_q^\ell$ *for some $n \geq 0$, $q \geq 0$.*

Proof: By induction on u, straightforwardly from Lemma 3.12 and Lemma 3.13. □

So, intuitively the Σ_H-normal terms are of the form:



consisting of a spine of $\mathtt{ev}$, Q and $\circ$ operators (on the left, not necessarily in the order shown). Following down the spine, we eventually meet some term w of the form 1^ℓ, $u \star^\ell v$ or $\lambda^\ell u$. Moreover, each $\mathtt{ev}$ or $\circ$ operator also has a stack ($u_{11}, \ldots, u_{1n_1}$ or $u_{21}, \ldots, u_{2n_2}$ in the figure) of terms of the same form, and reduction can only happen in the following ways: inside some u_{ij}, inside w, by reducing w if it is a (β) or (β^ℓ) redex. In the latter case, we shall have a $\succ'_{\lambda \mathtt{ev} Q}$-decreasing contraction.

Lemma 3.15 *For every $k \leq k'$, $T^k \subseteq T^{k'}$ and $S^k \subseteq S^{k'}$.*

Proof: By structural induction on $u \in T^k$ (resp. $u \in S^k$), we show that $u \in T^{k'}$ (resp. $u \in S^{k'}$).

If $u = \lambda^\ell v$, then $k \geq \ell + 1$ or $v \in T^k$. In the first case, $k' \geq \ell + 1$, so $u \in T^{k'}$; in the second case, by induction hypothesis $v \in T^{k'}$, so $v \in T^{k'}$.

If $u = v \star^\ell w$, then $k \geq \ell + 1$ or $v \in T^k$ or $w \in T^k$. In the first case $k' \geq \ell + 1$, so $u \in T^{k'}$; in the second case, $v \in T^{k'}$ by induction hypothesis, so $u \in T^{k'}$; similarly in the third case, using w instead of v.

If $u = 1^\ell$, then $k \geq \ell + 1$, so $k' \geq \ell + 1$ and therefore $u \in T^{k'}$. Similarly when $u = \uparrow^\ell$ is in S^k, then $u \in S^{k'}$. If $u = id^\ell$ is in S^k, then $k \geq \ell$, so $k' \geq \ell$ and therefore $u \in S^{k'}$.

If $u = v \bullet^\ell w$ is in S^k, then $k \geq \ell + 1$ or $v \in T^k$ or $w \in S^k$. In the first case, $k' \geq \ell + 1$; in the second case $v \in T^{k'}$ by induction hypothesis; and in the third case $w \in S^{k'}$ by induction hypothesis. In any case, $u \in S^{k'}$.

If $u = \Uparrow^\ell v$ is in S^k, then $k \geq \ell$, and if $k = \ell$ then $v \in S^k$. In particular, $k' \geq \ell$; and if $k' = \ell$, then since $k' \geq k \geq \ell$, we have in fact $k' = k' = \ell$, so it follows that $v \in S^k$, hence $v \in S^{k'}$ (since $k = k'$): so $u \in S^{k'}$.

If $u = v \circ^\ell_T w$, then either $k \geq \ell + 1$ and $v \in T^k$, or $k \leq \ell$ and $v \in T^\ell$ and $w \in S^k$. In the first case, $k' \geq \ell + 1$ and $v \in T^{k'}$ by induction hypothesis, so $u \in T^{k'}$. In the second case, if $k' \leq \ell$ then $v \in T^\ell$ (trivially) and $w \in S^{k'}$ (by induction hypothesis), so $u \in T^{k'}$; and if $k' \geq \ell + 1$, then $v \in T^{k'}$ (by induction hypothesis, using the facts that $v \in T^\ell$ and $\ell \leq k'$), so again $u \in T^{k'}$. Similarly when $u = v \circ^\ell_S w$ is in S^k, then $u \in S^{k'}$.

If $u = \mathtt{ev}^\ell_T vw$, then either $k \geq \ell$ and $v \in T^{k+1}$, or $k < \ell$ and $v \in T^\ell$ and $w \in S^k$. In the first case $k' \geq \ell$ and $v \in T^{k'+1}$ by induction hypothesis, so $u \in T^{k'}$. In the second case, if $k' < \ell$, then $v \in T^\ell$ (trivially) and $w \in S^{k'}$ (by induction hypothesis), so $u \in T^{k'}$; and if $k' \geq \ell$, then $v \in T^{k'+1}$ by induction hypothesis, using the facts that $v \in T^\ell$ and $\ell \leq k' + 1$. Similarly when $u = \mathtt{ev}^\ell_S vw$ is in S^k, then $u \in S^{k'}$.

If $u = Q^\ell_T v$, then $k \geq \ell + 1$ and $v \in T^{k-1}$. So $k' \geq \ell + 1$ and $v \in T^{k'-1}$ by induction hypothesis, and therefore $u \in T^{k'}$. Similarly when $u = Q^\ell_S v$ is in S^k, then $u \in S^{k'}$. □

Lemma 3.16 *Let u be a Σ-normal or Σ_H-normal typed $\lambda \mathtt{ev} Q^+$-term at level ℓ. If u is of sort T, then $u \in T^{\ell+1}$.*

Proof: Recall that the level of a $\lambda \mathtt{ev} Q^+$-term $f^\ell(\ldots)$ is ℓ unless f is $\mathtt{ev}_S$ or $\mathtt{ev}_T$, in which case it is $\ell - 1$.

By definition (see Definition 3.9), if $u = 1^\ell$, then $u \in T^{\ell+1}$; if $u = \lambda^\ell v$, then $u \in T^{\ell+1}$; and if $u = v \star^\ell w$, then $u \in T^{\ell+1}$.

If $u = v \circ^\ell_T w$, then $u \in T^{\ell+1}$ provided that $v \in T^{\ell+1}$. Now since u is a $\lambda \mathtt{ev} Q^+$-term, v must be of the form $f^k(v_1, \ldots, v_n)$ for some operator f (λ, $\star$, 1, $\circ_T$, $\mathtt{ev}_T$ or Q_T). Then $k \leq \ell$, otherwise some rule would apply ($(\lambda^k \circ^\ell)$, $(\star^k \circ^\ell)$, $(1^k \circ^\ell)$, $(\circ^k \circ^\ell)$, $(\mathtt{ev}^k \circ^\ell)$, or $(Q^k \circ^\ell)$ respectively). And the level of v is either k (if f is not $\mathtt{ev}_T$) or $k - 1$ (if f is $\mathtt{ev}_T$). By induction hypothesis, $v \in T^{k+1}$ or in T^k. Since $k \leq \ell$ and by Lemma 3.15, $v \in T^{\ell+1}$ in any case. So $u \in T^{\ell+1}$.

If $u = \mathtt{ev}_T^{\ell+1} vw$ (which is at level ℓ), then $u \in T^{\ell+1}$ provided that $v \in T^{\ell+2}$. Now since u is a $\lambda\mathtt{ev}Q^+$-term, v must be of the form $f^k(v_1, \ldots, v_n)$ for some operator f (λ, $\star$, 1, $\circ_T$, $\mathtt{ev}_T$ or Q_T). Then $k \leq \ell+1$, otherwise some rule would apply (($\mathtt{ev}^{\ell+1}\lambda^k$), ($\mathtt{ev}^{\ell+1}\star^k$), ($\mathtt{ev}^{\ell+1}1^k$), ($\mathtt{ev}^{\ell+1}\circ^k$), ($\mathtt{ev}^{\ell+1}\mathtt{ev}^k$), or ($\mathtt{ev}^{\ell+1}Q^k$) respectively). And the level of v is either k (if f is not $\mathtt{ev}_T$) or $k-1$ (if f is $\mathtt{ev}_T$). By induction hypothesis, $v \in T^{k+1}$ or in T^k. Since $k \leq \ell+1$ and by Lemma 3.15, $v \in T^{\ell+2}$ in any case. So $u \in T^{\ell+1}$.

If $u = Q_T^\ell v$, then $u \in T^{\ell+1}$ if and only if $v \in T^\ell$. Now since u is a $\lambda\mathtt{ev}Q^+$-term, v must be of the form $f^k(v_1, \ldots, v_n)$ for some operator f (λ, $\star$, 1, $\circ_T$, $\mathtt{ev}_T$ or Q_T). Then $k < \ell$, otherwise some rule would apply (($Q^\ell\lambda^{k+1}$), ($Q^\ell\star^{k+1}$), ($Q^\ell 1^{k+1}$), ($Q^\ell\circ^{k+1}$), ($Q^\ell\mathtt{ev}^{k+1}$), or ($Q^\ell Q^{k+1}$) respectively). And the level of v is either k (if f is not $\mathtt{ev}_T$) or $k-1$ (if f is $\mathtt{ev}_T$). By induction hypothesis, $v \in T^{k+1}$ or in T^k. Since $k < \ell$, $k \leq \ell - 1$ and by Lemma 3.15, $v \in T^\ell$ in any case. So $u \in T^{\ell+1}$. □

Lemma 3.17 *If $\mathcal{C}[u]$ is a typed Σ_H-normal $\lambda\mathtt{ev}Q^+$-term, where u is of sort T, then $\mathcal{C}$ is a syntactically safe$^+$ context.*

Proof: By structural induction on $\mathcal{C}$. If $\mathcal{C} = []$, this is clear. We then deal with the cases where $\mathcal{C}$ is a non-empty context such that $\mathcal{C}[u]$ is of sort T.

If $\mathcal{C} = \lambda^\ell \mathcal{C}_1$ or $\mathcal{C} = \mathcal{C}_1 \star^\ell w$ or $\mathcal{C} = w \star^\ell \mathcal{C}_1$, then this follows directly from the induction hypothesis. Similarly if $\mathcal{C} = \mathcal{C}_1 \circ_T^\ell w$, $\mathcal{C} = \mathtt{ev}_T^\ell \mathcal{C}_1 w$, or $\mathcal{C} = Q_T^\ell \mathcal{C}_1$.

If $\mathcal{C} = w \circ_T^\ell \mathcal{C}_1$, then since $\mathcal{C}[u]$ is Σ_H-normal, w is of the form $f^k(w_1, \ldots, w_n)$ with $k \leq \ell$ (as in the proof of Lemma 3.16) and f being λ, $\star$, 1, $\circ_T$, $\mathtt{ev}_T$ or Q_T. In fact we claim that $k < \ell$ as soon as $f \neq \mathtt{ev}_T$. Indeed, if $k = \ell$, and f is λ, $\star$, $\circ_T$ or Q_T, then $\mathcal{C}[u] = f^\ell(w_1, \ldots, w_n) \circ_T^\ell \mathcal{C}_1[u]$ would be reducible by (λ^ℓ), ($\star^\ell$), ($\circ^\ell$), or ($Q\circ^\ell$) respectively; and if $f = 1$, then notice that $\mathcal{C}_1[u]$ is by definition not T-free, so by Lemma 3.13 $\mathcal{C}_1[u] = u_1 \bullet^\ell u_2$ for some u_1, u_2 (and with the same ℓ, by typing), but then $\mathcal{C}[u] = 1^\ell \circ_T^\ell \mathcal{C}_1[u] = 1^\ell \circ_T^\ell (u_1 \bullet^\ell u_2)$ would be reducible by (1^ℓ). So if $f \neq \mathtt{ev}_T$, then $k < \ell$, therefore the level of w is $\leq \ell - 1$; and if $f = \mathtt{ev}_T$, then $k \leq \ell$ and therefore the level of $w = \mathtt{ev}_T^k w_1 w_2$ is again at most $\ell - 1$. By Lemma 3.16, $w \in T^{\mathcal{L}(w)+1}$, where $\mathcal{L}(w) \leq \ell - 1$ (recall that $\mathcal{L}(w)$ denotes the level of w). By Lemma 3.15, $w \in T^\ell$. By induction hypothesis, $\mathcal{C}_1$ is syntactically safe (i.e., in Sf_S^+). It follows that $\mathcal{C} = w \circ_T^\ell \mathcal{C}_1$ is in Sf_T^+.

If $\mathcal{C} = \mathtt{ev}_T^\ell w \mathcal{C}_1$, then since $\mathcal{C}[u]$ is Σ_H-normal, w is of the form $f^k(w_1, \ldots, w_n)$ with $k \leq \ell$ (as in the proof of Lemma 3.16) and f being λ, $\star$, 1, $\circ_T$, $\mathtt{ev}_T$ or Q_T. In fact we claim that $k < \ell$ as soon as $f \neq \mathtt{ev}_T$. Indeed, if $k = \ell$, and f is λ, $\star$, 1, $\circ_T$ or Q_T, then $\mathcal{C}[u] = \mathtt{ev}_T^\ell f^\ell(w_1, \ldots, w_n)\mathcal{C}_1[u]$ would be reducible by ($\mathtt{ev}\lambda^\ell$), ($\mathtt{ev}\star^\ell$), ($\mathtt{ev}1^\ell$), ($\mathtt{ev}\circ^\ell$), or ($\mathtt{ev}Q^\ell$) respectively. So if $f \neq \mathtt{ev}_T$, then $k < \ell$, therefore the level of w is $\leq \ell - 1$; and if $f = \mathtt{ev}_T$, then $k \leq \ell$ and therefore the level of $w = \mathtt{ev}_T^k w_1 w_2$ is again at most $\ell - 1$. By Lemma 3.16, $w \in T^{\mathcal{L}(w)+1}$, where $\mathcal{L}(w) \leq \ell - 1$. By Lemma 3.15, $w \in T^\ell$. By induction hypothesis, $\mathcal{C}_1$ is syntactically safe (i.e., in Sf_S^+). It follows that $\mathcal{C} = w \circ_T^\ell \mathcal{C}_1$ is in Sf_T^+.

Then, we deal with the cases when $\mathcal{C}[u]$ is of sort S. By Lemma 3.14 and the fact that $\mathcal{C}[u]$ is a $\lambda\mathtt{ev}Q^+$-term (not just a $\lambda\mathtt{ev}Q$-term), $\mathcal{C}[u]$ has the form $s_1 \bullet^\ell \ldots \bullet^\ell s_n \bullet^\ell \mathsf{pop}_q^\ell$. Since u has sort T, $\mathcal{C}$ must equal $s_1 \bullet^\ell \ldots \bullet^\ell s_{i-1} \bullet^\ell \mathcal{C}_1 \bullet^\ell s_{i+1} \bullet^\ell \ldots \bullet^\ell s_n \bullet^\ell \mathsf{pop}_q^\ell$ for some i, $1 \leq i \leq n$. By induction hypothesis $\mathcal{C}_1$ is syntactically safe$^+$, so an easy induction on i shows that $\mathcal{C}$ is in Sf_S^+ as well. □

Lemma 3.18 *For every $\lambda\mathtt{ev}Q$-terms u, v_1, ..., v_n, if u, v_1, ..., v_n are Σ-normal (resp. Σ_H-normal with u not of the form $\mathtt{ev}^1 xw$ where x is some variable), and v_1, ..., v_n are at level 0, then $(u`\rho)[v_1/x_1, \ldots, v_n/x_n]$ is Σ-normal (resp. Σ_H-normal) for any environment ρ.*

Proof: As for Lemma 4.4, Part IIIa (which stated the same thing with $\lambda\mathtt{ev}Q$ instead of Σ and $\lambda\mathtt{ev}Q_H$ instead of Σ_H). □

Theorem 3.19 *Every typed $\lambda\mathtt{ev}Q$-term u is Σ_H-RSN for the $\lambda\mathtt{ev}Q_H$ notion of reduction.*

Proof: We first show that: (A) every typed $\lambda\mathtt{ev}Q^+$ term u is Σ_H-RSN. Whenever $u \longrightarrow v$ in $\lambda\mathtt{ev}Q_H$, v is also a $\lambda\mathtt{ev}Q^+$-term and:

- either we have used a rule in Σ_H; in this case $u \succeq_{\lambda\mathtt{ev}Q} v$ by Lemma 3.7 and Lemma 3.3, and $u \succ_\lambda v$ or $u \succeq_\lambda v$ and $u \succ_{eq} v$ by Figures 6 and 7, Part IIIa;

- or we have used rule (β^ℓ); in this case, because we consider only Σ_H-eager rewrites, u must be Σ_H-normal, and by Lemma 3.17 the contracted (β^ℓ)-redex occurs under a syntactically safe$^+$ context. By Lemma 3.11, this context is safe$^+$: by definition, therefore, $u \succ_{\lambda\mathsf{ev}Q} v$.

So any reduction step in a Σ_H-eager rewrite decreases the terms in the lexicographic product of $\succ_{\lambda\mathsf{ev}Q}$, $\succ_\lambda$ and $\succ_{eq}$. Since this order is well-founded, all Σ_H-eager rewrites must be finite.

Now consider any Σ_H-eager rewrite (in $\lambda\mathsf{ev}Q_H$) starting from a typed $\lambda\mathsf{ev}Q$-term u (not just a $\lambda\mathsf{ev}Q_H$-term). For all environments ρ whose domain contains all free variables of u, $u`\rho$ is a typed $\lambda\mathsf{ev}Q^+$-term, and any Σ_H-eager rewrite sequence:

$$u = u_0 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_k \longrightarrow \ldots \tag{1}$$

induces a rewrite sequence:

$$u_0`\rho \longrightarrow^+ u_1`\rho \longrightarrow^+ \ldots \longrightarrow^+ u_k`\rho \longrightarrow^+ \ldots \tag{2}$$

by Theorem 3.18, Part II. Furthermore, whenever u_i contracts to u_{i+1} by (β) or (β^ℓ) in the rewrite sequence (1), then u_i is Σ_H-normal since this is a Σ_H-eager rewrite. By Lemma 3.18, $u_i`\rho$ is also Σ_H-normal. As i is arbitrary, the rewrite sequence (2) is also Σ_H-eager.

By property (A) above, the rewrite sequence (2) is finite. It follows that the rewrite sequence (1) is finite as well.

□

Observe that we have in fact proved that not only Σ_H-eager rewrites terminate, but reducing under any safe context is safe for termination as well, and this is strictly more general. A weak termination proof based on reducibility à la Girard-Tait would probably also show that Σ_H-eager rewrites terminate, but it would certainly be less clear how we could relax the constraint of Σ_H-eagerness.

3.4 The λ_{S4} and λ_{S4H}-calculi

It turns out that the technique of safe contexts allows us to give a new proof of (weak) termination of the typed λ_{S4} and λ_{S4H}-calculi. This proof is certainly not the simplest (compare with that of Part I), but it is reassuring that a technique like this is capable of reproving the termination of such a subsystem. Moreover, the proof will characterize a minimal subset of $\lambda\mathsf{ev}Q$-terms that we need to interpret cut elimination in S4, namely the set of stackless terms:

Lemma 3.20 *We say that a $\lambda\mathsf{ev}Q$-term u is* stackless *if and only if every subterm of u of sort S is of the form $()$ or pop_k^ℓ, $\ell \geq 1$, $k \geq 0$.*

If u is stackless, then so is $u`\rho$ for every environment ρ.

Proof: By structural induction on u. If u is a variable x, either x is not in the domain of ρ, then $u`\rho = Q_T^1 x$ is clearly stackless; or $x$ is in the domain of $\rho$, then $u`\rho = \mathsf{get}_k^1$ for some k, and this is also stackless. If $u = ()$, then $u`\rho = \mathsf{pop}_k^1$ for some $k$, hence is stackless as well. If $u = \mathsf{pop}_k^\ell$, then $u`\rho = \mathsf{pop}_k^{\ell+1}$, so u is stackless again. There are no other cases when u is a stack. All other cases, when u is of sort T, are trivial (see Figure 2, Part II). □

Lemma 3.21 *For any λ_{S4}-term u, $G(u)$ is stackless.*

Proof: By structural induction on u. If u is a variable x, then this is obvious. If $u = u_1 u_2$, then $G(u) = G(u_1)G(u_2)$ and this is again obvious, similarly when $u = \lambda x \cdot u_1$ (since $G(u) = \lambda x \cdot G(u_1)$) or when $u = \mathsf{unbox}\ u_1$ (since $G(u) = \mathsf{ev}_T^1 G(u_1)()$, where the only additional stack in $G(u)$ compared to $G(u_1)$ is $()$). When $u = \mathsf{box}\ u_1\ \mathsf{with}\ v_1, \ldots, v_n\ \mathsf{for}\ x_1, \ldots, x_n$, $G(u) = (G(u_1)`[])[G(v_1)/x_1, \ldots, G(v_n)/x_n]$. By Lemma 3.20 and by induction hypothesis, $G(u_1)`[]$ is stackless; since by induction hypothesis, $G(v_1)$, …, $G(v_n)$ are stackless as well, $G(u)$ is stackless. □

Lemma 3.22 *Every context in a stackless term is (syntactically) safe.*

Proof: Obvious, by Definition 3.10. $\square$

It follows a weaker form of Theorem 5.5, Part I:

Theorem 3.23 *The G-translations of reductions in the typed λ_{S4} and λ_{S4H}-calculi (see Theorem 3.29 and Theorem 4.11, Part II) terminate. It follows that all* $(\mathsf{gc}), (\mathsf{ctract})$*-eager reductions in the typed λ_{S4} and λ_{S4H}-calculi terminate.*

Proof: If $u \longrightarrow v$ in λ_{S4} or λ_{S4H} (defining a reduction $G(u) \longrightarrow^+ G(v)$ in $\lambda\mathsf{ev}Q$ or $\lambda\mathsf{ev}Q_H$), then we claim that $G(u) \succ'_{\lambda\mathsf{ev}Q} G(v)$, or $G(u) \succeq'_{\lambda\mathsf{ev}Q} G(v)$ and $G(u) \succ_\lambda G(v)$. Indeed, the only cases where we apply rule (β) or (β^ℓ) of $\lambda\mathsf{ev}Q$ is to reduce $G(u)$ to $G(v)$ when v was gotten from u by rule (β) of λ_{S4}. Then $G(u) \overset{\beta,\beta^\ell}{\longrightarrow}{}^* u' \overset{\Sigma}{\longrightarrow}{}^* G(v)$, all contracted (β) and (β^ℓ)-redex occur at disjoint positions (i.e., none is above another), so by Lemma 3.22, they are done under safe contexts. So the claim is proved. Therefore, each step in the G-translation of a reduction in the typed λ_{S4} and λ_{S4H}-calculi makes the term decrease in the well-founded lexicographic product of $\succ'_{\lambda\mathsf{ev}Q}$ and $\succ_\lambda$. This proves the first part of the Theorem.

Now, we show that $(\mathsf{gc}), (\mathsf{ctract})$-eager rewrites in λ_{S4} or λ_{S4H} are terminating. Any step of the form $u \overset{R}{\longrightarrow} v \overset{(\mathsf{gc}),(\mathsf{ctract})}{\longrightarrow}{}^* w$, where u and w are $(\mathsf{gc}), (\mathsf{ctract})$-normal and R is some rule of λ_{S4H} other than (gc) and (ctract), then $G(u) \longrightarrow^* G(v) = G(w)$ in $\lambda\mathsf{ev}Q_H$ (by Theorem 3.29 and Theorem 4.11, Part II, and $G(v) = G(w)$ by Lemma 3.10 and Lemma 3.11, Part II). Now by Lemma 4.3, Part IIIa, G is injective l from $\lambda_{S4}^{\approx}$ to $\lambda\mathsf{ev}Q$, so since $u \neq v$, $G(u) \neq G(v)$ and in particular $G(u) \longrightarrow^+ G(v)$ in $\lambda\mathsf{ev}Q_H$. So every $(\mathsf{gc}), (\mathsf{ctract})$-eager rewrite terminates, by the first part of the Theorem. $\square$

3.5 The $\lambda\mathrm{ev}Q$-Calculus

The case of the $\lambda\mathsf{ev}Q$-calculus is more complicated, because we do not have the rules of group (H). In fact, we have not succeeded in showing that it terminates weakly. Let's see how we would try to show this by the same techniques as above.

Instead of using S_H to normalize stacks, first, we must use S (see Definition 3.11). But the S-normal stacks are more complicated (see Lemma 3.25 below), and in fact they are so complicated that some contexts with a hole of sort T are necessarily unsafe. Consider for example the context $\mathcal{C} = \Uparrow^\ell \uparrow^\ell \circ^\ell \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} ([] \bullet^{\ell+1} v))w$, where, say, $v = id^{\ell+1}$ and $w = id^\ell$. Then, whenever $u : \overline{\varsigma^{\ell-1} \overset{\square}{\Rightarrow}} \tau \times \varsigma \overset{\square}{\Rightarrow} \varsigma \overset{\square}{\Rightarrow} \tau'$, $\mathcal{C}[u]$ has type $\overline{\varsigma^{\ell-1} \overset{\square}{\Rightarrow}} \tau \times \varsigma \overset{\square}{\Rightarrow} \tau \times \varsigma$, so this is typable. Moreover, whenever u is Σ-normal, so is $\mathcal{C}[u]$. Unfortunately:

$$
\begin{aligned}
&[\![\mathcal{C}[u]]\!]' s_1 \ldots s_\ell = [\![\Uparrow^\ell \uparrow^\ell \circ^\ell \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} (u \bullet^{\ell+1} v))w]\!]' s_1 \ldots s_\ell \\
&= [\![\Uparrow^\ell \uparrow^\ell]\!]' s_1 \ldots s_{\ell-1}([\![\mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} (u \bullet^{\ell+1} v))w]\!]' s_1 \ldots s_\ell) \\
&= [\![\Uparrow^\ell \uparrow^\ell]\!]' s_1 \ldots s_{\ell-1}([\![\Uparrow^{\ell+1} (u \bullet^{\ell+1} v)]\!]' s_1 \ldots s_\ell([\![w]\!]' s_1 \ldots s_\ell)) \\
&= [\![\Uparrow^\ell \uparrow^\ell]\!]' s_1 \ldots s_{\ell-1}\langle \mathrm{fst}\ [\![w]\!]' s_1 \ldots s_\ell, [\![u \bullet^{\ell+1} v]\!]' s_1 \ldots s_\ell(\mathrm{snd}\ [\![w]\!]' s_1 \ldots s_\ell)\rangle \\
&= \langle \mathrm{fst}\ [\![w]\!]' s_1 \ldots s_\ell, [\![\uparrow^\ell]\!]' s_1 \ldots s_{\ell-1}([\![u \bullet^{\ell+1} v]\!]' s_1 \ldots s_\ell(\mathrm{snd}\ [\![w]\!]' s_1 \ldots s_\ell))\rangle \\
&= \langle \mathrm{fst}\ [\![w]\!]' s_1 \ldots s_\ell, \mathrm{snd}\ ([\![u \bullet^{\ell+1} v]\!]' s_1 \ldots s_\ell(\mathrm{snd}\ [\![w]\!]' s_1 \ldots s_\ell))\rangle \\
&= \langle \mathrm{fst}\ [\![w]\!]' s_1 \ldots s_\ell, [\![v]\!]' s_1 \ldots s_\ell(\mathrm{snd}\ [\![w]\!]' s_1 \ldots s_\ell)\rangle
\end{aligned}
$$

does not depend on $[\![u]\!]'$. That is, $\mathcal{C}$ is unsafe.

Let's determine the general form of Σ-normal stacks:

Lemma 3.24 *Let u be a typed $\lambda\mathsf{ev}Q$-term such that:*

(1) *u has type $\overline{\varsigma^\ell \overset{\square}{\Rightarrow}} \varsigma_{\ell+1}$ for some $\ell \geq 0$, and some stack types $\varsigma_1, \ldots, \varsigma_\ell, \varsigma_{\ell+1}$;*

(2) *u is S-normal;*

(3) *and u is* not *T-free.*

Then (4): u is of one of the following forms:

(i) $u_1 \bullet^\ell u_2$,

(ii) $\Uparrow^\ell u_1$,

(iii) $\Uparrow^\ell u_1 \circ_S^\ell u_2$,

(iv) *or* $\mathtt{ev}_S^{\ell+1}(\Uparrow^{\ell+1} u_1)u_2$.

where $\ell \geq 0$, *and* *(i)* *is the only possible case when* $\ell = 0$.

Proof: By structural induction on u:

- $u = ()$: impossible by $(3)[u]$;
- $u = s \bullet t$: claim proved if $\ell = 0$, impossible otherwise;
- $u = \uparrow s$: by $(1)[u]$, $\ell = 0$; also, s obeys (1), (2) and (3) so $(4)[s]$ holds, and in fact $(i)[s]$. Then u would be reducible by rule $(\uparrow)$, which contradicts $(2)[u]$
- $u = s \circ_S^n t$: by $(1)[u]$, we must have (a) $n \leq \ell$. If $\ell = 0$, then, this case cannot happen.

  Assume that $\ell \geq 1$. Then, s also obeys (1) with some type $\varsigma_1 \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{n-1} \overset{\Box}{\Rightarrow} \varsigma'_n \overset{\Box}{\Rightarrow} \varsigma_{n+1} \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{\ell+1}$, and obeys (2). We then have two cases, according to whether s is or is not T-free.

  If s is T-free, by Lemma 3.12, either $s = ()$ or $s = \mathsf{pop}_k^\ell$ for some $k \geq 0$. By (a) $\ell \geq n \geq 1$, so $s = \mathsf{pop}_k^\ell$. If $n < \ell$, then u is reducible by $(id^\ell \circ^n)$ (if $k = 0$), $(\uparrow^\ell \circ^n)$ (if $k = 1$) or $(\circ^\ell \circ^n)$ (if $k \geq 2$); so by $(2)[u]$ $n \geq \ell$, and by (a) in fact $n = \ell$. By $(3)[u]$, t cannot be T-free, so t obeys (1), (2) and (3), and by induction hypothesis t obeys (i), (ii), (iii) or (iv). In any case, k cannot be 0, by $(1)[u]$ (typing). If $k = 1$, then u is reducible by rule $(\uparrow^\ell)$ (if $(i)(t)$), or $(\uparrow\Uparrow^\ell)$ (if $(ii)(t)$), or $(\uparrow\Uparrow \circ^\ell)$ (if $(iii)(t)$), or $(\uparrow \mathtt{ev} \Uparrow^\ell)$ (if $(iv)(t)$). And if $k \geq 2$, then u is reducible by rule $(\circ^\ell)$; so in any case we get a contradiction with $(2)[u]$.

  It follows that s is not T-free, i.e. $(3)[s]$ holds: by induction hypothesis, one of the statements $(i)[s]$, $(ii)[s]$, $(iii)[s]$ and $(iv)[s]$ holds. If $(i)[s]$ holds, then u is reducible by rule $(\bullet^\ell \circ^n)$ if $n < \ell$, and by $(\bullet^\ell)$ if $n = \ell$; by (a) there is no other possibility, so $(i)[s]$ does not hold. If $(ii)[s]$ holds, then either $n < \ell$, so that $(\Uparrow^\ell \circ^n)$ applies, contradicting $(2)[u]$; or $n = \ell$, and then $(iii)[u]$ holds. If $(iii)[s]$ holds, then u is reducible by $(\circ^\ell \circ^n)$ if $n < \ell$, and by $(\circ^\ell)$ if $n = \ell$; by a there is no other possibility, so $(iii)[s]$ does not hold. If $(iv)[s]$ holds, then u is reducible by $(\mathtt{ev}_S^{\ell+1} \circ_S^n)$ since by (a) $n < \ell + 1$, contradicting (2) again.
- $u = id^n$: impossible by (3);
- $u = s \bullet^n t$: by $(1)[u]$, $n = \ell$, then $(i)[u]$ holds;
- $u = \uparrow^n$: impossible by (3);
- $u = \Uparrow^n s$: by $(1)[u]$, $n = \ell$, then $(ii)[u]$ holds if $\ell \geq 1$; if $\ell = 0$, this case is impossible;
- $u = \mathtt{ev}_S^n st$: by $(1)[u]$, we must have (a) $n \leq \ell + 1$. Moreover, s must have a type of the form $\varsigma_1 \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{n-1} \overset{\Box}{\Rightarrow} \varsigma' \overset{\Box}{\Rightarrow} \varsigma_n \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{\ell+1}$.

  If s if T-free, then by Lemma 3.12 and typing considerations, $s = \mathsf{pop}_k^{\ell+1}$ for some $k \geq 0$. If $n < \ell + 1$, then u is reducible by $(\mathtt{ev}^n id^{\ell+1})$ if $k = 0$, by $(\mathtt{ev}^n \uparrow^{\ell+1})$ if $k = 1$, and by $(\mathtt{ev}^n \circ^{\ell+1})$ if $k \geq 2$. If $n = \ell + 1$, then u is reducible by $(\mathtt{ev} id^{\ell+1})$ if $k = 0$, by $(\mathtt{ev} \uparrow^{\ell+1})$, and by $(\mathtt{ev} \circ^{\ell+1})$ if $k \geq 2$. By (a), there is no other case; by $(2)[u]$, all these cases are impossible.

  So s is not T-free, i.e. obeys (3). Since s also obeys (1) and (2), by induction hypothesis one of $(i)[s]$, $(ii)[s]$, $(iii)[s]$ and $(iv)[s]$ holds. If $(i)[s]$ holds, then $s = s_1 \bullet^{\ell+1} s_2$ for some terms s_1 and s_2, so u is reducible by rule $(\mathtt{ev}^n \bullet^{\ell+1})$ if $n < \ell + 1$, and by $(\mathtt{ev} \bullet^{\ell+1})$ if $n = \ell + 1$; by (a), there is no other possibility, so $(i)[s]$ cannot hold.

  If $(ii)[s]$ holds, then $s = \Uparrow^{\ell+1} s_1$ for some s_1; if $n < \ell + 1$, then rule $(\mathtt{ev}^n \Uparrow^{\ell+1})$ applies, contradicting $(2)[u]$; if $n = \ell + 1$, then $(iv)[u]$ holds; by (a) there is no other possibility. Moreover, if $\ell = 0$, then by induction hypothesis on t, $t = t_1 \bullet t_2$ for some terms t_1 and t_2 (the case $t = ()$ is excluded by typing), so $u = \mathtt{ev}^1(\Uparrow^1 s_1)(t_1 \bullet t_2)$, which is reducible by rule $(\mathtt{ev} \Uparrow^1)$, contradicting $(2)[u]$: $(iv)[u]$ therefore holds only if $\ell \geq 1$.

If $(iii)[s]$ holds, then $s = \Uparrow^{\ell+1} s_1 \circ_S^{\ell+1} s_2$ for some s_1 and s_2; by (a), rule $(\mathsf{ev}_S^n \circ^{\ell+1})$ applies if $n < \ell + 1$, otherwise rule $(\mathsf{ev}_S \circ^{\ell+1})$ applies; in any case this contradicts $(2)[u]$, so $(iii)[s]$ cannot hold. If $(iv)[s]$ holds, then $s = \mathsf{ev}^{\ell+2}(\Uparrow^{\ell+2} s_1)s_2$ for some s_1 and s_2; by (a), rule $(\mathsf{ev}_S^n \mathsf{ev}_S^{\ell+2})$ applies, contradicting $(2)[u]$.

- $u = Q_S^n t$: by $(1)[u]$, we must have (a) $n \leq \ell - 1$. Moreover, t must have a type of the form $\varsigma_1 \overset{\Box}{\Rightarrow} \ldots \overset{\Box}{\Rightarrow} \varsigma_{n-1} \overset{\Box}{\Rightarrow} \varsigma_n \overset{\Box}{\Rightarrow} \ldots \varsigma_\ell \overset{\Box}{\Rightarrow} \varsigma_{\ell+1}$. Since t obeys (1), (2) and (3), by induction hypothesis one of $(i)[t]$, $(ii)[t]$, $(iii)[t]$ and $(iv)[t]$ holds. If $(i)[t]$ holds, then $t = t_1 \bullet^{\ell-1} t_2$ for some terms t_1 and t_2; but then u is reducible by rule $(Q^n \bullet^\ell)$ by (a), which contradicts $(2)[u]$. If $(ii)[t]$ holds, then u is reducible by $(Q_S^n \Uparrow^{\ell-1})$. If $(iii)[t]$ holds, then u is reducible by $(Q_S^n \circ^{\ell-1})$. And if $(iv)[t]$ holds, then u is reducible by $(Q_S^n \mathsf{ev}_S^{\ell-1})$. In any case, $(2)[u]$ is contradicted; so u cannot be of the form $Q_S^n t$.

□

The general form of typed Σ-normal stacks is then given by:

Lemma 3.25 *The typed S-normal stacks u of types of the form $\overline{\varsigma^\ell \overset{\Box}{\Rightarrow}} \varsigma$, $\ell \geq 0$, have the following form:*

- *$s_1 \bullet \ldots \bullet s_n \bullet ()$ for some $n \geq 0$, if $\ell = 0$;*
- *$s_1 \bullet^\ell \ldots \bullet^\ell s_n \bullet^\ell v$ for some $n \geq 0$, and with v is a term of the form $\mathcal{C}_1[\ldots[\mathcal{C}_p[w]]\ldots]$, where $p \geq 0$, $q \geq 0$, $w = \mathsf{pop}_q^\ell$ with $q \geq 0$ or $w = \Uparrow^\ell s_{n+p+1}$ for some s_{n+p+1}, and the contexts $\mathcal{C}_i$, $1 \leq i \leq p$, alternate between the forms $\Uparrow^\ell s_{n+i} \circ_S^\ell []$ and $\mathsf{ev}_S^{\ell+1}(\Uparrow^{\ell+1} s_{n+i})[]$. Moreover, if $p \geq 1$ and $\mathcal{C}_p = \Uparrow^\ell s_{n+p} \circ_S^\ell []$, then cannot be a lift, i.e. $w = \mathsf{pop}_q^\ell$.*

Proof: By induction on u. If $\ell = 0$, the result follows straightforwardly from Lemma 3.12 and Lemma 3.24.

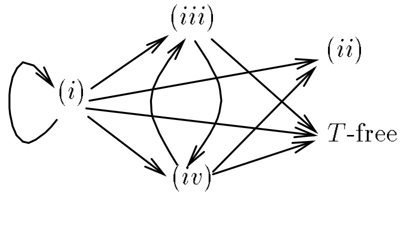
Let therefore $\ell \geq 1$. If u is T-free, then by Lemma 3.12 $u = \mathsf{pop}_q^\ell$ for some $q \geq 0$. Otherwise u has one of the forms (i), (ii), (iii), or (iv) of Lemma 3.24. When u is not of the form $\Uparrow^\ell s_1$ (case (ii)), we examine the possible forms for s_2:

- (i): $u = s_1 \bullet^\ell s_2$, then s_2 may be any stack term of the right type.
- (iii): $u = \Uparrow^\ell s_1 \circ^\ell s_2$. Then s_2 cannot be of the form (i) (by rule $(\Uparrow \bullet^\ell)$), or (ii) (by rule $(\Uparrow\Uparrow^\ell)$) or (iii) (by rule $(\Uparrow\Uparrow \circ^\ell)$). So s_2 must be of the form (iv).
- (iv): $u = \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} s_1)s_2$. Then s_2 cannot be of the form (i) (by rule $(\mathsf{ev} \Uparrow^{\ell+1})$) or (iv) (by rule $(\mathsf{ev} \Uparrow\Uparrow^{\ell+1})$), so only (ii) and (iii) remain. Observe also that s_2 must have a type of the form $\overline{\varsigma'^\ell \overset{\Box}{\Rightarrow}} \varsigma'$ with the same ℓ, so that the induction hypothesis applies.

The set of S-normal stacks having a type of the form $\overline{\varsigma^\ell \overset{\Box}{\Rightarrow}} \varsigma$ with given $\ell \geq 1$ is then (included in the set) described by the non-terminal U^ℓ of the following grammar:

$$
\begin{array}{ll}
U^\ell & ::= \text{Type-}(i)^\ell \mid \text{Type-}(iii)^\ell \mid \text{Type-}(iv)^\ell \mid Fin^\ell \\
\text{Type-}(i)^\ell & ::= T \bullet^\ell U^\ell \\
\text{Type-}(iii)^\ell & ::= \Uparrow^\ell U^\ell \circ_S^\ell \text{Type-}(iv)^\ell \mid \Uparrow^\ell U^\ell \circ_S^\ell T\text{-free}^\ell \\
\text{Type-}(iv)^\ell & ::= \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} U^{\ell+1})\text{Type-}(iii)^\ell \mid \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} U^{\ell+1})Fin^\ell \\
Fin^\ell & ::= \text{Type-}(ii)^\ell \mid T\text{-free}^\ell \\
\text{Type-}(ii)^\ell & ::= \Uparrow^\ell U^\ell \\
T\text{-free}^\ell & ::= \mathsf{pop}_k^\ell \quad (k \geq 0)
\end{array}
$$

where T is the sort of element terms (of the right types). Informally, this is represented by the following automaton, where all points are starting states, (ii) and T-free are the two accepting states, and transitions denote the choice of an s_2 term:



The claim then follows. Observe that yet another way to put it is to say that u must have the form:

$$\begin{array}{rl} u = & s_1 \bullet^\ell \ldots \bullet^\ell s_n \bullet^\ell \\ & \Bigg(\ldots \Uparrow^\ell s_{n+i} \circ^\ell \Bigg(\mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} s_{n+i+1}) \Bigg(\Uparrow^\ell s_{n+i+2} \circ^\ell \ldots \\ & \qquad \ldots \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} s_{n+p'}) \left(\left\{ \begin{array}{l} \mathsf{pop}_q^\ell \\ \text{or } \Uparrow^\ell s_{n+p'+1} \\ \text{or } \Uparrow^\ell s_{n+p'+1} \circ_S^\ell \mathsf{pop}_q^\ell \end{array} \right. \right) \Bigg) \Bigg) \ldots \Bigg) \end{array}$$

(where $p = p'$ in the first case, and $p = p' + 1$ in the last two cases). □

Observe that, although some S-normal stacks produce unsafe contexts, reduction is relatively well behaved:

Lemma 3.26 *Let u be an S-normal typed stack other than id^ℓ for any $\ell \geq 1$. Then u does not rewrite to any id^ℓ, $\ell \geq 1$.*

More precisely, if u is T-free, then u is normal; and otherwise, if u is of type (i) (resp. (ii), (iii), (iv): see Lemma 3.24), then it rewrites to type (i) (resp. (ii), (iii), (iv)) terms only:

- (i) *if $u = u_1 \bullet^\ell u_2 \longrightarrow^* v$, then $v = v_1 \bullet^\ell v_2$ with $u_1 \longrightarrow^* v_1$ and $u_2 \longrightarrow^* v_2$;*
- (ii) *if $u = \Uparrow^\ell u_1 \longrightarrow^* v$, then $v = \Uparrow^\ell v_1$ with $u_1 \longrightarrow^* v_1$;*
- (iii) *if $u = \Uparrow^\ell u_1 \circ_S^\ell u_2$, then $v = \Uparrow^\ell v_1 \circ_S^\ell v_2$ with $u_1 \longrightarrow^* v_1$ and $u_2 \longrightarrow^* v_2$;*
- (iv) *if $u = \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} u_1)u_2$, then $v = \mathsf{ev}^{\ell+1}(\Uparrow^{\ell+1} v_1)v_2$ with $u_1 \longrightarrow^* v_1$ and $u_2 \longrightarrow^* v_2$.*

Proof: By structural induction on u, using Lemma 3.12 and Lemma 3.24. If u is T-free, by Lemma 3.12 $u = ()$ or $u = \mathsf{pop}_k^\ell$ for some $\ell \geq 1$, $k \geq 0$. In any case u is normal, so that the claim is obvious.

Consider therefore the cases when u is not T-free. By Lemma 3.24, we have four cases:

- (i) $u = u_1 \bullet^\ell u_2$ for some $\ell \geq 0$. Then any sequence of rewrite steps starting from u must occur entirely inside u_1 and u_2, since no term of the form $s_1 \bullet^\ell s_2$ is a redex in $\lambda\mathsf{ev}Q$. In particular u rewrites only to type (i) terms, and cannot rewrite to any $id^{\ell'}$, $\ell' \geq 1$.
- (ii) $u = \Uparrow^\ell u_1$. Consider a rewrite sequence starting from u. Either it occurs entirely inside u_1, and the claim is proved: only terms of type (ii), and in particular no term of the form $id^{\ell'}$ occurs. Or at some point, some step rewrites at the top. This means that we have:

  $$u = \Uparrow^\ell u_1 \longrightarrow \Uparrow^\ell u_2 \longrightarrow \ldots \longrightarrow \Uparrow^\ell u_k \longrightarrow v$$

  where each $\longrightarrow$ step is either an S-normalization step (if some S-redex is present) or else some other step in $\lambda\mathsf{ev}Q$. We may choose k to be minimal such that $\Uparrow^\ell u_k$ is the redex that we reduce at step k. But the only rule that we can apply on $\Uparrow^\ell u_k$ that rewrites at the top is $(\Uparrow id^\ell)$, so u_k must equal id^ℓ. By minimality of the sequence, u_1 must rewrite to $u_k = id^\ell$. Therefore, by induction hypothesis $u_1 = \Uparrow^\ell id^\ell$. However, this would imply that u_1 is not S-normal, a contradiction.
- (iii) $u = \Uparrow^\ell u_1' \circ_S^\ell u_1''$. By Lemma 3.25, u_1'' is of type (iv) or is T-free. Consider a minimal rewrite sequence:

  $$u = \Uparrow^\ell u_1' \circ_S^\ell u_1'' \longrightarrow \Uparrow^\ell u_2' \circ_S^\ell u_2'' \longrightarrow \ldots \longrightarrow \Uparrow^\ell u_k' \circ_S^\ell u_k'' \longrightarrow v$$

  where again each arrow reduces an S-redex as soon as there is one, and the redex in the kth step is $\Uparrow^\ell u_k'$ or $\Uparrow^\ell u_k' \circ_S^\ell u_k''$. (Since the sequence is minimal, u_1' rewrites to u_k' and u_1'' rewrites to u_k''.) If the redex is $\Uparrow^\ell u_k'$, then as above the applied rule can only be $(\Uparrow id^\ell)$, so $u_k' = id^\ell$, so $u_1' = id^\ell$ by induction hypothesis; this implies that u would be reducible by rule $(\Uparrow id^\ell)$, a contradiction. So the redex is $\Uparrow^\ell u_k' \circ_S^\ell u_k''$. However, u_1'' is of type (iv) or T-free, so by induction hypothesis u_k'' is also of type (iv) or T-free, from which it appears that no rule applies to $\Uparrow^\ell u_k' \circ_S^\ell u_k''$, a contradiction again. So reduction must actually tale place entirely inside u_1' and u_1'', and every reduct of u is of type (iii).

- (iv) the argument is similar to case (iii).

□

Lemma 3.27 *If u_1, u_2 are S-SN, then for any $\ell \geq 1$:*

(i) *$u_1 \bullet u_2$ and $u_1 \bullet^\ell u_2$ are S-SN;*

(ii) *$\Uparrow^\ell u_1$, if S-normal, is S-SN;*

(iii) *$\Uparrow^\ell u_1 \circ^\ell u_2$, if S-normal, is S-SN;*

(iv) *$\mathtt{ev}^{\ell+1}(\Uparrow^{\ell+1} u_1)u_2$, if S-normal, is S-SN.*

Proof: In any case, by Lemma 3.26, any rewrite sequence starting from u must take place entirely inside u_1 and u_2. If the sequence starting from u is infinite, then so must be one of the induced sequences starting from u_1 and from u_2, contradicting the fact that they are S-SN. □

Definition 3.12 *Let A be any set of terms in N.*

The set $D(A)$ is the smallest set of S-normal typed terms of the form described in Lemma 3.25, with $s_1 \in A, \ldots, s_n \in A, s_{n+1} \in D(A), \ldots, s_{n+p} \in D(A)$.

Lemma 3.28 *$D(S\text{-}SN) \subseteq S\text{-}SN$.*

Proof: Easy structural induction on $u \in D(S\text{-SN})$, using Lemma 3.27. □

Definition 3.13 *Define the following rewrite relation $\rightsquigarrow$ on Σ-normal terms: $u \rightsquigarrow v$ if and only if $u \xrightarrow{\beta,\beta^\ell} u' \xrightarrow{\Sigma}{}^* v$, that is if v is obtained from u by contracting a (β) or (β^ℓ)-redex, $\ell \geq 1$, yielding u', and then taking the Σ-normal form v of u'.*

Any infinite Σ-eager rewrite sequence starting from a given term u is of the form:

$$u \xrightarrow{\Sigma}{}^* u_0 \longrightarrow v_0 \xrightarrow{\Sigma}{}^* u_1 \longrightarrow v_1 \xrightarrow{\Sigma}{}^* \ldots \xrightarrow{\Sigma}{}^* u_i \longrightarrow v_i \xrightarrow{\Sigma}{}^* \ldots$$

where each u_i is Σ-normal, and rewrites to v_i by (β) or (β^ℓ), $\ell \geq 1$. So we have:

$$u_0 \rightsquigarrow u_1 \rightsquigarrow \ldots \rightsquigarrow u_i \rightsquigarrow \ldots$$

and therefore u is $\rightsquigarrow$-terminating if and only if it is Σ-RSN-terminating.

In particular, we have:

Theorem 3.29 *A Σ-normal typed $\lambda\mathtt{ev}Q$-term u is Σ-RSN for the $\lambda\mathtt{ev}Q$ notion of reduction if and only if all innermost $\rightsquigarrow$-rewrites starting from i terminate.*

Therefore, the typed $\lambda\mathtt{ev}Q$-calculus is Σ-RSN if and only if $\rightsquigarrow$ is innermost-terminating.

Proof: Recall now that a locally confluent orthogonal term rewriting system is terminating if and only if it is innermost-terminating, a result of Gramlich [Gra92]. More generally, a locally confluent overlaying term rewriting system is terminating if and only if innermost rewriting always leads to a normal form ([DH95], Proposition 1). (Recall that a term rewriting system is *overlaying* if and only if for any two rules $l_1 \to r_1$ and $l_2 \to r_2$, the only critical pairs are at the top, namely when l_1 and l_2 unify. An *innermost* rewrite is the contracted redexes are minimal for the subterm ordering, i.e. the contracted redexes have no redex as proper subterms. Strictly speaking, the theorem above is for untyped term rewriting systems, but the proof in [DH95] extends to typed systems without difficulty.)

Now, any infinite Σ-eager rewrite sequence is of the form:

$$u \xrightarrow{\Sigma}{}^* u_0 \longrightarrow v_0 \xrightarrow{\Sigma}{}^* u_1 \longrightarrow v_1 \xrightarrow{\Sigma}{}^* \ldots \xrightarrow{\Sigma}{}^* u_i \longrightarrow v_i \xrightarrow{\Sigma}{}^* \ldots$$

where each u_i is Σ-normal, and rewrites to v_i by (β) or (β^ℓ), $\ell \geq 1$. So we have:

$$u_0 \rightsquigarrow u_1 \rightsquigarrow \ldots \rightsquigarrow u_i \rightsquigarrow \ldots$$

and by Gramlich's Theorem, there must be an infinite innermost rewrite sequence starting from u_0, since $\leadsto$ is clearly orthogonal.

In particular, if u is Σ-normal (i.e. $u = u_0$), then there is an infinite innermost $\leadsto$-rewrite starting from u. The converse holds: whenever we have an infinite innermost $\leadsto$-rewrite starting from u and u is Σ-normal (i.e. $u = u_0$), then this is also an infinite Σ-eager rewrite sequence. $\Box$

This reduces the cases to investigate to show weak termination or to find a counter-example. On the other hand, it is simple to produce a calculus $\lambda\mathtt{ev}Q'$ that does not have this defect, namely $\lambda\mathtt{ev}Q$ plus the rules of group (H) except $(\eta\mathtt{ev}^\ell)$. The latter indeed terminates weakly, is confluent in the typed case —but not in the untyped case—, and interprets the λ_{S4} notion of reduction but not the one of $\lambda_{\mathrm{S4}H}$; indeed, all proofs are as in the $\lambda\mathtt{ev}Q_H$ case (check it!).

A final note: all techniques developed here and in previous parts adapt readily to higher-order type systems, notably to variants of System F, where we add the following typing rules:

$$\frac{\Gamma \vdash u : \tau}{\Gamma \vdash \forall\alpha \cdot \tau}\ (\alpha \text{ not free in } \Gamma) \qquad \frac{\Gamma \vdash u : \forall\alpha \cdot \tau}{\Gamma \vdash u : \tau[\tau'/\alpha]}$$

where α is taken a set of so-called type variables, which we can see as a subset of the base types; and, in the case of $\lambda\mathtt{ev}Q$ and $\lambda\mathtt{ev}Q_H$ the following extra rules as well:

$$\frac{\Gamma \vdash u : \overline{\varsigma^\ell \overset{\Box}{\Rightarrow}}\tau}{\Gamma \vdash \overline{\varsigma^\ell \overset{\Box}{\Rightarrow}}\forall\alpha \cdot \tau}\ (\alpha \text{ not free in } \Gamma, \varsigma_1, \ldots, \varsigma_\ell) \qquad \frac{\Gamma \vdash u : \overline{\varsigma^\ell \overset{\Box}{\Rightarrow}}\forall\alpha \cdot \tau}{\Gamma \vdash u : \overline{\varsigma^\ell \overset{\Box}{\Rightarrow}}\tau[\tau'/\alpha]}$$

The only thing we have to do is, because terms may not have a unique type, is to index the $[\![\_]\!]$ and $[\![\_]\!]'$ interpretations by the types of the terms in argument. However, there remains a major difficulty, in that subject reduction does *not* hold for $\lambda\mathtt{ev}Q'$ or $\lambda\mathtt{ev}Q_H$ in this system, and this was a necessary condition for our techniques to work.

Indeed, consider $1^\ell \bullet^\ell\uparrow^\ell$: one type for this term is $\tau \times \varsigma \overset{\Box}{\Rightarrow} \tau \times \varsigma$, but another one is $\tau \times \varsigma \overset{\Box}{\Rightarrow} (\forall\alpha \cdot \tau) \times \varsigma$, where α is not free in $\tau \times \sigma$. Now $1^\ell \bullet^\ell\uparrow^\ell$ rewrites to id^ℓ, which does not admit $\tau \times \varsigma \overset{\Box}{\Rightarrow} (\forall\alpha \cdot \tau) \times \varsigma$ as a type. One solution to this problem is to restrict ourselves to a weaker variant of λ-calculus with explicit substitutions [MH97b, MH97a]. Another possibility is to use a more perspicuous type system, such as a variant of Raffalli's F^η [Raf96], which was designed to overcome a similar problem with the η rule in system F.

References

[DH95] Nachum Dershowitz and Charles Hoot. Natural termination. *Theoretical Computer Science*, 142(2):179–207, 1995.

[GL96a] Jean Goubault-Larrecq. On computational interpretations of the modal logic S4 I. Cut elimination. Technical report, University of Karlsruhe, 1996. Available on `ftp://theory.doc.ic.ac.uk/theory/guests/GoubaultJ/`.

[GL96b] Jean Goubault-Larrecq. On computational interpretations of the modal logic S4 II. The $\lambda\mathtt{ev}Q$-calculus. Technical report, University of Karlsruhe, 1996. Available on `ftp://theory.doc.ic.ac.uk/theory/guests/GoubaultJ/`.

[GL96c] Jean Goubault-Larrecq. On computational interpretations of the modal logic S4 III. Termination, confluence, conservativity of $\lambda\mathtt{ev}Q$ and $\lambda\mathtt{ev}Q_H$. Technical report, University of Karlsruhe, 1996. Available soon on `ftp://theory.doc.ic.ac.uk/theory/guests/GoubaultJ/`.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[Gra92] Bernd Gramlich. Relating innermost, weak, uniform and modular termination of term rewriting systems. In *LPAR'92*, pages 285–296. Springer Verlag LNAI 624, 1992.

[HS88] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and λ-Calculus*, volume 1 of *London Mathematical Society Student Texts*. Cambridge University Press, 1988.

[JR96a] Jean-Pierre Jouannaud and Albert Rubio. A recursive path ordering for higher-order terms compatible with $\beta\eta$-reductions. In *RTA'96 (submitted)*, 1996.

[JR96b] Jean-Pierre Jouannaud and Albert Rubio. A recursive path ordering for higher-order terms in η-long β-normal form. In *RTA'96*, 1996.

[MH97a] César Augusto Muñoz Hurtado. Dependent types with explicit substitutions: A meta-theoretical development. Submitted to the TYPES'96 workshop, 1997.

[MH97b] César Augusto Muñoz Hurtado. Meta-theoretical properties of λ_ϕ: A left-linear variant of λ_σ. Technical Report RR-3107, Inria, 1997.

[MS95] Michael Marz and Peter Selinger. A very well known theorem. *SemSoc Newsletter*, 4:7–8, May 1995. Available at `http://lix.polytechnique.fr/~mackie/SEMSOC.html`.

[Raf96] Christophe Raffalli. Type checking in system F^η. Available at `http://hypatia.dcs.qmw.ac.uk/authors/R/RaffalliC/papers/F-eta.dvi.gz`, 1996.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399