Conjunctive Types and SKInT

Jean Goubault-Larrecq*

G.I.E. Dyade, INRIA, Inria bâtiment 30, F-78153 Le Chesnay Cedex,
Jean.Goubault@inria.fr

1 Introduction

The λ-calculus and its typed versions are important tools for studying most fundamental computation and deduction paradigms. However, the non-trivial nature of substitution, as used in the definition of λ-reduction notably, has spurred the design of various first-order languages representing λ-terms, λ-reduction and λ-conversion, where computation is simple, first-order rewriting, and substitution becomes an easy notion again. Let us cite Curry's combinators [7], Curien's categorical combinators [5], the myriad of so-called λ-calculi with explicit substitutions, among which $\lambda\sigma$ [1], $\lambda\sigma_{\Uparrow}$ [12], $\lambda\upsilon$ [17], $\lambda\zeta$ [19], etc. Unfortunately, each one of these calculi has defects: Curry's combinators do not model λ-conversion fully, categorical combinators, $\lambda\sigma$ and $\lambda\sigma_{\Uparrow}$ do not normalize strongly in the typed case [18], $\lambda\upsilon$ is not confluent in the presence of free variables (a.k.a. meta-variables), $\lambda\zeta$ models λ-conversion but not λ-reduction, etc.

SKInT [9] is a first-order language and rewrite system that does not have these defects. In particular, SKInT models reduction in the λ-calculus, in the sense that there is a mapping L^* from λ-terms to SKInT such that whenever u rewrites to v in the λ-calculus, then $L^*(u)$ rewrites to $L^*(v)$ in SKInT. SKInT is confluent even on open terms, i.e. terms with meta-variables; L^* defines a conservative embedding of the λ-calculus inside SKInT, in that u and v are λ-convertible if and only if $L^*(u)$ and $L^*(v)$ are convertible in SKInT; reduction in SKInT standardizes; L^* preserves weak normalization, i.e., if u is a weakly normalizing λ-term, then $L^*(u)$ is a weakly normalizing SKInT-term. SKInT also enjoys a simple type discipline corresponding to that of the λ-calculus, that is, if u is a λ-term of type τ, then $L^*(u)$ is a SKInT-term of some type $L^*(\tau)$ easily computed from τ; reduction in SKInT obeys subject reduction, and every simply-typed SKInT-term normalizes strongly [9].

The aim of this paper is to extend our result on preservation of weak normalization to show that SKInT, like $\lambda\upsilon$ and $\lambda\zeta$, also preserves strong normalization, and also solvability; this is done by showing that, just like in the λ-calculus, strongly normalizing, weakly normalizing and solvable terms are characterized as terms that are typable in various conjunctive type disciplines [20, 4].

The plan of the paper is as follows: we introduce the required notions and notations in Section 2, then we attack our goal in Section 3. We show that any reasonable translation from the λ-calculus to SKInT preserves solvability, weak

* This work has been done in the context of Dyade (Bull/Inria R&D joint venture).

normalization and strong normalization (Corollary 1), where "reasonable" means that it preserves typability in certain systems of conjunctive types; this holds in particular for the translations of [9]. In turn, this result follows from a result stating that all terms that are typable in particular systems of conjunctive types (to be defined in Section 3) have the corresponding normalization properties (Theorem 1). The converse also holds (Corollary 2), just as in the λ-calculus.

The proofs have been reduced to keep the paper short and reasonably legible. Full proofs can be found in [11].

2 SKInT and the λ-Calculus

Recall that the syntax of the λ-calculus is [3]:

$$t ::= x \mid tt \mid \lambda x \cdot t$$

where x ranges over an infinite set of so-called variables, and terms s and t that are α-equivalent are considered equal; we denote λ-terms by s, t, …, and variables by x, y, z, etc. α-equivalence is the compatible closure of $\lambda x \cdot (t[x/y]) = \lambda y \cdot t$ and $t[s/x]$ denotes the usual capture-avoiding substitution of s for x in t. We shall write $=$ for α-equivalence; in the first-order calculi to come, $=$ will denote syntactic equality.

The basic computation rule is β-reduction, the compatible closure of:

$$(\beta) \quad (\lambda x \cdot t)s \to t[s/x]$$

The relation $\longrightarrow$ is the compatible closure of this relation, $\longrightarrow^*$ is the reflexive-transitive closure of the latter, and $\longrightarrow^+$ is its transitive closure. We shall use $\longrightarrow$, $\longrightarrow^*$, $\longrightarrow^+$ ambiguously in other calculi as well, taking care to make clear which is intended.

We shall also add the following η-reduction rule:

$$(\eta) \quad \lambda x \cdot tx \to x \quad (x \text{ not free in } t)$$

to the λ-calculus, yielding the so-called λ_η-calculus. The corresponding compatible closure relation will sometimes be noted $\longrightarrow_\eta$ to distinguish it from $\longrightarrow$, and similarly for the calculi to come and their respective η rules.

The terms of SKInT, and of its companion calculus SKIn [9], on the other hand, are defined by the grammar:

$$u ::= x \mid I_\ell \mid S_\ell(u, u) \mid K_\ell(u)$$

where ℓ ranges over $\mathbb{N}$. This is an infinitary first-order language. The reduction rules of SKInT are shown in Figure 1, thus defining an infinite rewrite system. The semantical idea behind SKInT, or SKIn, will be made clear by stating an informal translation from SKInT (or SKIn) to the λ-calculus. Intuitively:

$$\begin{aligned} I_\ell &\sim \lambda x_0 \cdot \ldots \cdot \lambda x_{\ell-1} \cdot \lambda x_\ell \cdot x_\ell \\ S_\ell(u, v) &\sim \lambda x_0 \cdot \ldots \cdot \lambda x_{\ell-1} \cdot u x_0 \ldots x_{\ell-1}(v x_0 \ldots x_{\ell-1}) \\ K_\ell(u) &\sim \lambda x_0 \cdot \ldots \cdot \lambda x_{\ell-1} \cdot \lambda x_\ell \cdot u x_0 \ldots x_{\ell-1} \end{aligned}$$

So I_ℓ, S_ℓ, K_ℓ generalize Curry's combinators I, S and K respectively.

SKIn is defined as SKInT, except that rule $(K_\ell S_{\mathcal{L}+1})$ is replaced by $(K_\ell S_\mathcal{L})$: $K_\ell(S_{\mathcal{L}-1}(u,v)) \to S_\mathcal{L}(K_\ell(u), K_\ell(v))$; conversely, SKInT is as SKIn, except that rule $(K_\ell S_\mathcal{L})$ is restricted to the case $\ell < \mathcal{L} - 1$.

$$(SI_\ell) \quad S_\ell(I_\ell, w) \to w \qquad (SK_\ell) \quad S_\ell(K_\ell(u), w) \to u$$

$$\begin{array}{ll|ll}
(S_\ell I_\mathcal{L}) & S_\ell(I_\mathcal{L}, w) \to I_{\mathcal{L}-1} & (K_\ell I_\mathcal{L}) & K_\ell(I_{\mathcal{L}-1}) \to I_\mathcal{L} \\
(S_\ell K_\mathcal{L}) & S_\ell(K_\mathcal{L}(u), w) \to K_{\mathcal{L}-1}(S_\ell(u,w)) & (K_\ell K_\mathcal{L}) & K_\ell(K_{\mathcal{L}-1}(u)) \to K_\mathcal{L}(K_\ell(u)) \\
(S_\ell S_\mathcal{L}) & S_\ell(S_\mathcal{L}(u,v), w) & (K_\ell S_{\mathcal{L}+1}) & K_\ell(S_\mathcal{L}(u,w)) \to S_{\mathcal{L}+1}(K_\ell(u), K_\ell(w)) \\
 & \quad \to S_{\mathcal{L}-1}(S_\ell(u,w), S_\ell(v,w)) & &
\end{array}$$

Fig. 1. SKInT reduction rules (for every $0 \leq \ell < \mathcal{L}$)

Both SKIn and SKInT are confluent and standardize. (See Section 3 for the definition of standard reductions.)

We can split SKInT in two: the set of all rules (SI_ℓ), $\ell \geq 0$, corresponds somehow to the actual β-reduction rule of the λ-calculus, or more precisely to βI-reduction (the notion of reduction of λI), and we shall call this group of rules βI. All other rules essentially correspond to the propagation of substitutions in the λ-calculus, and we call the set of these rules ΣT. Similarly, Σ is SKIn minus βI. It turns out that both Σ and ΣT are confluent, but ΣT terminates while Σ only normalizes weakly (even in a typed setting, see [9]).

We shall also consider SKInT$_\eta$, which is SKInT plus the following group η:

$$(\eta S_\ell) \quad S_{\ell+1}(K_\ell(u), I_\ell) \to u \quad (\ell \geq 0)$$

SKInT$_\eta$ is also confluent, and η-reductions can be postponed after all other reductions, just like in the λ-calculus.

The natural translation from the λ-calculus to SKInT, resp. SKIn, is $t \mapsto t^*$, defined in Figure 2. Whenever $u \longrightarrow v$ in the λ-calculus, $u^* \longrightarrow^+ v^*$ in SKIn, but not in SKInT. In the case of SKInT, we have to use a more complicated translation, like L^*: then $u \longrightarrow v$ implies $L^*(u) \longrightarrow^+ L^*(v)$. $L^*(s)$ is defined as $(L(s))^*$, where $L(s)$ is $\lambda z \cdot L_z(s)$ with z a fresh variable, and $L_z(x) =_{\mathrm{d}f} xz$, $L_z(\lambda x \cdot s) =_{\mathrm{d}f} \lambda x \cdot L_z(s)$, $L_z(st) =_{\mathrm{d}f} L_z(s)(L(t))$ (see [9] for an explanation).

The simple type discipline for the λ-calculus is defined by judgments $\Gamma \vdash t : \tau$, where t is a λ-term, τ is a simple type, i.e. a term in the following language:

$$\tau ::= B \mid \tau \to \tau$$

where B is a given non-empty set of so-called base types. Finally, Γ is a context, namely a finite map from variables to types; $\Gamma, x : \tau$ denotes Γ enriched by mapping x to τ, where x is outside the domain of Γ.

$$
\begin{array}{lll}
 & & [x]x = I_0 \\
x^* & = x & [x]y = K_0(y) \qquad (y \neq x) \\
(st)^* & = S_0(s^*, t^*) & [x](I_\ell) = I_{\ell+1} \\
(\lambda x \cdot t)^* & = [x](t^*) & [x](S_\ell(u,v)) = S_{\ell+1}([x]u, [x]v) \\
 & & [x](K_\ell(u)) = K_{\ell+1}([x]u)
\end{array}
$$

Fig. 2. Translation from the λ-calculus to SKIn

The typing rules for the λ-calculus are:

$$
\frac{}{\Gamma, x : \tau \vdash x : \tau} \qquad \frac{\Gamma \vdash s : \tau_1 \to \tau_2 \quad \Gamma \vdash t : \tau_1}{\Gamma \vdash st : \tau_2} \qquad \frac{\Gamma, x : \tau_1 \vdash s : \tau_2}{\Gamma \vdash \lambda x \cdot s : \tau_1 \to \tau_2}
$$

Those for SKInT are shown in Figure 3. Notice that in typing $K_\ell(u)$, the new type τ' is always inserted before an arrow type: this is intentional (see [9]), and is related to insights in proof systems for the modal logic S4.

$$
\frac{}{\Gamma, x : \tau \vdash x : \tau} \qquad \frac{}{\Gamma \vdash I_\ell : \tau_0 \to \ldots \to \tau_{\ell-1} \to \tau_\ell \to \tau_\ell}
$$

$$
\frac{\begin{array}{l}\Gamma \vdash u : \tau_0 \to \ldots \to \tau_{\ell-1} \to \tau_\ell \to \tau \\ \Gamma \vdash v : \tau_0 \to \ldots \to \tau_{\ell-1} \to \tau_\ell\end{array}}{\Gamma \vdash S_\ell(u,v) : \tau_0 \to \ldots \to \tau_{\ell-1} \to \tau} \qquad \frac{\Gamma \vdash u : \tau_0 \to \ldots \to \tau_{\ell-1} \to \tau_\ell \to \tau}{\Gamma \vdash K_\ell(u) : \tau_0 \to \ldots \to \tau_{\ell-1} \to \tau' \to \tau_\ell \to \tau}
$$

Fig. 3. Simple types for SKInT

Both the λ-calculus and SKInT enjoy subject reduction and normalize strongly on simply-typed terms. Moreover, we have the following meta-theorem on SKInT. Call a context Γ *arrowed* if and only if Γ maps every variable in its domain to an arrow type (of the form $\tau_1 \to \tau_2$); then for every arrowed context Γ, if $\Gamma, x : \tau_1 \vdash u : \tau_2$, then $\Gamma \vdash [x]u : \tau_1 \to \tau_2$. The restriction to arrowed contexts is necessary: consider the case where u is a variable other than x.

3 Conjunctive Types

We now turn to the relationship between conjunctive type systems for SKInT and termination properties, à la Sallé-Coppo [20, 4]. We first recall a few notions from [9]. Define the *spines* S by the grammar:

$$
S ::= x \mid I_\ell \mid S_\ell S \mid K_\ell S \qquad (\ell \geq 0)
$$

The *arity* of a spine is the number of operators of the form S_ℓ, $\ell \geq 0$, in it. If n is the arity of S, and $v_1, \ldots, v_n$ are n SKInT-terms, then the term $S[v_1, \ldots, v_n]$

is defined by:

$$\begin{array}{ll} x[] =_{\mathrm{d}f} x & (S_\ell S)[v_1, \ldots, v_n] =_{\mathrm{d}f} S_\ell(S[v_1, \ldots, v_{n-1}], v_n) \\ I_\ell[] =_{\mathrm{d}f} I_\ell & (K_\ell S)[v_1, \ldots, v_n] =_{\mathrm{d}f} K_\ell(S[v_1, \ldots, v_n]) \end{array}$$

Every term u can be written in a unique way as $S[v_1, \ldots, v_n]$: the spine S is the sequence of operators occurring along the leftmost branch of u, read top-down. Then the terms $v_1, \ldots, v_n$ are the second arguments of operators of the form S_ℓ, $\ell \geq 0$, on the spine, read bottom-up. The terms $v_1, \ldots, v_n$ are called the *arguments* of u. Iterating this decomposition of terms in spine and arguments allows us to see terms as trees of spines.

Call a *one-step spine reduction* $u \longrightarrow^s v$ any one-step reduction of a redex occurring on the spine of u. A *spine reduction* $u \longrightarrow^{s*} v$ is a sequence of one-step spine reductions. A term that has no one-step spine contractum is called *spine-normal.* Spine reductions play the role of head reductions in the λ-calculus. (However, spine reductions are not unique.)

Define *standard reductions* $u \longrightarrow^{std*} v$ by induction on v viewed as a tree of spines, if and only if $u \longrightarrow^{s*} S[u_1, \ldots, u_n]$, and $v = S[v_1, \ldots, v_n]$, where $u_i \longrightarrow^{std*} v_i$ for each i, $1 \leq i \leq n$. SKInT standardizes [9], in that $u \longrightarrow^* v$ (in SKInT) implies $u \longrightarrow^{std*} v$. In particular, a SKInT-term u is weakly normalizable if and only if it has a normalizing standard reduction.

Definition 1 *u is* solvable *if any of the following equivalent conditions hold:*

(i) *All* SKInT*-spine reductions starting from u terminate;*
(ii) *Some* SKInT*-spine reduction starting from u terminates.*

Proof. That (i) implies (ii) is clear. Conversely, write $u \Rightarrow v$ when u is ΣT-normal, has a spine βI-redex $S_\ell(I_\ell, w)$, and v is the unique ΣT-normal form of the spine βI-contraction of u. (Notice that any term has at most one spine βI-redex, hence $\Rightarrow$-reductions are unique.) By examining how rules commute, we can show that if $u \longrightarrow^{s*} v$ by using βI n times, then $\Sigma T(u) \Rightarrow^* \Sigma T(v)$ in exactly n steps (see the appendices to [11]). It follows that, if (ii) holds (with termination in n spine βI-steps), then any $\Rightarrow$-reduction starting from u terminates (in exactly n steps), and therefore all spine reductions do exactly n spine βI-steps; as ΣT terminates, all these spine reductions must be finite. $\square$

We wish to characterize solvable, weakly normalizing and strongly normalizing as terms that can be typed in some systems of conjunctive types. It will be profitable to use a simplified format for conjunctive types, due to S. van Bakel [2] (see also [21]), where the type of any given term is unique—modulo the choice of types for occurrences of variables—and has a well-defined arity: this is in contrast with the usual sort of conjunctive types (see e.g. [15, 8]). Define the *strict intersection types* τ by:

$$\tau ::= B \mid \mu \to \tau \qquad \mu ::= [\tau_1, \ldots, \tau_n] \qquad (n \geq 0)$$

where $[\tau_1, \ldots, \tau_n]$ is the multiset of types $\tau_1, \ldots, \tau_n$. Intuitively, $[\tau_1, \ldots, \tau_n]$ denotes the intersection of $\tau_1, \ldots, \tau_n$. If $n = 0$, $[]$ denotes the set of all terms. We shall also write ω for $[]$, and $\mu_1 \wedge \mu_2$ for the multiset union of μ_1 and μ_2.

Strongly normalizing λ-terms are those that are typable in system $\mathcal{S}$, defined as follows. Call $\mathcal{S}$*-types* the types generated by the grammar:

$$\tau ::= B \mid \mu \to \tau \qquad \mu ::= [\tau_1, \ldots, \tau_n] \qquad (n \geq 1)$$

That is, multisets of types are now restricted to be non-empty. We define system $\mathcal{S}$ as the system of Figure 4 (again), but where τ and μ-types are restricted to be $\mathcal{S}$-types.

$$\frac{}{\Gamma, x : \mu \wedge \tau \vdash x : \tau} \qquad \frac{\Gamma \vdash u : [\tau_1, \ldots, \tau_n] \to \tau \quad \Gamma \vdash v : \tau_1 \quad \ldots \quad \Gamma \vdash v : \tau_n}{\Gamma \vdash uv : \tau} \qquad \frac{\Gamma, x : \mu \vdash u : \tau}{\Gamma \vdash \lambda x \cdot u : \mu \to \tau}$$

Fig. 4. Conjunctive types for the λ-calculus

Correspondingly, we endow SKInT with the typing rules of Figure 5, yielding a typing system that we call $S\omega$ (when types are $\mathcal{S}\omega$-types), or S (when types are $\mathcal{S}$-types).

$$\frac{}{\Gamma, x : \mu \wedge \tau \vdash x : \tau} \qquad \frac{}{\Gamma \vdash I_\ell : \mu_0 \to \ldots \to \mu_{\ell-1} \to \mu_\ell \wedge \tau \to \tau}$$

$$\frac{\begin{array}{l}\Gamma \vdash u : \mu_0 \to \ldots \to \mu_{\ell-1} \to [\tau_1, \ldots, \tau_n] \to \tau \\ \Gamma \vdash v : \mu_0 \to \ldots \to \mu_{\ell-1} \to \tau_1 \\ \ldots \\ \Gamma \vdash v : \mu_0 \to \ldots \to \mu_{\ell-1} \to \tau_n\end{array}}{\Gamma \vdash S_\ell(u, v) : \mu_0 \to \ldots \to \mu_{\ell-1} \to \tau} \qquad \frac{\Gamma \vdash u : \mu_0 \to \ldots \to \mu_{\ell-1} \to \mu_\ell \to \tau}{\Gamma \vdash K_\ell(u) : \mu_0 \to \ldots \to \mu_{\ell-1} \to \mu \to \mu_\ell \to \tau}$$

Fig. 5. The system of conjunctive types for SKInT

Call an $\mathcal{S}\omega$-type τ *definite positive* if and only if ω only occurs negatively in τ. More formally, the definite positive types τ^+ and the definite negative types τ^- are defined by the grammar:

$$\begin{array}{lll} \tau^+ ::= B \mid \mu^- \to \tau^+ & \mu^+ ::= [\tau_1^+, \ldots, \tau_n^+] & (n \geq 1) \\ \tau^- ::= B \mid \mu^+ \to \tau^- & \mu^- ::= [\tau_1^-, \ldots, \tau_n^-] & (n \geq 0) \end{array}$$

A context Γ is *definite negative* if and only if every binding in Γ is of the form $x : \mu^-$ with μ^- definite negative. We say that the typing judgement $\Gamma \vdash s : \tau$ is *definite positive* if and only if Γ is definite negative and τ is definite positive.

We first have the following, which we leave to the reader to check:

Lemma 1 (Subject Reduction). *If $\Gamma \vdash u : \tau$ in $S\omega$, resp. S, and $u \longrightarrow^* v$ in* SKInT *or* SKInT$_\eta$*, then $\Gamma \vdash v : \tau$ in $S\omega$, resp. S.*

Theorem 1 (Normalization). *The following holds:*

(i) *If $\Gamma \vdash u : \tau$ is derivable in system $S\omega$, then u is solvable;*
(ii) *If $\Gamma^- \vdash u : \tau^+$ is definite positive and derivable in system $S\omega$, then u is weakly normalizing in* SKInT *and in* SKInT$_\eta$*;*
(iii) *If $\Gamma \vdash u : \tau$ is derivable in system S, then u is strongly normalizing in* SKInT *and even in* SKInT$_\eta$.

Proof. The idea is to translate the term u to a λ-term, and to use the corresponding results for the λ-calculus. We modify the $u \mapsto [\![u]\!](s_0, \ldots, s_{n_1})$ translation of [9] to map SKInT-terms to $\lambda_{\oplus\epsilon}$*-terms*: the idea is that instead of dropping some arguments (like s_ℓ in the definition of $[\![K_\ell(u)]\!](s_0, \ldots, s_{n-1})$ for $n \geq \ell$), we shall keep them on the left of some binary operator $\oplus$ such that $s \oplus t$ is semantically equivalent to t alone.

The $\lambda_{\oplus\epsilon}$-calculus is defined by the grammar:

$$t ::= x \mid tt \mid \lambda x \cdot t \mid \epsilon t \mid t \oplus t$$

and its reduction rules are $\beta\eta$-reduction plus:

$$(\epsilon) \quad \epsilon t \to t \qquad (\oplus-) \quad t_1 \oplus t_2 \to t_2 \qquad (\oplus) \quad (t_1 \oplus t_2) \oplus t_3 \to t_1 \oplus (t_2 \oplus t_3)$$

We define type systems that we call again $\mathcal{S}\omega$, resp. $\mathcal{S}$, defined on $\mathcal{S}\omega$-types, resp. $\mathcal{S}$-types, and whose typing rules are those of Figure 4, plus:

$$\frac{\Gamma \vdash t : \tau}{\Gamma \vdash \epsilon t : \tau} \quad \frac{[\Gamma \vdash t_1 : \tau_1] \quad \Gamma \vdash t_2 : \tau_2}{\Gamma \vdash t_1 \oplus t_2 : \tau_2}$$

where the bracketed premise $\Gamma \vdash t_1 : \tau_1$ is included in $\mathcal{S}$, but omitted in $\mathcal{S}\omega$.

There is an erasing translation $t \mapsto |t|$ from $\lambda_{\oplus\epsilon}$ to the λ-calculus (with β-reduction): $|\epsilon t| = |t|$, $|t_1 \oplus t_2| = |t_2|$, $|x| = x$, $|t_1 t_2| = |t_1|\ |t_2|$, $|\lambda x \cdot tx| = |t|$ if x is not free in t, and $|\lambda x \cdot t| = \lambda x \cdot |t|$ if t is not of the form $t'x$ with x not free in t'. $\lambda_{\oplus\epsilon}$ has the subject reduction property, and for every $\lambda_{\oplus\epsilon}$-term t:

(a) If $\Gamma \vdash t : \tau$ in $\mathcal{S}\omega$, then t is solvable, i.e., all head-reductions starting from t terminate. We call *head-reduction* in $\lambda_{\oplus\epsilon}$ any (ϵ), $(\oplus-)$, $(\oplus)$ or (η)-step, or any (β)-reduction step $s \longrightarrow t$ such that $|s| \longrightarrow |t|$ by a head (β)-reduction step in the λ-calculus (i.e., the $\lambda_{\oplus\epsilon}$-redex does not get erased).
(b) If $\Gamma \vdash t : \tau$ in $\mathcal{S}$, then t is strongly normalizing.

The proofs are by appealing to the same properties in the λ-calculus, using the erasing translation above (see [11]), or by reducibility methods (for (b)): see [10].

Define the *dimension* $\dim u$ of a SKInT-term u by:

$$\begin{array}{rlrl} \dim x =_{df} & -1 & \dim I_\ell =_{df} & \ell \\ \dim K_\ell(u) =_{df} & \max(\ell, \dim u) + 1 & \dim S_\ell(u,v) =_{df} & \max(\ell, \dim u) - 1 \end{array}$$

$$
[\![u]\!]_\oplus(s_0,\ldots,s_{n-1}) =_{\mathrm{d}f} \begin{cases} x s_0 \ldots s_{n-1} & \text{if } u = x \\ \epsilon(s_0 \oplus \ldots \oplus s_{\ell-1} \oplus s_\ell) s_{\ell+1} \ldots s_{n-1} & \text{if } u = I_\ell \\ [\![v]\!]_\oplus(s_0,\ldots,s_{\ell-1}, s_\ell \oplus s_{\ell+1}, s_{\ell+2},\ldots,s_{n-1}) & \text{if } u = K_\ell(v) \\ [\![v]\!]_\oplus(s_0,\ldots,s_{\ell-1}, \\ \qquad [\![w]\!]_\oplus(s_0,\ldots,s_{\ell-1}), s_\ell,\ldots,s_{n-1}) & \text{if } u = S_\ell(v,w) \end{cases}
$$

when $n > \dim u$

$$
[\![u]\!]_\oplus(s_0,\ldots,s_{n-1}) =_{\mathrm{d}f} \lambda x_n \cdot \ldots \cdot \lambda x_{m-1} \cdot [\![u]\!]_\oplus(s_0,\ldots,s_{n-1}, x_n,\ldots,x_{m-1})
$$

when $n < m = \dim u + 1$

Fig. 6. Interpretation of SKInT-terms as $\lambda_{\oplus\epsilon}$-terms

The new translation $u \mapsto [\![u]\!]_\oplus(s_0,\ldots,s_{n-1})$, parameterized by a list s_0, ..., s_{n-1} of $\lambda_{\oplus\epsilon}$-terms, is described in Figure 6.

By abuse of language, say that $\Gamma \vdash s : \mu$ is derivable in $\mathcal{S}\omega$, resp. $\mathcal{S}$, where $\mu = [\tau_1,\ldots,\tau_k]$, if and only if $\Gamma \vdash s : \tau_i$ is derivable in $\mathcal{S}\omega$, resp. $\mathcal{S}$, for every i, $1 \leq i \leq k$. An easy structural induction on u (see [11]) shows that, if $\Gamma \vdash u : \mu_0 \to \ldots \to \mu_{n-1} \to \tau$ in $S\omega$, resp. S, and $\Gamma \vdash s_i : \mu_i$ in $\mathcal{S}\omega$, resp. $\mathcal{S}$, for every i, $0 \leq i < n$, then $\Gamma \vdash [\![u]\!]_\oplus(s_0,\ldots,s_{n-1}) : \tau$ in $\mathcal{S}\omega$, resp. $\mathcal{S}$.

A tedious check now shows that whenever $u \longrightarrow v$ in SKInT_η, then for every sequence s_0, ..., s_{n-1} of $\lambda_{\oplus\epsilon}$-terms, $[\![u]\!]_\oplus(s_0,\ldots,s_{n-1}) \longrightarrow^* [\![v]\!]_\oplus(s_0,\ldots,s_{n-1})$ in $\lambda_{\oplus\epsilon}$, resp. $\longrightarrow^+$ in the case of (SI_ℓ), (ηS_ℓ) and a few other rules (see [11]).

By (b), any reduction R in SKInT, resp. SKInT_η, starting from a typable term in system S uses only finitely many instances of rules (SI_ℓ) and (ηS_ℓ), $\ell \geq 0$. But since ΣT terminates, there are finitely many reduction steps (in ΣT) inbetween two (SI_ℓ) or (ηS_ℓ)-steps. So R is finite, proving (iii).

To show (i), first notice that if $u \longrightarrow^s v$ in SKInT (or SKInT_η), then $[\![u]\!]_\oplus(s_0,\ldots,s_{n-1}) \longrightarrow^* [\![v]\!]_\oplus(s_0,\ldots,s_{n-1})$ (resp. $\longrightarrow^+$ if $u \longrightarrow^s v$ by (SI_ℓ), (ηS_ℓ) and a few other rules) by head-reductions in $\lambda_{\oplus\epsilon}$. By (a), any spine-reduction step in SKInT, resp. SKInT_η, starting from a typable term in $S\omega$ has only spine-reductions that use finitely many instances of (SI_ℓ) or (ηS_ℓ), $\ell \geq 0$. Since ΣT terminates, (i) follows.

To show (ii), we would like to use a similar argument, but any $\lambda_{\oplus\epsilon}$-term with a definite positive typing normalizes only weakly, and then we need to show that we can lift back the given normalization strategy in $\lambda_{\oplus\epsilon}$ to some normalization strategy in SKInT; this is not easy. Instead, we observe the following. Let u' be a spine-normal SKInT-term, and write u' as $S[u_1,\ldots,u_k]$. We may then write S as a word of the form $K_{i_{01}} \ldots K_{i_{0n_0}} S_{j_1} K_{i_{11}} \ldots K_{i_{1n_1}} S_{j_2} K_{i_{21}} \ldots K_{i_{2n_2}} \ldots S_{j_k} K_{i_{k1}} \ldots K_{i_{kn_k}} L$, with $L = I_j$ or L a variable (in which case we let $j =_{\mathrm{d}f} -1$), with:

$$
i_{01} > \ldots > i_{0n_0} \geq j_1 > i_{11} > \ldots > i_{1n_1} \geq j_2 > \ldots \geq j_k > i_{k1} > \ldots > i_{kn_k} > j \geq -1
$$

and $k \geq 0$, $n_0 \geq 0$, $n_1 \geq 0$, ..., $n_k \geq 0$, and when $n_i = 0$, the notation $j_i > i_{i1} > \ldots > i_{in_i} \geq j_{i+1}$ means $j_i \geq j_{i+1}$. If $n > \dim u'$, then we have:

$$
|[\![u']\!]_\oplus(s_0,\ldots,s_{n-1})| \tag{1}
$$

$$
\begin{aligned}
&= L'|s_{j+1}|\ldots|\widehat{s_{i_{kn_k}}}|\ldots|\widehat{s_{i_{k1}}}|\ldots|s_{j_k-1}| \\
&\quad |[\![u_1]\!]_\oplus(s_0,\ldots,s_{j_k-1})||s_{j_k}|\ldots|\widehat{s_{i_{(k-1)n_{k-1}}}}|\ldots|\widehat{s_{i_{(k-1)1}}}|\ldots|s_{j_{k-1}-1}| \\
&\quad |[\![u_2]\!]_\oplus(s_0,\ldots,s_{j_{k-1}-1})|\ldots \\
&\quad \ldots \\
&\quad |[\![u_k]\!]_\oplus(s_0,\ldots,s_{j_1-1})||s_{j_1}|\ldots|\widehat{s_{0n_0}}|\ldots|\widehat{s_{01}}|\ldots s_{n-1}
\end{aligned}
$$

where $L' = |s_j|$ if $L = I_j$, or $L' = x$ if L is a variable x, and $s_i \ldots \hat{s}_{i_1} \ldots \hat{s}_{i_p} \ldots s_j$ denotes the sequence of terms s_i, $\ldots$, s_j from which the terms s_{i_1}, $\ldots$, s_{i_p} have been omitted. If $n \leq \dim u'$, then $|[\![u']\!]_\oplus(s_0,\ldots,s_{n-1})|$ is the η-normal form of $\lambda s_n \cdot \ldots \cdot \lambda s_{\dim u'} \cdot |[\![u']\!]_\oplus(s_0,\ldots,s_{\dim u'})|$, and is therefore of the form $\lambda s_n \cdot \ldots \cdot \lambda s_{m-1} \cdot |[\![u']\!]_\oplus(s_0,\ldots,s_{m-1})|$.

As far as types are concerned, let u' have a definite positive typing. Any (definite positive) $S\omega$-type of u' must be of the form $\mu_0 \to \ldots \to \mu_{i_{01}} \to \mu_{i_{01}+1} \to \tau$, and the typing derivation leading to this type must map each u_i, $1 \leq i \leq k$, to the types $\mu_0 \to \ldots \to \mu_{j_{n+1-i}-1} \to \tau_{ip}$, $1 \leq p \leq m_i$, for some types τ_{ip}. Let μ'_i be $[\tau_{i1},\ldots,\tau_{im_i}]$. Then this typing derivation also gave L the type:

$$
\begin{aligned}
&\mu_0 \to \ldots \to \hat{\mu}_{i_{kn_k}} \to \ldots \to \hat{\mu}_{i_{k1}} \to \ldots \to \mu_{j_k-1} \\
&\to \mu'_1 \to \mu_{j_k} \to \ldots \to \hat{\mu}_{i_{(k-1)n_{k-1}}} \to \ldots \to \hat{\mu}_{i_{(k-1)1}} \to \ldots \to \mu_{j_{k-1}-1} \\
&\to \ldots \\
&\to \mu'_{k-1} \to \mu_{j_2} \to \ldots \to \hat{\mu}_{i_{1n_1}} \to \ldots \to \hat{\mu}_{i_{11}} \to \ldots \to \mu_{j_1-1} \\
&\to \mu'_k \to \mu_{j_1} \to \ldots \to \hat{\mu}_{i_{0n_0}} \to \ldots \to \hat{\mu}_{i_{01}} \to \mu_{i_{01}+1} \to \tau
\end{aligned} \tag{2}
$$

If L is a variable, since the typing context is definite negative, the type above must be definite negative, so in particular every μ'_i, $1 \leq i \leq k$, is definite positive. Since the type of u' was assumed definite positive, every μ_j is definite negative, so the types $\mu_0 \to \ldots \to \mu_{j_{n+1-i}-1} \to \tau_{ip}$ of u_i are definite positive. Similarly, if L is I_j, then recall that $j < i_{kn_k}$ (if $n_k \neq 0$) or $j < j_k$ (if $n_k = 0$ and $k \neq 0$), and by the form of the typing rule for I_j, every μ'_i, $1 \leq i \leq k$ must occur negatively in μ_ℓ, hence is definite positive (if $k \neq 0$; this is trivial if $k = 0$), hence again the types assigned to u_i are all definite positive.

Having made these remarks, let u be a SKInT-term with a definite positive $S\omega$-typing. Then $|[\![u]\!]_\oplus(s_0,\ldots,s_{n-1})|$ is a λ-term with a definite positive $\mathcal{S}\omega$-typing, for any sequence s_0, $\ldots$, s_{n-1} of the right types, hence it β-normalizes weakly. Recall that a weakly normalizing λ-term t has a finite Böhm tree: let $h(t)$ be the height of this tree. We show that, under the assumption that u has a definite positive $S\omega$-typing and that $|[\![u]\!]_\oplus(x_0,\ldots,x_{n-1})|$ normalizes weakly for some sequence of variables x_0, $\ldots$, x_{n-1}, then u SKInT-normalizes weakly. This is by induction on $h(t)$, where $t = |[\![u]\!]_\oplus(x_0,\ldots,x_{n-1})|$. First, by (i) and since u has an $S\omega$-typing, u is SKInT-solvable: let $u' =_{\mathrm{d}f} S[u_1,\ldots,u_k]$ be any spine-normal form of u. Since $u \longrightarrow^{s*} u'$, as in case (i), $|[\![u]\!]_\oplus(x_0,\ldots,x_{n-1})|$ head-rewrites to $|[\![u']\!]_\oplus(x_0,\ldots,x_{n-1})|$, and the latter is head-normal, since x_j is a variable (by inspection of Equation 1). By the remark on the types of spine-normal forms (Equation 2), each u_i, $1 \leq i \leq k$, also has a definite positive typing. Let now

t_i be $|\llbracket u_i \rrbracket_{\oplus}(x_0, \ldots, x_{j'_{k+1-i}})|$: by Equation 1, $h(t_i) < h(|\llbracket u' \rrbracket_{\oplus}(x_0, \ldots, x_{n-1})|) = h(|\llbracket u \rrbracket_{\oplus}(x_0, \ldots, x_{n-1})|)$. So the induction hypothesis applies: each u_i SKInT-normalizes weakly, say to some term v_i. Therefore u SKInT-normalizes weakly to $S[v_1, \ldots, v_k]$. This proves (ii) in the case of SKInT-reduction.

For SKInT$_\eta$-reduction, notice that every SKInT-normalizable term is also SKInT$_\eta$-normalizable. Indeed, observe that if $u \longrightarrow v$ by the η-rule (ηS_ℓ), and u is SKInT-normal, then so is v. □

The Normalization Theorem has an important corollary. Recall that a translation from λ-terms to a given language (say, SKInT) *preserves* strong normalization (resp. weak normalization, solvability) if and only if the translation of every strongly normalizing λ-term (resp. weakly normalizing, resp. solvable) is strongly normalizing (resp. weakly normalizing, resp. solvable).

Corollary 1 (Preservation of Normalization Properties). *Every translation mapping $\mathcal{S}$-typable λ-terms to S-typable* SKInT*-terms preserves strong normalization. Every translation mapping $\mathcal{S}\omega$-typable λ-terms to $S\omega$-typable* SKInT*-terms preserves solvability. Every translation mapping λ-terms having a definite positive typing in $\mathcal{S}\omega$ to* SKInT*-terms having a definite positive typing in $S\omega$ preserves weak normalization.*

Proof. Notice that these translations need not map λ-terms to SKInT-terms of the same type: we just need to preserve typability, not the types themselves. Every strongly normalizing (resp. weakly normalizing, solvable) λ-term is typable in system $\mathcal{S}$ (resp. in $\mathcal{S}\omega$, in $\mathcal{S}\omega$ with a definite positive typing) [2, 21]; the result then follows from Theorem 1. □

It follows that the L^* translation of [9], in particular, preserves strong normalization, weak normalization and solvability. This works also in the presence of η-rules. Also, Corollary 1 is stronger: essentially, any reasonable translation from the λ-calculus to SKInT will preserve all three normalization properties.

Corollary 1 depends on the fact that strongly normalizing, resp. weakly normalizing, resp. solvable terms in the λ-calculus are all characterized in terms of types. We end this section by showing that the same holds in SKInT.

First, define S, resp. *$S\omega$-type substitutions* θ as finite maps from type variables, a.k.a. base types in B, to S, resp. $S\omega$-types. For any type or context a, $a\theta$ denotes the result of applying θ to a; $[\tau/b]$ denotes the substitution mapping b to τ. The following is easy:

Lemma 2. *If $\Gamma \vdash u : \tau$ is derivable in S, then for every S-type substitution θ, $\Gamma\theta \vdash u : \tau\theta$ is derivable in S.*

If $\Gamma \vdash u : \tau$ is derivable in $S\omega$, then for every $S\omega$-type substitution θ, $\Gamma\theta \vdash u : \tau\theta$ is derivable in $S\omega$.

If $\Gamma^- \vdash u : \tau^+$ is definite positive and derivable in $S\omega$, then for every S-type substitution θ, $\Gamma^-\theta \vdash u : \tau^+\theta$ is definite positive and derivable in $S\omega$.

Lemma 3. *Every* SKInT*-normal term u has a typing in system S. Every spine-normal* SKInT*-term u has a typing in $S\omega$.*

Proof. We first observe that every type can be written uniquely $\mu_0 \to \ldots \to \mu_{n-1} \to b$, where $b \in B$. We call n the *arity* of the type. By extension, we call arity of a typing $\Gamma \vdash u : \tau$ the arity of τ. We call a typing as above *S-normal* if $n \geq 1$ and $\mu_{n-1} = [b']$ with b' a base type other than b; we call it *$S\omega$-normal* if $n \geq 1$ and $\mu_{n-1} = \omega$.

Let the *degree* $d(u)$ of a SKInT-term u be defined by: $d(x) =_{\mathrm{d}f} 0$, $d(I_\ell) =_{\mathrm{d}f} \ell + 1$, $d(S_\ell(v, w)) =_{\mathrm{d}f} \ell$, $d(K_\ell(v)) =_{\mathrm{d}f} \ell + 1$. We show the more general claims that: (i) every SKInT-normal term u has a normal S-typing of arity $d(u) + 1$, and: (ii) every spine-normal term u has a normal $S\omega$-typing of arity $d(u) + 1$. This is by structural induction on u.

Notice first that: $(*)$ whenever a term u has an S-normal, resp. $S\omega$-normal typing, then it also has S-normal, resp. $S\omega$-normal typings $\Gamma \vdash u : \tau$ of arbitrary higher arities in the same system: indeed, if u has a normal typing $\Gamma \vdash u : \tau$ of arity n as above (with b the base type at the end), then it also has a normal typing of arity $n+1$, namely $\Gamma\theta \vdash u : \tau\theta$, by Lemma 2, with $\theta =_{\mathrm{d}f} [[b''] \to b'/b]$, $b'' \neq b'$ in the case of system S, $\theta =_{\mathrm{d}f} [\omega \to b'/b]$ in the case of $S\omega$. Claim $(*)$ then follows by an easy induction on n.

If u is a variable, then the normal typing $x : [b'] \to b \vdash x : [b'] \to b$ establishes (i), and $x : \omega \to b \vdash x : \omega \to b$ establishes (ii). If u is of the form I_ℓ, then we can choose the typing $\vdash I_\ell : \mu_0 \to \ldots \to \mu_{\ell-1} \to [[b'] \to b] \to [b'] \to b$ with $\mu_0, \ldots, \mu_{\ell-1}$ any S-types and $b' \neq b$ for (i), and $\vdash I_\ell : \mu_0 \to \ldots \to \mu_{\ell-1} \to [\omega \to b] \to \omega \to b$ for (ii).

When u is of the form $S_\ell(v, w)$, then, first, define the conjunction $\Gamma' \wedge \Gamma''$ of two contexts Γ' and Γ'' as the collection of bindings $x : \mu' \wedge \mu''$ (when $x : \mu' \in \Gamma'$ and $x : \mu'' \in \Gamma''$), $x : \mu'$ (if $x : \mu' \in \Gamma'$ but x does not appear in Γ''), and $x : \mu''$ (if $x : \mu'' \in \Gamma''$ but x does not appear in Γ'). Notice also that (by examination of the reduction rules), if u is spine-normal (in particular, normal), then $d(v) \leq \ell$. We show claim (i) as follows: by induction, v has an S-normal typing of arity $d(v) + 1 \leq \ell + 1$, hence by $(*)$ it has an S-normal typing of arity exactly $\ell + 1$, say $\Gamma' \vdash v : \mu'_0 \to \ldots \to \mu'_{\ell-1} \to [b'] \to b$; by induction hypothesis again, w has an S-typing, hence by $(*)$ we may assume w.l.o.g. that w has an S-typing of arity at least ℓ, say $\Gamma' \vdash w : \mu''_0 \to \ldots \to \mu''_{\ell-1} \to \tau''$; then, we can derive $\Gamma'[\tau''/b'] \vdash v : \mu_0 \to \ldots \to \mu_{\ell-1} \to [\tau''] \to b$, where $\mu_i =_{\mathrm{d}f} \mu'_i[\tau''/b']$, $0 \leq i < n$, by Lemma 2; then $\Gamma'[\tau''/b'] \wedge \Gamma'' \vdash u : \mu_0 \to \ldots \to \mu_{\ell-1} \to b$ is an S-typing of arity $d(u) = \ell$; substituting b for, say, $[b''] \to b'''$ (using Lemma 2), we get the required S-normal typing of arity $d(u) + 1$. Showing (ii) is easier: by induction, v has an $S\omega$-normal typing of arity $d(v) + 1 \leq \ell + 1$, hence by $(*)$ it has an $S\omega$-normal typing of arity $\ell + 1$, say $\Gamma' \vdash v : \mu'_0 \to \ldots \to \mu'_{\ell-1} \to \omega \to b$; then $\Gamma \vdash u : \mu_0 \to \ldots \to \mu_{\ell-1} \to \omega \to b'$ is the required $S\omega$-normal typing of arity $\ell + 1$, where $\Gamma =_{\mathrm{d}f} \Gamma'[\omega \to b'/b]$, and $\mu_i =_{\mathrm{d}f} \mu'_i[\omega \to b'/b]$, $0 \leq i < \ell$.

When u is of the form $K_\ell(v)$, then observe that (by examination of the reduction rules), if u is spine-normal (or normal), then $d(v) \leq \ell$. We show (i) as follows: by induction v has an S-normal typing of arity $d(v) + 1$, hence by $(*)$ an S-normal typing $\Gamma' \vdash v : \mu'_0 \to \ldots \to \mu'_{\ell-1} \to [b'] \to b$ of arity $\ell + 1$; then $\Gamma' \vdash K_\ell(v) : \mu'_0 \to \ldots \to \mu'_{\ell-1} \to \mu \to [b'] \to b$ is the required S-normal typing for u, for any S-type μ. And (ii) follows by a similar construction, replacing $[b']$ by ω and letting μ be any $S\omega$-type. $\square$

Contrarily to what happens in the λ-calculus, types in $S\omega$ are not preserved by the inverse of SKInT-reduction. However:

Lemma 4. *If $u \longrightarrow^* v$ in* SKInT_η*, and $\Gamma \vdash v : \tau$ in $S\omega$, then $\Gamma\theta \vdash u : \tau\theta$ for some S-type substitution θ.*

Proof. More concisely, we shall say that whenever v has some $S\omega$-type τ, then u has type $\tau\theta$ for some S-substitution θ (where the contexts Γ and $\Gamma\theta$ are understood). We first show this when $u = l$, $v = r$, and $l \to r$ is any of the reduction rules. There are only three interesting rules:

- (SI_ℓ): $l = S_\ell(I_\ell, w)$, $r = w$. If τ has arity $n \geq \ell$, then τ is of the form $\mu_0 \to \ldots \to \mu_{\ell-1} \to \tau'$, and we can take I_ℓ of type $\mu_0 \to \ldots \to \mu_{\ell-1} \to [\tau'] \to \tau'$, so that l has type τ. If τ has arity $n < \ell$, then let τ be $\mu_0 \to \ldots \to \mu_{n-1} \to b$, and θ be $[\tau'/b]$, where τ' is any S-type of arity at least $\ell - n$: by Lemma 2, r has type $\tau\theta$, and since $\tau\theta$ has arity at least ℓ, then as above l has type $\tau\theta$.
- (SK_ℓ): $l = S_\ell(K_\ell(u), w)$, $r = u$. If τ has arity $n \geq \ell + 1$, then write τ as $\mu_0 \to \ldots \to \mu_{\ell-1} \to \mu_\ell \to \tau'$: we can give $K_\ell(u)$ the $S\omega$-type $\mu_0 \to \ldots \to \mu_{\ell-1} \to \omega \to \mu_\ell \to \tau'$, and therefore l has type τ as well (notice that we need not give w a type). Otherwise, as in the (SI_ℓ) case, r and l have type $\tau\theta$ for some S-substitution θ of arity $\ell + 1 - n$.
- (ηS_ℓ): $l = S_{\ell+1}(K_\ell(u), I_\ell)$, $r = u$. If the type τ of r has arity $n \geq \ell + 1$, then write τ as $\mu_0 \to \ldots \to \mu_\ell \to \tau'$, and μ_ℓ as $[\tau_1, \ldots, \tau_k]$. We can give I_ℓ all the types $\mu_0 \to \ldots \to \mu_{\ell-1} \to \mu_\ell \to \tau_i$, $1 \leq i \leq k$, and we can give $K_\ell(u)$ the type $\mu_0 \to \ldots \to \mu_{\ell-1} \to \mu_\ell \to [\tau_1, \ldots, \tau_k] \to \tau'$, so l can be given type τ again. If $n < \ell + 1$, then, if τ is of the form $\mu_0 \to \ldots \to \mu_{n-1} \to b$, let θ be $[\tau'/b]$ for any S-type τ' of arity at least $\ell + 1 - n$: then l has type $\tau\theta$.

For all the other rules, we can choose the identity substitution for θ (see [11]).

We now claim that whenever $u \longrightarrow v$—namely, when $u = \mathcal{C}[l]$, $v = \mathcal{C}[r]$, and $l \to r$ is some reduction rule—and v has type τ in $S\omega$, then u has some type $\tau\theta$ in $S\omega$, where θ is an S-substitution. This is a straightforward structural induction on $\mathcal{C}$, using Lemma 2. The Lemma then follows by induction on the length of the reduction $u \longrightarrow^* v$. $\square$

Theorem 2. *Every solvable* SKInT*-term has an $S\omega$-typing. Every* SKInT*-weakly normalizing, resp.* SKInT_η*-weakly normalizing term has a definite positive $S\omega$-typing.*

Proof. Let u be solvable. Then $u \longrightarrow^{s*} v$, where v is spine-normal; by Lemma 3, v has an $S\omega$-typing $\Gamma \vdash v : \tau$; by Lemma 4, u has a typing of the form $\Gamma\theta \vdash u : \tau\theta$, where θ is an S-substitution. This is clearly an $S\omega$-typing.

On the other hand, if u is weakly normalizing (in SKInT or in SKInT_η), then $u \longrightarrow^* v$ for some SKInT-normal term v; by Lemma 3, v has a definite positive $S\omega$-typing $\Gamma^- \vdash v : \tau^+$; by Lemma 4, u has a typing of the form $\Gamma^-\theta \vdash u : \tau^+\theta$, where θ is an S-substitution. By Lemma 2, this typing is therefore not only an $S\omega$-typing, but is also definite positive. $\square$

In fact, the proof even shows that every weakly normalizing term, whether in SKInT or in SKInT_η, has an $S\omega$-typing where ω does not occur at all (but ω may occur in the typing *derivation*). This is similar to [8], Theorem 6.12.

Theorem 3. *If u is strongly normalizing in* SKInT*, resp.* SKInT_η*, then it has an S-typing.*

Proof. As u is strongly normalizing, any normalization strategy terminates. Choose any *innermost* strategy, i.e. any strategy that reduces only redexes whose strict subterms are all normal. (In particular, the redex $S_\ell(K_\ell(u_1), u_2)$ can only be reduced when u_1 and $K_\ell(u_1)$ are normal.) Let $\nu(u)$ denote the length of the longest reduction sequence in SKInT starting from u according to this strategy.

We show the claim by induction on $\nu(u)$. If $\nu(u) = 0$, then this is by Lemma 3. Otherwise, assume that $u \longrightarrow v$ (so that $\nu(v) < \nu(u)$, hence by induction v has an S-typing $\Gamma \vdash v : \tau$). If the reduction from u to v is by any rule except (SK_ℓ), $\ell \geq 0$, then u has an S-typing $\Gamma\theta \vdash u : \tau\theta$, where θ is an S-substitution: this is as in the proof of Lemma 4.

In case $u \longrightarrow v$ by (SK_ℓ), this does not work any longer, since we cannot use ω (not an S-type). Instead, observe that u can be written as $\mathcal{C}[S_\ell(K_\ell(u_1), u_2)]$, and that $v = \mathcal{C}[u_1]$. By induction hypothesis, and since $\nu(u_2) < \nu(u)$, u_2 has an S-typing $\Gamma_2 \vdash u_2 : \tau_2$, and w.l.o.g. we may assume that τ_2 has arity at least ℓ, i.e. that $\tau_2 = \mu''_0 \to \ldots \to \mu''_{\ell-1} \to \tau''$. Since the chosen reduction strategy is innermost, u_1 is normal, and by Lemma 3 (more precisely, by Claim (i) in its proof), u_1 has an S-normal typing of degree exactly $d(u_1) + 1$; but since the reduction is innermost again, $K_\ell(u_1)$ is normal as well, so $d(u_1) \leq \ell$, and (by Remark $(*)$ in the proof of Lemma 3) therefore u_1 has an S-normal typing of arity $\ell + 1$, say v $\Gamma_1 \vdash u_1 : \mu'_0 \to \ldots \to \mu'_{\ell-1} \to [b'] \to b$. So we may now derive $\Gamma \vdash S_\ell(K_\ell(u_1), u_2) : \mu_0 \to \ldots \to \mu_{\ell-1} \to b$, where $\Gamma =_{\mathrm{d}f} \Gamma_1[\tau''/b'] \wedge \Gamma_2$, $\mu_i =_{\mathrm{d}f} \mu'_i[\tau''/b'] \wedge \mu''_i$ for each i, $0 \leq i < \ell$. It follows that u itself has an S-typing (which is an instance of the latter), by a straightforward induction on $\mathcal{C}$.

The case of SKInT_η-strongly normalizing terms is completely similar. $\square$

Corollary 2. *The following equivalences hold, for any* SKInT*-term u:*

- *u solvable $\Leftrightarrow$ u typable in $S\omega$*
- *u* SKInT*-weakly normalizing $\Leftrightarrow$ u* SKInT_η*-weakly normalizing*
  $\Leftrightarrow$ u has a definite positive $S\omega$-typing
- *u* SKInT*-strongly normalizing $\Leftrightarrow$ u* SKInT_η*-strongly normalizing*
  $\Leftrightarrow$ u typable in S.

4 Conclusion

We have shown that SKInT enjoyed exactly the same properties as the λ-calculus, as far as the relationship between conjunctive types and various normalization properties is concerned. This implies that SKInT preserves solvability, weak normalization and strong normalization, for any reasonable, i.e. typability-preserving translation from the λ-calculus to SKInT, in particular for the L^* translation of [9].

It is then interesting to compare SKInT with other calculi of explicit substitutions. Indeed, although SKInT was not presented as a calculus of explicit substitutions in [9], it is definitely so: I_ℓ is de Bruijn index $\ell \geq 0$, and the de Bruijn substitution $[0 := v_0, \ldots, n := v_n]$ applied to u is $(\ldots((u \circ_0 v_0) \circ_1 v_1) \ldots) \circ_n v_n$, where the $\circ_\ell$ operator is defined by: $u \circ_\ell v =_{\mathrm{d}f} S_{\ell+1}(K_\ell(u), v)$, as the reader is invited to check. Then ΣT plays the role of the substitution calculus, and βI (i.e., (SI_ℓ)) more or less plays the role of the β rule (the connection is not as direct as in more traditional calculi of explicit substitutions, though).

Together with the results of [9], this advances the table of properties of calculi with explicit substitutions proposed by Lang and Rose [16][1] to the following ($\supseteq \beta$ means "simulates β-reduction", CRM is "is Church-Rosser in the presence of meta-variables", SSN is "has a strongly normalizing substitution subcalculus", PSN means "preserves strong normalization").

Name	#	$\supseteq \beta$	first-order?	unconditional?	CRM	SSN	PSN	Reference
$\lambda\sigma$	11	yes	yes	yes	no	yes	no	[1]
$\lambda\sigma_{\Uparrow}$	21	yes	yes	yes	yes	yes	no	[6]
$\lambda\upsilon$	8	yes	yes	yes	no	yes	yes	[17]
λs	∞	yes	yes	yes	no	yes	yes	[13]
λs_e	∞	yes	yes	yes	yes	?	no	[14]
$\lambda\zeta$	13	no	yes	yes	yes		yes	[19]
λxci	∞	yes	no	no	yes	yes	?	[16]
$\lambda\mathrm{x}_B$ci	∞	yes	yes	no	no	yes	?	[16]
SKIn	∞	yes	yes	yes	yes	no	no	[9]
SKInT	∞	yes	yes	yes	yes	yes	yes	[9]
$\lambda\beta$	∞	yes	no		yes		yes	Church, see [3]

The only defect we know of SKInT is that it encodes the λ-calculus in a slightly complicated, and non-unique way [9]: e.g., you may wish to use the L^* translation, or any other. We believe that this drawback is offset by the fact that SKInT has basically all the properties of the λ-calculus: confluence, even on open terms; standardization; strong normalization of simply-typed terms; a conservative extension of the λ-calculus with β, resp. $\beta\eta$-conversion (see [9]); preservation of solvability, weak and strong normalization; characterization of solvable, weakly and strongly normalizable terms by conjunctive typings (this paper); all this in an infinite but regular first-order equational formulation.

Further work should investigate whether easier, direct proofs of the results presented here are possible, using variants of the reducibility method [15, 8].

[1] F. Lang notes that some of the results of this paper are wrong, and we have followed his remarks in the table (see `http://www.ens-lyon.fr/~flang/papiers.html`).

References

[1] M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit substitutions. In *POPL'90*, pages 31–46, 1990.

[2] S. van Bakel. Complete restrictions of the intersection type discipline. *Theoretical Computer Science*, 102(1):135–163, 1992.

[3] H. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984.

[4] F. Cardone and M. Coppo. Two extensions of Curry's type inference system. In P. Odifreddi, editor, *Logic and Computer Science*, volume 31 of *The APIC Series*, pages 19–76. Academic Press, 1990.

[5] P.-L. Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming*. Pitman, London, 1986.

[6] P.-L. Curien, T. Hardin, and J.-J. Lévy. Confluence properties of weak and strong calculi of explicit substitutions. *J. ACM*, 43(2):362–397, 1996.

[7] H. B. Curry and R. Feys. *Combinatory Logic*, volume 1. North-Holland, 1958.

[8] J. Gallier. Typing untyped λ-terms, or reducibility strikes again! *Annals of Pure and Applied Logic*, 91(2–3):231–270, 1998.

[9] H. Goguen and J. Goubault-Larrecq. Sequent combinators: A Hilbert system for the lambda calculus. *Mathematical Structures in Computer Science*, 1999.

[10] J. Goubault-Larrecq. On computational interpretations of the modal logic S4 IIIb. Confluence and conservativity of the $\lambda\mathtt{ev}Q_H$-calculus. Research report, Inria, 1997.

[11] J. Goubault-Larrecq. A few remarks on SKInT. Research report RR-3475, Inria, 1998.

[12] T. Hardin and J.-J. Lévy. A confluent calculus of substitutions. In *France-Japan Artificial Intelligence and Computer Science Symposium*, 1989.

[13] F. Kamareddine and A. Ríos. A λ-calculus à la de Bruijn with explicit substitutions. In *PLILP'95*, pages 45–62. Springer Verlag LNCS 982, 1995.

[14] F. Kamareddine and A. Ríos. Extending a λ-calculus with explicit substitution which preserves strong normalisation into a confluent calculus on open terms. *Journal of Functional Programming*, 7:395–420, 1997.

[15] J.-L. Krivine. *Lambda-calcul, types et modèles*. Masson, 1992.

[16] F. Lang and K. H. Rose. Two equivalent calculi of explicit substitution with confluence on meta-terms and preservation of strong normalization (one with names and one first-order). Presented at WESTAPP'98, 1998.

[17] P. Lescanne and J. Rouyer-Degli. From $\lambda\sigma$ to $\lambda\upsilon$: a journey through calculi of explicit substitutions. In *POPL'94*, 1994.

[18] P.-A. Melliès. Typed lambda-calculi with explicit substitutions may not terminate. In *TLCA'95*, pages 328–334. Springer Verlag LNCS 902, 1995.

[19] C. A. Muñoz Hurtado. Confluence and preservation of strong normalization in an explicit substitutions calculus. In *LICS'96*. IEEE, 1996.

[20] P. Sallé. Une extension de la théorie des types en λ-calcul. In *5th ICALP*, pages 398–410. Springer Verlag LNCS 62, 1978.

[21] É. Sayag. *Types intersections simples*. PhD thesis, Université Paris VII, 1997.