Strict Functionals for Termination Proofs

Jaco van de Pol and Helmut Schwichtenberg [1]
Mathematisches Institut, Universität München
jaco@phil.ruu.nl schwicht@rz.mathematik.uni-muenchen.de

A semantical method to prove termination of higher order rewrite systems (HRS) is presented. Its main tool is the notion of a strict functional, which is a variant of Gandy's notion of a hereditarily monotonic functional [1]. The main advantage of the method is that it makes it possible to transfer ones intuitions about why an HRS should be terminating into a proof: one has to find a "strict" interpretation of the constants involved in such a way that the left hand side of any rewrite rule gets a bigger value than the right hand side. The applicability of the method is demonstrated in three examples.

- An HRS involving map and append.
- The usual rules for higher order primitive recursion in Gödel's T.
- Derivation terms for natural deduction systems. We prove termination of the rules for β–conversion and permutative conversion for logical rules including introduction and elimination rules for the existential quantifier. This has already been proved by Prawitz in [5]; however, our proof seems to be more perspicuous.

Technically we build on [7]. There a notion of a strict functional and simultaneously of a strict greater–than relation $>_{\mathrm{str}}$ between monotonic functionals is introduced. The main result then is the following. Let M be a term in β normal form and $\Box \in \mathrm{FV}(M)$. Then for any strict environment U and all monotonic $\mathbf{f}$ and $\mathbf{g}$, one has $\mathbf{f} >_{\mathrm{mon}} \mathbf{g} \implies [\![M]\!]_{U[\Box\mapsto\mathbf{f}]} >_{\mathrm{str}} [\![M]\!]_{U[\Box\mapsto\mathbf{g}]}$. From this van de Pol derives the technique described above for proving termination of higher order term rewrite systems, generalizing a similar approach for first order rewrite systems (cf. [3, p. 367]). Interesting applications are given in [7].

Here a slight change in the definition of strictness is exploited (against the original conference paper; cf. [7, Footnote p. 316]). This makes it possible to deal with rewrite rules involving types of level > 2 too, and in particular with proof theoretic applications. In order to do this some theory of strict functionals is developed. We also add product types, which are necessary to treat e.g. the existential quantifier.

[1] Both authors are partially supported by the Science Twinning Contract SC1*–CT91–0724 of the European Community.

1. Monotonicity and Strictness

Let ρ, σ, τ denote simple types over some base types ι (containing at least o), composed with $\to$ and $\times$. For simplicity we consider the sets $\mathcal{T}_\rho$ of all functionals of type ρ over some ground domains $\mathcal{T}_\iota$. The ground domains are provided with some partial order $>_\iota$.

Definition. *For any type ρ we define the set $\mathcal{M}_\rho \subseteq \mathcal{T}_\rho$ of monotonic functionals of type ρ and simultaneously a relation $\geq$ on $\mathcal{T}_\rho$.*

(i) (a) $\mathcal{M}_\iota = \mathcal{T}_\iota$.

(b) $\mathbf{f} \in \mathcal{M}_{\sigma\to\tau} \iff$ *for all* $\mathbf{x}, \mathbf{y} \in \mathcal{M}_\sigma$, $\mathbf{f}(\mathbf{x}) \in \mathcal{M}_\tau$ *and if* $\mathbf{x} \geq \mathbf{y}$ *then* $\mathbf{f}(\mathbf{x}) \geq \mathbf{f}(\mathbf{y})$.

(c) $\mathcal{M}_{\sigma\times\tau} = \mathcal{M}_\sigma \times \mathcal{M}_\tau$.

(ii) (a) $n \geq_\iota m \iff n >_\iota m$ *or* $n = m$.

(b) $\mathbf{f} \geq_{\sigma\to\tau} \mathbf{g} \iff$ *for all* $\mathbf{x} \in \mathcal{M}_\sigma, \mathbf{f}(\mathbf{x}) \geq_\tau \mathbf{g}(\mathbf{x})$.

(c) $\langle \mathbf{a}, \mathbf{b}\rangle \geq_{\sigma\times\tau} \langle \mathbf{c}, \mathbf{d}\rangle \iff \mathbf{a} \geq_\sigma \mathbf{c}$ *and* $\mathbf{b} \geq_\tau \mathbf{d}$.

We will use the following notation: $\vec{\sigma} \to \rho$ denotes the type $\sigma_1 \to \cdots \sigma_n \to \rho$. Let $\mathbf{x}(0)$ denote the left component of the pair $\mathbf{x}$ and $\mathbf{x}(1)$ its right component. This allows us to write projections and applications in a uniform way. Furthermore, simply typed terms M, N are introduced as usual: Typed variables x, y, z, application MN, abstraction $\lambda x\, M$, pairing $\langle M, N\rangle$ and projections $\pi_i(M)$ for $i = 0, 1$. Projections are also written $M0$ and $M1$. We use standard notions of free and bound variables, substitution, interpretation of M in a domain under environment U (denoted by $[\![M]\!]_U$). Using the new notation for projections, the previous definition can be written very compactly as:

Definition. *For any type ρ we define the set $\mathcal{M}_\rho \subseteq \mathcal{T}_\rho$ of monotonic functionals of type ρ and simultaneously a relation $\geq$ on $\mathcal{T}_\rho$.*

(i) $\mathbf{f} \in \mathcal{M} \iff$ *for all* $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathcal{M} \cup \{0, 1\}$, *if* $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$, *then* $\mathbf{f}(\vec{\mathbf{x}}) \geq \mathbf{f}(\vec{\mathbf{y}})$.

(ii) $\mathbf{f} \geq \mathbf{g} \iff$ *for all* $\vec{\mathbf{x}} \in \mathcal{M} \cup \{0, 1\}$, $\mathbf{f}(\vec{\mathbf{x}}) \geq \mathbf{g}(\vec{\mathbf{x}})$.

Here $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ only range over vectors for which $\mathbf{f}(\vec{\mathbf{x}})$ and $\mathbf{f}(\vec{\mathbf{y}})$ are of base type. $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$ means: For all i such that $\mathbf{x}_i \in \mathcal{M}$, $\mathbf{x}_i \geq \mathbf{y}_i$.

Lemma 1. *For any term M of the simply typed λ–calculus we have*

(i) $[\![M]\!]_U \in \mathcal{M}$ *for any monotonic environment U.*

(ii) $U \geq V \implies [\![M]\!]_U \geq [\![M]\!]_V$ *for monotonic environments U, V.*

Proof by simultaneous induction on M (standard). □

Definition. $\mathbf{f} >_{\mathrm{mon}} \mathbf{g} \iff$ *for all* $\vec{\mathbf{x}} \in \mathcal{M} \cup \{0,1\}$, $\mathbf{f}(\vec{\mathbf{x}}) > \mathbf{g}(\vec{\mathbf{x}})$.

Remark. Gandy's definition of hereditarily monotonic functionals from [1] has the following form. For any type ρ he defines the set $\mathcal{G}_\rho \subseteq \mathcal{T}_\rho$ of hereditarily monotonic functionals of type ρ and simultaneously a relation $>_{\mathrm{Gandy}}$ on $\mathcal{T}_\rho$ by

(i) $\mathbf{f} \in \mathcal{G} \iff$ for all $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathcal{G}$, if $\vec{\mathbf{x}} >_{\mathrm{Gandy}} \vec{\mathbf{y}}$, then $\mathbf{f}(\vec{\mathbf{x}}) > \mathbf{f}(\vec{\mathbf{y}})$. Here $\vec{\mathbf{x}} >_{\mathrm{Gandy}} \vec{\mathbf{y}}$ means that at least once we have $\mathbf{x}_i >_{\mathrm{Gandy}} \mathbf{y}_i$ and otherwise $\mathbf{x}_j = \mathbf{y}_j$.

(ii) $\mathbf{f} >_{\mathrm{Gandy}} \mathbf{g} \iff$ for all $\vec{\mathbf{x}} \in \mathcal{G}$, $\mathbf{f}(\vec{\mathbf{x}}) > \mathbf{g}(\vec{\mathbf{x}})$.

This definition is not well suited for termination proofs. Consider e.g. the term $xz(\lambda y\, 0)$, where x is interpreted by $\mathbf{x} \in \mathcal{G}$. Then also in the case $[\![M]\!] >_{\mathrm{Gandy}} [\![N]\!]$ one cannot conclude $[\![xM(\lambda y\, 0)]\!] > [\![xN(\lambda y\, 0)]\!]$, since $[\![\lambda y\, 0]\!] \notin \mathcal{G}$. Hence Gandy in [1] had to restrict himself to λ–I–terms. As an alternative it is tempting to replace "for all $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathcal{G}$" in (i) by "for all $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathcal{M}$". Furthermore it turns out to be useful to add $\mathbf{f} \in \mathcal{M}$ to the right hand side of (i) and also $\mathbf{f} \geq \mathbf{g}$ to the right hand side of (ii). On pairs, the order $>_{\mathrm{Gandy}}$ is defined pointwise in [1]. We propose a change to obtain a more well suited order for termination proofs. If in a pair $\langle M, N\rangle$, M rewrites to M', with $[\![M]\!] > [\![M']\!]$, one wants to conclude that the corresponding interpretation gets smaller. These considerations motivate the following definition:

Definition. *For any type* ρ *we define the set* $\mathcal{S}_\rho \subseteq \mathcal{M}_\rho$ *of strict functionals of type* ρ *and simultaneously a relation* $>_{\mathrm{str}}$ *on* $\mathcal{T}_\rho$.

(i) $\mathbf{f} \in \mathcal{S} \iff \mathbf{f} \in \mathcal{M}$, *and for all* $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathcal{M} \cup \{0,1\}$, *if* $\vec{\mathbf{x}} >_{\mathrm{str}} \vec{\mathbf{y}}$, *then* $\mathbf{f}(\vec{\mathbf{x}}) > \mathbf{f}(\vec{\mathbf{y}})$. *Here* $\vec{\mathbf{x}} >_{\mathrm{str}} \vec{\mathbf{y}}$ *means that at least once we have* $\mathbf{x}_i >_{\mathrm{str}} \mathbf{y}_i$ *and otherwise* $\mathbf{x}_j \geq \mathbf{y}_j$.

(ii) $\mathbf{f} >_{\mathrm{str}} \mathbf{g} \iff \mathbf{f} \geq \mathbf{g}$ *and*

- *(a) (base type)* $\mathbf{f} > \mathbf{g}$*; or*
- *(b) (arrow type) for all* $\mathbf{x} \in \mathcal{S}$, $\mathbf{f}(\mathbf{x}) >_{\mathrm{str}} \mathbf{g}(\mathbf{x})$*; or*
- *(c) (product type)* $\mathbf{f}(0) >_{\mathrm{str}} \mathbf{g}(0)$ *or* $\mathbf{f}(1) >_{\mathrm{str}} \mathbf{g}(1)$.

Remark. In [7] a very similar modification of Gandy's definition is used. In a preliminary version, the requirement $\mathbf{f} \geq \mathbf{g}$ in (ii) was missing. For the examples considered in [7], which only concern rewrite rules for constants of level ≤ 2, this makes no difference. However, if one considers higher order rewrite rules like those for the primitive recursion operators in Gödel's T, then it is necessary to be able to infer $\mathbf{f} \geq \mathbf{g}$ from $\mathbf{f} >_{\mathrm{str}} \mathbf{g}$. This property is not satisfied without this requirement. (For the proof consider two functionals $\mathbf{f}, \mathbf{g}$ of level 2 satisfying for all $\mathbf{x} \in \mathcal{S}$ the inequality $\mathbf{f}(\mathbf{x}) > \mathbf{g}(\mathbf{x})$. Now modify these functionals on the non–strict, but monotonic functions, e.g. by giving $\mathbf{f}$ on $[\![\lambda z\, 0]\!]$ the value 0 and $\mathbf{g}$ on $[\![\lambda z\, 0]\!]$ the value 1.)

From the definition it is clear that from $\mathbf{f} \in \mathcal{S}$ and $\mathbf{x} \in \mathcal{M} \cup \{0,1\}$ we can conclude $\mathbf{f}(\mathbf{x}) \in \mathcal{S}$. Furthermore from $\mathcal{S} \subseteq \mathcal{M}$ we get immediately $\mathbf{f} >_{\mathrm{mon}} \mathbf{g} \implies \mathbf{f} >_{\mathrm{str}} \mathbf{g}$.

Theorem. *Let M be a term in β normal form and $\Box \in \mathrm{FV}(M)$. Then for any strict environment U and all $\mathbf{f}, \mathbf{g} \in \mathcal{M}$*

$$\mathbf{f} >_{\mathrm{mon}} \mathbf{g} \implies [\![M]\!]_{U[\Box \mapsto \mathbf{f}]} >_{\mathrm{str}} [\![M]\!]_{U[\Box \mapsto \mathbf{g}]}.$$

Proof by induction on M. Let M be in long normal form. Let $\mathbf{f}, \mathbf{g} \in \mathcal{M}$ with $\mathbf{f} >_{\mathrm{mon}} \mathbf{g}$ be given. Then $\geq$ holds by Lemma 1(ii).

Case $\lambda\vec{x}.\Box\vec{M}$. Let $\vec{\mathbf{x}} \in \mathcal{S}$ and $V := U[\vec{x} \mapsto \vec{\mathbf{x}}]$. From $\mathbf{f} >_{\mathrm{mon}} \mathbf{g}$ we get

$$\mathbf{f}([\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{f}]}) >_{\mathrm{mon}} \mathbf{g}([\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{f}]})$$

and therefore also $>_{\mathrm{str}}$. Furthermore from $\mathbf{f} >_{\mathrm{mon}} \mathbf{g}$ we obtain $\mathbf{f} \geq \mathbf{g}$, hence $[\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{f}]} \geq [\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{g}]}$. Now $\mathbf{g}([\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{f}]}) \geq \mathbf{g}([\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{g}]})$ follows because $\mathbf{g} \in \mathcal{M}$.

Case $\lambda\vec{x}.y\vec{M}$ with $y \neq \Box$. Let $\vec{\mathbf{x}} \in \mathcal{S}$ and $V := U[\vec{x} \mapsto \vec{\mathbf{x}}]$. For any i with $\Box \in \mathrm{FV}(M_i)$ we have $[\![M_i]\!]_{V[\Box \mapsto \mathbf{f}]} >_{\mathrm{str}} [\![M_i]\!]_{V[\Box \mapsto \mathbf{g}]}$ by IH, hence $[\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{f}]} >_{\mathrm{str}} [\![\vec{M}]\!]_{V[\Box \mapsto \mathbf{g}]}$. Since $V(y) \in \mathcal{S}$, we obtain $[\![y\vec{M}]\!]_{V[\Box \mapsto \mathbf{f}]} > [\![y\vec{M}]\!]_{V[\Box \mapsto \mathbf{g}]}$.

Case $\lambda\vec{x}.\langle M_0, M_1\rangle$. Then $\Box \in \mathrm{FV}(M_i)$ for some $i \in \{0,1\}$. Let $\vec{\mathbf{x}} \in \mathcal{S}$ and $V := U[\vec{x} \mapsto \vec{\mathbf{x}}]$. By IH $[\![M_i]\!]_{V[\Box \mapsto \mathbf{f}]} >_{\mathrm{str}} [\![M_i]\!]_{V[\Box \mapsto \mathbf{g}]}$ for this i. □

This theorem shows that the strict functionals form an interesting class. In the rest of this section we will explore the strict functionals and in the next section it will be shown how to use them in termination proofs. The first question is of course, whether there exist such functionals at all. To construct strict functionals, we surely need them on the base types. Hence we assume that for any tuple $\iota_1, \ldots, \iota_n, \iota$ of base types we are given a strict function $+$ of type $\iota_1 \to \ldots \to \iota_n \to \iota$ (written in infix notation, or as prefix $\sum$; we will write 0^ι for $+^\iota$.) Using this $+$, we simultaneously define special functionals $\mathbf{S}^\sigma$ (a *strict* functional of type σ for any σ) and $\mathbf{M}_\sigma$ (a *measure* functional of type $\sigma \to o$), where o is one of the base types. In this definition, $\vec{\mathbf{S}}^{\vec{\rho}}$ denotes $\mathbf{S}^{\rho_1}, \cdots, \mathbf{S}^{\rho_n}$, and $\vec{\mathbf{M}}(\vec{\mathbf{f}})$ is to be read as $\mathbf{M}(\mathbf{f}_{i_1}), \cdots, \mathbf{M}(\mathbf{f}_{i_k})$, where $\mathbf{f}_{i_1}, \cdots, \mathbf{f}_{i_k}$ are the proper arguments among $\vec{\mathbf{f}}$, i.e. not the 0 and 1 used for projections. These shortcuts will be used frequently. In the last equation $\mathbf{S}^\sigma(\vec{\mathbf{f}})$ is to be of base type.

Definition.

$$\begin{aligned}
\mathbf{M}_{\vec{\sigma} \to \iota}(\mathbf{f}) &:= +^{\iota \to o}(\mathbf{f}(\vec{\mathbf{S}}^{\vec{\sigma}})) \\
\mathbf{M}_{\vec{\sigma} \to \rho \times \tau}(\mathbf{f}) &:= \mathbf{M}_\rho(\mathbf{f}(\vec{\mathbf{S}}^{\vec{\sigma}}, 0)) + \mathbf{M}_\tau(\mathbf{f}(\vec{\mathbf{S}}^{\vec{\sigma}}, 1)) \\
\mathbf{S}^\sigma(\vec{\mathbf{f}}) &:= \sum \vec{\mathbf{M}}(\vec{\mathbf{f}})
\end{aligned}$$

In examples, we assume that the $+^{\vec{\iota} \to \iota}$ are chosen in such a way that $0^{\iota_1} + \cdots + 0^{\iota_n} = 0^\iota$ holds for any combination of base types and $+^{\iota \to \iota}$ is the identity. For instance, we may take 0 in $\mathbb{N}$ with usual order and addition, or else take the empty list in $\mathbb{N}^*$ and let $+$ be concatenation and $>$ be the comparison of lengths. Under

these assumptions $\mathbf{M}_\sigma = \mathbf{S}^{\sigma\to o}$. By induction on the types one can see immediately that $\mathbf{M}(\mathbf{S}) = 0$.

Here are some examples:

$$\begin{aligned}
\mathbf{S}^\iota &= 0^\iota,\\
\mathbf{S}^{\iota\to\iota}(\mathbf{x}) &= \mathbf{x},\\
\mathbf{S}^{(\iota\to\iota)\to\iota}(\mathbf{f}) &= \mathbf{f}(0),\\
\mathbf{S}^{((\iota\to\iota)\to\iota\to\iota)\to(\iota\to\iota)\to\iota\to\iota}(\mathbf{F},\mathbf{f},\mathbf{x}) &= \mathbf{F}(\mathbf{S}^{\iota\to\iota},0) + \mathbf{f}(0) + \mathbf{x},\\
\mathbf{S}^{\iota\times\iota\to\iota}\langle\mathbf{x},\mathbf{y}\rangle &= \mathbf{x}+\mathbf{y},\\
\mathbf{S}^{\iota\times\iota\to\iota\times\iota}\langle\mathbf{x},\mathbf{y}\rangle &= \langle\mathbf{x}+\mathbf{y},\mathbf{x}+\mathbf{y}\rangle,\\
\mathbf{S}^{\sigma\times\tau} &= \langle\mathbf{S}^\sigma,\mathbf{S}^\tau\rangle.
\end{aligned}$$

Lemma 2. *For any type ρ, both $\mathbf{M}_\rho$ and $\mathbf{S}^\rho$ are strict.*

Proof by simultaneous induction on the type ρ. If ρ is a base type, then $\mathbf{S}^\iota = 0^\iota$ and $\mathbf{M}_\iota(n) = +^{\iota\to o}(n)$. Strictness is clear. So let ρ be some compound type.

Let $\vec{\mathbf{x}},\vec{\mathbf{y}} \in \mathcal{M}\cup\{0,1\}$ be given, with $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$ and $\mathbf{S}^\rho(\vec{\mathbf{x}})$ of base type. The $\vec{\mathbf{M}}$ in $\vec{\mathbf{M}}(\vec{\mathbf{x}})$ all have type smaller than ρ, so they are strict by IH, hence monotonic. This yields that $\vec{\mathbf{M}}(\vec{\mathbf{x}}) \geq \vec{\mathbf{M}}(\vec{\mathbf{y}})$, so also $\sum\vec{\mathbf{M}}(\vec{\mathbf{x}}) \geq \sum\vec{\mathbf{M}}(\vec{\mathbf{y}})$. This proves monotonicity of $\mathbf{S}^\rho$. Next, assume that $\vec{\mathbf{x}} >_{\mathrm{str}} \vec{\mathbf{y}}$ holds. Then $\geq$ holds, and for some i, $\mathbf{x}_i >_{\mathrm{str}} \mathbf{y}_i$. By IH $\mathbf{M}(\mathbf{x}_i) > \mathbf{M}(\mathbf{y}_i)$, so $\sum\vec{\mathbf{M}}(\vec{\mathbf{x}}) > \sum\vec{\mathbf{M}}(\vec{\mathbf{y}})$. This proves that $\mathbf{S}^\rho$ is strict.

Next we prove strictness of $\mathbf{M}_\rho$. Let $\rho = \vec{\rho}\to\tau$, with τ not an arrow type. Let $\mathbf{f},\mathbf{g} \in \mathcal{M}_\rho$ be given. Note that the $\vec{\mathbf{S}}^{\vec{\rho}}$ are strict by IH, hence they are monotonic too. So if $\mathbf{f} \geq \mathbf{g}$, then $\mathbf{f}(\vec{\mathbf{S}}^{\vec{\rho}}) \geq \mathbf{g}(\vec{\mathbf{S}}^{\vec{\rho}})$. Moreover, if $\mathbf{f} >_{\mathrm{str}} \mathbf{g}$ then $\mathbf{f}(\vec{\mathbf{S}}^{\vec{\rho}}) >_{\mathrm{str}} \mathbf{g}(\vec{\mathbf{S}}^{\vec{\rho}})$. In case τ is a base type, this proves both monotonicity and strictness of $\mathbf{M}_\rho$. Otherwise, $\tau = \tau_0\times\tau_1$, and we use that $\mathbf{M}_{\tau_0}$ and $\mathbf{M}_{\tau_1}$ are strict by IH, and hence monotonic too. The monotonicity of $\mathbf{M}_\rho$ then follows from monotonicity of the projections and $+$. For strictness, note that either $\mathbf{f}(\vec{\mathbf{S}},0) >_{\mathrm{str}} \mathbf{g}(\vec{\mathbf{S}},0)$ or $\mathbf{f}(\vec{\mathbf{S}},1) >_{\mathrm{str}} \mathbf{g}(\vec{\mathbf{S}},1)$. For the other component $\geq$ holds. Now strictness of $\mathbf{M}_\rho$ follows from strictness of the $\mathbf{M}_{\tau_i}$ and of $+^{o\to o\to o}$. □

The success of the method, to be developed in Section 2, depends on finding strict functionals. By now, we have only seen the $\mathbf{S}$ functionals as examples. The following lemma enables us to find a lot more strict functionals:

Lemma 3. *For any strict functional $\mathbf{G}$ and monotonic functional $\mathbf{H}$, the functional $\mathbf{F}$ defined by $\mathbf{F}(\vec{\mathbf{x}}) := \mathbf{G}(\vec{\mathbf{x}}) + \mathbf{H}(\vec{\mathbf{x}})$, is strict.*

Proof. Let $\vec{\mathbf{x}} >_{\mathrm{str}} \vec{\mathbf{y}}$ for some monotonic $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$. Then $\mathbf{G}(\vec{\mathbf{x}}) > \mathbf{G}(\vec{\mathbf{y}})$ (by strictness of $\mathbf{G}$). By the definition of $>_{\mathrm{str}}$, we obtain $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$, hence by monotonicity of $\mathbf{H}$, $\mathbf{H}(\vec{\mathbf{x}}) \geq \mathbf{H}(\vec{\mathbf{y}})$. This yields $\mathbf{F}(\vec{\mathbf{x}}) > \mathbf{F}(\vec{\mathbf{y}})$. □

Note that this result doesn't hold if one drops the requirement $\mathbf{f} \geq \mathbf{g}$ in the definition of $\mathbf{f} >_{\mathrm{str}} \mathbf{g}$. So this addition is motivated by the fact that it enables us to find more strict functionals easily. We proceed with showing that one cannot get smaller strict functionals.

Lemma 4. *Consider the special case that the only ground domain is* $\mathbb{N}$ *with usual ordering and addition. Then for any* $\mathbf{f} \in \mathcal{S}_\sigma$, $\mathbf{f} \geq \mathbf{S}^\sigma$.

Proof. We use an operation $\mathbf{L}_\sigma$ (*lower* by 1) on functionals, defined by induction on the type. $\mathbf{L}_\sigma$ takes two arguments, a functional $\mathbf{f}$ of type σ and a sequence $\vec{\mathbf{a}}$ in $\mathcal{M} \cup \{0,1\}$, such that $\mathbf{f}(\vec{\mathbf{a}})$ is of base type. The result of $\mathbf{L}_\sigma(\mathbf{f}, \vec{\mathbf{a}})$ will be of type σ. We will write $\mathbf{L}_{\vec{\mathbf{a}}}(\mathbf{f})$ for $\mathbf{L}_\sigma(\mathbf{f}, \vec{\mathbf{a}})$.

$$\begin{array}{rcl} \mathbf{L}_\varepsilon(n) & := & \begin{cases} 0 & \text{if } n = 0 \\ n-1 & \text{otherwise} \end{cases} \\ \mathbf{L}_{\langle \mathbf{a}, \vec{\mathbf{a}} \rangle}(\mathbf{f}, \mathbf{x}) & := & \mathbf{L}_{\vec{\mathbf{a}}}(\mathbf{f}(\mathbf{x})) \\ \mathbf{L}_{\langle 0, \vec{\mathbf{a}} \rangle}(\langle \mathbf{x}, \mathbf{y} \rangle) & := & \langle \mathbf{L}_{\vec{\mathbf{a}}}(\mathbf{x}), \mathbf{y} \rangle \\ \mathbf{L}_{\langle \mathbf{1}, \vec{\mathbf{a}} \rangle}(\langle \mathbf{x}, \mathbf{y} \rangle) & := & \langle \mathbf{x}, \mathbf{L}_{\vec{\mathbf{a}}}(\mathbf{y}) \rangle. \end{array}$$

Note that the $\vec{\mathbf{a}}$ is only used to know which of the components of a product to lower. With induction on the types, it is easy to see that for any $\vec{\mathbf{a}}$ and monotonic $\mathbf{x}$,

(i) $\mathbf{L}_{\vec{\mathbf{a}}}$ is monotonic, and

(ii) $\mathbf{M}(\mathbf{L}_{\vec{\mathbf{a}}}(\mathbf{x})) = \mathbf{L}_\varepsilon(\mathbf{M}(\mathbf{x}))$.

We now prove the lemma by a main induction on σ. For the base type, we have to show that $m \geq 0$ for $m \in \mathbb{N}$, which clearly holds. If $\sigma = \rho \times \tau$, observe that by IH, for any strict pair $\langle \mathbf{x}, \mathbf{y} \rangle$, $\langle \mathbf{x}, \mathbf{y} \rangle \geq \langle \mathbf{S}^\rho, \mathbf{S}^\tau \rangle$, and that the latter equals $\mathbf{S}^\sigma$. If $\sigma = \rho \to \tau$, we have to prove that for monotonic $\mathbf{x}$, $\mathbf{f}(\mathbf{x}) \geq \mathbf{S}^\sigma(\mathbf{x})$. This is proved with induction on $\mathbf{M}(\mathbf{x})$.

If $\mathbf{M}(\mathbf{x}) = 0$, we use that $\mathbf{f}(\mathbf{x})$ is strict, hence $\mathbf{f}(\mathbf{x}) \geq \mathbf{S}^\tau$ (main IH). Now for monotonic $\vec{\mathbf{x}}$ we obtain $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq \mathbf{S}^\tau(\vec{\mathbf{x}}) = \mathbf{M}(\mathbf{x}) + \mathbf{S}^\tau(\vec{\mathbf{x}}) = \mathbf{S}^\sigma(\mathbf{x}, \vec{\mathbf{x}})$.

If $\mathbf{M}(\mathbf{x}) = n + 1$, we can find $\vec{\mathbf{a}}$ with elements among $\mathbf{S}$ and 0 and 1, such that $\mathbf{x}(\vec{\mathbf{a}}) \geq 1$. Define $\mathbf{y} := \mathbf{L}_{\vec{\mathbf{a}}}(\mathbf{x})$. By (i) above, $\mathbf{y}$ is monotonic. We first show, that $\mathbf{x} >_{\text{str}} \mathbf{y}$. It suffices to show that $\mathbf{x}(\vec{\mathbf{z}}) > \mathbf{y}(\vec{\mathbf{z}})$, where $\vec{\mathbf{z}}$ is obtained from $\vec{\mathbf{a}}$ by replacing the real arguments by arbitrary strict functionals. (i.e. the 0 and 1s for projections are not replaced.) By IH, we have that $\vec{\mathbf{z}} \geq \vec{\mathbf{S}}$, hence $\mathbf{x}(\vec{\mathbf{z}}) \geq \mathbf{x}(\vec{\mathbf{a}}) \geq 1$. Hence $\mathbf{y}(\vec{\mathbf{z}}) = \mathbf{x}(\vec{\mathbf{z}}) - 1$.

Now we show that for monotonic $\vec{\mathbf{x}}$, $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq \mathbf{S}^\sigma(\mathbf{x}, \vec{\mathbf{x}})$. Note that $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) > \mathbf{f}(\mathbf{y}, \vec{\mathbf{x}})$, because $\mathbf{f}$ is strict. By (ii) above $\mathbf{M}(\mathbf{y}) = n$. Hence we can apply the inner IH, and obtain $\mathbf{f}(\mathbf{y}, \vec{\mathbf{x}}) \geq \mathbf{S}^\sigma(\mathbf{y}, \vec{\mathbf{x}}) = n + \mathbf{S}(\vec{\mathbf{x}})$, hence $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq n + 1 + \mathbf{S}(\vec{\mathbf{x}}) = \mathbf{S}^\sigma(\mathbf{x}, \vec{\mathbf{x}})$. □

So we have found out that $\mathbf{S}(\vec{\mathbf{x}}) + \mathbf{H}(\vec{\mathbf{x}})$ is strict in $\vec{\mathbf{x}}$ for monotonic $\mathbf{H}$ and that $\mathbf{S}$ is a minimal strict one. One might wonder if all strict functionals have the form $\mathbf{S}$ + monotonic. However, this is not the case. Consider $\mathbf{F}(\mathbf{f}) := \mathbf{f}(1)$, of type $(o \to o) \to o$. This is clearly strict. But the difference between $\mathbf{F}$ and $\mathbf{S}$ is not monotonic: Put $\mathbf{f}(n) := \max(1, n)$ and $\mathbf{g}(n) := n$. Then $\mathbf{f}$ and $\mathbf{g}$ are both monotonic, and $\mathbf{f} \geq \mathbf{g}$. But $\mathbf{g}(1) - \mathbf{g}(0) > \mathbf{f}(1) - \mathbf{f}(0)$.

2. Termination

To be able to apply the theorem above to prove termination we of course need to know that $>_{\mathrm{str}}$ is a well–founded partial ordering on any $\mathcal{T}_\rho$. This can be proved if we assume that for the base types ι we are given domains $\mathcal{T}_\iota$ together with well–founded (partial) orderings $>_\iota$.

Proposition. *$>_{\mathrm{str}}$ is well-founded on any $\mathcal{T}_\rho$.*

Proof. Let $(\mathbf{x}_i)_{i\in\mathbb{N}}$ of type ρ be given. Consider $(\mathbf{M}_\rho(\mathbf{x}_i))_{i\in\mathbb{N}}$. □

Following [7] we define a higher order term rewrite system (HRS) to be given by rules $L \mapsto R$ with closed terms L, R of the same type ρ. Then $M_1 \to M_2$ (M_1 rewrites to M_2) is defined to mean that we have a β–normal term M with $\Box \in \mathrm{FV}(M)$ such that for some rule $L \mapsto R$

$$M_1 = M[L/\Box]{\downarrow_\beta} \qquad \text{and} \qquad M_2 = M[R/\Box]{\downarrow_\beta}.$$

Here $N{\downarrow_\beta}$ denotes the β–normal form of the term N; β is defined as usual for arrow types, and for product types by the two rules $\pi_i\langle M_0, M_1\rangle \mapsto M_i$.

Note that we only require from L, R that they are closed terms of the same type. Closedness is not a restriction, but it avoids substitutions in the definition of a rewrite step. If L and R are not closed, one can simulate the step $M[L^\sigma] \to M[R^\sigma]$ by $M[(\lambda\vec{x}.L)\vec{x}^\sigma] \to M[(\lambda\vec{x}.R)\vec{x}^\sigma]$, where $\vec{x}$ is the list of variables occurring in l or r. Hence this notion of an HRS is quite liberal (and e.g. strictly includes the one given by Nipkow in [4]). The reason for this liberality is of course that termination results get stronger that way. See [8] for a comparison with other higher-order rewrite formats.

Example. Consider the rule $\lambda x.x + x \mapsto \lambda x\, x$. Then

$$\lambda u, v.c(\lambda w.wu + wu)(v + v) \to \lambda u, v.c(\lambda w.wu)v$$

using the term $M := \lambda u, v.c(\lambda w.\Box(wu))(\Box v)$.

Now we obtain as in [7] the following method to prove termination of higher order rewrite systems.

(1) For the base types ι choose domains $\mathcal{T}_\iota$ together with well–founded (partial) orders $>_\iota$. Furthermore find for any tuple $\iota_1, \ldots, \iota_n, \iota$ of base types a strict function $+$ of type $\iota_1 \to \ldots \to \iota_n \to \iota$.

(2) Find an appropriate strict interpretation of the constants.

(3) For any rule $L \mapsto R$ of the higher order rewrite system show that

$$[\![L]\!] >_{\mathrm{mon}} [\![R]\!].$$

Theorem. *Any HRS satisfying (1)–(3) is terminating.*

Proof. Assume that we have $(M_i)_{i\in\mathbb{N}}$ such that $M_i \to M_{i+1}$ for all $i \in \mathbb{N}$. Let U be a strict interpretation. Then we obtain

$$\begin{array}{rcll} [\![M_i]\!]_U & = & [\![M[L/\Box]]\!]_U & \\ & = & [\![M]\!]_{U[\Box \mapsto [\![L]\!]]} & \\ & >_{\mathrm{str}} & [\![M]\!]_{U[\Box \mapsto [\![R]\!]]} & \text{since } [\![L]\!] >_{\mathrm{mon}} [\![R]\!] \\ & = & [\![M[R/\Box]]\!]_U & \\ & = & [\![M_{i+1}]\!]_U. & \end{array}$$

This contradicts the well–foundedness of $>_{\mathrm{str}}$. □

In Section 3, termination of Gödel's T is proved using this method. Section 4 contains a termination proof for the proper reductions and permutative conversions on derivation terms of first order logic. We first treat a well known small example, to illustrate the use of the proposed strategy to prove termination of HRSs.

Consider terms built up from the constants

$$\begin{array}{rlrl} \mathrm{nil} & : o & \mathrm{append} & : o \to o \to o \\ \mathrm{cons} & : o \to o \to o & \mathrm{map} & : (o \to o) \to o \to o. \end{array}$$

The types are chosen such that e.g. $\mathrm{map}(\lambda x\, \mathrm{append}(x, x), \ell)$ is well typed. Terms of type o represent finite lists of lists. The functions map and append are defined via the following rewrite rules (for readability, we drop the initial λs):

$$\begin{array}{rcll} \mathrm{append}(\mathrm{nil}, \ell) & \mapsto & \ell & \text{(i)} \\ \mathrm{append}(\mathrm{cons}(k, \ell), m) & \mapsto & \mathrm{cons}(k, \mathrm{append}(\ell, m)) & \text{(ii)} \\ \mathrm{map}(f, \mathrm{nil}) & \mapsto & \mathrm{nil} & \text{(iii)} \\ \mathrm{map}(f, \mathrm{cons}(k, \ell)) & \mapsto & \mathrm{cons}(f(k), \mathrm{map}(f, \ell)) & \text{(iv)} \\ \mathrm{append}(\mathrm{append}(k, \ell), m) & \mapsto & \mathrm{append}(k, \mathrm{append}(\ell, m)) & \text{(v)} \\ \mathrm{map}(f, \mathrm{append}(\ell, k)) & \mapsto & \mathrm{append}(\mathrm{map}(f, \ell), \mathrm{map}(f, k)) & \text{(vi)} \end{array}$$

To prove termination, we have to satisfy (1), (2) and (3) above. For the ground domain, we choose $\mathbb{N}$, with the usual order and addition. The interpretation of the constants is specified in the following way:

$$\begin{array}{rlrl} [\![\mathrm{nil}]\!] & := 1 & & \\ [\![\mathrm{cons}]\!](m, n) & := m + n + 1 & [\![\mathrm{map}]\!](f, n) & := \sum_{i=0}^{n} f(i) + 3n + 1 \\ [\![\mathrm{append}]\!](m, n) & := 2m + n + 2 & & \end{array}$$

The interpretations of nil, cons and append are obviously strict. Strictness of $[\![\mathrm{map}]\!]$ follows e.g. by Lemma 3, if we write its definition as

$$\big(f(0) + n\big) + \big(\sum_{i=1}^{n} f(i) + 2n + 1\big).$$

Hence (1) and (2) are fulfilled. We still have to check (3). In the sequel k, ℓ, m, f are arbitrary values for the corresponding variables. Note that f ranges over

monotonic functionals. For rule (v) e.g. the check boils down to the true inequality $2 \cdot (2\ell + k + 2) + m + 2 > 2\ell + (2k + m + 2) + 2$. We don't present all calculations here, but let us yet verify the most difficult one, rule (vi):

$$\begin{aligned}
& [\![\mathrm{map}(f, \mathrm{append}(\ell, k))]\!] \\
= & \sum_{i=0}^{2\ell+k+2} f(i) + 3 \cdot (2\ell + k + 2) + 1 \\
= & \sum_{i=0}^{\ell} f(i) + \sum_{i=\ell+1}^{2\ell+1} f(i) + \sum_{i=2\ell+2}^{2\ell+k+2} f(i) + 6\ell + 3k + 7 \\
> & \sum_{i=0}^{\ell} f(i) + \sum_{i=0}^{\ell} f(i) + \sum_{i=0}^{k} f(i) + 6\ell + 3k + 5 \quad \text{because } f \text{ is monotonic} \\
= & \; 2 \cdot (\sum_{i=0}^{\ell} f(i) + 3\ell + 1) + (\sum_{i=0}^{k} f(i) + 3k + 1) + 2 \\
= & \; [\![\mathrm{append}(\mathrm{map}(f, \ell), \mathrm{map}(f, k))]\!]
\end{aligned}$$

For all rules, this relation between left- and right hand side hold. Therefore the HRS under consideration is terminating.

3. Example: Higher order primitive recursion

We now apply this method to prove termination for the canonical rules associated with higher order primitive recursion from Gödel's T. These are based on constants Rec of type $\rho \to (o \to \rho \to \rho) \to o \to \rho$, for any type ρ.

$$\begin{aligned}
\mathrm{Rec}(g, h, 0) &\mapsto g, \\
\mathrm{Rec}(g, h, s(x)) &\mapsto h(x, \mathrm{Rec}(g, h, x)).
\end{aligned}$$

As ground domain we choose $\mathbb{N}$ with the usual addition $+$ and the usual ordering $>$. Then (1) is clearly satisfied. For (2) we choose a strict interpretation of the constants Rec, as follows.

$$\begin{aligned}
[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, 0, \vec{\mathbf{x}}) &= \mathbf{g}(\vec{\mathbf{x}}) + \mathbf{S}(\mathbf{g}, \mathbf{h}, \vec{\mathbf{x}}) + 1, \\
[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n+1, \vec{\mathbf{x}}) &= \mathbf{h}(n, [\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n), \vec{\mathbf{x}}) + [\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}}) + 1.
\end{aligned}$$

The strictness of $[\![\mathrm{Rec}]\!]$ can be seen as follows.

First we show that $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n)$ for $\mathbf{g}, \mathbf{h} \in \mathcal{M}$ and any n is monotonic, by induction on n. *Case* 0. $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, 0)$ is monotonic, since $\mathbf{g}$ is. *Case* $n+1$. $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n+1)$ is monotonic, since $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n)$ and $\mathbf{h}$ are monotonic.

Hence we get $[\![\mathrm{Rec}]\!] \in \mathcal{M}$ as follows. Let $\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}} \in \mathcal{M}$. It suffices to show that by decreasing these arguments in $\mathcal{M}$ in the sense of $\geq$ the value $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}})$ will get at most smaller. This clearly holds if n is decreased. For the other possibilities we fix n. In the case $n = 0$ the claim is obvious, in case $n+1$ we need the monotonicity of $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n)$.

Now we can show that $[\![\mathrm{Rec}]\!]$ is strict. $[\![\mathrm{Rec}]\!] \in \mathcal{M}$ has already been proved. Let $\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}} \in \mathcal{M}$. It remains to show that by decreasing exactly one of these arguments in $\mathcal{M}$ in the sense of $>_{\mathrm{str}}$ the value $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}})$ gets strictly smaller. This again clearly holds if n is decreased. For the other possibilities we fix n and use Lemma 3:

First note that $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}}) = \mathbf{S}(\mathbf{g}, \mathbf{h}, \vec{\mathbf{x}}) + \mathbf{H}(\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}})$, where $\mathbf{H}$ is defined by

$$\begin{aligned} \mathbf{H}(\mathbf{g}, \mathbf{h}, 0, \vec{\mathbf{x}}) &= \mathbf{g}(\vec{\mathbf{x}}) + 1, \\ \mathbf{H}(\mathbf{g}, \mathbf{h}, n+1, \vec{\mathbf{x}}) &= \mathbf{h}(n, \mathbf{S}(\mathbf{g}, \mathbf{h}) \oplus \mathbf{H}(\mathbf{g}, \mathbf{h}, n), \vec{\mathbf{x}}) + \mathbf{H}(\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}}) + 1; \end{aligned}$$

here we have written $\mathbf{x} \oplus \mathbf{y}$ for the functional which takes the value $\mathbf{x}(\vec{\mathbf{z}}) + \mathbf{y}(\vec{\mathbf{z}})$ on $\vec{\mathbf{z}}$. This can be proved easily by induction on n. Since $\mathbf{H} \in \mathcal{M}$ can be proved just as we proved $[\![\mathrm{Rec}]\!] \in \mathcal{M}$ above, it follows from Lemma 3 that $[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n, \vec{\mathbf{x}})$ is strict for fixed n.

For the proof of (3) let us first consider the rule $\mathrm{Rec}(g, h, 0) \mapsto g$. We have to show that for monotonic $\mathbf{g}, \mathbf{h}, \vec{\mathbf{x}}$ we have

$$[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, 0, \vec{\mathbf{x}}) > \mathbf{g}(\vec{\mathbf{x}}).$$

This holds because of the summand 1 in the first defining equation for $[\![\mathrm{Rec}]\!]$. For the rule $\mathrm{Rec}(g, h, s(x)) \mapsto h(x, \mathrm{Rec}(g, h, x))$ we have to show that for monotonic $\mathbf{g}, \mathbf{h}, \vec{\mathbf{x}}$ we have

$$[\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n+1, \vec{\mathbf{x}}) > \mathbf{h}(n, [\![\mathrm{Rec}]\!](\mathbf{g}, \mathbf{h}, n), \vec{\mathbf{x}}).$$

This clearly holds because of the summand 1 in the second equation of the definition for $[\![\mathrm{Rec}]\!]$.

4. Example: Permutative Conversions

The next example comes from proof theory in the style of Prawitz. In [5] several reductions are given, to bring proofs into a certain normal form. These are divided in *proper reductions* and *permutative conversions*. Strong normalization is then proved via a refined notion of strong computability, *strong validity*. In [1] also examples taken from proof theory occur. There a normalization proof is given via hereditarily monotonic functionals, but the permutative conversions are not dealt with. We also refer to [2] for another adaptation of Gandy's approach, which can be extended to the full calculus including permutative conversions (See [2, Exc. 2.C.10]). Instead of bounding reduction lengths by functionals, Girard uses the length of a specific reduction path, given by a weak normalization theorem for the full calculus.

We present a termination proof for the whole calculus, including the permutative conversions. However, for simplicity we don't include disjunction. We first reduce the calculus with derivation terms to an HRS. Termination of this HRS is proved

using the method of Section 2. The translation to an HRS is such that termination of the derivation terms immediately follows.

Definition. *Derivation terms are defined simultaneously with the set of free assumption variables* (FA) *occurring in them. We use A, B, C for formulae; d, e, f for derivation terms; r, s for object terms; x, y for object variables and u, v for assumption variables; i ranges over 0, 1.*

$$
\begin{array}{ll}
u^A & (\lambda x\, d^A)^{\forall x A}, \\
(\lambda u^A\, d^B)^{A\to B} & \quad \textit{provided } x \notin \mathrm{FV}(B) \textit{ for any } u^B \in \mathrm{FA}(d) \\
(d^{A\to B} e^A)^B & (d^{\forall x A(x)} r)^{A(r)} \\
\langle d^A, e^B\rangle^{A\wedge B} & \langle r, d^{A(r)}\rangle^{\exists x A(x)} \\
\pi_i(d^{A_0\wedge A_1})^{A_i} & (\varepsilon x u^A.d^{\exists x A} e^B)^B, \textit{ provided } x \notin \mathrm{FV}(B) \textit{ and} \\
 & \quad x \notin \mathrm{FV}(C) \textit{ for any } v^C \in \mathrm{FA}(e) \setminus \{u\}.
\end{array}
$$

We define $\mathrm{FA}(\varepsilon x u.de) := \mathrm{FA}(d) \cup (\mathrm{FA}(e) \setminus \{u\})$. *In the other cases the set of free assumption variables is defined as usual.*

The following conversion rules are taken from [5]. The first four are the *proper reductions*, the last four are called *permutative conversions*. Again i ranges over 0, 1.

$$
\begin{array}{rlrl}
(\lambda u\, d)e & \mapsto d[u := e] & (\varepsilon x u.de)f & \mapsto \varepsilon x u.d\,(ef) \\
\pi_i\langle d_0, d_1\rangle & \mapsto d_i & \pi_i(\varepsilon x u.de) & \mapsto \varepsilon x u.d\,\pi_i(e) \\
(\lambda x\, d)r & \mapsto d[x := r] & (\varepsilon x u.de)r & \mapsto \varepsilon x u.d\,(er) \\
\varepsilon x u.(\langle r, d\rangle e) & \mapsto e[x, u := r, d] & \varepsilon x u.(\varepsilon y v.de)f & \mapsto \varepsilon x u.d\,\varepsilon y v.ef
\end{array}
$$

To translate this calculus into an HRS, we first have to transform formulae into types. This is done by removing the dependencies on object terms, also called *collapsing*. This technique is also used in [6, p. 560]. Collapsing A will be denoted by A^*. In the following definition, P is a predicate symbol.

$$
\begin{array}{rlrl}
P(\vec{t})^* & = o & (\exists x A)^* & = o \times A^* \\
(A \to B)^* & = A^* \to B^* & (\forall x A)^* & = o \to A^* \\
(A \wedge B)^* & = A^* \times B^* &
\end{array}
$$

Clearly, A^* is a type for any formula A. The difference between implication and quantification disappears. Existential quantifiers and conjunctions are translated into product types.

The derivation terms are translated too. We introduce a new constant $\exists^-$ to model the ε-construct. In the definition of a rewrite step, β-normalization is performed implicitly. To avoid these implicit steps, we introduce another constant I, to block the β-redexes. So for any type σ (and τ) we have the following constants, which make the signature of the HRS we are constructing:

$$
I_\sigma \quad : \sigma \to \sigma \qquad\qquad \exists^-_{\sigma,\tau} \quad : o \times \sigma \to (o \to \sigma \to \tau) \to \tau
$$

To describe the translation precisely, we extend the collapse function on derivation

terms:

$$\begin{array}{rlrl}
(u^A)^* & = u^{A^*} & (d^{A\to B}e)^* & = I_{A^*\to B^*}(d^*, e^*)\\
(\lambda u^A\, d)^* & = \lambda u^{A^*}\, d^* & \pi_i(d^{A\wedge B})^* & = I_{A^*\wedge B^*}(d^*, i)\\
\langle d, e\rangle^* & = \langle d^*, e^*\rangle & (d^{\forall x A}r)^* & = I_{o\to A^*}(d^*, r)\\
(\lambda x\, d)^* & = \lambda x^o\, d^* & (\varepsilon x u^A.de^B)^* & = \exists^-_{A^*,B^*}(d^*, \lambda x^o u^{A^*}.e^*)\\
\langle r, d\rangle^* & = \langle r, d^*\rangle &&
\end{array}$$

Clearly $(d^A)^*$ gives a term of type A^* for any derivation term d. Due to the blocking I, d^* cannot contain subterms MN, with M an assumption variable, an abstraction or a pair. So d^* is in β-normal form, even after substituting β-normal terms for free assumption variables. Furthermore, it is easy to see that $(d[u := e])^* = d^*[u := e^*]$.

Finally, we present the rewrite rules of the HRS. These are all well typed instances of the following schemata; i ranges over 0, 1.

$$\begin{array}{rcll}
I_\sigma(x) & \mapsto & x & \text{(i)}\\
\exists^-_{\sigma,\tau}(\langle r, d\rangle, e) & \mapsto & e(r, d) & \text{(ii)}\\
I_{\sigma\to\tau}(\exists^-_{\rho,\sigma\to\tau}(d, e), f) & \mapsto & \exists^-_{\rho,\tau}(d, \lambda x^o\, u^\rho.I_{\sigma\to\tau}(e(x, u), f)) & \text{(iii)}\\
\pi_i(I_{\sigma_0\times\sigma_1}(\exists^-_{\rho,\sigma_0\times\sigma_1}(d, e))) & \mapsto & \exists^-_{\rho,\sigma_i}(d, \lambda x^o\, u^\rho.\pi_i(I_{\sigma_0\times\sigma_1}(e(x, u)))) & \text{(iv)}\\
\exists^-_{\sigma,\tau}(\exists^-_{\rho,o\times\sigma}(d, e), f) & \mapsto & \exists^-_{\rho,\tau}(d, \lambda x^o\, u^\rho.\exists^-_{\sigma,\tau}(e(x, u), f)) & \text{(v)}
\end{array}$$

It is not difficult to check that if $d \to e$ for derivation terms d and e, then also $d^* \to e^*$ with the rules just described. The first rule deals with proper reductions for $\to$, $\wedge$ and $\forall$; the second with the proper $\exists$–reduction. The third takes care of permutative conversions with $\to$ and $\forall$, the fourth with $\wedge$ and the last rule deals with the permutative conversion for $\exists$. We give as an example the proper $\to$–reduction. Consider the rewrite step $(\lambda u\, d)e \to d[u := e]$. The first derivation term translates to $I(\lambda u\, d^*, e^*)$. Now rule (i) is applicable. Literal replacement yields $(\lambda u\, d^*)e^*$, which has to be rewritten to β-normal form, due to the definition of a rewrite step. This normal form is $d^*[u := e^*]$, which is exactly the translation of the second derivation term.

Next we prove termination of the HRS, by carrying out the strategy of Section 2. As domain we (again) choose $\mathbb{N}$ (with standard order and addition). The interpretation of I is defined by

$$[\![I]\!](\mathbf{f}, \vec{\mathbf{z}}) := \mathbf{S}(\mathbf{f}, \vec{\mathbf{z}}) + \mathbf{f}(\vec{\mathbf{z}}) + 1.$$

This is strict by Lemma 3, and clearly $[\![I]\!](\mathbf{x}) >_{\mathrm{mon}} \mathbf{x}$ for any monotonic $\mathbf{x}$. This already proves termination of the proper reduction rules for $\to$, $\wedge$ and $\forall$ and in particular of the simply typed lambda calculus with products. (Note however, that we used the unique β-normal form of simply typed terms. In fact, weak normalization suffices at meta–level.)

Due to the presence of the permutative conversions, it is more difficult to find a well-suited interpretation of $\exists^-$. We first need auxiliary functionals $\mathbf{A}_\sigma$ of type $\sigma \to \sigma$, which calculate the price of repeated $\to$ and $\times$–eliminations. Here the value

of the blocking constant has to be taken into account. This leads to the following definition:

$$\begin{aligned}
\mathbf{A}_o(n) &:= n+1,\\
\mathbf{A}_{\sigma\to\tau}(\mathbf{f},\mathbf{x}) &:= \mathbf{A}_\tau([\![I]\!](\mathbf{f},\mathbf{x})),\\
\mathbf{A}_{\rho_0\times\rho_1}(\mathbf{f},i) &:= \mathbf{A}_{\rho_i}([\![I]\!](\mathbf{f},i)),\ \text{for } i=0,1.
\end{aligned}$$

With induction on the type and using strictness of $[\![I]\!]$, one easily checks that $\mathbf{A}$ is strict. Also $\mathbf{A}(\mathbf{x}) >_{\mathrm{mon}} \mathbf{x}$ can be proved with induction. Let $\mathbf{A}^n(\mathbf{x})$ denote the n–fold application of $\mathbf{A}$ on $\mathbf{x}$. We write $\mathbf{x}\oplus\mathbf{y}$ for the functional which takes the value $\mathbf{x}(\vec{\mathbf{z}})+\mathbf{y}(\vec{\mathbf{z}})$ on $\vec{\mathbf{z}}$. Now we can define

$$[\![\exists^-_{\sigma,\tau}]\!](\mathbf{d},\mathbf{e}) = \mathbf{A}_\tau^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{e}(\pi_0(\mathbf{d}),\mathbf{S}^\sigma\oplus\pi_1(\mathbf{d}))).$$

Let us first explain the intuition behind this interpretation. Due to the β-rule for $\exists^-$, we need a subterm $\mathbf{e}(\pi_\mathbf{0}(\mathbf{d}),\pi_\mathbf{1}(\mathbf{d}))$. The summand $\mathbf{S}\oplus$ is added to achieve strictness in $\mathbf{e}$. With a permutative conversion, the second argument of the $\exists^-$ gets bigger. After an application of rule (iii), the argument f appears inside the $\exists^-$. Note however, that the type of the involved $\exists^-$ goes down. So the value of an $\exists^-$ of higher type has to count for the value of f, which is still raised by the value of the blocking I. This explains the occurrence of $\mathbf{A}$ (which is defined by induction on the types). The same intuition applies to rule (iv). The last permutative conversion is still more involved. Here the type doesn't go down. The only thing which goes down is the left argument of the $\exists^-$–symbols involved. So the value of $\exists^-$ has to weigh its first argument rather high, to compensate for the increasing second argument. This explains the $2^{\mathbf{S}(\mathbf{d})}$ in the previous definition.

Monotonicity of $[\![\exists^-]\!]$ follows from monotonicity of $\mathbf{A}$. Next strictness is proved. Let $\mathbf{e},\mathbf{f},\mathbf{x},\mathbf{y}$ be monotonic. If $\mathbf{x} >_{\mathrm{str}} \mathbf{y}$, then by monotonicity of $\mathbf{e}$, $\mathbf{e}(\pi_0(\mathbf{x}),\mathbf{S}\oplus\pi_1(\mathbf{x})) \geq \mathbf{e}(\pi_0(\mathbf{y}),\mathbf{S}\oplus\pi_1(\mathbf{y}))$. Furthermore $2^{\mathbf{S}(\mathbf{x})} > 2^{\mathbf{S}(\mathbf{y})}$. Because $\mathbf{A}(\mathbf{x}) >_{\mathrm{mon}} \mathbf{x}$ for all $\mathbf{x}$, it follows that $[\![\exists^-]\!](\mathbf{x},\mathbf{e}) >_{\mathrm{mon}} [\![\exists^-]\!](\mathbf{y},\mathbf{e})$. This proves strictness in the first argument. Next, assume that $\mathbf{e} >_{\mathrm{str}} \mathbf{f}$. Note that both $\pi_0(\mathbf{x})$ and $\mathbf{S}\oplus\pi_1(\mathbf{x})$ are strict (the first is of base type, the second by Lemma 3). Hence $\mathbf{e}(\pi_0(\mathbf{x}),\mathbf{S}\oplus\pi_1(\mathbf{x})) >_{\mathrm{str}} \mathbf{f}(\pi_0(\mathbf{x}),\mathbf{S}\oplus\pi_1(\mathbf{x}))$. Now $[\![\exists^-]\!](\mathbf{x},\mathbf{e}) >_{\mathrm{mon}} [\![\exists^-]\!](\mathbf{x},\mathbf{f})$ follows from strictness of $\mathbf{A}$. This proves strictness in the second argument. Strictness in the next arguments directly follows from strictness of $\mathbf{A}$.

Now we verify condition (3) from Section 2 for the last three rules. First we show this for the proper $\exists^-$–rule. Let $\mathbf{r}$, $\mathbf{d}$ and $\mathbf{e}$ be monotonic. Then, using $\mathbf{A}(\mathbf{x}) >_{\mathrm{mon}} \mathbf{x}$ for monotonic $\mathbf{x}$, we get:

$$[\![\exists^-]\!](\langle\mathbf{r},\mathbf{d}\rangle,\mathbf{e}) >_{\mathrm{mon}} \mathbf{e}(\mathbf{r},\mathbf{S}\oplus\mathbf{d}) \geq \mathbf{e}(\mathbf{r},\mathbf{d}).$$

Hence, in any monotonic environment $[\![\exists^-\langle r,d\rangle e]\!] >_{\mathrm{mon}} [\![erd]\!]$.

Next we verify the same relation for rules (iii) and (iv), permutative conversions for $\to$, $\forall$ and $\wedge$. These two rules can be written as:

$$I(\exists^-(d,e),f) \mapsto \exists^-(d,\lambda x\,u.I(e(x,u),f)),$$

where f is a term or 0 or 1 for the projections.

Let $\mathbf{d}, \mathbf{e}, \mathbf{f}, \vec{\mathbf{z}}$ be monotonic. Put $\mathbf{a} := [\![\exists^-]\!](\mathbf{d}, \mathbf{e})$; $\mathbf{b}(\mathbf{x}, \mathbf{u}) := [\![I]\!](\mathbf{e}(\mathbf{x}, \mathbf{u}), \mathbf{f})$ and $\mathbf{c} := \mathbf{e}(\pi_0(\mathbf{d}), \mathbf{S} \oplus \pi_1(\mathbf{d}))$. Note that $\mathbf{a} \geq \mathbf{A}(\mathbf{c}) >_{\mathrm{mon}} \mathbf{c}$. We have to show that $[\![I]\!](\mathbf{a}, \mathbf{f}, \vec{\mathbf{z}}) > [\![\exists^-]\!](\mathbf{d}, \mathbf{b}, \vec{\mathbf{z}})$.

$$\begin{array}{rcl}
[\![I]\!](\mathbf{a}, \mathbf{f}, \vec{\mathbf{z}}) & = & \mathbf{S}(\mathbf{a}, \mathbf{f}, \vec{\mathbf{z}}) + \mathbf{a}(\mathbf{f}, \vec{\mathbf{z}}) + 1 \\
& > & [\![\exists^-]\!](\mathbf{d}, \mathbf{e}, \mathbf{f}, \vec{\mathbf{z}}) \\
& = & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{c})(\mathbf{f}, \vec{\mathbf{z}}). \\
[\![\exists^-]\!](\mathbf{d}, \mathbf{b}, \vec{\mathbf{z}}) & = & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{b}(\pi_0(\mathbf{d}), \mathbf{S} \oplus \pi_1(\mathbf{d})))(\vec{\mathbf{z}}) \\
& = & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}([\![I]\!](\mathbf{c}, \mathbf{f}))(\vec{\mathbf{z}}).
\end{array}$$

So it suffices to prove that $\mathbf{A}^{n+1}(\mathbf{c})(\mathbf{f}) \geq \mathbf{A}^{n+1}([\![I]\!](\mathbf{c}, \mathbf{f}))$. This is proved by induction on n. If $n = 0$, both terms are equal by definition of $\mathbf{A}$. The successor case uses that $[\![I]\!](\mathbf{x}) >_{\mathrm{mon}} \mathbf{x}$, for all monotonic $\mathbf{x}$:

$$\begin{array}{rcll}
\mathbf{A}^{n+2}(\mathbf{c})(\mathbf{f}) & = & \mathbf{A}(\mathbf{A}^{n+1}(\mathbf{c}), \mathbf{f}) & \\
& = & \mathbf{A}([\![I]\!](\mathbf{A}^{n+1}(\mathbf{c}), \mathbf{f})) & \text{by definition of } \mathbf{A} \\
& > & \mathbf{A}(\mathbf{A}^{n+1}(\mathbf{c})(\mathbf{f})) & \\
& \geq & \mathbf{A}(\mathbf{A}^{n+1}([\![I]\!](\mathbf{c}, \mathbf{f}))) & \text{by IH} \\
& = & \mathbf{A}^{n+2}([\![I]\!](\mathbf{c}, \mathbf{f})). &
\end{array}$$

Finally, we have to prove condition (3) for the $\exists^-\exists^-$ permutative conversion,

$$\exists^-_{\sigma,\tau}(\exists^-_{\rho,o\times\sigma}(d, e), f) \mapsto \exists^-_{\rho,\tau}(d, \lambda x^o\, u^\rho.\exists^-_{\sigma,\tau}(e(x, u), f)).$$

Let $\mathbf{d}, \mathbf{e}, \mathbf{f}$ be monotonic. Put $\mathbf{a} := [\![\exists^-]\!](\mathbf{d}, \mathbf{e})$; $\mathbf{b}(\mathbf{x}, \mathbf{u}) := [\![\exists^-]\!](\mathbf{e}(\mathbf{x}, \mathbf{u}), \mathbf{f})$ and $\mathbf{c} := \mathbf{e}(\pi_0(\mathbf{d}), \mathbf{S} \oplus \pi_1(\mathbf{d}))$. We have to show that $[\![\exists^-]\!](\mathbf{a}, \mathbf{f}) >_{\mathrm{mon}} [\![\exists^-]\!](\mathbf{d}, \mathbf{b})$. Again we have $\mathbf{a} \geq \mathbf{A}(\mathbf{c}) >_{\mathrm{mon}} \mathbf{c}$, so $\mathbf{S}(\mathbf{a}) > \mathbf{S}(\mathbf{c})$. From the left hand side of the rule it is clear that $\mathbf{a}$ is of product type. Hence, $\mathbf{S}(\mathbf{a}) = \mathbf{a}(0) + \mathbf{S}(\mathbf{a}(1))$. Because $\mathbf{S}(\mathbf{a}(1)) > 0$, we obtain $\mathbf{S}(\mathbf{a}) > \mathbf{a}(\mathbf{0}) = \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{c})(\mathbf{0}) \geq 2^{\mathbf{S}(\mathbf{d})} \geq \mathbf{S}(\mathbf{d}) + 1$. Hence

$$2^{\mathbf{S}(\mathbf{a})} \geq 2^{\max\{\mathbf{S}(\mathbf{d})+1, \mathbf{S}(\mathbf{c})\}+1} \geq 1 + 2^{\mathbf{S}(\mathbf{d})} + 2^{\mathbf{S}(\mathbf{c})}.$$

Now we can compute:

$$\begin{array}{rcl}
[\![\exists^-]\!](\mathbf{a}, \mathbf{f}) & = & \mathbf{A}^{2^{\mathbf{S}(\mathbf{a})}}(\mathbf{f}(\pi_0(\mathbf{a}), \mathbf{S} \oplus \pi_1(\mathbf{a}))) \\
& >_{\mathrm{mon}} & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{A}^{2^{\mathbf{S}(\mathbf{c})}}(\mathbf{f}(\pi_0(\mathbf{a}), \mathbf{S} \oplus \pi_1(\mathbf{a})))) \\
& \geq & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{A}^{2^{\mathbf{S}(\mathbf{c})}}(\mathbf{f}(\pi_0(\mathbf{c}), \mathbf{S} \oplus \pi_1(\mathbf{c})))) \\
& = & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}([\![\exists^-]\!](\mathbf{c}, \mathbf{f})) \\
& = & \mathbf{A}^{2^{\mathbf{S}(\mathbf{d})}}(\mathbf{b}(\pi_0(\mathbf{d}), \mathbf{S} \oplus \pi_1(\mathbf{d}))) \\
& = & [\![\exists^-]\!](\mathbf{d}, \mathbf{b}).
\end{array}$$

We have shown that for all rules, the left hand side is greater than the right hand side. Hence the HRS is terminating. This directly implies termination for the calculus with derivation terms presented at the beginning of this section.

References

[1] Robin O. Gandy. Proofs of strong normalization. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 457–477. Academic Press, 1980.

[2] Jean-Yves Girard. *Proof Theory and Logical Complexity.* Bibliopolis, Napoli, 1987.

[3] Gerard Huet and Derek Oppen. Equations and rewrite rules — a survey. In *Formal Language Theory — Perspectives and Open Problems*, pages 349–405. Academic Press, 1980.

[4] Tobias Nipkow. Orthogonal higher–order rewrite systems are confluent. In M. Bezem and J.F. Groote, editors, *Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*, pages 306–317, Berlin, 1993. Springer.

[5] Dag Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 235–307. North–Holland, Amsterdam, 1971.

[6] Anne S. Troelstra and Dirk van Dalen. *Constructivism in Mathematics. An Introduction*, volume 121, 123 of *Studies in Logic and the Foundations of Mathematics.* North–Holland, Amsterdam, 1988.

[7] Jaco van de Pol. Termination proofs for higher–order rewrite systems. In J. Heering, K. Meinke, B. Möller, and T. Nipkow, editors, *Higher–Order Algebra, Logic and Term Rewriting (HOA '93)*, volume 816 of *Lecture Notes in Computer Science*, pages 305–325, Berlin, 1994. Springer.

[8] Vincent van Oostrom. *Confluence for Abstract and Higher–Order Rewriting.* PhD thesis, Vrije Universiteit, Amsterdam, 1994.