[ATTE]MPTS for GENERALISING the RECURSIVE PATH ORDERINGS. → PL

Sam Kamin – Jean-Jacques Lévy

① FUNCTIONALS ON ORDERINGS:

We would like to prove that the following Ackermann system is noetherian.

$$\begin{cases} A(sx, sy) \to A(x, A(sx, y)) \\ A(sx, 0) \to A(x, s0) \\ A(0, x) \to sx \end{cases}$$

With the usual recursive path ordering, this is not possible because $\{sx, sy\} \not\gg \{x, A(sx,y)\}$. Here some lexicographic ordering seems necessary. And surely $(sx, sy) \gg_{lex} (x, A(sx,y))$. However, this is not sufficient since this will work too with

$A(sx, sy) \to A(x, A(sx, sy))$.

But one property of simplifications is that:

$t > u = f\vec{u}$ implies $t > u_i$ for all arguments u_i.

Hence, what could work is bounded lexicographic ordering, i.e.:

$f\vec{t} > f\vec{u}$ if $\vec{t} \gg_{lex} \vec{u}$ and $f\vec{t} > u_i$ for all i.

In that case, the Ackermann system is OK, since both $(sx, sy) \gg_{lex} (x, A(sx,y))$ and $A(sx,sy) > A(sx,y)$ are true. For generalising this remark, one can say that the problem is to [t]ake the recursi[ve] path ordering and to generalise it to vectors of terms (and this way of comparing vectors may depend upon the function symbol of which they are arguments.

For instance, we would like that $f\vec{t} > f\vec{u}$ if $\vec{t} \gg_{lexic} \vec{u}$, and $g\vec{t} > g\vec{u}$ if $\vec{t} \gg_{multiset} \vec{u}$.
In order to say that, we can imagine that we have a functional on relations. Suppose $\mathcal{T}$ is the set of terms, a functional on relations would be a function $\rhd$ from $2^{(\mathcal{T}^2)}$ in $2^{(\mathcal{T}^2)}$. Suppose furthermore that $>$ is a relation on terms, we write $\rhd\!\!>$ for the result of $\rhd$ to argument $>$.
Exemples of such a functional could be:

$f\vec{t} \rhd\!\!> f\vec{u}$ iff $(t_1, t_2, \cdots t_n) >_{lexicographic} (u_1, u_2, \cdots u_n)$

$g\vec{t} \rhd\!\!> g\vec{u}$ iff $(u_1, u_2, \cdots u_n) >_{multiset} (u_1, u_2, \cdots u_n)$

Now, we can induce some recursive path orderi[ng] as follows:

Definition 1: Suppose $\gg$ is an order (strict) on function symbols. Then $t > u$ is defined inductively by the union of the following three cases:

1) $t = f\vec{t}$, $u = g\vec{u}$, $f \gg g$, $t > u_i$ for all i,
2) $t = f\vec{t}$, $t_i \geq u$ for some i,

3) $t = f\vec{t} \rhd\rhd f\vec{u} = u$, $t > u_i$ for all i.

Then, one can show that $>$ is a simplification as soon as the four following conditions on the functional $\rhd$ are true:

a) $\rhd$ preserves transitivity,

b) $\rhd$ preserves irreflexivity,

c) $\rhd$ is continuous (with respect to the subset topology)

d) $t > u$ implies $f(\cdots t \cdots) \rhd\rhd f(\cdots u \cdots)$

Maybe condition (c) needs further explanations (and that's Sam who got the idea of continuity). That is a very weak condition. Let write $>_1 \subset >_2$ if the relation $>_1$ is contained in $>_2$, i.e. $t >_1 u$ implies $t >_2 u$ for all terms t and u (as usual). Then $\rhd$ is continuous iff $t \rhd\rhd u$ implies there is a finite subset of $>$, say $>_f \subset >$, such that $t \rhd >_f u$. That is that, in order to check $t \rhd\rhd u$, one needs to look only at a finite number of pairs $t' > u'$. Conditions (a), (b), (c), (d) are surely true when:

$f\vec{t} = t \rhd\rhd u = f\vec{u}$ iff $\vec{t} \gg_{lexico.} \vec{u}$

$g\vec{t} = t \rhd\rhd u = g\vec{u}$ iff $\{t_1, \cdots t_n\} \gg_{multiset} \{u_1, \cdots u_n\}$

Conditions a, b, c are used for proving that $>$ defined by 1, 2, 3 is a strict order. Condition d is required for showing the monotonicity aspect of $>$ with respect to the structure of terms, i.e. $t > u \Rightarrow f(\cdots t \cdots) > f(\cdots u \cdots)$. It thus can be weakened to be true only when $>$ is the order defined by 1-2-3.

The proof can be sketched by doing first some remarks on fixpoints. Suppose $\tau(>_1, >_2)$ is the continuous functional corresponding to 1, 2, 3. More precisely,

$$t = f\vec{t} \;\; \tau(>_1, >_2) \;\; g\vec{u} = u$$

iff

1) $f > g$ and $t >_1 u_i$ for all i,

2) $t_i \geq_1 u$ for some i,

3) $f = g$, $t >_1 u_i$ for all i and $t \geq_2 u$.

Then the wanted ordering is the least fixpoint of the functional $\lambda > . \tau(>, \rhd>)$, written $\mu > . \tau(>, \rhd>)$. Now (see Manna-Ness-Vuillemin), since both τ and $\rhd$ are continuous, one has:

$$\mu > . \tau(>, \rhd>) = \mu >_1 . \mu >_2 . \tau(>_2, \rhd>_1).$$

In a less esoteric language, it means (using Kleene sequence) that the desired ordering $>$ is the infinite union $\bigcup_{n=0}^{\infty} >^n$ of orderings $>^n$ such that:

$$>^0 = \emptyset, \quad >^{n+1} = \mu > . \tau(>, \rhd >^n) \text{ for } n \geq 0.$$

Jain, this means that $>^{n+1}$ is defined inductively by: $t = f\vec{t} >^{n+1} g\vec{u} = u$ iff

1) $f \rhd g$ and $t >^{n+1} u_i$ for all i,
2) $t_i \geqslant^{n+1} u$ for some i,
3) $t \rhd\!\!> ^n u$, $f = g$, $t >^{n+1} u_i$ for all i.

Now the proof is as follows:

<u>Transitivity</u>: $t > u > v$ implies $t > v$. It is sufficient to prove $>^n$ transitive for all $n \geqslant 0$. Then, if $t > u > v$, $t >^m u >^n v$ for some $m, n \geqslant 0$. Therefore $t >^p u >^p v$ if $p = \max\{m, n\}$. And $t >^p v$. Thus $t > v$.

Therefore, we show $>^n$ transitive. That is:

$t >^n u >^n v$ implies $t >^n v$

by induction on $\langle n, \|t\|, \|u\|, \|v\| \rangle$ ordered in lexicographic ordering. The case $n=0$ is trivial. Thus we try $n > 0$. There are 9 cases with respect to use of rules 1, 2, 3.

$\boxed{1.1}$: $t = f\vec{t}$, $u = g\vec{u}$, $v = h\vec{v}$ with $f \rhd g \rhd h$, $t >^n u_i$ for all i and $u > v_j$ for all j. Then $t >^n u >^n v_i$ and $t >^n v_i$ for all i by induction. Since $f \rhd h$, $t > v$ by rule 1.

$\boxed{1.2}$ $t = f\vec{t}$, $u = g\vec{u}$ with $f \rhd g$, $t >^n u_i$ for all i and $u_j \geqslant^n v$ for some j. Then $t >^n u_i \geqslant^n v$. Therefore $t >^n v$ by induction.

And so on ... until

$\boxed{3.3}$ $t = f\vec{t}$, $u = f\vec{u}$, $v = f\vec{v}$ with $t >^n u_i$ for all i, $u >^n v_i$ for all i and $t \rhd\!\!>^{n-1} u \rhd\!\!>^{n-1} v$. By induction,

we know that $>^{n-1}$ is transitive. Thus, by condition (a) $\rhd\!\!>^{n-1}$ is also transitive. Therefore $t \rhd\!\!>^{n-1} v$. Now $t >^n u >^n v_i$ for all i. Again by induction, $t >^n v_i$ for all i. Thus $t >^n v$ by rule 3. □

<u>Irreflexivity</u>: It is again enough to prove $>^n$ irreflexive for all n. Therefore by induction on $\langle n, \|t\| \rangle$. The case $n=0$ is obvious. Suppose now $t >^n t$. One has 3 cases:

$\boxed{1}$ $t = f\vec{t} >^n t = f\vec{t}$ by rule 1. Impossible, since $\rhd$ is irreflexive.

$\boxed{2}$ $t = f\vec{t} >^n t$ since $t_i \geqslant^n t$ for some i. But $t >^n t_i$ also by rule 2. Therefore $t_i >^n t_i$ by transitivity. Impossible by induction.

$\boxed{3}$ $t = f\vec{t} >^n t = f\vec{t}$ since $t >^n t_i$ for all i and $t \rhd\!\!>^{n-1} t$. But $>^{n-1}$ is irreflexive by induction. Therefore $\rhd\!\!>^{n-1}$ is irreflexive by condition (b). This case is thus impossible. □

<u>Subexpression</u>: $f(\cdots t \cdots) > t$ by rule 2. □

<u>Monotonicity</u>: Suppose $t > u$. Then $f(\cdots t \cdots) \rhd\!\!> f(\cdots u \cdots)$ by condition (d). Since $f(\cdots t \cdots) > t_i$ for all arguments and $t > u$. Then $f(\cdots t \cdots) > u$ by transitivity. Therefore $f(\cdots t \cdots) > f(\cdots u \cdots)$ by rule 3. □

revoir preuve ind // sur Trans, Irreflex.

Now, there are funny examples of systems.

example 1: Ackermann (already done)

Then $A \triangleright s$ is the partial order on function symbols and

$A(t,u) \triangleright\triangleright A(t',u')$ if $t > t'$ or $(t = t', u > u')$

$s\,t \triangleright\triangleright s\,t'$ if $t > t'$

Then

$A(sx, sy) > A(x, A(sx,y))$ by rule 3 if the following subgoals are true

a) $sx > x$ ok by rule 2

b) $A(sx, sy) > x$ ok since $A(sx, sy) > sx > x$.

c) $A(sx, sy) > A(sx, y)$ by rule 3 if again the subgoals:

c1) $sy > y$ ok rule 2

c2) $A(sx, sy) > sx$ ok rule 2.

c3) $A(sx, sy) > y$ ok rule 2 (twice).

Again:

$A(sx, 0) > A(x, s0)$

since $A(sx, 0) > x$, $A(sx, 0) > s0$, $sx > x$.

example 2: Group theory (Knuth-Bendix)

1: $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$ (rule 3)

2: $e \cdot x \to x$ (rule 2)

3: $I(x) \cdot x \to e$ (rule 1) with $\cdot \triangleright e$

4: $I(x) \cdot (x \cdot y) \to y$ (rule 2)

5: $x \cdot (I(x) \cdot y) \to y$ (rule 2)

6: $x \cdot e \to x$ (rule 2)

7: $I(e) \to e$ (rule 2)

8: $I(I x) \to x$ (rule 2)

9: $x \cdot I(x) \to e$ (rule 1)

10: $I(x \cdot y) \to I(y) \cdot I(x)$ (rule 1 + 3) with $I \triangleright \cdot$

The only interesting case is $(x \cdot y) \cdot z \to x \cdot (y \cdot z)$

This is ok if $t \cdot u \triangleright\triangleright t' \cdot u'$ when $t > t'$ or $(t = t', u > u')$ (left lexicographic ordering). Then; it is sufficient to show

a) $x \cdot y > x$ (rule 2)

b) $(x \cdot y) \cdot z > x$ (rule 2 - twice)

c) $(x \cdot y) \cdot z > y \cdot z$ (rule 3)

Again 3 subgoals:

c1) $x \cdot y > y$ (rule 2)

c2) $(x \cdot y) \cdot z > y$ (rule 2 - twice)

c3) $(x \cdot y) \cdot z > z$ (rule 2).

Nice ?? Not ??

example 3: Some Alberto Pettorossi problem in combinatory logic ("A property which guarantees termination in weak combinatory logic and subtree replacement system

Un[illegible]sità di Roma, Istituto di Automatico, R. 78-2 (Nov 78). In combinatory logic, the binary application operator $A(t,u)$ is not usually written. Thus $t\,u$ means $A(t,u)$. More generally:

$t\,u\,u_1\,u_2 \cdots u_n$ means $A(\cdots A(A(A(t,u),u_1),u_2)\cdots,u_n)$

Alberto had the following property and could not prove it in its full generality.

Let $\mathcal{T}$ be the set of combinatory logic terms. Rules considered by Alberto are all of the form:

$F x_1 x_2 \cdots x_n \to t \in \mathcal{T}_n$

where F is a combinator symbol, $x_1, x_2 \cdots x_n$ are n distinct variables and $\mathcal{T}_n$ is the subset of terms defined inductively by:

$\mathcal{T}_0 = \emptyset, \quad \mathcal{T}_n = \{x_1, x_2, \cdots x_n\} \cup \{t\,u \mid t \in \mathcal{T}_{n-1}, u \in \mathcal{T}_n\}$

As examples:

$I x \to x$ is OK

$K x y \to x$ "

$B x y z \to x(yz)$ "

$S x y z \to (xz)(yz)$ is not OK, since z goes too far to the left of an application node. In fact, if $A = SSS$, one can show that AAA is not terminating. Now, one can show in 5 lines that the Alberto conjecture is correct. (although not too much interesting from a programming point of view ??)

One has just to take $t\,u > t'\,u'$ if $t > t'$ or $t = t'$ and $u > u'$. (In fact a stronger proposition may be shown by relaxing the construction of $\mathcal{T}_n$)

<u>example 4</u>: Plaisted (Sept 78) examples with the rules:

$(x ** y) ** z \to x ** (y * z)$

$x + (y + z) \to (x + y) + z$

$x * (y * z) \to (x * y) * z$

Again lexicographic ordering (but right to left)

② <u>QUASI-ORDERS on FUNCTIONS SYMBOLS</u>:

This is a very minor point. Instead of having an order $>$ on function symbols, one has a quasi-order. Rules are the same, except rule 3 which permits $f \doteq g$ instead of $f = g$. This allows two further examples:

<u>example 1</u>: Mutual recursion.

$f a \to a$	$g a \to a$
$f b \to b$	$g b \to b$
$f(x \cdot y) \to g x$	$g(x \cdot y) \to f y$

Then $f \doteq g$ is OK (≃ bullshit, but maybe interesting in practise).

example 2: your FOCS example 3.

$$\begin{cases} \neg\neg p \to p \\ \neg(p \vee q) \to \neg\neg\neg p \wedge \neg\neg\neg q \\ \neg(p \wedge q) \to \neg\neg\neg p \vee \neg\neg\neg q \end{cases}$$

Then $t \rhd\!\rhd u$ iff $[\![t]\!] > [\![u]\!]$ or $([\![t]\!] = [\![u]\!]$ and $\{t_1, \dots t_n\} \gg_{\text{multiset}} \{u_1, \cdots u_m\}$. Let also make all the function symbols equivalent by $\doteq$. (By $[\![t]\!]$ one means the number of symbols different from $\neg$ in t).

Then one shows $t > u \Rightarrow [\![t]\!] \geq [\![u]\!]$. Thus $t > u$ implies $f(\cdots t \cdots) \rhd\!\rhd f(\cdots u \cdots)$. Therefore condition (d) ~~[illegible]~~ (the weak one) is true. (Remark that rule 1 is never apply)

Therefore: $\neg\neg p > p$ by rule 2 (twice).

$\neg(p \vee q) > \neg\neg\neg p \wedge \neg\neg\neg q$, by rule 3 since $\neg \doteq \wedge$ and $[\![\neg(p\vee q)]\!] = [\![\neg\neg\neg p \wedge \neg\neg\neg q]\!]$, $p \vee q > \neg\neg\neg p$ and $p \vee q > \neg\neg\neg q$ and $\neg(p \vee q) > \neg\neg\neg p$, $\neg(p\vee q) > \neg\neg\neg q$ (ouff!)
For instance, $p \vee q > \neg\neg\neg p$ since $\vee \doteq \neg$ and $[\![p \vee q]\!] > [\![\neg\neg\neg p]\!]$, $p \vee q > \neg\neg p$. Etc.....

③ CONCLUSION:

It seems also possible to work with quasi-simplifications by having two functionals (a reflexive one and an ireflexive one). But we have no convincing examples there. The main trouble is that one needs to keep schematic the strict ordering associated to the quasi-simplification. The only gain could be with a new weakening of condition (d), which does not really need the strict ordering to be compatible with the terms structure, as you remarked in your FOCS paper.

One short remark: (esoteric but which could impress all λ-calculus people). Can you prove termination of typed λ-calculus with simplifications? The Kruskal theorem must be true even with binders, and the usual Tait computability (see the Stenlund book) is short, but quite hard to understand. So maybe the argument can go through simplifications.

Some more sensible (?) remark: Is it possible to leave simplifications and to get "semi-simplication"? That is: the Kruskal theorem says that, in order, to show that the strict order $>$ is well-founded, it is sufficient to show that $> \cap \hookrightarrow = \phi$ (where $\hookrightarrow$ means the embedding relation). But, as you remark in your proof of the Kruskal theorem, $>$ is well-founded iff $> \cap \hookrightarrow$ is noetherian. Thus, one would like to have orderings $>$, which permits only finite chains such that $t_n > t_{n+1}$ and $t_n \hookrightarrow t_{n+1}$ at the same time. For instance, the factorial function terminates (via simplifications) when it is written as following:

$$\begin{cases} f(sx) \to sx * f(x) \\ f\,0 \to s\,0 \end{cases}$$

But, it is hard to believe that it is harder to prove the termination of factorial written as:

$$\begin{cases} f(sx) \to sx * f(p(sx)) \\ f\,0 \to s\,0 \\ p\,s\,x \to x. \end{cases}$$

Then $f(sx)$ is temporarily embedded in $f(p\,sx)$. But this not too serious since $f(psx) \to fx$. For treating such an example, we would like to use the semantical fact that $[\![sx]\!] > [\![p\,s\,x]\!]$. So the problem could be stated, as how to mix semantics and the syntactic recursive path ordering? If this is possible, it is not hard to believe that a more realistic version of factorial, namely

$$f\,x \to \underline{\text{if}}\ x=0\ \underline{\text{then}}\ s0\ \underline{\text{else}}\ x * f(px)$$

could be proved to terminate. (with termination in the usual sence, i.e. evaluating the test of the conditional first).

5. Post-scriptum: Mixing semantics and recursive path orderings:

Assume that one has:

(A) some well-founded ordering $\geqslant$ on terms and $\doteq$ is some equivalence relation compatible with $\geqslant$, i.e.

$$\underset{\substack{\uparrow \\ \text{subset inclusion}}}{\geqslant \doteq \subset} \geqslant \qquad \text{and} \qquad \underset{\substack{\uparrow \\ \text{subset inclusion.}}}{\doteq \geqslant \subset} \geqslant$$

(B) some functional $\triangleright$ on orders as previously with conditions:

(a) preserving transitivity,
(b) " irreflexivity,) strict-ordering.
(c) continuity.
(d) monotonicity, i.e. $t > u$ implies $f(\cdots t \cdots) \gg f(\cdots u \cdots)$

(C) $\doteq$ contain the internal reduction $\underset{int}{\to}$ relation, i.e:

$$t \to u \quad \text{implies} \quad f(\cdots t \cdots) \geqq f(\cdots u \cdots).$$

Then consider the following definition of some "semantical" recursive path ordering, s.r.p.o. in short.

Definition of s.r.p.o: $t = f\vec{t} > g\vec{u} = u$ iff
1) $t \gtrdot u$ and $t > u_i$ for all i,
2) $t_i \geq u$ for some i,
3) $t \doteq u$, $t \rhd\!\!\rhd u$ and $t > u_i$ for all i.

Then one can show as previously, that $>$ is a strict order, satisfying:
(e) - $f(\cdots t \cdots) > t$,
(f) - $t \to u$ and $t > u$ implies $f(\cdots t \cdots) > f(\cdots u \cdots)$.

Thus $>$ is not monotonic, but its intersection $> \cap \to$ with $\to$ is. Therefore, rule (f) shows that, for testing that $t \to u$ implies $t > u$, it is sufficient to test $\sigma\alpha_i > \sigma\beta_i$ for all instances $\sigma\alpha_i$ and $\sigma\beta_i$ of left hand sides and corresponding right hand sides. So $>$ may not be a simplification ordering.

Now, one can show that $>$ is a well-founded ordering, provided some further condition is true on $\rhd$:

(g) For any infinite sequence $(t^i)_i$ such that $t^i \rhd\!\!\rhd t^{i+1}$ for all $i \geq 0$, there is an infinite subsequence $(u^i)_i$ such that, for all i there is an argument $u^i_{k_i}$ of $u^i = f_i(u^i_1, \cdots u^i_{p_i})$ satisfying

$$u^i_{k_i} > u^{i+1}_{k_{i+1}}$$

This is not very nice (quite ugly in fact) and there is maybe one way of weakening it. Roughly speaking, it says that $\rhd\!\!\rhd$ preserves the noetherian aspect of $>$ (on the arguments), but this is true when:

$$t = f\vec{t} \rhd\!\!\rhd f\vec{u} = u \quad \text{if} \quad \vec{t} >_{\text{lexicographic}} \vec{u},$$

$$t = g\vec{t} \rhd\!\!\rhd g\vec{u} = u \quad \text{if} \quad \{t_1, \cdots t_n\} \gg_{\text{multiset}} \{u_1, \cdots u_p\}.$$

Proof that $>$ is well-founded: (As the Kruskal theorem) Suppose $>$ is not well-founded. We take a minimal counterexample: $t_1 > t_2 > t_3 > \cdots > t_n > \cdots$ (i.e. t_n is at each step minimal in size of all counterexamples starting with $t_1, t_2, \cdots t_{n-1}$. We remark first that rule 2 is never applied, otherwise this is not a minimal counterexample. Thus, only rules 1 and 3 are applied. Now $\gtrdot$ is well-founded, thus after some while only rule 3 is applied. But condition (g) tells us this is not possible. Because, since $t_m \rhd\!\!\rhd t_{m+1} \rhd\!\!\rhd t_{m+2} \rhd\!\!\rhd \cdots$ after some m, there is an infinite subsequence of arguments $(u^i_{k_i})_i$ by (g) such that $u^i_{k_i} > u^{i+1}_{k_{i+1}} > \cdots$.
Say that $u^i_{k_i}$ is an argument of t_n. Since $t_n > u^i_{k_i}$, the counterexample is not minimal, because

$$t_1 > t_2 > \cdots > t_{n-1} > u^i_{k_i} > u^{i+1}_{k_{i+1}} > \cdots$$

is a "shorter" counterexample. Contradiction. $\square$

Now, let look as some examples. First, we want to follow Plaisted's idea, And say there is some partial ordering ⓢ on function symbols (strict-order for simplifying) and we also suppose that the ~~alphabet~~ function symbols alphabet is finite. (also for simplifying). Now, we consider some interpretation $[\![t]\!]$ of any term t, which satisfies the rewriting system. Thus:

(h) $t \to u$ implies $[\![t]\!] = [\![u]\!]$

Finally, let $[\![\vec{t}]\!]$ be an abbreviation for the vector $([\![t_1]\!], [\![t_2]\!], \cdots [\![t_n]\!])$ of the semantics values of $(t_1, t_2, \cdots t_n)$. And suppose, we assume that we have some well-founded order $\geq_s$ on vectors $[\![\vec{t}]\!]$. The well-founded ordering needed in the definition of s.r.p.o is done by stating:

(k) $f\vec{t} = t > u = g\vec{u}$ iff
- either f ⓢ g
- or $f = g$ and $[\![\vec{t}]\!] >_s [\![\vec{u}]\!]$

(l) $f\vec{t} = t \doteq u = g\vec{u}$ iff $f = g$ and $[\![\vec{t}]\!] =_s [\![\vec{u}]\!]$.

Since the interpretation makes valid the rewrite rules, it is straightforward to check that $t \to u$ implies $f(\cdots t \cdots) \doteq f(\cdots u \cdots)$.

It is maybe better to rewrite the definition of the now considered s.r.p.o.

Definition 2: $t = f\vec{t} > g\vec{u} = u$ iff
① f ⓢ g, and $t > u_i$ for all i,
② $f = g$, and $[\![\vec{t}]\!] \geq_s [\![\vec{u}]\!]$, and $t > u_i$ for all i,
③ $t_i \geq u$ for some i,
④ $f = g$, and $[\![\vec{t}]\!] =_s [\![\vec{u}]\!]$, and $t \rhd\!\!> u$, and $t > u_i$ for all i.

Notice that the choice of $\geq_s$ can be parameterised by the function symbol f. Similarly for $\rhd\!\!>$ (in fact stronger: it may depend also on the equivalence class modulo the semantics equivalence). Consider now examples:

example 1: Factorial:

$$\begin{cases} f(sx) \to sx * f(psx) \\ f(0) \to s0 \\ psx \to x \end{cases}$$

Then, suppose f ⓢ $*$, f ⓢ p, f ⓢ s,
In order to show that $fsx > sx * f(psx)$, by rule 1, one has two subgoals:

$fsx > sx$ (true by rule 3)
$fsx > f(psx)$

Now, we use the well-founded usual ordering on $\mathbb{N}$, and we have $[\![sx]\!] \geq_s [\![psx]\!]$. Therefore, by rule 2, we show $fsx > f(psx)$, if $fsx > psx$. But

f is more complicated than p ($f \circledS p$). Therefore, by rule 1, one has: $f\,s\,x > s\,x$ as subgoal, which is true by 3.

Now $f(0) > s\,0$, since $f \circledS s$ and $f\,0 > 0$ by rule 3.

Finally $p\,s\,x > x$, since $s\,x > x$ by rule 3 (twice).

Some remarks: we use only the interpretation of p and s. Therefore, some possible interpretation could be:

$[\![f\,t]\!] = f_I [\![t]\!] = 1$ for all $[\![t]\!]$,
$[\![t * u]\!] = 1$ for all t, u,
$[\![s\,t]\!] = [\![t]\!] + 1$, $[\![0]\!] = 0$,
$[\![p\,t]\!] = [\![t]\!] - 1$, if $[\![t]\!] \geq 1$
$[\![p\,0]\!] = 4$

Remark too that, in case the interpretation is the trivial one (everything equal). Then definition 2 does no more than the usual (syntactic) recursive path ordering. And the trivial interpretation surely satisfies the rewriting rules. Thus, there can be some kind of tuning in defining the interpretation, taking care only of what is really needed in the termination proof. For instance, having not interpreted at all $*$ and $+$, we can add all the axioms of the Plaisted's paper (Sept 78) like:

$x * (y * z) \to (x * y) * z$, ...

and keep the totally syntactic proof.

also by acting on $\geq_s$ and $=_s$

6 - Post scriptum 2: Usual recursive programs with the conditional:

Now we want to consider the true factorial:

$f(x) \to$ if $x = 0$ then $s\,0$ else $x * f(p\,x)$.

We can define the evaluator by rules of inference:

Axioms: $\left\{ \begin{array}{l} \sigma\alpha_i \to \sigma\beta_i \\ \\ \text{if } tt \text{ then } t \text{ else } u \to t \\ \text{if } ff \text{ then } t \text{ else } u \to u \end{array} \right\}$ rewrite rules instances.

(tt, ff are supposed to be normal forms)

Inference rules:

$$(\mathrm{I})\ \frac{t \to u}{f(\cdots t \cdots) \to f(\cdots u \cdots)} \qquad (\text{for all } f \neq \text{if})$$

$$(\mathrm{II})\ \frac{p \to q}{\text{if } p \text{ then } t \text{ else } u \to \text{if } q \text{ then } t \text{ else } u}$$

Thus, the evaluation of the predicate part of the conditional is forced before evaluation of the alternatives. Now, you can guess how to change rules 1, 2, 3, 4 and get the following definition:

For simplifying, we suppose the *if* function symbol to be the least complicated. Furthermore, it will be easier to write $[\![p]\!] = \text{true}$ implies $t > u$, as $p \models t > u$. Similarly for $\neg p \models t > u$. Thus

Definition 3: We keep 1.2.3.4 when f and g are not *if*. We add $t = f\vec{t} > g\vec{u} = u$ if

(1') $f \neq$ *if*, $u =$ *if* p *then* u_1 *else* u_2, $t > p$, $p \models t > u_1$ and $\neg p \models t > u_2$,

(3') $t =$ *if* p *then* t_1 *else* t_2, $p \geqslant u$ or $p \models t_1 \geqslant u$ or $\neg p \models t_2 \geqslant u$,

(4') $t =$ *if* p *then* t_1 *else* t_2, $u =$ *if* q *then* u_1 *else* u_2, $p > q$, $q \models t > u_1$, $\neg q \models t > u_2$.

Notice that, only in case (3'), there are disjunctions. All the rest are conjonctions. Now it is routine (?) to test that $>$ is still transitive, irreflexive, well-founded and satisfying:

(m) *if* p *then* t_1 *else* $t_2 > p$

(n) $p \models$ *if* p *then* t_1 *else* $t_2 > t_1$

(p) $\neg p \models$ *if* p *then* t_1 *else* $t_2 > t_2$

Furthermore, the monotonicity rule is still true. In fact, a weakened version of it is true, but sufficient for proving what is really needed (that is that showing $\tau\alpha_i > \tau\beta_i$ for any instance of an axiom of the rewriting system satisfies $t \to u$ for all t and u). In other words, we want to validate the inference rules. Thus:

- rule I is still valid, using rule 2 of definition 3 (i.e.

- rule II now: Suppose $p \to q$ and $p > q$. Then we want to apply rule 4' in order to show that $t =$ *if* p *then* t_1 *else* $t_2 > u =$ *if* q *then* t_1 *else* t_2. Since $p \to q$, we have $[\![p]\!] = [\![q]\!]$. Thus $[\![p]\!] = \text{true}$ iff $[\![q]\!] = \text{true}$. By rule 3', we know that $p \models t > t_1$ and $\neg p \models t > t_2$. Therefore $q \models t > t_1$ and $\neg q \models t > t_2$. And rule 4' permits now $t > u$.

<u>example 1: factorial</u>

$$\begin{cases} f(x) \to \textit{if } x = 0 \textit{ then } s0 \textit{ else } x * f(px) \\ psx \to x \\ \vdots \end{cases}$$

Then $psx > x$ by rule 3 (twice). Now, we have to show $fx >$ *if* $x = 0$ *then* $s0$ *else* $x * f(px)$. We try rule 1'. Therefore, we have three subgoals.

1) $fx > x = 0$. OK if $f > =$ and $f > 0$

2) $fx > s0$. Idem.

3) $x \neq 0 \models fx > x * f(px)$.

We come back to the method of page 18. By interpreting correctly p and s and taking the well-founded usual order on integer. Notice that we need to interpret also the test for zero.

<u>example 2 : The 91-function :</u>

$$f(x) \to \text{if } x > 100 \text{ then } x-10 \text{ else } f(f(x+11))$$

Here there is a big trouble for making a difference between the recursive path ordering $>$ and the symbol $>$ of $x > 100$. We do as for factorial, but we need more information on a possible interpretation of f satisfying the rewriting system. Take for interpreting f, the function $g(x) = \text{if } x > 100 \text{ then } x-10 \text{ else } 91$ (which is <u>one</u> solution - Remark that we do not know it is <u>the</u> solution). Then the proof is as for factorial, generating at some points :

$$x \leq 100 \models x \geq_s g(x+11) \;\&\; x \geq_s x+11$$

Which is ok for finding some well-founded order $\geq_s$, i.e. $x \geq_s y$ when $x < y \leq 111$ (as in your paper with Zohar Manna on multisets for proving termination).

I agree there is nothing new under the sun (as we say here). But what is maybe interesting is the ability of mixing syntax and semantics ??? And the training between the real recursive path ordering and some cut point method ??? Although I am lacking of examples.

Finally, if there is some mistake (which makes empty these post-scriptum), do not charge Sam's responsibili[ty] who left meantime for the POPL conference. But that's (I believe) what we try to do before he left. Also, do n[ot] reproduce this too much until it could get some better form. Copies are also sent to Henk Barendreg[t], Gérard Berry, Leo Guibas, Jan-Willem Klop, Gérard Huet, Gordon Plotkin who could also be interested by that.

Sam Kamin - Jean-Jacques Lévy.
Feb 1st, 1980.

Addendum to the 1^st Feb note:

1§: On functionals of orderings:

In order to make possible $\rhd$ for the multiset ordering (page 3), one has to merge conditions a & b in:

(a') $\rhd$ preserves strict orderings

(otherwise the multiset functional does not preserve irreflexivity by itself) The proof still works, but with a parallel induction for proving transitivity and irreflexivity at same time.

2§: On the ugly condition (g) of page 15:

. Condition (g) can be replaced by the following (g'):

$$(g')\quad \left.\begin{cases} t \doteq u,\ t = f\vec{t},\ u = g\vec{u} \\ t_{k_i} \geq u_i \quad \text{for all } i \end{cases}\right\} \text{ implies } \quad f\vec{t} \rhd\!> g\vec{u} \quad \text{or } f\vec{t} = g\vec{u}$$

where $1 \leq k_1 < k_2 \cdots < k_n$. For doing this, we use both one theorem in Szpilrajn (see Birkhoff p.195) and the Kruskal theorem. The first one states that any well-founded partial order can be strenghtened to a well-(total)-ordering. Therefore, the partial well-founded ordering $\gg$ of page 15 (in fact its natural extension on the quotient set of terms by $\doteq$) can be extended to a well-ordering $\gg^t$ with still $\doteq$ being an equivalence compatible with it. Now as $\gg^t$ is a well-ordering, by the Kruskal theorem, its tree image is a well-partial ordering, i.e. in any infinite sequence $(t_i)_i$ of terms there is a pair t_i and t_j with $i < j$ such that $t_i \hookrightarrow t_j$, where $t \hookrightarrow u$ is true iff:

(1) $u = g\vec{u}$ and $t \hookrightarrow u_i$ for some i,
(2) or $t \leq^t u$ and $t_i \hookrightarrow u_{k_i}$ for all i with $1 \leq k_1 < k_2 < \cdots < k_n$.

Now, consider the s.r.p.o induced by $\gg^t$ (following definition of page 15), i.e. $t = f\vec{t} >^t g\vec{u} = u$ iff:

1) $t \gg^t u$ and $t >^t u_i$ for all i,
2) $t_i \geq^t u$ for some i,
3) $t \doteq u$, $t \rhd\!>^t u$ and $t >^t u_i$ for all i.

Then again the s.r.p.o $>^t$ is a strict ordering (same proof) and contains the s.r.p.o $>$ defined with the well-founded partial ordering $\gg$. If one shows that $>^t$ is well-founded, we have then showed that $>$ is also well-founded and then finished. But $>^t$ is surely well-founded, since in any infinite sequence $(t_i)_i$ of terms, there is $t_i \hookrightarrow t_j$. But then $t \leq^t u$, because $\hookrightarrow$ is contained in $\leq^t$ with (g). Thus $t_i >^t t_j$ and $t_i \leq^t t_j$ are not possible at same-time.

Feb 13^th