Une preuve effective de la bonne fondation de l'ordre récursif multi-ensemble sur les chemins

S. Coupet-Grimal[1] & W. Delobel[2]

1: Laboratoire d'Informatique Fondamentale de Marseille, UMR 6166
Université de Provence, CMI, 39 rue Joliot-Curie, F-13453, Marseille, France
Solange.Coupet@cmi.univ-mrs.fr
2: Laboratoire d'Informatique Fondamentale de Marseille, UMR 6166
Université de Provence, CMI, 39 rue Joliot-Curie, F-13453, Marseille, France
William.Delobel@cmi.univ-mrs.fr

Résumé

La contribution de cet article est une preuve effective de la bonne fondation de l'ordre récursif multi-ensemble sur les chemins (Multiset Path Ordering (MPO)), comme terme du Calcul des Constructions Inductives. Cette preuve est directe, courte et simple. Elle ne fait appel qu'à des résultats préliminaires élémentaires et s'applique à des termes contenant des variables, construits sur une signature non nécessairement finie de symboles fonctionnels d'arité variable. Toutes les preuves présentées ici ont été vérifiées par l'assistant de preuves Coq.

1. Introduction

La terminaison est une propriété importante des systèmes de réécriture de termes (Term Rewriting Systems (TRS)). Cette propriété est en général indécidable comme l'ont montré Huet et Lankford dans [HL78]. Une méthode classique pour prouver la terminaison d'un TRS donné est d'exhiber un ordre bien fondé $>$ sur les termes, tel que $s > t$ pour chaque pas de réécriture $s \rightarrow t$. Les Ordres Récursifs sur les Chemins (Recursive Path Orderings (RPO)), introduits par Dershowitz [Der82], sont des ordres de simplification, compatibles avec les contextes et les substitutions. Il est donc suffisant, pour de tels ordres, de vérifier que le membre gauche de toute règle de réduction est supérieur au membre droit. Ces ordres ont d'autres caractéristiques intéressantes : par exemple, le problème de savoir si la terminaison d'un TRS fini sur une signature finie peut être prouvée en utilisant un RPO est décidable (on pourra se reporter à [BN98] pour une description de ces propriétés).
Ces ordres comparent les termes en commençant par comparer leurs racines, puis la liste des sous-termes immédiats. Ces listes peuvent être comparées soit en les considérant comme des multi-ensembles - on parle alors d'ordre multi-ensemble sur les chemins (MPO) - soit lexicographiquement dans le cas de l'ordre lexicographique sur les chemins (LPO), ou encore en combinant les deux approches précédentes (RPO avec statut).
La contribution de cet article est une preuve effective de la bonne fondation de MPO, comme un terme du Calcul des Constructions Inductives (CIC). Cette preuve a été vérifiée par Coq [Tea04, Coq]) et est intégrée à *CoLoR*, la bibliothèque Coq dédiée à la réécriture et à la terminaison [CoL]. Cette preuve est directe dans le sens où elle n'est pas obtenue par application d'un des théorèmes plus généraux récemment établis (voir section 5). Elle est courte (30 lignes de Coq), simple et s'appuie exclusivement sur des inductions imbriquées. Enfin, elle requiert comme seuls résultats préliminaires la transitivité

de MPO et le fait que les multi-ensembles finis dont les éléments sont accessibles pour la relation de base sont eux-mêmes accessibles pour l'ordre multi-ensemble. Notre spécification de MPO est générale dans le sens où les termes que nous considérons ne sont pas nécessairement clos ni la signature finie. Cet article ne suppose aucun pré-requis et il est structuré comme suit. La section 2 est consacrée à la spécification dans le CIC des notions clés : accessibilité, bonne fondation et induction bien fondée. La section 3 traite des multi-ensembles finis et de l'ordre multi-ensemble. Dans la section 4, on définit MPO et on prouve sa bonne fondation. En section 5, nous présentons les travaux connexes à cette étude avant de conclure.

2. Bonne fondation

Soit $(A, <)$ un ensemble muni d'une relation binaire. La bonne fondation de la relation $<$ fait intervenir la notion d'*accessibilité*. Intuitivement, un élément a de A est accessible pour la relation $<$, ce que l'on note $(acc_< \; x)$, si et seulement si toute chaîne descendante commençant par x est finie. Dans le calcul des constructions, ceci s'exprime à l'aide d'une définition inductive généralisée (1), à laquelle est associé le principe d'induction (2) :

$$\frac{\forall y : A,\; y < x \;\rightarrow\; (acc_< \; y)}{(acc_< \; x)} \tag{1}$$

$$\frac{(acc_< \; x) \quad (\forall y : A,\; y < x \;\rightarrow\; (P \; y)) \;\rightarrow\; (P \; x)}{(P \; x)} \tag{2}$$

Les éléments minimaux sont clairement accessibles, ce qui fournit le cas de base de la définition récursive (1). En ce qui concerne le principe d'induction, il permet de prouver qu'un prédicat P est vérifié par un élément x, sous l'hypothèse $(acc_< \; x)$, en utilisant comme hypothèse d'induction le fait que P est vérifié par tout élément y inférieur à x.

La relation $<$ est bien fondée si et seulement si toute chaîne descendante est finie, c'est-à-dire si et seulement si tout élément de A est accessible. On définit ainsi le prédicat WF :

$$(WF \;\; <) := \forall x : A,\; (acc_< \; x) \tag{3}$$

On en déduit la validité du principe d'*induction bien fondée* suivant, dans le cas où la relation $<$ est bien fondée :

$$\frac{\forall x : A,\; (\forall y : A,\; y < x \;\rightarrow\; (P \; y)) \;\rightarrow\; (P \; x)}{\forall x : A,\; (P \; x)} \tag{4}$$

3. Ordre multi-ensemble

3.1. Multi-ensembles finis

Considérons un setoïde $(A, \sim_A)$, où A est un ensemble et $\sim_A$ une relation d'équivalence sur A. Nous supposerons ici que cette relation est décidable :

$$\forall a, b : A,\; (a \sim_A b) \;\vee\; \neg(a \sim_A b)$$

Un multi-ensemble M de A est une application de A dans l'ensemble des entiers naturels, compatible avec la relation $\sim_A$. Elle induit donc une application quotient $M/\sim_A$ de $A/\sim_A$ dans IN. Pour tout élément a de A, l'entier $M(a)$ est appelé multiplicité de a dans M. On dit qu'un élément a est dans

M, si sa multiplicité est non nulle. Quand seul un nombre fini de classes modulo $\sim_A$ ont une image non nulle par $M/\sim_A$, on dit que M est fini. On convient de représenter tout multi-ensemble fini en faisant figurer entre doubles accolades la classe de chacun de ses éléments, autant de fois que sa multiplicité. Par exemple, $M = \{\!\{1,1,5,5,5,6,6,7\}\!\}$ est le multi-ensemble du setoïde $(IN, =)$ des entiers naturels défini par : $M(1) = 2$, $M(5) = 3$, $M(6) = 2$, $M(7) = 1$, et M prend la valeur nulle partout ailleurs.

Ces multi-ensembles finis ont été implantés dans le CIC par Koprowski [Kop04]. Les deux sous-sections 3.1 et 3.2 font référence à ces bibliothèques Coq, que nous avons légèrement modifiées et complétées dans le cadre de notre étude.

L'auteur donne une définition axiomatique de la notion de multi-ensemble fini, dont il établit ensuite la cohérence en montrant que les listes finies en constituent un modèle. Cette axiomatisation comporte en particulier les opérations union et différence, le multi-ensemble vide $\emptyset$ et une relation d'équivalence $\sim_{mul}$. Le fait que ces multi-ensembles soient finis s'exprime grâce au principe d'induction suivant, où P est un prédicat sur les multi-ensembles :

$$\frac{(P\ \emptyset) \qquad (\forall M\ : (Multiset\ A))(\forall a : A)\ (P\ M) \rightarrow (P\ M \cup \{\!\{a\}\!\})}{\forall M : (Multiset\ A)\ (P\ M)} \tag{5}$$

Il est à noter que le type $Multiset$ est paramétré par l'ensemble de base A. Dans toute la suite de cet article, les multi-ensembles considérés seront finis : nous omettrons donc de le préciser.

À partir de ces définitions axiomatiques, de nouveaux opérateurs sont ensuite introduits et diverses propriétés établies. En particulier, à partir d'une fonction *insert* qui ajoute un élément à un multi-ensemble, est définie une fonction de conversion *list2multiset*, qui transforme récursivement chaque liste en un multi-ensemble en insérant sa tête dans le multi-ensemble résultant de la transformation de sa queue.

Nous avons ajouté la fonction *multiset2list* qui construit une liste à partir d'un multi-ensemble M par induction sur M (principe 5), et nous avons prouvé que $(list2multiset\,(multiset2list\ M)) \sim_{mul} M$.

3.2. Ordre sur les multi-ensembles finis

Considérons maintenant une nouvelle relation binaire $>_A$ sur le setoïde $(A, \sim_A)$. Cette relation induit une relation $>_{mul,>_A}$ sur les multi-ensembles de A, définie par induction de la manière suivante :

$$\frac{\begin{array}{c} M\ \sim_{mul}\ Z \cup X \quad N \sim_{mul}\ Z \cup Y \quad \neg(X \sim_{mul} \emptyset) \\ (\forall y : A,\ y \in Y\ \rightarrow\ \exists x : A,\ x \in X\ \wedge\ x >_A y) \end{array}}{M\ >_{mul,>_A}\ N}$$

Notons $<_A$ et $<_{mul,>_A}$ les relations symétriques de $>_A$ et $>_{mul,>_A}$. On démontre les résultats suivants :

Lemme 1 *Si $<_A$ est transitive sur A, alors $<_{mul,>_A}$ est transitive sur (Multiset A).*

Nous établissons le résultat suivant, plus général, que nous avons ajouté à la bibliothèque existante :

Lemme 2 *Soit M un multi-ensemble sur A. Supposons que :*

$$\forall a\ :\ A,\ a \in M \rightarrow (\forall a_1, a_2\ :\ A,\ a_2 >_A a_1 \rightarrow a_1 >_A a \rightarrow a_2 >_A a)$$

Alors, pour tous multi-ensembles M_1, M_2 :

$$M_2 >_{mul,>_A} M_1 \rightarrow M_1 >_{mul,>_A} M \rightarrow M_2 >_{mul,>_A} M$$

Ce lemme est utile pour établir des résultats de transitivité par induction structurelle sur les éléments de l'ensemble, comme nous le ferons par la suite.

En supposant que la relation $>_A$ est transitive, on peut montrer que $>_{mul,>_A}$ est la clôture transitive de la relation $>_{red,>_A}$ définie par :

$$\frac{M \sim_{mul} Z \cup \{\!\{a\}\!\} \quad N \sim_{mul} Z \cup Y \quad (\forall y : A,\ y \in Y \rightarrow a >_A y)}{M >_{red,>_A} N}$$

Lemme 3 *Si $<_A$ est transitive, alors pour tous multi-ensembles M et N sur A : $M >_{mul,>_A} N \leftrightarrow M(>_{red,>_A})^+N$*

Preuve. La preuve fournie dans la bibliothèque Coq existante utilise une hypothèse de décidabilité $\forall a, b\ :\ A,\ (a >_A b) \vee \neg(a >_A b)$ non triviale à prouver dans la cas qui nous intéresse (MPO sur les termes du premier ordre). Mais en fait, cette hypothèse peut être affaiblie et remplacée par la décidabilité de $\sim_A$, qui était déjà exigée par le reste du développement. Nous avons modifié le lemme en ce sens. □

Lemme 4 *Pour tout multi-ensemble M sur A, si tout élément de M est accessible pour la relation $<_A$, alors M est accessible pour la relation $<_{mul,>_A}$.*

Une preuve inductive "papier-crayon" de ce lemme a été proposée par Buchholtz, présentée par Nipkow dans [Nip98] et implantée en Coq par Koprowski. Nous avons ajouté aux bibliothèques Coq la propriété réciproque :

Lemme 5 *Pour tout multi-ensemble M sur A, si M est accessible pour la relation $(<_{red,>_A})^+$, alors tout élément de M est accessible pour la relation $<_A$.*

Preuve. Cette preuve se fait par induction sur l'hypothèse d'accessibilité, selon le principe (2). On doit ainsi prouver que tout élément d'un multi-ensemble M est accessible sous l'hypothèse d'induction :

$$\forall N : (MultisetA), N(<_{red,<_A})^+M \ \rightarrow\ \forall n : A,\ n \in N \ \rightarrow\ (acc_{<_A}\, n) \qquad .$$

Soit m un élément de M. Prouver que m est accessible pour $<_A$, d'après la définition 1, revient à prouver que tout n tel que $n <_A m$ est accessible. Ceci s'établit en appliquant l'hypothèse d'induction en prenant pour N le multi-ensemble obtenu en remplaçant dans M une occurence de m par n. □

Mentionnons de plus qu'une conséquence immédiate du lemme 4 est que la relation $(<_{red,>_A})^+$ sur les multi-ensembles est bien fondée dès que la relation $<_A$ sur l'ensemble de base est bien fondée.

3.3. Ordre multi-ensemble sur les listes

Notre étude portant sur les termes du premier ordre, et ces termes étant codés par des symboles fonctionnels appliqués à la liste de leurs arguments (voir la section 4.1), nous sommes naturellement conduits à convertir l'ordre sur les multi-ensembles à une relation sur les listes. Ainsi, étant donné un setoïde $(A, \sim_A)$ muni de la relation d'ordre $>_A$, définissons une relation $\ll_{>_A}$ sur les listes d'éléments de A par :

$$\ll_{>_A} := \lambda\, l,\, l'\ : (list\, A).\, (list2multiset\, l)\ <_{mul,\,>_A}\ (list2multiset\, l')$$

Nous établissons en premier lieu un résultat légèrement plus général que la transitivité de $\ll_{>_A}$, sous une hypothèse plus faible que la transitivité de la relation $>_A$.

Lemme 6 *Soit l une liste d'éléments de A. Supposons que :*

$$\forall a : A,\, a \in l \rightarrow (\forall a_1,\, a_2 \,:\, A,\, a <_A a_1 \rightarrow a_1 <_A a_2 \rightarrow a <_A a_2)$$

Alors :

$$\forall l_1,\, l_2 \,:\, (list\, A),\, l \ll_{>_A} l_1 \rightarrow l_1 \ll_{>_A} l_2 \rightarrow l \ll_{>_A} l_2.$$

Preuve. Ce lemme est une reformulation du lemme 2. □

Les lemmes suivants sont les résultats clés pour prouver la bonne fondation de MPO.

Lemme 7 *Supposons que la relation $>_A$ soit transitive. Pour toute liste l d'éléments de A, si $(list2multiset\, l)$ est accessible par la relation $(<_{red,<_A})^+$, alors l est accessible par la relation $\ll_{>_A}$.*

Preuve. La relation $>_A$ étant transitive, $(<_{red,<_A})^+$ est équivalente à $<_{mul,>_A}$. Il suffit donc de montrer le résultat pour cette dernière relation. Il provient du fait que pour toute fonction f et toute relation $<$, si $(f\, x)$ est accessible par $<$, alors x est accessible par $(f^{-1}\ <)$ (ceci est prouvé dans les bibliothèques standard Coq). □

Le lemme suivant établit la propriété réciproque.

Lemme 8 *Supposons que la relation $>_A$ est transitive. Pour toute liste l d'éléments de A, si l est accessible par la relation $\ll_{>_A}$, alors $(list2multiset\, l)$ est accessible par la relation $(<_{red,<_A})^+$.*

Preuve. Clairement, $<_{mul,>_A}$ est l'image inverse de $\ll_{>_A}$. Par une approche analogue à celle du lemme précédent, nous obtiendrions que pour tout multi-ensemble M, si $(multiset2list\, M)$ est accessible par $\ll_{>_A}$, alors M est accessible par $<_{mul,>_A}$, donc par $(<_{red,<_A})^+$. Mais ceci ne permet pas de conclure, car la liste l n'est pas, en général, égale à $(multiset2list\,(list2multiset\, l))$). La fonction $list2multiset$ n'étant pas injective, son inverse est une relation non fonctionnelle. Par suite, on utilise le lemme suivant :

Lemme 9 *Soient $(\mathcal{L}, \ll)$ et $(\mathcal{M}, <)$ des ensembles munis de relations binaires et soit une relation $r : \mathcal{L} \rightarrow \mathcal{M} \rightarrow Prop$ telle que :*

$$(\forall l : \mathcal{L})\ (\forall M, M' : \mathcal{M})\ (r\ l\ M) \rightarrow M' < M \rightarrow (\exists l' : \mathcal{L})\ (r\, l'\, M') \wedge l' \ll l \quad .$$

Alors :

$$(\forall l : \mathcal{L})\ (\forall M : \mathcal{M})\ (r\ l\ M) \rightarrow\ (acc_{\ll}\ l)\ \rightarrow\ (acc_{<}\ M)$$

Preuve. La preuve se fait par induction sur $(acc_{\ll}\ l)$. □

La preuve du lemme 8 se fait alors en appliquant le lemme 9 avec $\mathcal{L} = (list\ A)$, $\mathcal{M} = (Multiset\ A)$, $r = \lambda l, M. M\ \sim_{mul}\ (list2multiset\ l)$. L'hypothèse du lemme 9 est satisfaite en choisissant $l' = (multiset2list\ M')$. □

Lemme 10 *Supposons $<_A$ transitive. Pour toute liste l d'éléments de A, si tout élément de l est accessible pour la relation $<_A$, alors l est accessible pour la relation $\ll_{>_A}$.*

Preuve. De la transitivité de $<_A$, on déduit l'équivalence de la relation $>_{mul,>_A}$ et de la clôture transitive de $>_{red,>_A}$ (lemme 3). Par suite, d'après le lemme 4, $(list2multiset\ l)$ est accessible par $(<_{red,<_A})^+$. Le résultat se déduit du lemme 7. □

Lemme 11 *Supposons $<_A$ transitive. Pour toute liste l d'éléments de A, si l est accessible pour la relation $\ll_{>_A}$, alors tout élément de l est accessible pour la relation $<_A$.*

Preuve. Elle est analogue à la précédente, mais repose sur les lemmes 5 et 8. □

Enfin, le lemme suivant, dont la preuve est immédiate, sera souvent utilisé par la suite :

Lemme 12 *Soient l_1 et l_2 deux listes d'éléments de A, telles que $l_1 \ll_{<_A} l_2$. Pour tout élément a_1 de l_1, il existe un élément a_2 de l_2 tel que $a_1 \leq_A a_2$.*

4. Ordre multi-ensemble sur les chemins (MPO)

MPO est une relation binaire sur les termes du premier ordre. Cette relation a été introduite par Dershowitz [Der82] pour prouver la terminaison de systèmes de réécriture.

4.1. Termes du premier ordre

Soient un ensemble F de symboles fonctionnels, muni d'une relation transitive bien fondée $<_F$, et X un ensemble de noms de variables. Le type *term* des termes du premier ordre sur la signature F peut être défini inductivement de la manière suivante :

$term\ :\ Set :=$
$\quad Var\ :\ X \rightarrow term\ |$
$\quad App\ :\ F \rightarrow (list\ term)\ \rightarrow\ term.$

où le type *list* est défini de la façon usuelle :

$(list\ term)\ :\ Set :=$
$\quad nil\ :\ (list\ term)\ |$
$\quad cons : term\ \rightarrow\ (list\ term)\ \rightarrow\ (list\ term).$

Dans la suite, nous utiliserons les notations simplifiées ci-dessous :
- $(s_1, \ldots, s_n)$ pour $(cons\ s_1(\ldots(cons\ s_n\ nil)\ldots))$
- $\{\!\{s_1, \ldots, s_n\}\!\}$ pour $(list2multiset\ (cons\ s_1(\ldots(cons\ s_n\ nil)\ldots)))$.
- $f(s_1, \ldots, s_n)$ pour $(App\ f\ (cons\ s_1(\ldots(cons\ s_n\ nil)\ldots))$
- x pour $(Var\ x)$
- $Vars(s)$ pour l'ensemble des variables qui apparaissent dans le terme s.
- $s \in ss$ pour exprimer que s est un élément de la liste ss.

Un principe d'induction associé au type *term* peut être décrit par la règle suivante, dans laquelle P est un prédicat sur *term* :

$$\frac{\begin{array}{c}\forall x\ :\ X,\ P\,(Var\ x)\\ \forall f : F, \forall ss : (list\ term),\ (\forall s_i : term,\ s_i \in ss\ \rightarrow\ (P\ s_i))\ \rightarrow\ (P\ f(ss))\end{array}}{\forall s\ :\ term,\ (P\ s)} \tag{6}$$

Ce principe découle immédiatement du lemme suivant :

Lemme 13 *Soit P un prédicat sur les termes. Sous les hypothèses*

(i) $\forall x\ :\ X,\ P\,(Var\ x)$

(ii) $\forall f : F, \forall ss : (list\ term),\ (\forall s_i : term,\ s_i \in ss\ \rightarrow\ (P\ s_i))\ \rightarrow\ (P\ f(ss))$

on peut prouver que $\forall n\ :\ nat,\ \forall s\ :\ term, |s|\ =\ n\ \rightarrow\ (P\ s)$ *où* $|s|$ *désigne la taille du terme s, c'est à dire le nombre de symboles fonctionnels apparaissant dans s.*

Preuve. En appliquant le principe (4) à l'ordre strict habituel sur les entiers naturels, on procède par induction sur la taille n du terme s. Si s est une variable, l'hypothèse (i) s'applique. Si s est de la forme $s = f(ss)$, le résultat découle de l'hypothèse (ii) et du fait que la taille de tout sous-terme immédiat de s est inférieure à n. □

4.2. Définition de MPO

L'Ordre Multi-ensemble sur les Chemins MPO (noté ici $<_{MPO}$) est une relation sur les termes définie inductivement par les 3 règles ci-dessous :

$$\frac{g <_F f \quad \forall i \in \{1,\ldots,m\},\, t_i <_{MPO}\ f(s_1,\ldots,s_n)}{g(t_1,\ldots,t_m) <_{MPO}\ f(s_1,\ldots,s_n)} \quad (MPO_1)$$

$$\frac{\{\!\{t_1,\ldots,t_m\}\!\} <_{mul,<_{MPO}}\ \{\!\{s_1,\ldots,s_n\}\!\}}{f(t_1,\ldots,t_m) <_{MPO}\ f(s_1,\ldots,s_n)} \quad (MPO_2)$$

$$\frac{\exists i \in \{1,\ldots,n\},\ t \leq_{MPO}\ s_i}{t <_{MPO}\ f(s_1,\ldots,s_n)} \quad (MPO_3)$$

Remarquons que la règle (MPO_2) est récursive, puisque la relation $<_{mul,<_{MPO}}$ sur les multi-ensembles de termes dépend de la relation $<_{MPO}$ sur les termes. Par la suite, nous utiliserons une notation simplifiée $<_{mul}$ à la place de $<_{mul,<_{MPO}}$. De plus, nous définissons :

$$\leq_{MPO} := \lambda s,\, t : term,\ s <_{MPO} t \vee s = t \quad .$$

4.3. Comportement des variables pour MPO

Cette partie est consacrée à quelques résultats concernant le comportement des variables vis-à-vis de la relation $<_{MPO}$.

Lemme 14 *Les variables sont les termes minimaux pour la relation $<_{MPO}$.*

Preuve. Ceci est immédiat car aucune règle de la définition de $<_{MPO}$ ne permet de dériver $s\ <_{MPO}\ x$, où s est un terme et x est une variable. □

Lemme 15 *Pour tout terme s et toute variable x, si $x <_{MPO} s$, alors $x \in Vars(s)$.*

Preuve. Par induction sur la structure de s. □

Lemme 16 *Pour tout terme s et toute variable x, si $x \in Vars(s)$ et $x \neq s$, alors $x <_{MPO} s$.*

Preuve. Par induction sur la structure de s. □

Lemme 17 *Soient s et t deux termes. Si $t\ \leq_{MPO}\ s$, alors $Vars(t) \subset Vars(s)$.*

Preuve. On doit montrer $\forall t : term,\ (P\ t)$ avec :

$$P := \lambda t : term. \forall s : term,\ t\ \leq_{MPO}\ s\ \rightarrow\ Vars(t) \subset Vars(s) \quad .$$

Procédons par induction sur le terme t, en appliquant le principe (6) de la section 4.1.
- **Cas de base** Prouvons $(P\ x)$ pour n'importe quelle variable x. Soit s un terme, que l'on suppose tel que $x <_{MPO} s$. Par le lemme 15, x est une variable de s et le résultat est immédiat.
- **Etape d'induction** Soit g un symbole fonctionnel et ts une liste de termes. Sous l'hypothèse d'induction

HInd1 : $\forall t' \,:\, term,\ t' \in ts \,\rightarrow\, (P\ t')$

on doit montrer $(P\ g(ts))$, c'est-à-dire $\forall s : term,\ (Q\ s)$ où :

$Q := \lambda s : term.\, g(ts) \leq_{MPO}\ s \rightarrow Vars(g(ts) \subset Vars(s).$

Par induction sur le terme s, deux cas doivent être considérés.
- **Cas de base** Prouvons $(Q\ x)$ pour x une variable quelconque. Supposons $g(ts)\ \leq_{MPO}\ x$: il s'ensuit $g(ts)\ <_{MPO}\ x$, ce qui est impossible d'après le lemme (14). Ainsi, ce cas est résolu par contradiction.
- **Etape d'induction** Soit f un symbole fonctionnel et ss une liste de termes. On doit établir $(Q\ f(ss))$, sous l'hypothèse d'induction :

  HInd2 : $\forall s' \in ss,\ (Q\ s')$.

  Supposons $g(ts)\ \leq_{MPO}\ f(ss)$. Deux cas se présentent : soit $f(ss)\ =\ g(ts)$, et la conclusion est triviale, soit $g(ts)\ <_{MPO}\ f(ss)$. Dans ce dernier cas, la preuve se poursuit selon la règle appliquée pour obtenir l'inégalité :
  - Avec (MPO_1), on a **H** : $\forall t' \in ts,\, t'\ <_{MPO}\ f(ss)$. Soit $x \in Vars(g(ts))$, prouvons que $x \in Vars(f(ss))$. Puisque $x \in Vars(g(ts))$, il existe nécessairement un élément t' de la liste ts tel que $x \in Vars(t')$. Or, **HInd1** nous indique que $(P\ t')$ est vraie et de **H** on peut déduire $Vars(t') \subset Vars(f(ss))$.
  - Avec (MPO_2), on a $ts\ \ll_{>_{MPO}}\ ss$. Comme dans le cas précédent, une variable x de $Vars(g(ts))$ est dans $Vars(t)$ pour un certain terme t de ts. Or, d'après le lemme 12, il existe un terme s de ss tel que $t \leq_{MPO} s$. Si $t = s$, $Vars(t) = Vars(s)$; sinon, par **Hind1**, $Vars(t) \subset Vars(s)$. Dans les deux cas, on peut en déduire que x est dans $Vars(s)$, donc dans $Vars(f(ss))$.
  - Avec (MPO_3), il existe un terme s' de la liste ss tel que $g(ts) \leq_{MPO} s'$. Grâce à **Hind2**, on sait que $(Q\ s')$ est vrai. Ainsi, toute variable de $g(ts)$ appartient à $Vars(s') \subset Vars(f(ss))$. □

4.4. Transitivité

La transitivité de la relation $<_{MPO}$ est prouvée par trois inductions imbriquées sur les termes, suivant le principe (6) de la section 4.1. Cette preuve utilise le lemme 17 ci-dessus.

Lemme 18 *Pour tous termes u,t,s, si $u <_{MPO} t$ et $t <_{MPO} s$, alors $u <_{MPO} s$.*

Preuve. La preuve se fait par inductions successives sur les termes u, t et s. Tout d'abord, on doit établir que $\forall u : term,\ (P\ u)$ où :

$$P := \lambda u : term. \forall t : term, \forall s : term,\ u <_{MPO} t \wedge t <_{MPO} s \,\rightarrow\, u <_{MPO} s \qquad .$$

- **Cas de base** Le terme u est une variable x. D'après le lemme 15, x appartient à $Vars(t)$ et donc, d'après le lemme 17, $Vars(t) \subset Vars(s)$. Ainsi, x est une variable apparaîssant dans le terme s. De plus, s n'est pas une variable puisqu'il n'est pas un terme minimal. On en conclut donc, en utilisant le lemme 16, que $x\ <_{MPO}\ s$.
- **Etape d'induction** Soit h un symbole fonctionnel et us une liste de termes. On doit prouver $(P\ h(us))$ sous l'hypothèse d'induction :

HInd1 : $\forall u' : term,\ u' \in us \,\rightarrow\, (P\ u')$.

Plus précisément, il faut établir $\forall t : term,\ (Q\ t)$, où :

$Q := \lambda t : term,\ \forall s : term,\ h(us) <_{MPO} t \wedge t <_{MPO} s \rightarrow h(us) <_{MPO} s.$

- **Cas de base** Le terme t est alors une variable, ce qui est impossible puisque les variables sont des éléments minimaux.
- **Etape d'induction** Soient g un symbole fonctionnel, ts une liste de termes. Il faut prouver $(Q\ g(ts))$ sous l'hypothèse d'induction :

  HInd2 : $\forall t' : term,\ t' \in ts \rightarrow (Q\ t')$.

  Le but est de la forme $\forall s : term,\ (R\ s)$ où :

  $R := \lambda s : term, h(us) <_{MPO} g(ts) \wedge g(ts) <_{MPO} s \rightarrow h(us) <_{MPO} s.$

  - **Cas de base** Comme précédemment, le terme s est alors une variable, ce qui est impossible puisque les variables sont des éléments minimaux.
  - **Etape d'induction** Soient f un symbole fonctionnel, ss une liste de termes. Il faut prouver $(R\ f(ss))$ sous l'hypothèse d'induction :

    HInd3 : $\forall s' : term,\ s' \in ss \rightarrow (R\ s')$.

    Par définition de R, ceci revient à prouver
    G : $h(us) <_{MPO} f(ss)$
    sous les deux hypothèses
    H1 : $h(us) <_{MPO} g(ts)$
    H2 : $g(ts) <_{MPO} f(ss)$.
    Chacune de ces hypothèses amène à considérer trois cas ; ainsi, nous avons à examiner neuf cas. Nous ne détaillons ici que trois d'entre eux, les autres ne présentant pas de difficulté particulière.

    Cas 1 Supposons par exemple que **H1** découle de (MPO_1) et **H2** de (MPO_2). Alors :
    (i) $h <_F g$
    (ii)$\forall u' : term,\ u' \in us \rightarrow u' <_{MPO} g(ts)$
    (iii) $g = f$
    De (i) et (iii), $h <_F f$. Pour u' un élément quelconque de us, montrons que $u' <_{MPO} f(ss)$. D'après (ii), **H2** et **HInd1**, on déduit $u' <_{MPO} f(ts)$. Le but est atteint en appliquant (MPO_1).

    Cas 2 Si **H1** et **H2** sont obtenues via (MPO_2), on a $f = g = h$ et $us \ll_{>_{MPO}} ts \ll_{>_{MPO}} ss$. On conclut que $us \ll_{>_{MPO}} ss$ par application du lemme 6 (voir section 3.3) et grâce à **HInd1**.

    Cas 3 Si **H1** vient de (MPO_3) et **H2** de (MPO_2), on a :
    (i) $\exists t : term,\ t \in ts \wedge h(us) \leq_{MPO} t$
    (ii) $g = f$
    (iii) $ts \ll_{>_{MPO}} ss$
    D'après le lemme 12 et (iii), il existe un terme s de ss tel que $t \leq_{MPO} s$. Si $t = s$, (i) permet d'écrire $h(us) \leq_{MPO} s$. Sinon, de **Hind2** et (i) on peut déduire $h(us) <_{MPO} s$. Dans les deux cas, (MPO_3) permet de conclure $h(us) \leq_{MPO} f(ss)$.

    Précisons que, dans le cas où **H1** et **H2** sont obtenues par application de la règle (MPO_1), la transitivité de $<_F$ est nécessaire.

4.5. Bonne fondation de MPO

Nous avons maintenant à notre disposition tous les outils nécessaires à la preuve de bonne fondation de MPO.

Théorème 1 *La relation $<_{MPO}$ est bien fondée.*

Preuve. Soit s un terme, prouvons que s est accessible pour $<_{MPO}$. On procède par induction sur s (principe 6).

- **Cas de base** Si s est une variable, d'après le lemme 14, s est un élément minimal, et donc est accessible.
- **Etape d'induction** Soient f un symbole fonctionnel et ss une liste de termes. On doit prouver $(acc_{<_{MPO}}\ f(ss))$ sous l'hypothèse d'induction :

  HInd1 : $(accs\ ss)$

  où $accs\ :=\ \lambda ss:\ (list\ term).\forall s' : term, s' \in ss \rightarrow (acc_{<_{MPO}}\ s')$.

  Ceci revient à prouver que $\forall f : F,\ (P\ f)$, où le prédicat P est défini par :

  $P\ :=\ \lambda f : F.\forall ss:\ (list\ term),\ (accs\, ss)\ \rightarrow (acc_{<_{MPO}}\ f(ss))$.

  Soit f un symbole fonctionnel. Puisque $<_F$ est supposée bien fondée, f est accessible pour cette relation et donc, en appliquant le principe d'induction 2, on est conduit à prouver $(P\ f)$ sous l'hypothèse :

  Hind2 : $\forall g : F,\ g <_F f\ \rightarrow\ (P\ g)$.

  Le but $(P\ f)$ peut s'écrire sous la forme $(\forall ss:\ (list\ term))\ (Q\ ss)$ avec Q le prédicat défini par :

  $Q\ :=\ \lambda ss : (list\ term).\,(accs\ ss)\ \rightarrow (acc_{<_{MPO}}\ f(ss))$

  Soit alors ss une liste de termes telle que $(accs\ ss)$. D'après le lemme 10 de la section 3.3, ss est accessible pour la relation $\ll_{<_{MPO}}$. Par conséquent, on peut montrer que ss satisfait Q sous l'hypothèse d'induction :

  HInd3 : $\forall ts:\ (list\ term),\ ts\ \ll_{<_{MPO}}\ ss\ \rightarrow\ (Q\ ts)$.

  Par définition du prédicat acc (règle 1), prouver $(Q\ \ ss)$ revient à prouver la proposition $(accs\ ss) \rightarrow \forall t : term, (R\, t)$, où R est défini par :

  $R\ :=\ \lambda t : term.\, t\ <_{MPO}\ f(ss)\ \rightarrow\ (acc_{<_{MPO}}\ t)$

  Ainsi, à ce stade, il faut prouver $(R\ t)$ sous l'hypothèse **Hind1**. Procédons par induction sur le terme t.
  - **Cas de base** Si t est une variable, d'après le lemme 14, s est un élément minimal, et donc est accessible.
  - **Etape d'induction** Soient g un symbole fonctionnel et ts une liste de termes. Prouvons $(R\ g(ts))$ sous l'hypothèse d'induction :

HInd4 : $\forall t' : term,\ t' \in ts, \rightarrow (R\ t')$.

Par définition de R, il faut montrer $(acc_{<_{MPO}}\ g(ts))$ sous l'hypothèse **H** : $g(ts) <_{MPO} f(ss)$. Ceci nous amène à considérer trois cas, suivant la règle (MPO$_i$) utilisée pour obtenir l'inégalité.

Cas MPO$_1$. On sait que : (i) $g <_F f$
(ii) $\forall t' : term,\ t' \in ts \rightarrow t' <_{MPO} f(ss)$

D'après (i) et l'hypothèse **Hind2**, g satisfait le prédicat P. Ainsi, pour démontrer l'accessibilité de $g(ts)$, il suffit de prouver $(accs\ ts)$. Mais tout élément t' de ts est inférieur à $f(ss)$ d'après (ii), et ainsi est accessible d'après **HInd4**.

Cas MPO$_2$. Les hypothèses sont alors : (i) $f = g$
(ii) $ts \ll_{MPO} ss$

D'après (i), le but devient $(acc_{<_{MPO}}\ f(ts))$. Les hypothèses (ii) et **HInd3** permettent de déduire que ts satisfait Q. Ainsi, est-on ramené à prouver que tout terme t' de ts est accessible. Or d'après **HInd4**, t' est accessible dès qu'il est inférieur à $f(ss)$. Mais, d'après (MPO$_3$), $t' <_{MPO} f(ts)$ et, d'après **H**, $f(ts) <_{MPO} f(ss)$ soit, par transitivité de MPO, $t' <_{MPO} f(ss)$

Cas MPO$_3$. Dans ce dernier cas, $g(ts)$ est inférieur ou égal à un élément s' de ss. Puisque, d'après **HInd1**, s' est accessible, il en va de même pour $g(ts)$. □

5. Travaux connexes et conclusion

La preuve de bonne fondation de RPO donnée dans [Der82] s'appuie sur le fait que les ordres de simplification sont bien fondés puisque, d'après le théorème de Kruskal, ils contiennent un plongement homéomorphique qui est un bon ordre partiel, et donc est bien fondé. Néanmoins, la preuve du théorème de Kruskal n'est pas constructive et se fait sous l'hypothèse d'une signature finie.

Dans [Les82], Lescanne introduit un ordre de décomposition sur les termes, puis prouve que cet ordre est bien fondé et équivalent à MPO. Sa preuve de bonne fondation, bien qu'élémentaire n'est pas réellement constructive, telle qu'elle est présentée dans l'article. Il semblerait qu'on puisse avec quelques efforts, en donner une version constructive. Elle est cependant beaucoup moins directe que celle que nous présentons ici. Il est à noter toutefois qu'elle ne fait pas appel à la transitivité. De plus, un autre intérêt de cette approche réside dans le fait qu'elle fournit un algorithme efficace pour comparer deux termes.

Dans [FZ95], Ferreira et Zantema démontrent plusieurs théorèmes à propos de la bonne fondation d'ordres sur les termes du premier ordre. Ces résultats sont généraux et même complets pour ce qui est de la terminaison des systèmes de réécriture dans le cas d'une signature finie. Ils peuvent être appliqués naturellement à RPO, mais bien que leurs preuves n'utilisent pas le théorème de Kruskal, elles ne sont pas constructives.

Dans [JR99] et [JR03], Jouannaud et Rubio proposent une preuve constructive de terminaison au moyen d'un ordre récursif sur les chemins d'ordre supérieur (horpo), en utilisant la technique de Tait-Girard [GLT88]. La restriction de cette preuve aux termes du premier ordre revient à donner une preuve de bonne fondation de MPO par induction structurelle sur les termes, comme décrit dans [van01]. Notre preuve Coq utilise une simplification de cette preuve appliquée à MPO (dans le sens où nous évitons leur utilisation d'un ordre lexicographique auxiliaire sur des triplets).

Dans [GL01], Goubault-Larrecq démontre un théorème énonçant un résultat de bonne fondation, dont la preuve a été vérifiée en Coq. Le résultat est général en ce sens qu'il ne dépend pas de la structure

de terme et, par suite est applicable à d'autres algèbres. La preuve en reste élémentaire. Toutefois, la preuve que ce théorème généralise les résultats de Ferreira et Zantema, fait intervenir un argument non constructif. De plus, appliquer ce théorème à MPO, et en particulier montrer que l'hypothèse (iv) de son théorème est satisfaite, n'est pas plus simple que la preuve directe que nous présentons ici. En outre, pour appliquer directement ce théorème à MPO, la définition de cet ordre doit être modifiée : il faut ajouter la condition $\forall\ i \in \{1, \ldots, m\},\ t_i \ <_{MPO} \ f(s_1, \ldots, s_n)$ comme prémisse de la règle MPO_2 (voir la section 4.2), ce qui est gênant puisqu'il faut ensuite prouver que cette condition est superflue.
Dawson et Goré, dans [DG04], prouvent un théorème général pour établir la bonne fondation de relations closes par contexte ; cette preuve a été certifiée en Isabelle. Là également, s'assurer des hypothèses du théorème peut être difficile et requiert en particulier d'exhiber une relation auxiliaire bien-fondée non triviale. Les auteurs appliquent leur résultat à de nombreux exemples, parmi lesquels LPO. La preuve obtenue dans ce cas paraît raisonnablement facile et il serait intéressant de comparer leur approche dans le cas de MPO avec une approche directe telle que la notre.
Mentionnons enfin les travaux de Leclerc [Lec95], qui donne en Coq une preuve de terminaison de TRS en utilisant MPO. Néanmoins, cette preuve n'utilise pas la bonne fondation de MPO, qui n'est donc pas démontrée, mais plutôt un plongement des règles de réécriture dans un certain ordre bien fondé basé sur la hiérarchie de Grzegorzcyk de fonctions de la théorie des nombres.

Nous avons produit dans le CIC une preuve directe de la bonne fondation de MPO. Cette preuve est particulièrement courte, ne fait appel qu'à des résultats préliminaires élémentaires, et s'applique à des termes contenant des variables, construits sur une signature non nécessairement finie de symboles fonctionnels d'arité variable. Elle peut également s'appliquer dans le cas de symboles fonctionnels d'arité fixée (termes *algébriques*). En effet, Blanqui a intégré dans CoLoR [CoL] une bibliothèque comprenant notamment une conversion des termes algébriques vers les termes variadiques, dont il prouve qu'elle préserve la terminaison. Ce travail est une première étape vers l'extension aux cas de LPO et RPO avec statut.

Remerciements Les auteurs tiennent à remercier Frédéric Blanqui pour sa collaboration amicale et de qualité. Ils adressent également leurs remerciements à Pierre Lescanne pour sa relecture attentive de cet article et ses nombreux commentaires constructifs (dans l'acception vulgaire du terme !). Ils remercient enfin les rapporteurs pour leur remarques et conseils qui leur ont permis d'améliorer la version finale de ce travail.

Références

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, New York, 1998.

[CoL] CoLoR : a Coq Library on Rewriting and Termination. http ://color.loria.fr.

[Der82] Nachum Dershowitz. Orderings for Term Rewriting Systems. *Theoretical Computer Science*, 3(17) :279–301, 1982.

[DG04] E.Jeremy Dawson and Rajeev Goré. A General Theorem on Termination of Rewriting. In *Computer Science Logic, CSL'04*, number 3210 in LNCS, pages 100–114. Springer-Verlag, 2004.

[FZ95] Maria.C.F. Ferreira and Hans Zantema. Well-Foundedness of Term Orderings. In *4th International Workshop on Conditional Term Rewriting Systems (CTRS'94)*, number 968 in LNCS, pages 106–123. Springer-Verlag, 1995.

[GL01] Jean Goubault-Larrecq. Well-Founded Recursive Relations. In *15th Worshop on Computer Science Logic (CSL'01), Paris*, volume 2142 of *LNCS*, pages 484–497. Springer-Verlag, 2001.

[GLT88] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge Tracts in Theoritical Computer, Science 7, 1988.

[HL78] Gérard Huet and Dallas Lankford. On the Uniform Halting Problem for Term Rewriting Systems. Technical Report 283, IRIA, 1978.

[JR99] Jean-Pierre Jouannaud and Albert Rubio. The Higher-Order Recursive Path Ordering. In *Proceedings of the 14th annual IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 402–411, Trento, Italy, 1999.

[JR03] Jean-Pierre Jouannaud and Albert Rubio. Higher-Order Recursive Path Orderings 'à la carte'. Technical report, http ://www.lix.polytechnique.fr/Labo/Jean-Pierre.Jouannaud/biblio.html, 2003.

[Kop04] Adam Koprowski. Well-foundedness of the Higher-Order Recursive Path Ordering in Coq. Master thesis, Free University of Amsterdam (The Netherlands) and Warsaw University (Poland), 2004.

[Lec95] François Leclerc. Termination Proof of Term Rewriting System with the Multiset Path Ordering. A Complete Development in the System Coq. In *TLCA*, pages 312–327, 1995.

[Les82] Pierre Lescanne. Some Properties of Decomposition Ordering, a Simplification Ordering to Prove Termination of Rewriting Systems. *R.A.I.R.O. Theoretical Informatics*, 14(4) :331–347, 1982.

[Nip98] Tobias Nipkow. An Inductive Proof of the Well-Foundedness of the Multiset Order. Due to Wilfried Buchholz. Technical report, http ://www4.informatik.tu-muenchen.de/~nipkow/misc/index.html, 1998.

[Tea04] The Coq Development Team. The Coq Proof Assistant Reference Manual – Version V8.0. Technical report, LogiCal Project-INRIA, 2004.

[Coq] The Coq Proof Assistant. http ://coq.inria.fr.

[van01] Femke van Raamsdonk. On Termination of Higher-Order Rewriting. In *Proceedings of the 12th International Conference on Rewriting Techniques and Applications (RTA'01)*, pages 261–275, Utrecht, The Netherlands, 2001.