

DERSHOWITZERIES

JTL 23 octobre 1981.

Rappel: (Kruskaleries) Si \geqslant est un prébel ordre et \geqslant un préordre vérifiant :

$$\underline{\text{Ax1}}: f(\dots t \dots) \geqslant t,$$

Ax2: $(t_{i_1} \geqslant u_1, t_{i_2} \geqslant u_2, \dots, t_{i_n} \geqslant u_n)$ pour $i_1 < i_2 < \dots < i_n$
et $f\vec{t} \geqslant \vec{u}$ impliquent $f\vec{t} \geqslant \vec{u}$,

Alors \geqslant est bien fondé.

Définition: (Plaisted 78 - revue Dershowitz 78). Soit \geqslant un préordre donné sur les termes. Le "recursive path ordering" associé (en court $rpo(\geqslant)$) est défini par: $t = \vec{f}t' > g\vec{u} = u$ ssi

- (1) $t \geqslant u$ et $t > u_i$ pour tout i ,
- (2) $t \equiv u$ et $\{t_1, t_2, \dots, t_n\} \geqslant_m \{u_1, u_2, \dots, u_p\}$,
- (3) $t_i \geqslant u$ pour un i ,

où \geqslant_m est l'ordre sur les multi-ensembles associé à \geqslant et $t \geqslant u$ ssi $t > u$ ou $t \simeq u$, avec \simeq défini par:

$t \simeq u$ ssi $t \equiv u$, $t = \vec{f}t'$, $u = g\vec{u}$ et $\{t_1, t_2, \dots, t_n\} \geqslant_m \{u_1, u_2, \dots, u_p\}$

Nous allons montrer les théorèmes suivants :

Théorème 1: Si \geqslant est bien-fondé, alors $rpo(\geqslant)$ est aussi bien-fondé.

Théorème 2: Si \geqslant vérifie:

(μ) $t \rightarrow u$ implique $f(\dots t \dots) \geqslant f(\dots u \dots)$

et si un système $\Sigma = \{\alpha_i \rightarrow \beta_i\}$ vérifie $\sigma(\alpha_i) > \sigma(\beta_i)$ pour toute substitution close σ (où \geqslant est le $rpo(\geqslant)$), alors \rightarrow est noetherien.

Lemmes:

- (a) \simeq est une relation d'équivalence,
- (b) $t > u \simeq v \Rightarrow t > v$,
- (c) $t \simeq u > v \Rightarrow t > v$,
- (d) $t > u > v \Rightarrow t > v$,
- (e) $>$ est irréflexif,

Démonstrations:

(a) immédiat puis \equiv est une équivalence

(b) Par récurrence sur $\|t\| + \|u\| + \|v\|$. Trois cas selon la définition de $t > u$.

cas 1: $t \geq u$ et $t > u_i$ pour tout i . Or $u \equiv v$, puisque $u \approx v$. Donc $t \geq v$. Or, par définition de \approx_m , pour tout v_i , il existe un u_j tel que $u_j \approx_m v_i$. Or $t > u_j$. Donc $t > v_i$ par récurrence. Donc $t > v$ par la règle 1.

cas 2: $t \equiv u$ et $\{t_1, \dots, t_n\} \geq_m \{u_1, \dots, u_p\}$. Comme $u \approx v$, on a $u \equiv v$ et $\{u_1, \dots, u_p\} \approx_m \{v_1, \dots, v_q\}$. Donc $t \equiv v$ et par récurrence $\{t_1, \dots, t_n\} \geq_m \{v_1, \dots, v_q\}$. D'où $t > v$ par la règle 2.

cas 3: $t_i \geq u$ pour un i . Comme $u \approx v$, on a $t_i \geq v$ par récurrence. Donc $t > v$ par la règle 3.

(c) Par récurrence sur $\|t\| + \|u\| + \|v\|$. Trois cas selon la définition de $u > v$.

cas 1: $t \equiv u \geq v$. Donc $t \geq v$. De plus $u > v_i$ pour tout i . Donc $t > v_i$ par récurrence. Donc $t > v$ par la règle 1.

cas 2: $t \equiv u \equiv v$. Et $\{t_1, \dots, t_n\} \approx_m \{u_1, \dots, u_p\} \geq_m \{v_1, \dots, v_q\}$. Donc $\{t_1, \dots, t_n\} \geq_m \{v_1, \dots, v_q\}$ par récurrence. D'où $t > v$ par la règle 2.

cas 3: $u_i \geq v$ pour un i . Or $\{t_1, \dots, t_n\} \approx_m \{u_1, \dots, u_p\}$. Donc, il existe j tel que $t_j \approx_m u_i$. D'où $t_j \geq v$ par récurrence. Et $t > v$ par la règle 3.

(d) Même récurrence.

(e) $t > t$ est impossible, par récurrence sur $\|t\|$. Par la règle c'est impossible puisque $t \geq t$ est impossible. Par la règle 2, \geq_m est impossible puisque \geq_m est irréflexif, puisque par (d) et récurrence on sait que \geq est un ordre strict pour les termes de taille plus petite qu' $\|t\|$. Par la règle 3, on aurait $t_i \geq t$ pour un i .

Or $t > t_i$ par la règle 3 également. Donc $t_i > t_i$ par (c) et (d). Impossible par récurrence. \square

Démonstration du théorème 1: \Rightarrow est bien-fondé. Or on peut compléter \Rightarrow en \Rightarrow' bon ordre admettant \equiv aussi pour équivalence compatible (voir Birkhoff). Notons $t \geq' u$ pour $t \geq u$ ou $t \equiv u$. Les lemmes a,b,...e restent vrai pour le rpo associé à \Rightarrow' . Notons \geq' ce rpo. Et $t \geq' u$ pour $t \geq u$ et $t \not\sim u$. Les bons ordres étant des cas particuliers des prébon ordres, il reste à montrer que \geq' vérifie Ax1 et Ax2 (par rapport à \Rightarrow') pour affirmer que \geq' est bien fondé. Or, puisque \Rightarrow' est une extension de \Rightarrow , on a trivialement \geq' extension de \geq , et donc \geq est bien-fondé. Montrons donc Ax1 et Ax2. D'abord Ax1 est vrai par la règle 3 puisque $t \not\sim t$. Pour Ax2, on a deux cas. Si $t = \vec{f} \Rightarrow' \vec{g} = u$, alors $t \geq' u$ par la règle 1, puisque $t > t_{ik} \geq' u_k$ pour tout k . Si $t = \vec{f} = \vec{g} = u$, alors on a soit $\{t_1, \dots, t_p\} \geq_m^1 \{u_1, \dots, u_n\}$, soit $\{t_1, \dots, t_p\} \not\sim_m \{u_1, \dots, u_n\}$, i.e. soit $t > u$ par règle 2, soit $t \not\sim u$. Donc $t \geq' u$. \square

Démonstration du théorème 2: On va montrer que $\geq \cap \rightarrow$ est compatible avec la structure. Donc, supposons $t \rightarrow u$ et $t \geq u$ montrons que $f(\ldots t \ldots) \geq f(\ldots u \ldots)$. D'après l'axiome (x), on a $f(t) \geq f(u)$. Et on applique soit les règles 1 ou 2. \square

Exemples:

1) Posons $t = \vec{f} \vec{t} \Rightarrow g \vec{u} = u$ ssi $f \Rightarrow g$ dans un ordre supposé bien fondé sur les symboles de fonctions, et $t = \vec{f} \vec{t} \doteq g \vec{u} = u$ ssi $f = g$. Remarquons que \doteq est compatible avec \Rightarrow , et que (u) est trivialement vrai. On retrouve les vraies définitions de Plaisted ou Dershovitz.

$$\left\{ \begin{array}{l} \neg \vec{p} \rightarrow p \\ \neg(p \vee q) \rightarrow \neg p \wedge \neg q \\ \neg(p \wedge q) \rightarrow \neg p \vee \neg q \\ p \wedge (q \vee r) \rightarrow (p \wedge q) \vee (p \wedge r) \\ (p \vee q) \wedge r \rightarrow (p \wedge r) \vee (q \wedge r) \end{array} \right.$$

Prenons $\neg \Rightarrow \wedge \Rightarrow \vee$ comme ordre sur les symboles.

2) Factorielle:

$$\left\{ \begin{array}{l} f(sx) \rightarrow sx * f(p(sx)) \\ f(0) \rightarrow s0 \\ p(sx) \rightarrow x \end{array} \right.$$

Dénotons par $\llbracket t \rrbracket$ l'interprétation naturelle sur \mathbb{N} . Posons $f(t) \triangleright f(u)$ ssi $\llbracket t \rrbracket \geq_N \llbracket u \rrbracket$, $f(t) \triangleright u * v$, $f(t) \triangleright p(u)$, $f(t) \triangleright su$ pour tout t, u et v . Alors (u) est trivialement vérifié. Et les réécritures vérifient le rpo.

Dans cet exemple, remarquons que l'on a :

$$p(sx) \triangleright sx, \text{ mais } f(sx) \triangleright f(p(sx)).$$

Quelques remarques supplémentaires sur $>$:

1) termes ouverts: Le rpo $>$ a été défini dans le cas des termes clos. Pour les termes avec variables, il suffit de considérer les variables comme des symboles de fonctions non reliés par $>$ aux autres symboles ou variables. Alors, on montre que $>$ est fermé par substitution, et coïncide sur les termes fermés.

2) ambiguité de la définition: Les cas 1 et 2 de la définition du rpo sont exclusifs. Pas le cas 3 ! Ce qui peut être continu pour calculer si deux termes sont reliés. Montrons que la définition est équivalente si on change la règle 3 en:

(3') $t \not> u$ et $t_i \geq u$ pour un i .

Appelons $>'$ l'ordre défini avec (3'). Clairement, on a $>'$ inclus dans $>$. Montrons le contraire, c'est à dire que $t > u$ implique $t >' u$ par récurrence sur $\|t\| + \|u\|$.

Cas 1 et 2: immédiats.

Cas 3: On a $t_i \geq u$ pour un i . On a trois sous-cas, selon que $t > u$, $t = u$ ou $t \not> u$.

Cas 3.1: $t > u$. Comme $t_i \geq u$ et $u > u_j$ pour tout j . On a $t_i > u_j$. Comme $t > t_i$, on a $t > u_j$ pour tout j . Donc $t > u_j$ pour tout j par récurrence, et $t > u$ par règle 1.

Cas 3.2: $t = u$. De même $t_i > u_j$ pour tout j . Donc $t_i > u_j$ pour tout j par récurrence. D'où $t > u$ par règle 2.

Cas 3.3: $t \not> u$. Alors $t_i \geq u$ par récurrence. Donc $t > u$ par règle 3. \square

7

En conclusion, le rpo se calcule vite, si on sait calculer vite \geq_m ou \simeq_m . C'est un problème relatif aux multi ensembles avec des ordres partiels. (On peut vérifier que dans le cas d'un ordre total \geq un simple tri suffit)

Rpo avec ordres lexicographiques :

Définition : (Kamin + JJL 80). Soit \geq un préordre donné sur les termes. Le "lexicographic recursive path ordering" (lypo) associé est défini par : $t = \vec{f} \vec{t} > \vec{g} \vec{u} = u$ ssi

(1) $t \geq u$ et $t > u_i$ pour tout i ,

(2) $t = u$ et $\vec{f} \geq_{lex} \vec{u}$ et $t > u_i$ pour tout i ,

(3) $t_i \geq u$ pour un i ,

où \geq_{lex} est l'ordre lexicographique associé à \geq , et $t \geq u$ ssi $t \geq u$ et $t \not\simeq u$, où :

$t \simeq u$ ssi $t = u$, $t = \vec{f} \vec{t}$, $u = \vec{g} \vec{u}$ et $t_i \simeq u_i$ pour tout i .

On démontre aussi les théorèmes 1 et 2. La remarque sur les termes ouverts reste vraie. L'ordre marche mieux pour les exemples suivants :

Exemple 1 :

$$\left\{ \begin{array}{l} \neg p \rightarrow p \\ \neg(p \vee q) \rightarrow \neg p \wedge \neg q \\ \neg(p \wedge q) \rightarrow \neg p \vee \neg q \\ p \wedge (q \vee r) \rightarrow (p \wedge q) \vee (p \wedge r) \\ (p \vee q) \wedge r \rightarrow (p \wedge r) \vee (q \wedge r) \\ (p \wedge q) \wedge r \rightarrow p \wedge (q \wedge r) \\ (p \vee q) \vee r \rightarrow p \vee (q \vee r). \end{array} \right.$$

Exemple 2: les groupes (voir Knuth-Bendix)

Quelques remarques supplémentaires:

1) ambiguité de la définition: On peut remplacer (2) et (3) de la définition du bpo par: (en notant $\alpha(t)$ le nombre n de fils de $t = f(t_1, t_2, \dots, t_n)$): confusion des notations $\alpha(t)$ et n

(2') $t = u$ et $[\alpha(t) > \alpha(u)$ et $t_1 \simeq u_1, t_2 \simeq u_2, \dots, t_{\alpha(u)} \simeq u_{\alpha(u)}$
 $\exists i \leq \min(\alpha(u), \alpha(t))$
ou $[\alpha(t) = \alpha(u)$ et $t_1 \simeq u_1, t_2 \simeq u_2, \dots, t_{i-1} \simeq u_{i-1},$
 $t_i > u_i, t > u_{i+1}, t > u_{i+2}, \dots, t > u_n]$

(3') $[t \neq u$ ou $[t = u$ et $\alpha(t) < \alpha(u)]]$ et $t_i \gtrsim u$ pour un i]
ou $[t = u$ et $\alpha(t) = \alpha(u)$ et $t_1 \simeq u_1, t_2 \simeq u_2, \dots, t_{i-1} \simeq u_{i-1},$
 $t_i \not\simeq u_i$ et $t_{i+k} \gtrsim u$ pour un i .]

Cette définition compliquée peut être montrée équivalente à celle donnée par 1,2,3. (La démonstration est identique à celle effectuée pour les multiensembles). L'intérêt est que les cas 1,2',3' sont exclusifs et la complexité de calculer $t > u$ à l'air d'être $O(\|t\| \times \|u\|)$.

2) complexité de la définition avec les multiensembles:

Cette définition me semble curieusement avoir la même complexité. En effet, pour calculer $t > u$, le cas douloureux semble le cas 2. Or, pour calculer $\{t_1, t_2, \dots, t_n\} \geq_m \{u_1, \dots, u_p\}$, on peut éliminer les paires $t_i \simeq u_j$ en comparant toutes les

paires successivement, puis vérifier que tous les u_i restants sont dominés par un t_i restant, en recomparant à nouveau les ensembles qui restent. Donc :

$$\begin{aligned} C(t, u) &\leq k + \sum_i C(t, u_i) && (\text{règle 1}) \\ &\leq k + 2 \sum_{i,j} C(t_i, u_j) && (\text{règle 2}) * \\ &\leq k + 2 \sum_i C(t_i, u) && (\text{règle 3}') \end{aligned}$$

en partant du principe que $t > u$ et $t \approx u$ ont même complexité $C(t, u)$. Par récurrence, on en conclut que $C(t, u) = O(\|t\| \times \|u\|)$! Par ailleurs, pour calculer $\{t_1, \dots, t_m\} \geq_m \{u_1, \dots, u_n\}$, on a besoin de $m \times n$ comparaisons dans le cas pire où tous les éléments ne sont pas reliés. Donc $\|t\| \times \|u\|$ ne semble pas si mauvais comme complexité.

* par règle (2) en fait

$$\leq k + \sum_{i,j} C(t_i, u_j) + K^2 \quad \text{où } K \text{ est aussi maximale.}$$

Donc $C(t, u) = O(\|t\| \times \|u\|, K^2)$