Réécriture

Terminaison des systèmes de réécriture

version du October 27, 2004 – 13 h 31

Indécidabilité de la terminaison

Problème de correspondance de Post

Un ensemble $(\alpha_1, \beta_1), ..., (\alpha_n, \beta_n)$

où $\alpha_i \in A^*$ et $\beta_i \in A^*$

est appelé un problème de correspondance de Post.

Une holorime est formée

- d'un mot $w \in A^+$
- et d'une suite $(i_1, ..., i_p) \in [1..n]^+$

tels que

$$w = \alpha_{i_1} ... \alpha_{i_p} = \beta_{i_1} ... \beta_{i_p}.$$

Problème de correspondance de Post

1. (ala, tour)
2. (aman, dela)
3. (dela, rène)
4. (gal, galaman)
5. (magnanime, anime)
6. (rène, ala)
7. (tour, magn)

Une holorime est galamandelarènealatourmagnanime avec la suite $(4,2,3,6,1,7,5)$.

> Gal, amant de la reine, alla, tour magnanime,
> Galamment de l'arène, à la tour Magne, à Nîmes.
>
> Victor Hugo

Mais là c’est super-facile.

Une autre holorime est due à la poétesse Louis de Vilmorin:

> Étonnamment monotone et lasse
>
> Est ton âme en mon automne, hélas!

Des exemples

Les problèmes de correspondance ci-dessus :

i	α_i	β_i
1	010	101
2	00	000
3	101	10

1

i	α_i	β_i
1	101	10
2	11	011
3	011	101

2

ont-ils une holorime ?

Des exemples (encore)

Et ceux là?

i	α_i	β_i
1	011	1
2	1	0
3	0	011
4	10	001

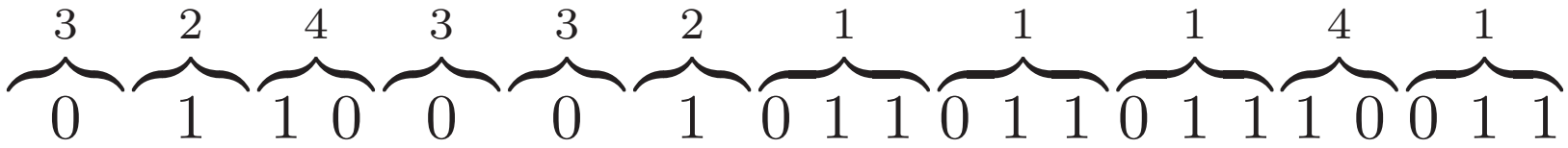
3

i	α_i	β_i
1	000	0
2	0	111
3	11	0
4	10	100

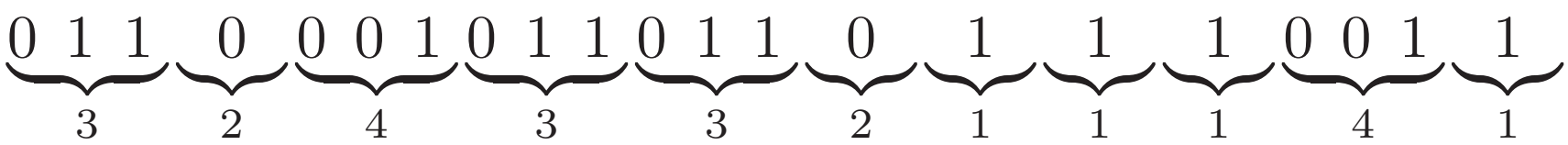
4

ont-ils une holorime ?

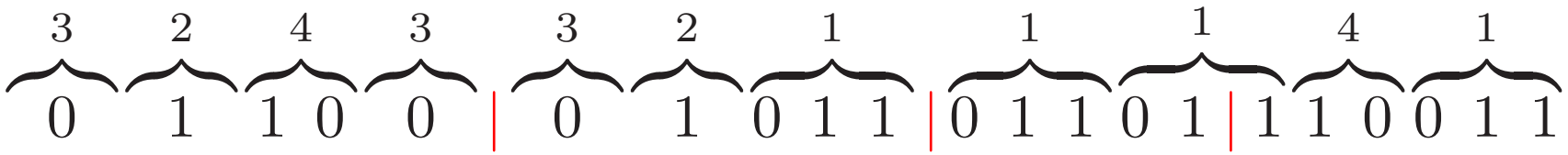
Le problème 3 a pour solution



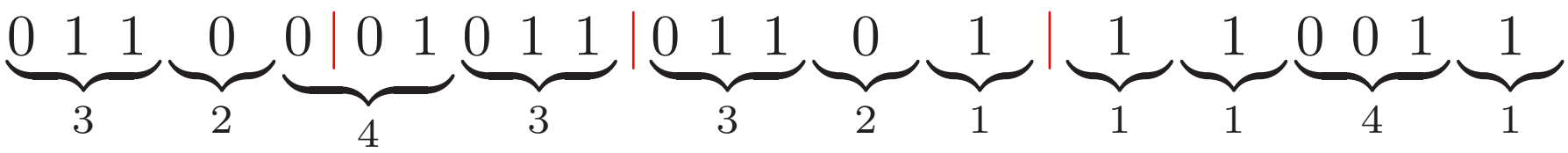
et



Le problème 3 a pour solution



et



La plus petite solution du problème 4 est de taille 204.

A vous de la trouver!

Le problème de correspondance de Post (ou PCP) est indécidable,

à partir de deux éléments dans A.

Autrement dit, il n'y a pas d'algorithme avec

- entrée : un PCP sur deux lettres
- sortie : le problème a une holorime ou le problème n'a pas d'holorime .

Réduction de la terminaison à PCP

On considère la signature

- $\Sigma_0 = \{\#\}$,
- $\Sigma_1 = A$,
- $\Sigma_2 = \emptyset$,
- $\Sigma_3 = \{f\}$,
- $\Sigma_n = \emptyset$ pour $n \geq 4$.

A chaque mot $a_1 a_2 ... a_n$ de A^* on peut associer un terme

dit monadique

$$(a_1(a_2(...(a_n(\#))...))) \in T(\Sigma).$$

que l'on note $a_1 a_2 ... a_n$.

Dans la suite $a_1 a_2 ... a_n x$ pour $x \in X$ est

$$(a_1(a_2(...(a_n(x))...))) \in T(\Sigma, X).$$

Étant donné un problème de correspondance de Post sur

$A \equiv \{a_1, ..., a_m\}$.

On considère le système de réécriture :

$$R_1 = \begin{cases} f(\alpha_1 x, \beta_1 y, z) & \longrightarrow & f(x, y, z) \\ & \vdots & \\ f(\alpha_n x, \beta_n y, z) & \longrightarrow & f(x, y, z) \end{cases}$$

$$R_2 = \begin{cases} f(\#, \#, a_1(y)) & \longrightarrow & f(a_1(y), a_1(y), a_1(y)) \\ & \vdots & \\ f(\#, \#, a_m(y)) & \longrightarrow & f(a_m(y), a_m(y), a_m(y)) \end{cases}$$

$\xrightarrow[R_1 \cup R_2]{}$ ne termine pas si et seulement si PCP a une holorime.

Si PCP a une holorime w, alors

$$f(\#, \#, w) \xrightarrow[R_2]{} f(w, w, w) \xrightarrow[R_1]{+} f(\#, \#, w).$$

Si $\xrightarrow[R_1 \cup R_2]{}$ ne termine pas, alors elle passe une infinité de fois par des règles $\xrightarrow[R_2]{}$. Et cela implique qu'il y a une holorime.

En effet, les règles $\xrightarrow[R_1]{}$ décroissent la taille des termes et ne peuvent pas contribuer seules à la non terminaison.

On a d'ailleurs montré plus fort : $\xrightarrow[R_1 \cup R_2]{}$ est cyclique si et seulement si PCP a une holorime.

Donc

- l'acyclicité,
- et la terminaison

sont indécidables.

Ordre de réduction

Un ordre de réécriture est une ordre sur $T(\Sigma, V)$ qui est

- compatible c'est-à-dire si $s > u$

  $f(t_1, ..., t_{i-1}, s, t_{i+1}, ..., t_n) > f(t_1, ..., t_{i-1}, u, t_{i+1}, ..., t_n)$

- clos par substitution c'est-à-dire si

  $s_1 > s_2 \quad \Rightarrow \quad (\forall \sigma \in Subst(T(\Sigma, V))) \quad \sigma(s_1) > \sigma(s_2)$.

Un ordre de réduction est un ordre de réécriture noethérien

(c'est-à-dire qui termine).

N.B. La taille du terme n'est pas un ordre de réduction.

Un système de réécriture R termine

si et seulement si

il existe un ordre de réduction $>$

tel que $l > r$ pour tout $l \longrightarrow r \in R$.

Si : $>$ est une relation noethérienne qui contient $\xrightarrow{+}$.

Donc $\xrightarrow{+}$ elle-même est noethérienne.

Seulement si : $\xrightarrow{+}$ est une ordre de réduction qui satisfait clairement la condition (d'ordonner les règles de réécriture, il est clair que l'on a $l \xrightarrow{+} r$, on peut donc prendre $\xrightarrow{+}$ comme ordre).

Interprétations

Algèbre ordonnée et ordre induit

Une algèbre ordonnée est la donnée d'une algèbre $\mathcal{A}$

et d'un ordre $>$ sur A.

L'ordre $>_{\mathcal{A}}$ sur $T(\Sigma, V)$ induit par cette algèbre est défini par :

$s >_{\mathcal{A}} t$ ssi $\pi(s) > \pi(t)$ pour tout morphisme $\pi : T(\Sigma, V) \to \mathcal{A}$.

Monotonie

Une fonction $F : A^n \to A$ est monotone si

$$b > c \quad \Rightarrow F(a_1, ..., a_{i-1}, b, a_{i+1}, ..., a_n) > F(a_1, ..., a_{i-1}, c, a_{i+1}, ..., a_n)$$

Si $\mathcal{A}$ est une algèbre ordonnée, munie d'un ordre noethérien,
si toutes les interprétations $f^{\mathcal{A}}$ sont monotones,
alors $>_{\mathcal{A}}$ est un ordre de réduction.

Polynômes monotones

$A = \mathbb{N}_\star$

Les $f_n^{\mathcal{A}} \in \Sigma_n^{\mathcal{A}}$ sont des polynômes à n indéterminées de $\mathbb{N}[X_1, ..., X_n]$.

Un polynôme est monotone

- s'il dépend de toutes ses indéterminées,
- c'est-à-dire que pour chaque i tel que $1 \leq i \leq n$,
  il contient un monôme avec une occurrence de X_i.

Interprétations polynomiales

Une interprétation polynomiale monotone est une interprétation dans laquelle toutes les interprétations sont des polynômes monotones.

L'ordre induit est un ordre de réduction, appelé ordre polynômial.

Exercice

Prouver par un ordre polynômial la terminaison des systèmes suivants.

$$\begin{array}{rcll} (x * y) * z & \longrightarrow & x * (y * z) & \text{(A)} \\ f(x * y) & \longrightarrow & f(x) * f(y) & \text{(E)} \end{array}$$

$$\begin{array}{rcll} (x * y) * z & \longrightarrow & x * (y * z) & \text{(A)} \\ f(x) * f(y) & \longrightarrow & f(x * y) & \text{(E)} \\ f(x) * (f(y) * z) & \longrightarrow & f(x * y) * z & \text{(EA)} \end{array}$$

Les limites des ordres polynômiaux I

Prouver la satisfaction d'un ensemble

$$(F_i(X_1, ..., X_n) > G_i(X_1, ..., X_n))_{1 \leq i \leq m}$$

- où $F_i \in \mathbb{N}[X_1, ..., X_n]$ et $G_i \in \mathbb{N}[X_1, ..., X_n]$
- et où les X_i parcourent $\mathbb{N}$

est indécidable.

Pour le prouver on se ramène au 10^{eme} problème de Hilbert.

10^{eme} problème de Hilbert

Dans son 10^{eme} problème (1900) Hilbert demandait un algorithme pour déterminer si les systèmes d'équations diophantiennes ont une solution ou non,

Un système d'équations diophantiennes est une suite $(P_1, ..., P_m)$ où les $P_i \in \mathbb{Z}[X_1, ..., X_n]$ sont des polynômes à coefficients entiers relatifs.

Une solution est un n-uplet $(a_1, ...a_n) \in \mathbb{Z}^n$ tel que

pour chaque $1 \leq i \leq m$, on a $P_i(a_1, ..., a_n) = 0$.

Youri Matijasevič (1970) a démontré que ce problème est indécidable.

Le dixième problème de Hilbert, son indécidabilité, Éditions Masson.

Réduction de la positivité des polynômes au 10^{eme} problème de Hilbert

Comme souvent pour démontrer qu'un problème est indécidable

on réduit le problème en question à un problème de référence connu pour être indécidable.

Dans notre cas, le problème de référence est le 10^{eme} problème de Hilbert.

Une suite de réductions

Nous n'allons pas faire une réduction en une seule étape, mais en trois que nous allons appeler:

- solutions dans $\mathbb{N}^n$,
- positivité dans $\mathbb{N}^n$,
- comparaison de polynômes dans $\mathbb{N}^n$.

Première réduction: solutions dans $\mathbb{N}^n$ (1/4)

Dans cette réduction, on montre que l'existence de zéros dans $\mathbb{N}^n$ pour les polynômes à coefficients dans $\mathbb{Z}$ se réduit à l'existence de zéros dans $\mathbb{Z}^n$ pour les polynômes à coefficients dans $\mathbb{Z}$.

J'appelle hyperquadrant de $\mathbb{Z}^n$ un ensemble $\{X_i \mid X_i \%_i 0\}$

où $\%_i$ est soit $>$, soit $<$.

Il y a 2^n hyperquadrants.

Première réduction: solutions dans $\mathbb{N}^n$ (2/4)

On peut transformer le système $(P_1, ..., P_m)$ en 2^n systèmes $(P_{1,k}, ..., P_{m,k})$ pour $1 \leq k \leq 2^n$ associés à chaque hyperquadrant.

Dans chaque polynôme $P(X_1, ..., X_n)$ on transforme

- X_j en X_j si on est dans un hyperquadrant où $X_j > 0$,
- et X_j en $-X_j$ si on est dans un hyperquadrant où $X_j < 0$.

Première réduction: solutions dans $\mathbb{N}^n$ (3/4)

Exemple :

$\{(X_1, X_2, X_3, X_4) \mid X_1 > 0, X_2 < 0, X_3 < 0, X_4 > 0\}$ est un hyperquadrant de $\mathbb{Z}^4$.

Dans ce quadrant on transforme le polynôme $P(X_1, X_2, X_3, X_4)$ en $P(X_1, -X_2, -X_3, X_4)$.

On voit que pour chaque système $(P_{1,k}, ..., P_{m,k})$, on peut chercher chercher les solutions dans $\mathbb{N}^n$ pour trouver les solutions du système $(P_1, ..., P_m)$ dans le $k^{\text{ème}}$ hyperquadrant.

Première réduction: solution dans $\mathbb{N}^n$ (4/4)

Donc si on a un algorithme pour décider si un système de polynômes $(P_1, ..., P_m)$ a une solution dans $\mathbb{N}^n$,

alors on a un algorithme pour décider si un système de polynômes $(P_1, ..., P_m)$ a une solution dans $\mathbb{Z}^n$,

Première réduction: solution dans $\mathbb{N}^n$ (4/4)

Donc si on a un algorithme pour décider si un système de polynômes $(P_1, ..., P_m)$ a une solution dans $\mathbb{N}^n$,

alors on a un algorithme pour décider si un système de polynômes $(P_1, ..., P_m)$ a une solution dans $\mathbb{Z}^n$,

alors on a un algorithme pour résoudre le 10^{eme} problème de Hilbert.

Deuxième réduction: positivité dans $\mathbb{N}^n$

Dans cette réduction, on montre que la positivité dans $\mathbb{N}^n$ pour les polynômes à coefficients dans $\mathbb{Z}$ se réduit à l'existence de zéros dans $\mathbb{N}^n$ pour les polynômes à coefficients dans $\mathbb{Z}$.

Le problème de la positivité des systèmes de polynômes est de savoir si un système de $(Q_1, ..., Q_m)$ où les $Q_i \in \mathbb{Z}[X_1, ..., X_n]$ est strictement positif pour tout $(a_1, ..., a_n) \in \mathbb{N}^n$.

Si l'on a un algorithme pour la positivité des systèmes de polynômes, alors pour chaque système de polynômes $(P_1, ..., P_m)$ où les $P_i \in \mathbb{Z}[X_1, ..., X_n]$, on est capable de répondre si le système $(P_1^2, ..., P_m^2)$ est strictement positif ou non sur $\mathbb{N}^n$, donc de décider s'il a une solution ou non dans $\mathbb{N}^n$.

Troisième réduction: comparaison de polynômes dans $\mathbb{N}^n$ (1/3)

Dans cette réduction, on montre que comparer des couples de polynômes à coefficients dans $\mathbb{N}$ se réduit à la positivité dans $\mathbb{N}^n$ pour les polynômes à coefficients dans $\mathbb{Z}$.

Troisième réduction: comparaison de polynômes dans $\mathbb{N}^n$ (2/3)

Si on a un algorithme pour prouver la satisfaction d'un ensemble

$$(F_i(X_1, ..., X_n) > G_i(X_1, ..., X_n))_{1 \leq i \leq m}$$

- où $F_i \in \mathbb{N}[X_1, ..., X_n]$ et $G_i \in \mathbb{N}[X_1, ..., X_n]$
- et où les X_i parcourent $\mathbb{N}$

alors on a un algorithme pour prouver la positivité des systèmes de $(Q_1, ..., Q_m)$ où les $Q_i \in \mathbb{Z}[X_1, ..., X_n]$.

Troisième réduction: comparaison de polynômes dans $\mathbb{N}^n$ (3/3)

En effet, chaque Q_i peut être décomposé en $Q_i = F_i - G_i$ où $F_i \in \mathbb{N}[X_1, ..., X_n]$ et $G_i \in \mathbb{N}[X_1, ..., X_n]$.

Et alors prouver la positivité de Q_i revient à prouver $(F_i(X_1, ..., X_n) > G_i(X_1, ..., X_n))_{1 \leq i \leq m}$.

Prouver la positivité par ordinateur

Puisque c'est indécidable on peut se ramener à la décidabilité sur les réels qui est décidable, mais inefficace (méthode d'élimination des quantificateurs de Tarski).

On peut utiliser des heuristiques. De toute façon la terminaison est indécidable!

En pratique on cherche des interprétations dans un espace plus petit:

- en limitant le degré,
- et en limitant la taille des coefficients.

Les limites des ordres polynômiaux II

La longueur maximale des réductions d'un système de réécriture dont la terminaison est prouvée par un ordre polynômial est doublement exponentielle.

Si R est un système de réécriture dont la preuve de terminaison est faite par une interprétation polynômial monotone, alors il existe c telle que la longueur de toute dérivation à partir de t soit majorée par $2^{2^{c\|t\|}}$.

Si R est un système de réécriture dont la preuve de terminaison est faite par une interprétation polynomiale monotone,

alors il existe c telle que la longueur de toute dérivation à partir de t soit majorée par $2^{2^{c\|t\|}}$.

On fait l'hypothèse que R n'a qu'un nombre fini de règles (on peut généraliser à un nombre infini de règles).

Soit t un terme et a un entier quelconque pour interpréter les variables de t. Soit π_a un morphisme qui prend la valeur a sur chaque variable.

Soit $t \longrightarrow t_1 \cdots \longrightarrow t_m$ une dérivation partant de t et de longueur m. On a $\pi_a(t) > \pi_a(t_1) > \cdots > \pi_a(t_m)$ donc $\pi_a(t) \geq m$.

Il suffit donc de majorer $\pi_a(t)$.

Soit $c \geq \log_2(k) + \log_2(\log_2(d))$ où pour chaque opérateur f apparaissant dans le système de réécriture

$$P_f(a_1, ..., a_n) \leq d \prod_{1 \leq i \leq n} a_i^k.$$

On procède par récurrence sur la structure de t.

Si t est une constante alors $\pi_a(t) \leq d \leq 2^{2^c}$.

Si $t = f(t_1, ..., t_n)$ alors

$$
\begin{aligned}
\pi_a(f(t_1,...,t_n)) &= P_f(\pi(t_1),...,\pi(t_n)) \\
&\leq d \cdot \prod_{1\leq i\leq n} \pi(t_i)^k \\
&\leq d \cdot \prod_{1\leq i\leq n} 2^{k\cdot 2^{c\cdot \|t_i\|}} = d \cdot 2^{k\cdot \Sigma_{1\leq i\leq n} 2^{c\cdot \|t_i\|}} \\
&\leq d \cdot 2^{k\cdot 2^{c\cdot \Sigma_{1\leq i\leq n}\|t_i\|}} = 2^{\log_2(d)} \cdot 2^{k\cdot 2^{c\cdot \Sigma_{1\leq i\leq n}\|t_i\|}} \\
&= 2^{2^{\log_2(\log_2(d))} + k\cdot 2^{c\cdot \Sigma_{1\leq i\leq n}\|t_i\|}} \\
&= 2^{2^{\log_2(\log_2(d))} + 2^{\log_2(k)+c\cdot \Sigma_{1\leq i\leq n}\|t_i\|}} \\
&\leq 2^{2^{c\cdot(1+\Sigma_{1\leq i\leq n}\|t_i\|)}} \\
&= 2^{2^{c\|t\|}}
\end{aligned}
$$

$\leq$ par définition de d et k, $\leq$ par récurrence, $\leq$ par convexité de $n \mapsto 2^n$.

La borne est atteinte

Exercice : Trouver un système de réécriture tel que la longueur de la plus longue chaîne de dérivation soit doublement exponentielle.

La borne est atteinte

Exercice : Trouver un système de réécriture tel que la longueur de la plus longue chaîne de dérivation soit doublement exponentielle.

Indication : Un système de réécriture

- qui “calcule” quelque chose comme une exponentielle (une puissance fixe d’un entier par exemple, cube, carré, . . .),
- dont la preuve de terminaison est faite par une interprétation polynomiale,
- dans lequel il y un terme court qui se normalise en temps doublement exponentiel.

Le système suivant calcule le carré d'un nombre

$$
\begin{array}{rcl}
0 + m & \longrightarrow & m \\
s(n) + m & \longrightarrow & s(n + m) \\
double(0) & \longrightarrow & 0 \\
double(s(n)) & \longrightarrow & s(s(double(n))) \\
carre(0) & \longrightarrow & 0 \\
carre(s(n)) & \longrightarrow & s(carre(n) + double(n))
\end{array}
$$

est prouvé terminer par un ordre polynomial.

Le système suivant calcule le carré d'un nombre

$$
\begin{aligned}
0 + m &\longrightarrow m \\
s(n) + m &\longrightarrow s(n + m) \\
double(0) &\longrightarrow 0 \\
double(s(n)) &\longrightarrow s(s(double(n))) \\
carre(0) &\longrightarrow 0 \\
carre(s(n)) &\longrightarrow s(carre(n) + double(n))
\end{aligned}
$$

réduit le terme $carre^{n+1}(s(s(0)))$ (de taille $n + 4$) vers $carre(s^{2^{2^n}}(0))$ dont la réduction demande au moins 2^{2^n} appels à la dernière règle du système.

Les limites des ordres polynômiaux III

Si un système de réécriture calcule une fonction entière (avec les deux constructeurs 0 et s) et a une preuve de terminaison fondée sur un ordre polynomiale,

alors cette fonction est à croissance polynomiale.

Exercice : Le système

$$
\begin{aligned}
0 + m &\longrightarrow m \\
s(n) + m &\longrightarrow s(n + m) \\
binom(n, 0) &\longrightarrow 1 \\
binom(0, s(p)) &\longrightarrow 0 \\
binom(s(n), s(p)) &\longrightarrow binom(n, s(p)) + binom(n, p)
\end{aligned}
$$

ne possède pas de preuve de terminaison fondée sur un ordre polynômial.

Comment faut-il faire, si on veut le faire par une algèbre ordonnée?

Les beaux ordres

Les bons ordres

Les bons ordres sont les ordres noethériens totaux.

Les bons ordres sont aussi appelés des ordinaux.

En réécriture, nous sommes intéressés par des ordres partiels.

Ordre noethérien et incrémentalité

Jusqu'à maintenant, on a montré qu'un ordre était noethérien parce qu'il était contenu dans un ordre qui était noethérien.

Mais un informaticien veut construire les ordres noethériens incrémentalement, c'est-à-dire en ajoutant de nouvelles paires à un ordre déjà connu.

C'est exactement ce qui se passe dans une procédure de complétion. On ajoute des paires nouvelles à l'ordre correspondant aux identités que l'on a ne sait pas orienter.

Il faut donc agrandir l'ordre à la volée.

Définition : Un ordre est incrémental si :

- il est noethérien,
- tout ordre qui le contient est noethérien.

Un ordre incrémental doit être “beau” :

il ne doit être ni trop grand, ni trop gros.

Ni trop grand : noethérien,

Ni trop gros : il n'y a pas d'antichaîne infinie.

Une antichaîne est un ensemble tel que $x \leq y \quad \Rightarrow \quad x = y$.

C'est-à-dire que dans une antichaîne, tous les éléments sont tous deux à deux incomparables.

Je rappelle

$\leq$ est un préordre donné

(c'est-à-dire un relation réflexive et transitive).

$x < y \quad \Leftrightarrow \quad x \leq y$ et $\neg(x \geq y)$

(irréflexive, antisymétrique et transitive).

$x \sim y \quad \Leftrightarrow \quad x \leq y$ et $\quad x \geq y$

(relation d'équivalence).

$x \# y \quad \Leftrightarrow \quad \neg(x \leq y)$ et $\quad \neg(x \geq y)$.

Une suite $(x_i)_{i\in\mathbb{N}}$ dans laquelle il existe i et j

tels que $i < j$ et $x_i \leq x_j$ est dite bonne.

Si une suite n'est pas bonne, elle est mauvaise.

Une sous-suite de $(x_i)_{i\in\mathbb{N}}$ est donnée

par une application $\varphi : \mathbb{N} \to \mathbb{N}$ croissante,

autrement dit la sous-suite est celle des $(x_{\varphi(i)})_{i\in\mathbb{N}}$

Les définitions suivantes sont équivalentes :

1. l'ordre est incrémental
2. l'ordre est noethérien et sans antichaîne infinie,
3. toute suite est bonne,
4. de toute suite $(x_i)_{i\in\mathbb{N}}$ on peut extraire une sous-suite croissante.

Les définitions suivantes sont équivalentes :

1. l'ordre est incrémental
2. l'ordre est noethérien et sans antichaîne infinie,
3. toute suite est bonne,
4. de toute suite $(x_i)_{i\in\mathbb{N}}$ on peut extraire une sous-suite croissante.

(1) $\Longrightarrow$ (2) (2) $\Longrightarrow$ (1)

(4) $\Longrightarrow$ (3) (2) $\Longrightarrow$ (3)

(3) $\Longrightarrow$ (2) (3) $\Longrightarrow$ (4)

(2) $\Longrightarrow$ (1)

Supposons que l'ordre est noethérien sans antichaîne infinie.

Soit un ordre $\sqsupset$ qui contient $>$ et une suite infinie $(x_i)_{i\in\mathbb{N}}$ décroissante pour $\sqsupset$.

Dans cette suite il existe une sous-suite $(x_{\varphi(i)})_{i\in\mathbb{N}}$ telle que

$$i > j \Rightarrow x_{\varphi(i)} \sqsupset x_{\varphi(j)} \wedge \neg(x_{\varphi(i)} > x_{\varphi(j)}).$$

Si on ne pouvait pas construire cette sous-suite, cela voudrait dire qu'à tout moment chaque élément serait plus grand pour $>$ que ses suivants.

ce qui serait une contradiction avec le fait que $>$ est noethérien.

$(x_{\varphi(i)})_{i\in\mathbb{N}}$ est un antichaîne infinie pour $>$.

(2) $\Longrightarrow$ (3)

Supposons qu'il n'y ait pas de suite infinie strictement décroissante.

Soit une suite mauvaise.

On considère la suite extraite $(x_{\varphi(i)})_{i\in\mathbb{N}}$ ainsi

- $x_{\varphi(0)}$ est tel que pour $j > \varphi(0)$ on a $x_{\varphi(0)} \# x_j$. C'est possible, car la suite est mauvaise et il n'y a pas de suite infinie strictement décroissante.
- $x_{\varphi(k+1)}$ est tel que $\varphi(k) < \varphi(k+1)$ et pour $j > \varphi(k+1)$ on a $x_{\varphi(k+1)} \# x_j$.

Cette suite extraite $(x_{\varphi(i)})_{i\in\mathbb{N}}$ est une antichaîne infinie.

Contradiction !

(3) $\Longrightarrow$ (4)

Soit une suite qui n'admet pas de sous-suite croissante.

Les sous-ensembles maximaux qui sont ordonnés par indices croissants et sont faiblement croissants pour $\leq$ sont tous finis.

- Ils sont en nombre infini.
- Ils ont tous un dernier élément.

La suite (infinie) $\left(x_{\varphi(i)}\right)_{i\in\mathbb{N}}$ de ces derniers éléments est bonne, donc il existe dans cette suite $i < j$ avec $x_{\varphi(i)} \leq x_{\varphi(j)}$.

En contradiction avec la supposition que $x_{\varphi(i)}$ est un dernier élément !

Beaux ordres

Les ordres qui satisfont les conditions du lemme précédent sont appelés aussi des beaux ordres.

Beaux ordres

Les ordres qui satisfont les conditions du lemme précédent sont appelés aussi des beaux ordres.

Le produit (composante par composante) de beaux ordres est un bel ordre.

Le produit d'ordre $(A_1, \leq_1) \times \cdots \times (A_n, \leq_n)$ est défini par

$$(a_1, ..., a_n) \leq_1 \times ... \times \leq_n (b_1, ..., b_n) \iff \bigwedge_{i=1}^{n} a_i \leq_i b_i$$

Le plongement

Le plongement

Considérons le système de réécriture $\mathcal{EMB}$ qui contient pour chaque $f \in \Sigma_n$ et chaque $1 \leq i \leq n$ une règle,

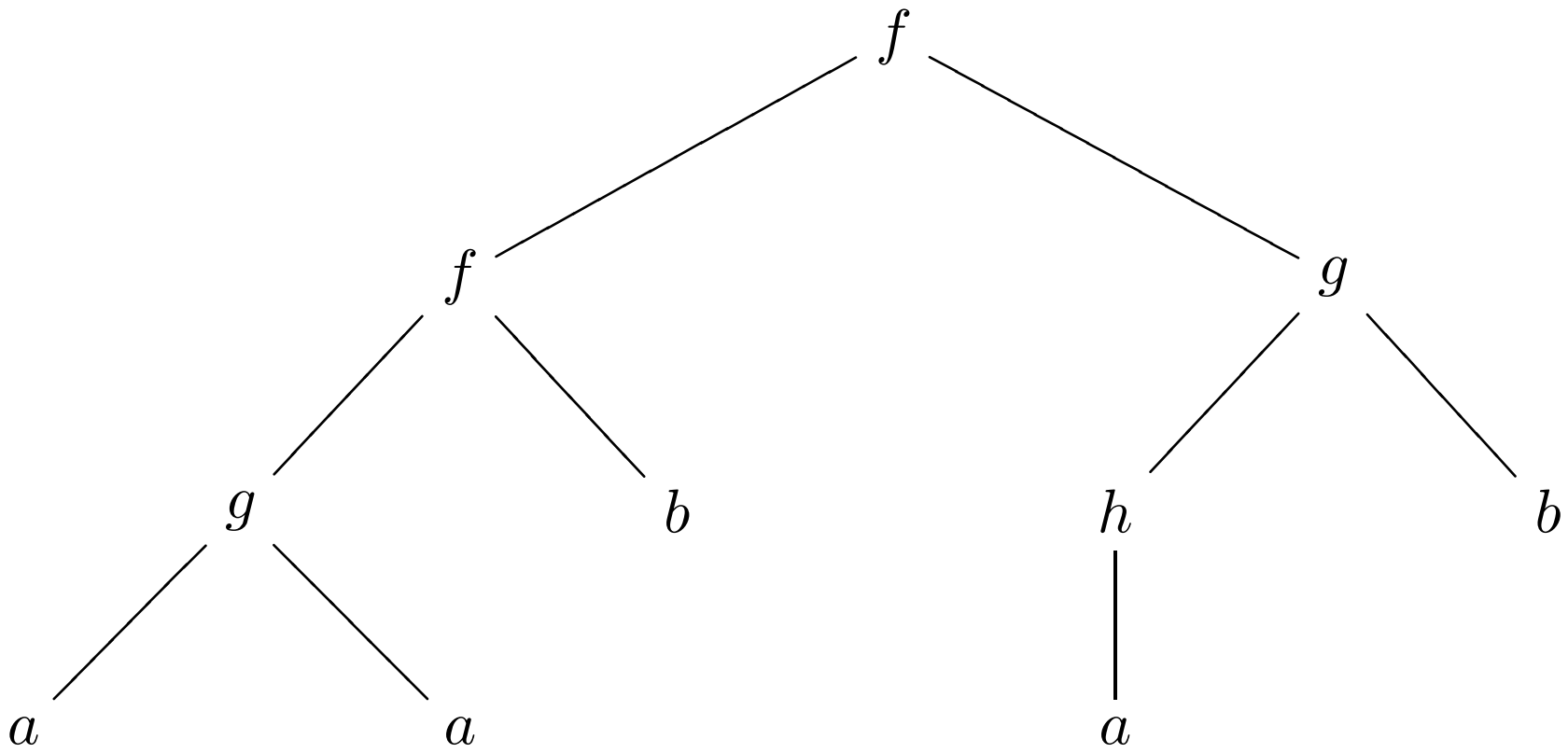
$$f(x_1, ..., x_n) \longrightarrow x_i$$

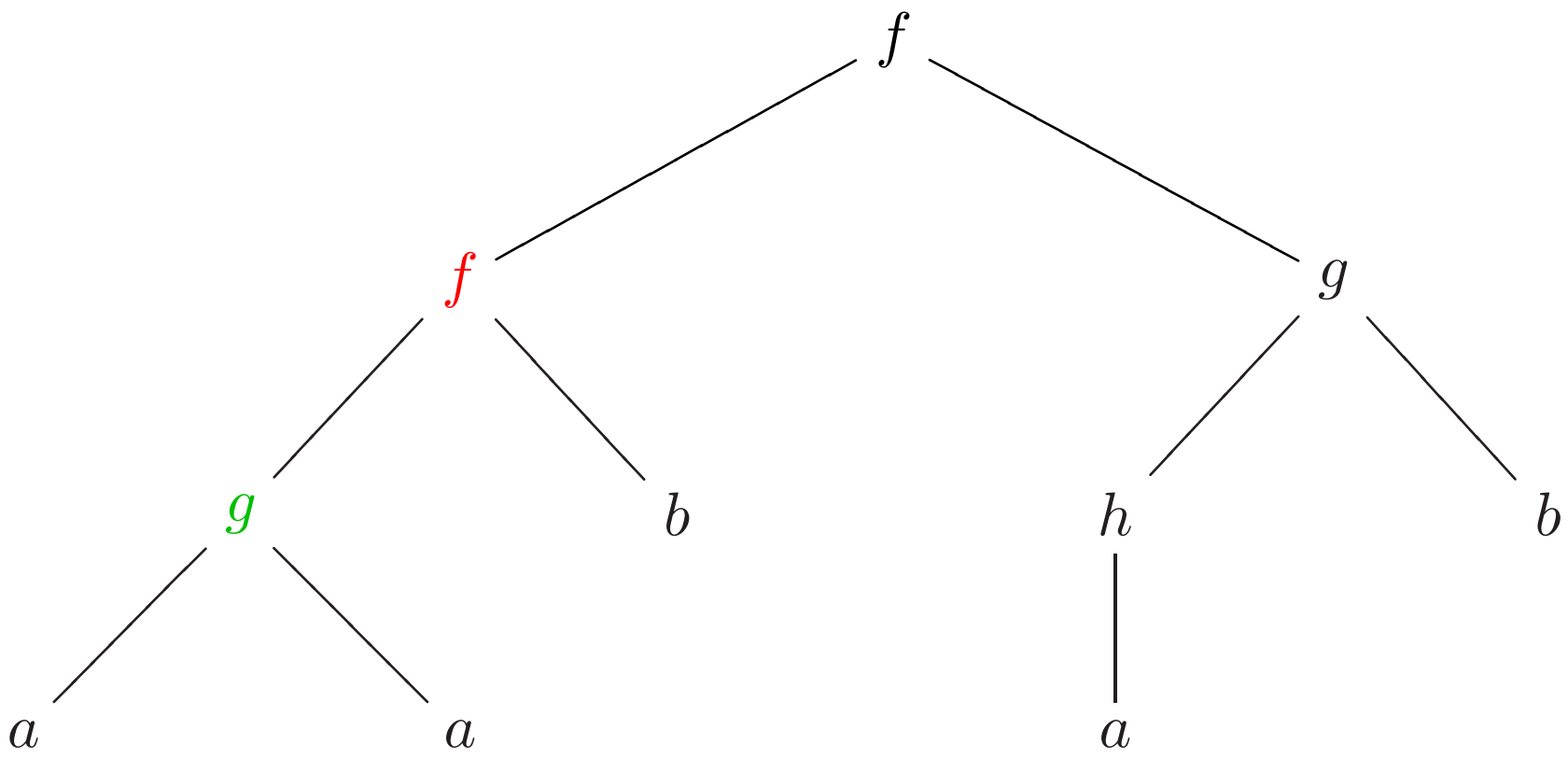
$\xrightarrow[\mathcal{EMB}]{+}$ est sans cycle.

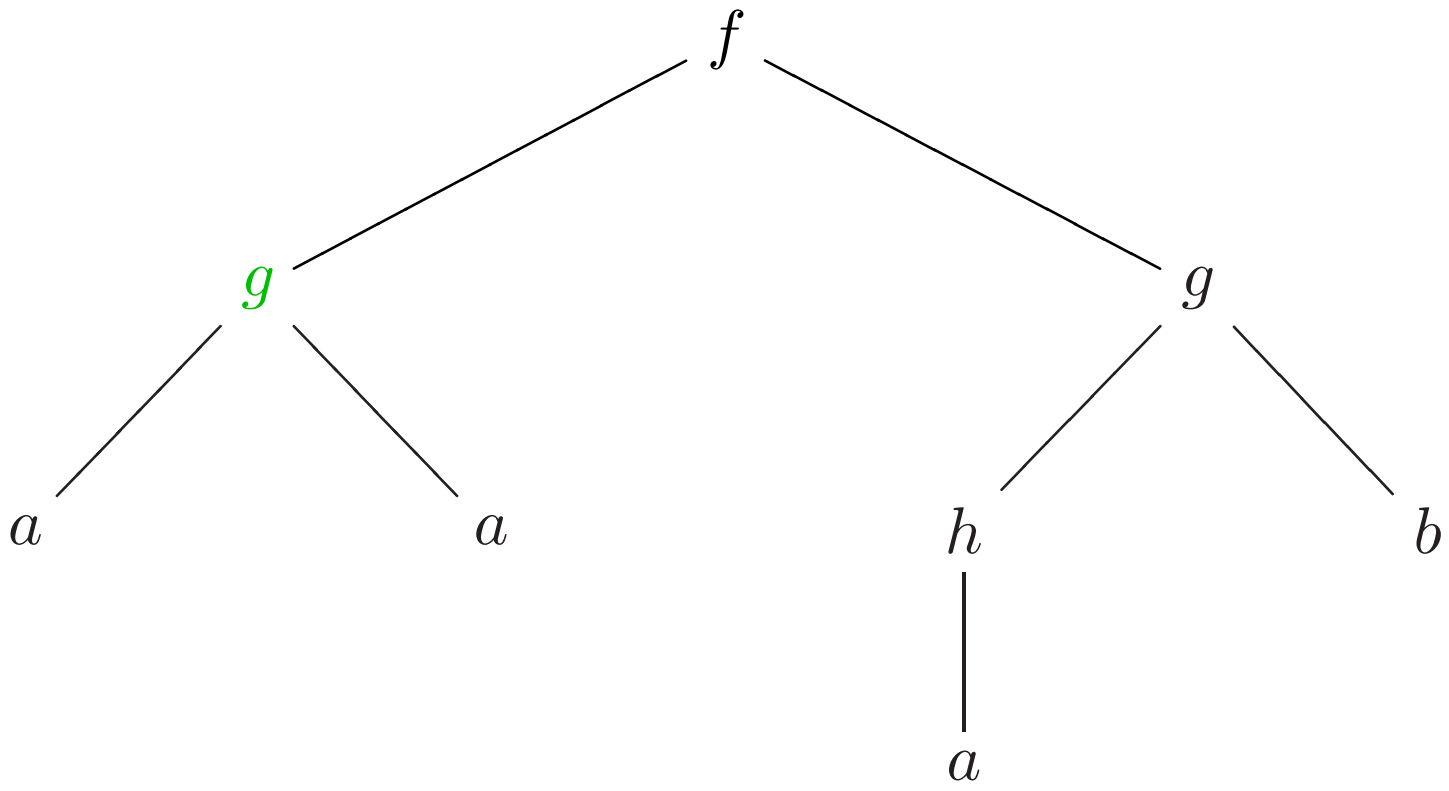
La relation $\xrightarrow[\mathcal{EMB}]{+}$ est un ordre strict noethérien qui s'appelle le plongement.

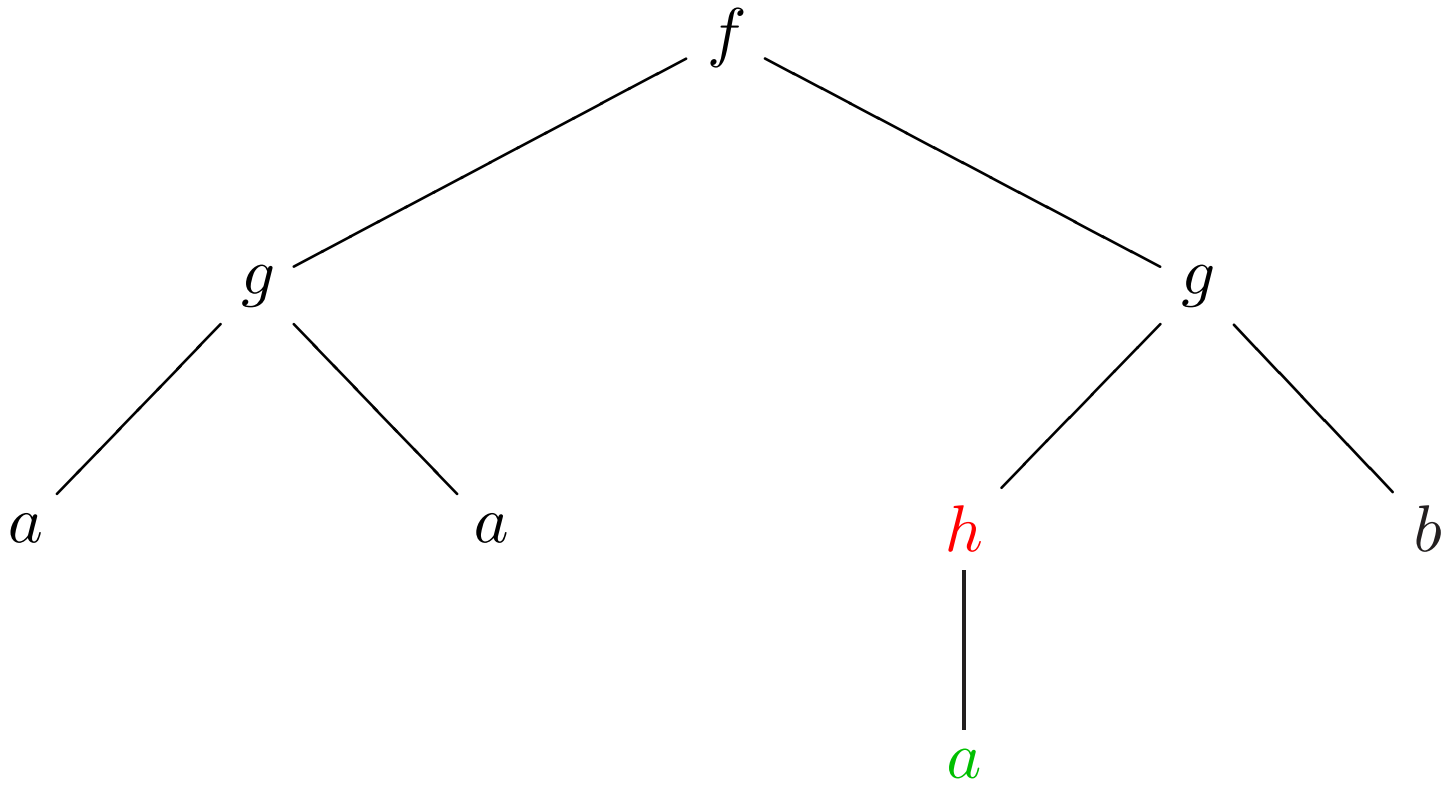
Le plongement

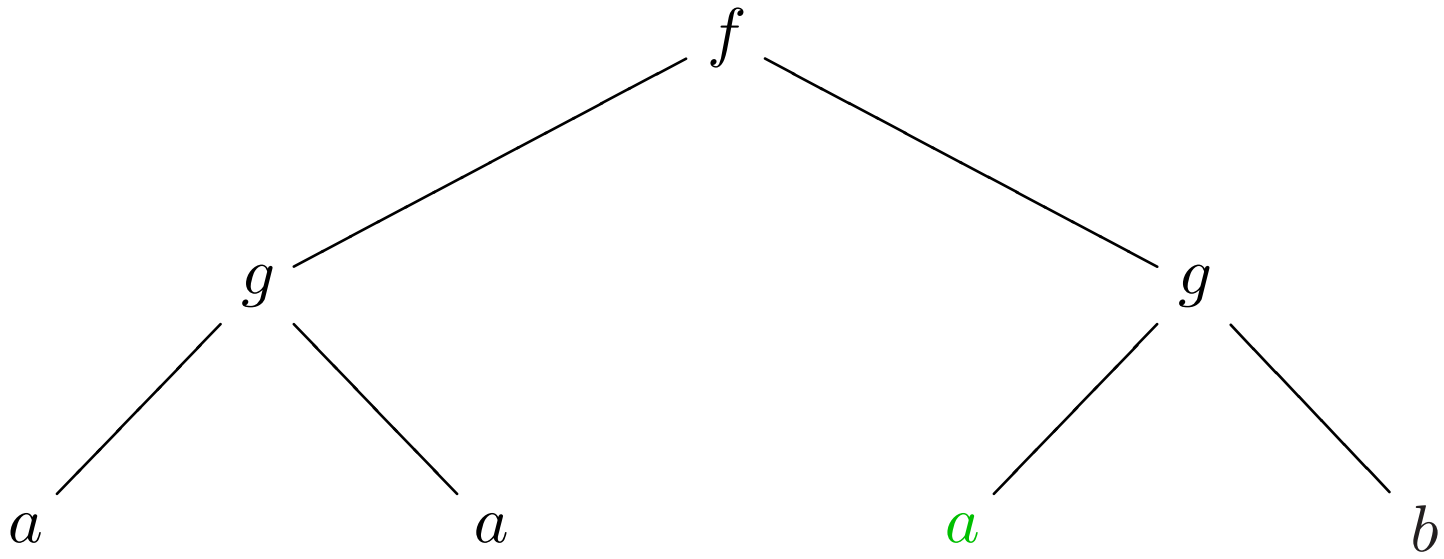
Intuitivement, un terme t est plongé dans un terme t', si $t' \xrightarrow[\mathcal{EMB}]{+} t$, c'est-à-dire si on passe de t' à t en effaçant des nœuds et en recollant un terme fils du nœud manquant à la place laissée libre. le plongement.


f
f
g
g
b
h
b
a
a
a


f
f
g
g
b
h
b
a
a
a


f
g
g
a
a
h
b
a


f
g
g
a
a
h
b
a


f
g
g
a
a
a
b

Le plongement dans A^*

Les termes de A^* sont les termes monadiques.

Le plongement est défini par

- $(\forall \alpha \in A^+) \quad \alpha \xrightarrow[\mathcal{EMB}]{+} \varepsilon,$
- $(\forall \alpha \in A^*)(\forall \beta \in A^*)(\forall a \in A)\ \alpha \xrightarrow[\mathcal{EMB}]{+} \beta \Rightarrow a\alpha \xrightarrow[\mathcal{EMB}]{+} a\beta$
- $(\forall \alpha \in A^*)(\forall \beta \in A^*)(\forall a \in A)\ \alpha \xrightarrow[\mathcal{EMB}]{*} \beta \Rightarrow a\alpha \xrightarrow[\mathcal{EMB}]{+} \beta$

Le Théorème de Higman

Théorème de Higman

Si A est fini, le plongement est un bel ordre sur A^*.

Théorème de Higman

Si A est fini, le plongement est un bel ordre sur A^*.

Supposons que $A^*, \xrightarrow[\mathcal{EMB}]{+}$ n'est pas un bel ordre.

Soit une suite $(\alpha_i)_{i\in\mathbb{N}}$ mauvaise et minimale définie de la façon suivante.

1. α_0 est un des plus petits éléments qui commence une mauvaise suite.
2. α_i est le plus petit élément en $i$ème position d'une mauvaise suite qui commence par $\alpha_0, \ldots \alpha_{i-1}$.

Cette suite ne contient aucun ε. Donc tous les termes sont de la forme $a_i \alpha'_i$.

On peut extraire une sous-suite $a\alpha'_{\varphi(i)}$, pour le même a (parce que A est fini).

La suite $\alpha_0, \ldots \alpha_{\varphi(0)-1}, \alpha'_{\varphi(0)}, \ldots, \alpha'_{\varphi(j)} \ldots$ est bonne.

Donc

1. soit $\alpha_k \xleftarrow[\mathcal{EMB}]{*} \alpha'_{\varphi(j)}$ pour $0 \leq k < \varphi(0)$ et $0 \leq j$,

   donc $\alpha_k \xleftarrow[\mathcal{EMB}]{*} a\alpha'_{\varphi(j)}$, contradiction!

2. soit $\alpha'_{\varphi(j)} \xleftarrow[\mathcal{EMB}]{*} \alpha'_{\varphi(j')}$ pour $0 \leq j < j'$,

   donc $a\alpha'_{\varphi(j)} \xleftarrow[\mathcal{EMB}]{*} a\alpha'_{\varphi(j')}$, contradiction!

Théorème de Higman généralisé

On a besoin du théorème de Higman sur un alphabet infini dénombrable.

Comment le généraliser?

Théorème de Higman généralisé

On a besoin du théorème de Higman sur un alphabet infini dénombrable.

Comment le généraliser?

- Il faut ordonner A par un bel ordre,
- Il faut généraliser l'énoncé.

Plongement avec restriction sur A^*

- $(\forall \alpha \in A^+) \quad \alpha \xrightarrow[\mathcal{EMB}]{+} \varepsilon,$
- $(\forall \alpha \in A^*)(\forall \beta \in A^*)(\forall a \in A) \qquad a \geq b \;\&\; \alpha \xrightarrow[\mathcal{EMB}]{+} \beta \;\Rightarrow\; a\alpha \xrightarrow[\mathcal{EMB}]{+} b\beta$
- $(\forall \alpha \in A^*)(\forall \beta \in A^*)(\forall a \in A)\ \alpha \xrightarrow[\mathcal{EMB}]{*} \beta \;\Rightarrow\; a\alpha \xrightarrow[\mathcal{EMB}]{+} \beta$

Théorème de Higman généralisé

Si $\geq$ est un bel ordre sur A

le plongement est un bel ordre sur A^*.

Là où on a dit

"On peut extraire une sous-suite $a\alpha'_{\varphi(i)}$, pour le même a.",

on dit

"On peut extraire une sous-suite $a_{\varphi(i)}\alpha'_{\varphi(i)}$

telle que la suite $a_{\varphi(i)}$ soit croissante."

Le reste de la démonstration est identique.

Le théorème de Kruskal

Théorème de Kruskal

Si Σ est fini, le plongement est un bel ordre sur $T(\Sigma, X)$.

Théorème de Kruskal

Si Σ est fini, le plongement est un bel ordre sur $T(\Sigma, X)$.

Supposons que le plongement n'est pas un bel ordre.

Considérons une mauvaise suite minimale, construite comme dans le théorème de Higman. (Exercice : refaire la construction.)

De cette suite on peut extraire une sous-suite de la forme $(f(s_1^i, ..., s_n^i))_{i \geq 0}$.

Les n suites $(s_j^i)_{i\geq 0}$ ainsi que leurs sous-suites sont bonnes.

De $(s_1^i)_{i\geq 0}$ on peut extraire une sous-suite croissante $(s_1^{\varphi_1(i)})_{i\geq 0}$.

De $(s_2^{\varphi_1(i)})_{i\geq 0}$ on peut extraire une sous-suite croissante $(s_2^{\varphi_2\varphi_1(i)})_{i\geq 0}$.

$\vdots$

De $(s_n^{\varphi_{n-1}\cdots\varphi_1(i)})_{i\geq 0}$ on peut extraire une sous-suite croissante $(s_n^{\varphi_n\cdots\varphi_1(i)})_{i\geq 0}$.

Clairement la suite $(f(s_1^{\varphi_n\cdots\varphi_1(i)},\ldots,s_n^{\varphi_n\cdots\varphi_1(i)}))_{i\geq 0}$ est croissante. Contradiction !

Plongement avec restriction sur $T(\Sigma, X)$

Un ordre sur Σ est appelé une précédence.

On se donne

- une précédence $\geq$
- et un système de réécriture appelé restriction

  tel que pour chaque f et chaque g avec $f \geq g$

  et chaque permutation $\pi : [1..n] \longrightarrow [1..n]$

$$f(x_1, ..., x_n) \longrightarrow g(x_{\pi(1)}, ..., x_{\pi(p)})$$

Plongement avec restriction sur $T(\Sigma, X)$

Le plongement avec restriction est défini par les règles de $\mathcal{EMB}$ (effacement) et les règles de restriction.

Théorème de Kruskal

Si $(\Sigma, \leq)$ est un bel ordre, le plongement avec restriction est un bel ordre sur $T(\Sigma, X)$.

Les ordres de simplification

Ordre de simplification

Un ordre de simplification est un ordre de réécriture $>$ qui satisfait la propriété de sous-terme, c'est-à-dire :

$$(\forall p \in Pos(t)) \qquad t > t_{|p}$$

Ordre de simplification

Un ordre de simplification est un ordre de réécriture $>$ qui satisfait la propriété de sous-terme, c'est-à-dire :

$$(\forall p \in Pos(t)) \qquad t > t_{|p}$$

Combiné avec la comptabilité, les ordres de simplification contiennent le plongement. Donc

les ordres de simplification sont des beaux ordres,

les ordres de simplification terminent,

les ordres de simplification sont des ordres de réduction.

L'ordre lexicographique sur les chemins

Supposons donné un préordre $\leq$ sur Σ appelé précédence.

$s <_{lpo} t$ ssi

(A) s est une variable x, t n'est pas une variable et $x \in Var(t)$

ou

(B1) $s \equiv f(s_1, ..., s_m)$ et $t \equiv g(t_1, ..., t_n)$

et $\forall i \in [1..m]\ s_i <_{lpo} t$

(B2)

(B21) $f < g$ ou

(B22) $f \sim g$ et $(s_1, ..., s_m) <_{lpo}^{lex} (t_1, ..., t_n)$ ou

(B23) $\exists j \in [1..n] s \leq_{lpo} t_j$

L'ordre lexicographique sur les chemins

$<_{lpo}$ est un ordre.

Antisymétrique

Transitivité.

Simplification.

L'ordre lexicographique sur les chemins

$<_{lpo}$ est un ordre de simplification.

$<_{lpo}$ est compatible et clos par substitution et satisfait la propriété de sous-terme.

Donc $<_{lpo}$ est un ordre de réduction.

Exercice : Fonction d'Ackermann-Peter

Prouver la terminaison de la fonction d'Ackermann-Peter

$$
\begin{aligned}
A(0, n) &\longrightarrow s(n) \\
A(s(m), 0) &\longrightarrow A(m, s(0)) \\
A(s(m), s(n)) &\longrightarrow A(m, A(s(m), n))
\end{aligned}
$$

Exercice : (E) + (A)

Premier cas :

$$(x * y) * z \longrightarrow x * (y * z) \qquad \text{(A)}$$
$$f(x * y) \longrightarrow f(x) * f(y) \qquad \text{(E)}$$

Deuxième cas :

$$(x * y) * z \longrightarrow x * (y * z) \qquad \text{(A)}$$
$$f(x) * f(y) \longrightarrow f(x * y) \qquad \text{(E)}$$
$$f(x * y) * z \longrightarrow f(x) * (f(y) * z) \qquad \text{(EA)}$$

Exercice : (E) + (A)

Troisième cas :

$$
\begin{aligned}
(x * y) * z &\longrightarrow x * (y * z) && \text{(A)} \\
f(x) * f(y) &\longrightarrow f(x * y) && \text{(E)} \\
f(x) * (f(y) * z) &\longrightarrow f(x * y) * z && \text{(EA)}
\end{aligned}
$$

Exercice : les groupes

$$
\begin{array}{rcl}
x * e & \longrightarrow & x \\
e * x & \longrightarrow & x \\
x * i(x) & \longrightarrow & e \\
i(x) * x & \longrightarrow & e \\
i(e) & \longrightarrow & e \\
i(i(x)) & \longrightarrow & x \\
i(x * y) & \longrightarrow & i(y) * i(x) \\
(x * y) * z & \longrightarrow & x * (y * z) \\
x * (i(x) * y) & \longrightarrow & y \\
i(x) * (x * y) & \longrightarrow & y
\end{array}
$$

Le statut des opérateurs

On peut choisir de prendre l'ordre lexicographique

de gauche à droite,

ou de droite à gauche.

Exercice : les groupes (variante avec division)

$$
\begin{array}{rcl}
i(e) & \longrightarrow & e \\
x/e & \longrightarrow & x \\
e/x & \longrightarrow & i(x) \\
x/x & \longrightarrow & e \\
i(x/y) & \longrightarrow & y/x \\
i(i(x)) & \longrightarrow & x \\
x/(y/z) & \longrightarrow & (x/i(z))/y \\
(x/i(y))/y & \longrightarrow & x \\
(x/y)/i(y) & \longrightarrow & x
\end{array}
$$

Exercice : les groupes (variante avec division)

$$
\begin{array}{ccc}
i(e) & \longrightarrow & e \\
x/e & \longrightarrow & x \\
e/x & \longrightarrow & i(x) \\
x/x & \longrightarrow & e \\
i(x/y) & \longrightarrow & y/x \\
i(i(x)) & \longrightarrow & x \\
x/(y/z) & \longrightarrow & (x/i(z))/y \\
(x/i(y))/y & \longrightarrow & x \\
(x/y)/i(y) & \longrightarrow & x
\end{array}
\qquad
x * y \longrightarrow x/i(y)
$$

Exercice d'école

$$f(f(x)) \quad \longrightarrow \quad f(g(f(x)))$$

Exercice : une définition tarabiscotée de l'identité

$$
\begin{aligned}
i(s(x)) &\longrightarrow s(i(p(s(x))))\\
i(0) &\longrightarrow 0\\
p(s(0)) &\longrightarrow 0\\
p(s(s(x)) &\longrightarrow s(x)
\end{aligned}
$$

Exercice : pgcd

$$
\begin{array}{rcl}
if(s(m), s(n), p, q) & \rightarrow & if(m, n, p, q) \\
if(s(m), 0, p, q) & \rightarrow & p \\
if(0, n, p, q) & \rightarrow & q \\
 & & \\
s(m) - s(n) & \rightarrow & m - n \\
s(m) - 0 & \rightarrow & m \\
0 - s(n) & \rightarrow & 0 \\
 & & \\
pgcd(m, 0) & \rightarrow & m \\
pgcd(0, m) & \rightarrow & m \\
pgcd(s(m), s(n)) & \rightarrow & if(m, n, pgcd(m - n, s(n)), pgcd(s(m), n - m))
\end{array}
$$