920 : Réécriture et formes normales. Exemples

[Exemp.]

Motivations: A partir de structures définies par des égalités, on souhaite pouvoir déduire d'autres égalités.

Exemple: Les entiers et l'addition : $x + 0 = x$
$x + s(y) = s(x+y)$

$s(0) + s(s(0)) = s(s(0) + s(0)) = s(s(s(0)) + 0) = s(s(s(0)))$
on a montré que $1 + 2 = 3$

Application : en programmation fonctionnelle

[3 ()]

I Termes et réécriture

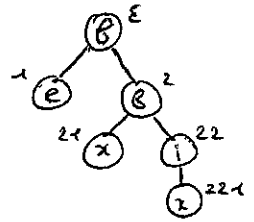
1) Termes

Def 1: Une signature Σ est un ensemble de fonctions $\Sigma = \bigcup_{n \geq 0} \Sigma^{(n)}$ avec $\Sigma^{(n)}$ ensemble de fonctions d'arité n. $\Sigma^{(0)}$ est l'ensemble des symboles de constantes.

Def 2: Soit Σ une signature et X ensemble de variables. L'ensemble des termes $T(\Sigma, X)$ est défini par induction par :
- $X \subseteq T(\Sigma, X)$
- Si $n \geq 0$, $f \in \Sigma^{(n)}$, $t_1, \dots, t_n \in T(\Sigma, X)$ alors $f(t_1, \dots, t_n) \in T(\Sigma, X)$

Def 3:
- L'ensemble des positions d'un terme s est défini inductivement par :
  - $s = x$, $Pos(s) = \{\varepsilon\}$
  - $s = f(s_1, \dots, s_n)$, $Pos(s) = \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{ip / p \in Pos(s_i)\}$
- la taille de s est $|s| = \# Pos(s)$
- $s_{|p}$ le sous-terme de s à la position p
- $s[t]_p$ le terme obtenu en remplaçant $s_{|p}$ par t
- Une substitution σ : $V \to T(\Sigma, X)$ est une fonction telle que $\sigma(x) \neq x$ pour un nombre fini de X. On peut étendre σ sur les termes par induction.

Ex 4:



représente le terme $f(e, f(x, i(x)))$ et $s_{|2} = f(x, i(x))$

2) Relation de réécriture

Def 5:
- Une Σ-égalité est une paire $(s, t) \in T(\Sigma, X)^2$ notée $s \approx t$
- Une théorie équationnelle est un ensemble de Σ-égalité.

Ex 6: $E_1 = \{ f(x, f(y,z)) \approx f(f(x,y), z),\ f(e, x) \approx x,\ f(i(x), x) \approx e \}$

Def 7: Soit E une théorie équationnelle.
$s \to_E t$ si $\exists (l, r) \in E$, $p \in Pos(s)$, σ une substitution tel que $s_{|p} = \sigma(l)$ et $t = s[\sigma(r)]_p$

Exemple 8 :
$f(i(e), f(e,e)) \to_{E_1} f(f(i(e), e), e) \to_{E_1} f(e, e) \to_{E_1} e$

Lemme 9 : $\to_E$ est close par substitution et compatible avec les Σ-opérations, i.e :

- $s \to_E t \Rightarrow \sigma(s) \to \sigma(t)$
- $s \to_E t \Rightarrow f(s_1, \dots, s_{i-1}, s, s_{i+1}, \dots, s_n) \to_E f(s_1, \dots, s_{i-1}, t, s_{i+1}, \dots, s_n)$

Notation 10 : On note $\overset{*}{\longleftrightarrow}_E$ la clôture réflexive transitive de $\to_E$.

[2. 1] **II Les systèmes de réécriture**

1) Définition

Def 11 : Un système de réécriture est une théorie équationnelle E finie telle que $\forall (e,r) \in E$ alors $Var(e) \supseteq Var(r)$ muni de la relation de réécriture.

Def 12 :
- s est réductible si $\exists t$ tel que $s \to_E t$
- s est une forme norme si il n'est pas réductible
- t est une forme normal de s si $s \overset{*}{\to}_E t$
- s et t sont joignables si $\exists v$ tel que $s \overset{*}{\to}_E v$, $t \overset{*}{\to}_E v$

Dans ce cas on écrit $s \downarrow t$

Ex 13 : $\Sigma = \Sigma^{(0)} = \mathbb{N} \setminus \{0,1\}$

$E = \{(m,n) \,/\, m > n$ et n divise $m\}$

alors m est une forme normale si et seulement si m est premier
$m \downarrow n$ si et seulement si m et n ne sont pas premiers entre eux

Définition 14 : $\to_E$ est
- de Church-Rosser si $s \overset{*}{\longleftrightarrow}_E t \Rightarrow s \downarrow t$

$$\begin{tikzcd} s \arrow[rr, leftrightarrow, "*"] \arrow[dr, dashed, "*"'] & & t \arrow[dl, dashed, "*"] \\ & v & \end{tikzcd}$$

- confluente si $s_1 \overset{*}{\underset{E}{\leftarrow}} t \overset{*}{\underset{E}{\to}} s_2 \Rightarrow s_1 \downarrow s_2$

$$\begin{tikzcd} t \arrow[r, "*"] \arrow[d, "*"'] & s_1 \arrow[d, dashed, "*"] \\ s_2 \arrow[r, dashed, "*"'] & v \end{tikzcd}$$

- noethérienne ou terminante si il n'y a pas de suite infinie $s_0 \to_E s_1 \to_E \dots$
- normalisante si chaque élément a une forme normale
- convergente si elle est confluente et noethérienne

2) Propriétés

Th 15 : Church-Rosser $\Leftrightarrow$ confluence

Cor 16 : Si $\to_E$ est confluente et que $s \overset{*}{\longleftrightarrow}_E t$ alors :
(i) $s \overset{*}{\to}_E t$ si t est une forme normale
(ii) $s = t$ si s et t sont des formes normales

Lemme 17 : Si $\to_E$ est noethérienne et confluente alors chaque élément admet une unique forme normale.

Th 18 : Si $\to_E$ est confluente et noethérienne alors $s \overset{*}{\longleftrightarrow}_E t \Leftrightarrow x$ et y ont même forme normale

Procédure de décision :

Si $\to_E$ est un système de réécriture terminant et confluent, on peut décider de la validité sémantique de $s \overset{*}{\longleftrightarrow}_E t$ en testant l'égalité des formes normales

III Prouver la terminaison et la confluence [5 et 6]

1) Prouver la terminaison

Th 19 : Les problèmes suivants sont indécidables

① Instance : un système de réécriture $\to_E$ et un terme t
Problème : Est-ce que toutes les réductions commençant par t terminent ?

② **Instance :** Un système de réécriture $\to_E$

Problème : Est-ce que $\to_E$ termine

Def 20 : Un ordre strict $>$ sur $T(\Sigma, X)$ est de réécriture si
- $>$ est compatible avec les opérations et closes par substitution.

Un ordre strict $>$ sur $T(\Sigma, X)$ est de réduction s'il est de réécriture et bien fondé, ie il n'existe pas de suites infinies décroissantes

Ex 21 : • $s > t$ si $|s| > |t|$ est un ordre strict qui n'est pas clos par substitution
• $s > t$ si $|s| > |t|$ et $\forall x \in X$, $|s|_x \geq |t|_x$ est un ordre de réduction

Th 22 : Un système de réécriture $\to_E$ est terminant si et seulement si il existe un ordre de réduction $>$ tel que $l > r$ pour tout $(l, r) \in E$.

2) Prouver la confluence

Th 23 : Le problème de savoir si $\to_E$ est confluent est indécidable.

[2] ⓐ **Confluence locale**

Def 24 : $\to_E$ est localement confluent si $s_1 \leftarrow_E t \to_E s_2 \Rightarrow s_1 \downarrow s_2$

$$\begin{tikzcd} t \arrow[r, "E"] \arrow[d, "E"'] & s_1 \arrow[d, dashed, "*"', "E"] \\ s_2 \arrow[r, dashed, "*", "E"'] & t' \end{tikzcd}$$

Th 25 : Si $\to_E$ est noethérien alors

[DEV 1] $\to_E$ est confluente $\Leftrightarrow$ $\to_E$ est localement confluente

ⓑ **Paires critiques**

Def 26 : Soit $l_1 \to r_1$ et $l_2 \to r_2$ deux règles de E sans variables communes. Soit $p \in Pos(l_1)$ tel que $l_1|_p \notin X$ et soit θ un unificateur principal de $l_1|_p$ et l_2.

Si
$$\begin{tikzcd} & \theta l_1 \arrow[dl] \arrow[dr] & \\ t_1 = \theta r_1 & & t_2 = (\theta l_1)[\theta r_2]_p \end{tikzcd}$$
alors (t_1, t_2) est une paire critique

Ex 27 :
$$\begin{tikzcd} & f(f(i(x_1), x_1), z) \arrow[dl] \arrow[dr] & \\ f(i(x_1), f(x_1, z)) & & f(e, z) \end{tikzcd}$$

$(f(i(x_1), f(x_1, z)), f(e, z))$ est une paire critique avec les règles

(1) $f(f(x, y), z) \to f(x, f(y, z))$ (2) $f(i(x_1), x_1) \to e$

Th 28 : (des paires critiques) **[DEV 2]**
Un système de réécriture est localement confluent si et seulement si ses paires critiques sont joignables

Cor 29 : Un système de réécriture terminant est confluent si et seulement si ses paires critiques sont joignables.

Cor 30 : La confluence d'un système de réécriture terminant est décidable

Algorithme de Knuth-Bendix :

L'algorithme prend en entrée une théorie équationnelle E et un ordre de réduction sur $T(\Sigma, X)$ et tente de trouver un système de réécriture R équivalent à E, ie $s \approx_E t$ si et seulement si $s \approx_R t$.

La terminaison de cet algorithme n'est pas assurée

LEMME DE NEWMAN

Lemme Soit $(E, \rightarrow_E)$ un système de réécriture.
Si $\rightarrow_E$ est noethérienne alors
$\rightarrow_E$ est confluente (ssi) $\rightarrow_E$ est localement confluente

Demonstration ($\Rightarrow$) Trivial : Une relation confluente est toujours localement confluente
($\Leftarrow$) Supposons $\rightarrow$ noethérienne et localement confluente.
Montrons que $\rightarrow$ est confluente par le principe d'induction bien fondée appliquée à la propriété :

$$P(x) = \text{"} \forall y, z, \quad y \overset{*}{\longleftarrow} x \overset{*}{\longrightarrow} z \implies y \downarrow z \text{"}$$

RAPPEL : règle d'inférence de l'induction bien fondée

$$\begin{array}{ll} B \left\{ \right. & \dfrac{\forall x \in A (\forall y \in A,\ x \overset{+}{\longrightarrow} y \Rightarrow P(y)) \Rightarrow P(x)}{\forall x \in A,\ P(x)} \\ C \left\{ \right. & \end{array}$$

Soit $x \in A$. Supposons que P est vérifiée pour tout successeur strict de x.

- Si $x = y$ (ou $x = z$ de la même manière)
  alors $y \overset{*}{\longrightarrow} z \overset{\circ}{\longleftarrow} z$
- Si $x \neq y$ et $x \neq z$, alors $\exists y_1, z_1$ tq $y \overset{*}{\longleftarrow} y_1 \longleftarrow x \longrightarrow z_1 \overset{*}{\longrightarrow} z$

$$\begin{tikzcd}
x \arrow[r] \arrow[d] \arrow[dr, phantom, "L.C"] & z_1 \arrow[r, "*"] \arrow[d, dashed, "*"] \arrow[ddr, phantom, "H.I"] & z \arrow[dd, dashed, "*"] \\
y_1 \arrow[r, dashed, "*"] \arrow[d, "*"] \arrow[dr, phantom, "H.I"] & u \arrow[d, dashed, "*"] & \\
y \arrow[r, dashed, "*"] & v \arrow[r, dashed] & w
\end{tikzcd}$$

- existence de u par hypothèse de locale confluence
- existence de v par hypothèse d'induction
- existence de w par hypothèse d'induction

Ainsi on a montré que $P(x)$ est vraie.

Par le principe d'induction bien fondée, on a prouvé que $\forall x \in A, P(x)$ est vraie.
C'est à dire $\rightarrow$ est confluente

QUESTION : Où le caractère noethérien intervient-il ?

CONTREX $\quad a \leftarrow 0 \rightleftarrows 1 \rightarrow b \quad$ relation localement confluente mais non noethérienne

Elle n'est pas confluente : $a \leftarrow 0 \xrightarrow{*} b$ et a et b ne sont pas joignables

$\Rightarrow$ Le caractère noethérien est nécessaire

Théorème. $\rightarrow$ est noethérienne ssi le principe d'induction bien fondée est satisfait pour $\rightarrow$

[2.2] p13-14

Démonstration $\Rightarrow$ Par contraposée,

Supposons que le principe d'induction bien fondée n'est pas satisfait pour $\rightarrow$

Alors il existe une propriété P tq (B $\Rightarrow$ non C)

et non C = $\exists a_0 \in A$ tq $\neg P(a_0)$

La prémisse B affirme que si P est vérifiée pour tous les successeurs stricts de a_0, alors elle l'est pour a_0. Comme $\neg P(a_0)$, $\exists a_1 \xleftarrow{+} a_0$ tq $\neg P(a_1)$

En itérant ce procédé, on construit une chaîne infinie $a_0 \xrightarrow{+} a_1 \xrightarrow{+} \dots \xrightarrow{+} a_n \xrightarrow{+} \dots$

C'est à dire la relation $\rightarrow$ n'est pas terminante

$\Leftarrow$ Supposons que le PIBF est satisfait pour $\rightarrow$

Notons $P(x)$ = "toute chaîne commençant par x termine"

B est alors trivialement vérifiée : si pour tout successeur strict y de x, toute chaîne commençant par y termine, alors toute chaîne commençant par x est terminante.

Ainsi $\forall x, P(x)$. C'est à dire $\rightarrow$ est noethérienne.

LE THEOREME DES PAIRES CRITIQUES

Théorème Un système de réécriture est localement confluent ssi toutes ses paires critiques sont joignables.

Démonstration $(\Rightarrow)$ Si le système est localement confluent, toute paire critique est joignable car issue de dérivations de la forme

$$\begin{array}{ccc} & \theta l_1 & \\ \swarrow & & \searrow \\ \theta r_1 & & (\theta l_1)[\theta r_2]_p \end{array}$$

$(\Leftarrow)$ Supposons maintenant que les paires critiques sont joignables

Soit s un terme tq

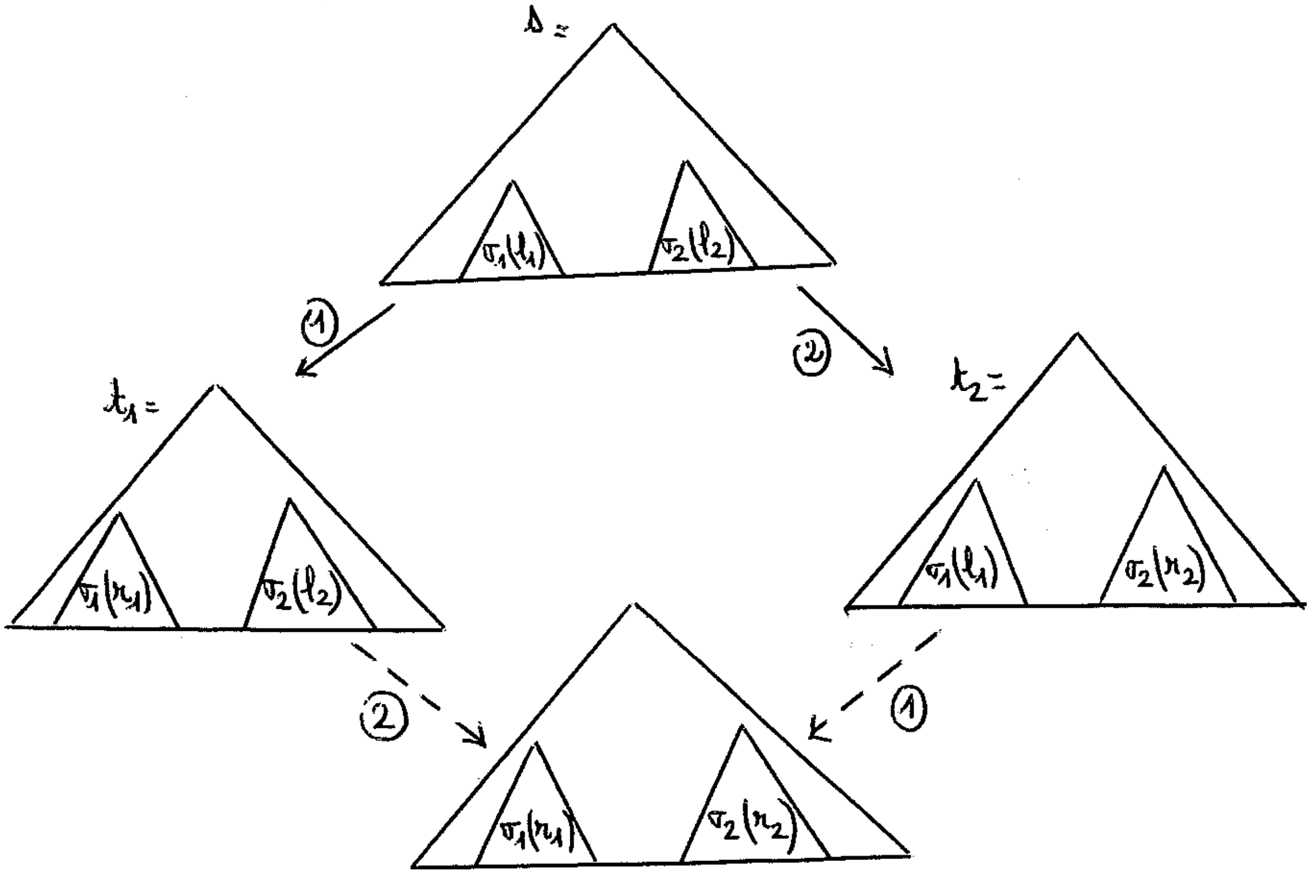
$$\begin{array}{ccc} & s & \\ {\scriptstyle 1}\swarrow & & \searrow{\scriptstyle 2} \\ t_1 & & t_2 \end{array} \quad \text{avec} \begin{cases} l_1 \xrightarrow{1} r_1 \\ l_2 \xrightarrow{2} r_2 \end{cases}$$

il faut mq t_1 et t_2 sont joignables.

ie $\exists p_1 \in Pos(s)$ tq $\begin{cases} s|_{p_1} = \sigma_1(l_1) \\ t_1 = s[\sigma_1(r_1)]_{p_1} \end{cases}$ $\exists p_2 \in Pos(s)$ tq $\begin{cases} s|_{p_2} = \sigma_2(l_2) \\ t_2 = s[\sigma_2(r_2)]_{p_2} \end{cases}$

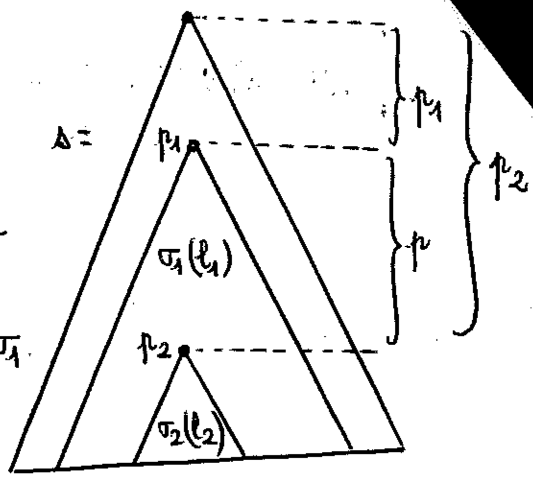
La suite de la démonstration dépend des positions de dérivation p_1 et p_2 :

1er Cas p_1 et p_2 sont des sous-arbres de s séparés



2ème Cas Si p_1 est un préfixe de p_2 $\exists p$ tq $p_2 = p_1 p$

Les deux radicaux $\sigma_1(l_1)$ et $\sigma_2(l_2)$ se chevauchent

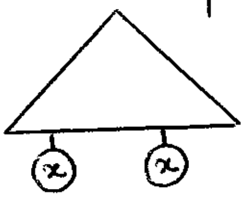


<u>Remarque</u>: dans s, seul le sous arbre à la position p_1 est modifié par les réductions. On représentera seulement ce sous-arbre par la suite

<u>a) Chevauchement non critique</u>

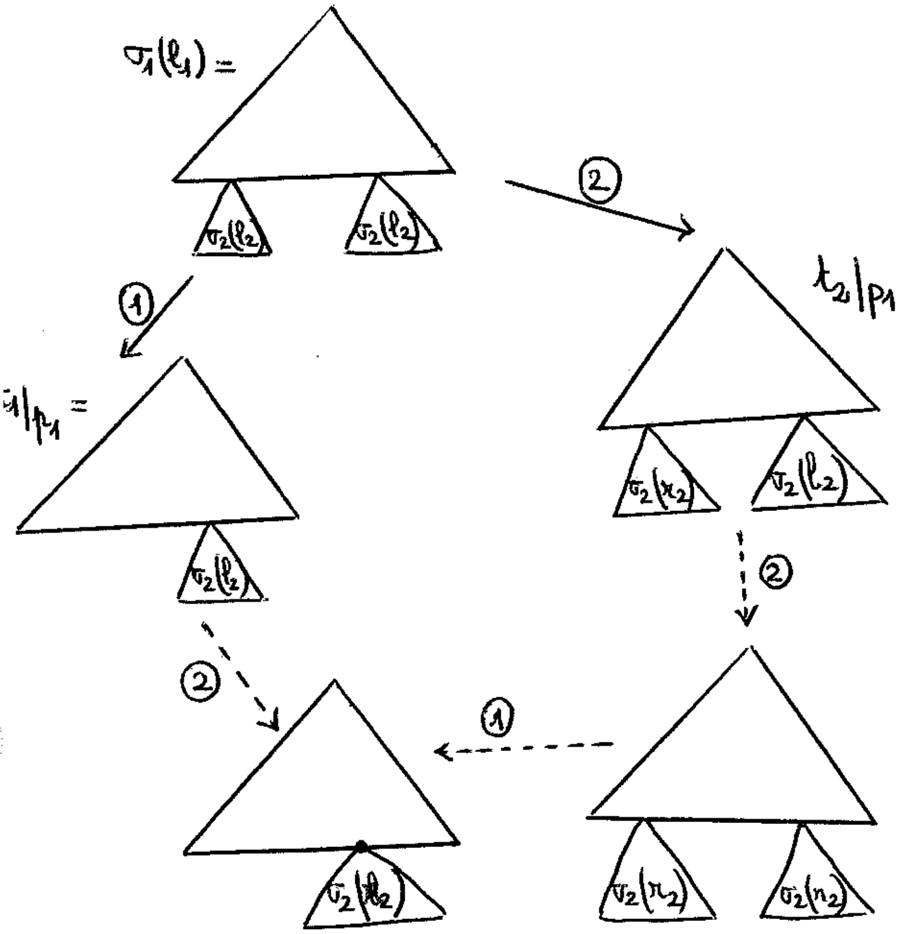
$\rightarrow$ $\sigma_2(l_2)$ ne chevauche pas l_1 lui-même mais est contenu dans σ_1.

l_1 =



(x) représente une variable du support de σ_1, c'est une feuille de l_1

On a alors :



Et t_1 et t_2 sont joignables

<u>b) Chevauchement critique</u> $\rightarrow$ $\sigma_2(l_2)$ chevauche l_1

Plus précisément, $p \in Pos(l_1)$, $l_1|_p$ n'est pas une variable et $\sigma_1(l_1|_p) = \sigma_2(l_2)$

Quitte à renommer, on peut supposer $Var(l_1, r_1) \cap Var(l_2, r_2) = \emptyset$. Notons $\sigma = \sigma_1 \cup \sigma_2$

On a $\sigma(l_1|_p) = \sigma_1(l_1|_p) = \sigma_2(l_2) = \sigma(l_2)$ Ainsi σ est un unificateur de $l_1|_p$ et l_2 et donc une instance de tout Θ unificateur principal de $l_1|_p$ et l_2.

$\langle \sigma(r_1), (\sigma(l_1))[\sigma(r_2)]_p \rangle$ est donc une instance de la paire critique $\langle \Theta(r_1), (\Theta(l_1))[\Theta(r_2)]_p \rangle$ qui est joignable par hypothèse.

$$\begin{array}{ccc} \sigma(r_1) & \text{et} & (\sigma(l_1))[\sigma(r_2)]_p \quad \text{sont joignables} \\ \shortparallel & & \shortparallel \\ \sigma_1(r_1) & & (\sigma_1(l_1))[\sigma_2(r_2)]_p \\ \shortparallel & & \shortparallel \\ t_1|_{p_1} & & t_2|_{p_1} \end{array}$$

donc t_1 et t_2 sont joignables

Réduction du problème de l'arrêt universel pour une machine de Turing au problème de terminaison d'un système de réécriture

Référence : Baader

Définition ★ Pour une machine de Turing $\mathcal{M}$, on définit la signature

$$\Sigma_{\mathcal{M}} = \{\overrightarrow{s_0}, \dots, \overrightarrow{s_m}, \overleftarrow{s_0}, \dots, \overleftarrow{s_m}\} \cup \{q_0, \dots, q_p\} \cup \{\overrightarrow{l}, \overleftarrow{r}\}$$

où chaque fonction est d'arité 1.

★ Soit x_0 une variable. Un terme de configuration t sur $\Sigma_{\mathcal{M}}$ est un terme de la forme : $t = \overrightarrow{l}(\overrightarrow{s_{i_k}}(\dots \overrightarrow{s_{i_1}}(q(\overleftarrow{s_{j_1}}(\dots \overleftarrow{s_{j_h}}(\overleftarrow{r}(x_0))\dots)))\dots))$

où $h, k \geq 0$, $\{i_1, \dots, i_k, j_1, \dots, j_h\} \subseteq \{0, \dots, m\}$ et $q \in Q$

★ Chaque terme de configuration décrit une unique configuration C_t de la machine $\mathcal{M}$:

- l'état de la machine est q
- la tête de lecture lit s_{j_1} si $h \geq 1$, ou le symbole blanc s_0 si $h = 0$
- à droite de la tête de lecture, il y a $s_{j_2}, \dots, s_{j_h}$ sur le ruban suivi d'une infinité de symboles blancs.
- à gauche de la tête de lecture, il y a $s_{i_k}, \dots, s_{i_1}$ sur le ruban.

Remarque : Réciproquement, à une configuration C de $\mathcal{M}$ correspond une infinité de termes de configuration qui diffèrent du nombre de symboles blancs s_0.

Notons Δ l'ensemble des transitions de la machine $\mathcal{M}$, les effets d'une transition sur une configuration C_t peuvent s'exprimer à l'aide d'un système de réécriture appliqué au terme de configuration t.

Définition : On définit le système de réécriture $R_{\mathcal{M}}$ par les règles suivantes :

- $\forall (q, s_i, q', s_j, r) \in \Delta$, $\quad q(\overleftarrow{s_i}(x)) \to \overrightarrow{s_j}(q'(x))$

  si $i = 0$, $\quad q(\overleftarrow{r}(x)) \to \overrightarrow{s_j}(q'(\overleftarrow{r}(x)))$

- $\forall (q, s_i, q', s_j, l) \in \Delta$, $\quad \overrightarrow{l}(q(\overleftarrow{s_i}(x))) \to \overrightarrow{l}(q'(\overleftarrow{s_j}(x)))$

  et $\quad \overrightarrow{s_k}(q(\overleftarrow{s_i}(x))) \to q'(\overleftarrow{s_k}(\overleftarrow{s_j}(x)))$

  si $i = 0$, $\quad \overrightarrow{l}(q(\overleftarrow{r}(x))) \to \overrightarrow{l}(q'(\overleftarrow{s_j}(\overleftarrow{r}(x))))$

  et $\quad \overrightarrow{s_k}(q(\overleftarrow{r}(x))) \to q'(\overleftarrow{s_k}(\overleftarrow{s_j}(\overleftarrow{r}(x))))$

Remarque Comme Δ et l'alphabet du ruban sont finis, R_M est fini !

> Lemme 1 : Soit M une machine de Turing et R_M le système de réécriture correspondant.
> 1. Pour chaque paire (t, t') de termes de configuration,
> $$\left(t \xrightarrow[R_M]{} t'\right) \Rightarrow \left(C_t \vdash_M C_{t'}\right)$$
> 2. Pour toute paire de configurations (C, C') et pour tout terme de configuration t tq $C = C_t$,
> $$\left(C \vdash_M C'\right) \Rightarrow \left(\exists t' \text{ terme de configuration tq } C' = C_{t'} \text{ et } t \xrightarrow[R_M]{} t'\right)$$

Démonstration découle des définitions.

Remarque Comme pour configuration C, il y a une infinité de termes de configuration correspondants, on ne peut pas simplifier (2) en une simple réciproque de (1) !

Conséquence Une exécution infinie dans M $\Leftrightarrow$ Une reduction infinie dans R_M

Le problème suivant est donc indécidable

> Entrée Un système de réécriture fini R et un terme t
> Problème Est-ce que toutes les reductions issues de t terminent ?

Mais ça ne suffit pas à montrer l'indécidabilité du problème de terminaison d'un système de réécriture.

PROBLÈME : Tous les termes sur Σ_M ne sont pas des termes de configuration !

> Lemme 2 : Soit t un terme quelconque sur Σ_M.
> S'il existe une réduction infinie à partir de t ($t \xrightarrow[R_M]{} t_1 \xrightarrow[R_M]{} t_2 \rightarrow \dots$), alors il existe un terme de configuration t' et une réduction infinie à partir de t'

Démonstration :

- Comme $\Sigma_{\mathcal{M}}$ ne contient que des symbôles de fonctions d'arité 1, on écrira $t = f_1(f_2(\dots f_k(x)\dots)) = w(x)$ avec $w = f_1 f_2 \dots f_k$ un mot sur "l'alphabet" $\Sigma_{\mathcal{M}}$

- En notant $\overrightarrow{\Gamma} = \{\overrightarrow{s_1}, \dots, \overrightarrow{s_m}\}$ et $\overleftarrow{\Gamma} = \{\overleftarrow{s_1}, \dots, \overleftarrow{s_m}\}$, on a

$$\Sigma_{\mathcal{M}} = \overrightarrow{\Gamma} \cup \overleftarrow{\Gamma} \cup Q \cup \{\overleftarrow{r}, \overrightarrow{l}\}$$

- Soit $w \in \Sigma_{\mathcal{M}}^*$, alors w peut s'écrire

$$w = u_1 v_1 u_2 v_2 \dots u_q v_q u_{q+1}$$

avec $\forall i \; u_i \in (\overrightarrow{\Gamma} \cup \overleftarrow{\Gamma} \cup \{\overleftarrow{r}, \overrightarrow{l}\})^*$

$\forall i \; v_i \in \overrightarrow{\Gamma}^* Q \overleftarrow{\Gamma}^*$

et $\forall i \; v_i$ est maximal dans le sens suivant :

$$\begin{cases} u_i \text{ ne termine pas par une lettre de } \overrightarrow{\Gamma} \\ u_{i+1} \text{ ne commence pas par une lettre de } \overleftarrow{\Gamma} \end{cases}$$

- Comme les règles de réécriture de $R_{\mathcal{M}}$ concernent chacune un symbole de Q, toute réduction de $w(x)$ va concerner seulement l'un des v_i. De façon plus précise :

Supposons que $w(x) \xrightarrow[R_{\mathcal{M}}]{} w'(x)$. Alors il existe $j \in [\![1, q]\!]$ tq
- $w' = u_1 v_1 \dots u_j v_j' u_{j+1} \dots v_q u_{q+1}$
- $\overrightarrow{l} v_j \overleftarrow{r}(x_0) \xrightarrow[R_{\mathcal{M}}]{} \overrightarrow{l} v_j' \overleftarrow{r}(x_0)$

Comme q est fini, si on dispose d'une reduction infinie à partir de $w(x_0)$ alors $\exists j \in [\![1, q]\!]$ tq on ait une réduction infinie à partir de $\overrightarrow{l} v_j \overleftarrow{r}(x_0)$ qui est un terme de configuration CQFD. ∎

Théorème : Le problème suivant est indécidable :
Entrée : Un système de réécriture fini R
Problème : R est-il terminant ?

Démonstration

- Notons $L_\forall = \{\langle M \rangle \,/\, M \text{ s'arrête à partir de n'importe quelle configuration}\}$ c'est le langage d'un problème indécidable

• Montrons que $L_{\mathcal{R}} = \{\langle \mathcal{R} \rangle \,/\, \mathcal{R}$ (est un système de réécriture fini) est terminant$\}$
est indécidable par réduction de $L_\forall$ à $L_{\mathcal{R}}$.
Par l'absurde, supposons que $L_{\mathcal{R}}$ est décidable.
Soient M une instance de $L_\forall$ et $\mathcal{R}_M$ le système de réécriture fini associé à M.
Considérons la procédure suivante :

- Appliquer la procédure de décision de $L_{\mathcal{R}}$ à $\langle \mathcal{R}_M \rangle$
- Si cette procédure renvoie vrai, renvoyer VRAI
- Sinon renvoyer FAUX.

Alors $\langle \mathcal{R}_M \rangle$ est terminant (ssi) M s'arrête pour toute configuration.
En effet (⇒) $\langle \mathcal{R}_M \rangle$ est terminant
donc il n'y a pas de suite infinie de réduction
donc (lemme 1) il n'y a pas d'exécution infinie dans M
(⇐) Par contraposée, si $\langle \mathcal{R}_M \rangle$ n'est pas terminant,
alors il existe un terme t sur Σ_M tq t admet une réduction infinie
donc (lemme 2) il existe une reduction infinie à partir de t' un terme de configuration de M qui admet une réduction infinie.
donc (lemme 1) il existe une exécution infinie à partir de la configuration $C_{t'}$
Absurde. ■

DEMONSTRATION de l'INDÉCIDABILITÉ Via le PCP sur l'alphabet $\{0,1\}$
"Term Rewriting Systems" Terese