Tristan Stérin
FDI
DM #2
07/01/2016

Exercice 2

On définit le problème *Term* (resp. $Conf$) de la manière suivante : étant donné un système de réécriture de termes $\mathcal{R}$, est ce que $\mathcal{R}$ est terminant (resp. confluent).

Question 1 Expliquer comment coder des mots dans un système de termes.

On suppose que les mots sont codés dans $\{0,1\}^*$.
On peut par exemple se donner deux symboles unaires $0(x)$ et $1(x)$ ainsi qu'un symbole constant $\circ$. Par exemple on code 01001 par $0(1(0(0(0(1(\circ))))))$.
De manière générale on code $u = u_0 \dots u_n$ par $u_0(\dots u_n(\circ)\dots)$.
On notera directement $u(x)$. ■

Question 2 Exhiber une réduction de PCP vers *Term* avec une signature contenant un seul symbole de fonction ternaire f (en plus de ce qui est nécessaire pour simuler des mots).

Soit une instance $< \alpha_i, \beta_i >$ de PCP. On va montrer qu'on peut lui associer un système de réécriture qui ne termine pas si et seulement si cette instance possède une solution.
On prend la signature $\sum = \{\circ,\ 0(x),\ 1(x),\ g(x,y,z)\}$. On définit les règles :

$$\begin{array}{rlcl} r_i : & g(\alpha_i(x),\ \beta_i(y),\ z) & \rightarrow & g(x,y,z) \\ r_{\circ,0} : & g(\circ,\ \circ,\ 0(x)) & \rightarrow & g(0(x),0(x),0(x)) \\ r_{\circ,1} : & g(\circ,\ \circ,\ 1(x)) & \rightarrow & g(1(x),1(x),1(x)) \end{array}$$

Grâce à ces règles on va pouvoir créer des cycles : $g(\circ,\circ,u(\circ)) \rightarrow g(u(\circ),u(\circ),u(\circ))$.

— Supposons que l'instance est une solution : $\alpha_{i_1}\dots\alpha_{i_m} \equiv \beta_{i_1}\dots\beta_{i_m}$, on pose $u = \alpha_{i_1}\dots\alpha_{i_m}$. Alors en utilisant les règles on va dépiler successivement les deux premiers arguments de g et lorsque $\circ$ est atteint on retrouve la situation initiale. On a bien :

$$g(u(\circ),u(\circ),u(\circ)) = g(\alpha_{i_1}\dots\alpha_{i_m}(\circ),\beta_{i_1}\dots\beta_{i_m}(\circ),u(\circ)) \rightarrow g(\circ,\circ,u(\circ)) \rightarrow g(u(\circ),u(\circ),u(\circ))$$

Cette réduction est cyclique, le système ne termine pas.

- On suppose que l'instance n'a pas de solution. On veut montrer que $\mathcal{R}$ termine. Ceci par induction sur les termes on montre que tout terme se réduit en un nombre fini d'étapes en une forme normale.
  - **Cas** x une variable : déjà en forme normale.
  - **Cas** $\circ$: idem
  - **Cas** $0(t)$: on ne peut appliquer aucune règles, donc par hypothèse d'induction sur t c'est bon.
  - **Cas** $1(t)$: idem
  - **Cas** $t = g(t_1, t_2, t_3)$: supposons que t possède une suite de réduction infinie. Par le cours de PROG on sait que la relation :

    $$s \succ t \equiv \exists p \ t = s_{|p} \vee s \rightarrow t$$

    conserve les termes fortements normalisants de $\rightarrow$.
    Donc ici t_1, t_2 et t_3 sont fortements normalisants pour $\succ$ par hypothèse d'induction.
    D'autre part, si $g(t_1, t_2, t_3) \rightarrow g(p_1, p_2, p_3)$ et que ce pas de ré-écriture n'est pas une application de $r_{\circ,0}$ ou $r_{\circ,1}$ à la position ϵ alors :
    - soit il s'agit d'un pas r_i à la position ϵ auquel cas les p_i sont des sous termes des t_i.
    - soit le pas de réécriture se fait à l'intérieur d'un des t_i et alors $t_i \rightarrow p_i$.

    Dans tous les cas $t_i \succeq p_i$ et la relation est stricte pour au moins un des i.
    Cela implique qu'une suite de réduction ne contenant aucune application de $r_{\circ,0}$ ou $r_{\circ,1}$ à la racine termine puisque $t \succ s \implies |t| > |s|$ et donc qu'au pire on se ramène à $g(\circ, \circ, \circ)$ sur lequel aucune règle est appliquable.
    Ainsi on a pour t le schéma suivant (où l'on prend la première apparition) :

    $$g(t_1, t_2, t_3) \rightarrow^* g(\circ, \circ, t_4) \rightarrow g(t_4, t_4, t_4)$$

    Et par le même argument :

    $$g(t_4, t_4, t_4) \rightarrow^* g(\circ, \circ, t_5)$$

    En remontant les calculs de cette dernière relation on a nécessairement que t_4 est de la forme $\alpha_{i_1} \dots \alpha_{i_m}(\circ)$ en effet par exemple pour le dernier pas, on a pris forcément une règle r_i et à l'étape d'avant on avait un $g(\alpha_i(\circ), \beta_i(\circ), t_5)$. Mais on a alors aussi $t_4 = \beta_{i_1} \dots \beta_{i_m}(\circ)$. Et donc une solution de PCP ! ! !
    Ce qui est absurde.

    On a donc montré $PCP \leq_m Term$ puisque la transformation d'une instance de *PCP* en une instance de *Term* est calculable par une machine de Turing.

∎

Question 3 Exhiber une réduction de PCP vers $Conf$ avec une signature contenant un seul symbole de fonction binaire g et deux constantes $Start$, End (en plus de ce qui est nécessaire pour simuler des mots).

On se place dans le même cadre qu'à la question précédente où l'on se donne une instance $< \alpha_i, \beta_i >$ de PCP et où l'on construit un TRS qui est confluent ssi cette instance est solution.

On propose la signature $\sum = \{g, 0, 1, \circ, Start, End\}$ et le TRS :

$$\begin{aligned}
r_{i,0} : \quad & Start & \to \quad & g(\alpha_i(\circ), \beta_i(\circ)) \\
r_{i,1} : \quad & g(x,y) & \to \quad & g(\alpha_i(x), \beta_i(y)) \\
r_{Start} : \quad & g(x,y) & \to \quad & Start \\
r_{End} : \quad & g(x,x) & \to \quad & End
\end{aligned}$$

Tout d'abord on constate que l'instance est solution de PCP si et seulement si $Start \to^* End$. En effet :

- Si l'instance est solution, il existe $u = \alpha_{i_1} \dots \alpha_{i_n} = \beta_{i_1} \dots \beta_{i_m}$ et alors pas à pas on peut faire :

  $$Start \to g(\alpha_{i_m}(\circ), \beta_{i_m}(\circ)) \to g(\alpha_{i_m}(\alpha_{i_{m-1}}(\circ)), \beta_{i_{m-1}}(\beta_{i_m}(\circ))) \to^* g(u,u) \to End$$

- Si $Start \to^* End$ si on prend une plus courte réduction, on utilise pas r_{Start} (qui ramène au début) et alors on a nécessairement utilisé $r_{i,0}$ au premier pas puis une succession fini de $r_{i,1}$ qui amène à r_{End} et donc à un mot u qui donne une solution de PCP (la forme des termes implique des réductions à la position ϵ à chaque étape).

Maintenant on montre que le TRS est confluent si et seulement si $Start \to^* End$:

- Si le système est confluent on a par r_{Start} et r_{End} : $g(x,x) \to Start$ et $g(x,x) \to End$. Or le système est confluent et End est une forme normale donc $Start \to^* End$.
- On suppose $Start \to^* End$
- On montre que tout terme est confluent par induction structurelle :
  - **Cas** End, $\circ$: formes normales, c'est bon.
  - **Cas** $Start$: d'après les hypothèses et ce qu'on a montré l'instance de PCP est ici solution. On a alors déjà montré qu'une séquence de réduction commençant par $Start$ peut se ramener à End. D'où la confluence pour $Start$.
  - **Cas** $0(t)$: supposons :

$$\begin{array}{ll}
0(t) \to^* t_1 \\
\downarrow * \\
t_2
\end{array}$$

Aucune règle ne peut s'appliquer à 0 en position ϵ donc on a aussi :

$$\begin{array}{l} t \to^* t_1 \\ \downarrow * \\ t_2 \end{array}$$

On conlut par hypothèse d'induction : t_1 et t_2 sont joignables.

— **Cas** $1(t)$: idem

— **Cas** $g(t_1, t_2)$:

  — **Cas** :

$$\begin{array}{l} g(t_1, t_2) \to^* Start \\ \downarrow * \\ Start \end{array}$$

  Immédiat.

  — **Cas** :

$$\begin{array}{l} g(t_1, t_2) \to^* End \\ \downarrow * \\ End \end{array}$$

  Immédiat.

  — **Cas** :

$$\begin{array}{l} g(t_1, t_2) \to^* Start \\ \downarrow * \\ End \end{array}$$

  Par hypothèse $Start \to^* End$.

  — **Cas** :

$$\begin{array}{l} g(t_1, t_2) \to^* g(t_3, t_4) \\ \downarrow * \\ Start \end{array}$$

  On a alors $Start \leftrightarrow^* g(t_3, t_4)$ on a montré la confluence pour $Start$ qui est équivalent à Church-Rosser donc $Start$ et $g(t_3, t_4)$ sont joignables.

  — **Cas** :

$$\begin{array}{l} g(t_1, t_2) \to^* g(t_3, t_4) \\ \downarrow * \\ End \end{array}$$

  Idem

— **Cas** :

$$\begin{array}{l} g(t_1, t_2) \rightarrow^* g(t_3, t_4) \\ \downarrow * \\ g(t_5, t_6) \end{array}$$

On a $g(t_3, t_4) \rightarrow Start$ et $g(t_5, t_6) \rightarrow Start$ par r_{Start}.
D'où le résultat.

Ce sont les seules possibilités de réductions pour $g(t_1, t_2)$ d'où le résultat.

D'où le résultat. Ainsi, comme à la question précédente on a $PCP \leq_m Conf$.

∎

Question 4 Conclure que la terminaison et la confluence d'un système de réécriture de terme sont indécidables.

On a montré :

— $PCP \leq_m Term$

— $PCP \leq_m Conf$

Or PCP est indécidable, donc $Term$ et $Conf$ sont indécidables.

∎