Indécidabilité de la confluence et de la terminaison

Dans ce développement, on montre que la confluence et la terminaison d'un système de réécriture de termes est indécidable dans le cas général, en se ramenant au problème de correspondance de Post.

Définition 1 (PCP). *Un Problème de Correspondance de Post est donné par :*
- *Un alphabet A*
- *Un ensemble de paires de mots sur A : $((u_i, v_i))_i, u_i, v_i \in A^*$ vérifiant $|u_i v_i| > 0$ pour tout i.*

Le problème admet une solution ssi il existe une suite d'indice $(i_j)_j$ tel que $u_{i_1}.u_{i_2}.\cdots.u_{i_n} =: w_u = w_v := v_{i_1}.v_{i_2}.\cdots.v_{i_n}$

Theorème 1 (Admis). *PCP est indécidable.*

Démonstration. Il est bon d'avoir une idée de la preuve du résultat même si on n'a pas le temps de la faire dans le développement. Pour montrer que PCP est indécidable, on réduit le problème de l'arrêt à une instance de PCP. On va encoder le déroulement de la machine dans PCP : des $\sharp$ vont servir à séparer des pas de calculs différents. On sait comment aller d'un pas de calcul à l'autre puisque les changements sont locaux (changement d'état, déplacement de la tête de lecture). On construit une paire $(\sharp, \sharp c_0)$ où c_0 est la situation initiale de la machine de Turing. Les autres paires sont de la forme (u, v) avec v qui est le "pas de calcul suivant" de u, et u, v décrivent localement la configuration (bout de bande de taille au plus 3). □

Theorème 2. *La terminaison d'un système de réécriture de termes est indécidable.*

Démonstration. On va donc réduire PCP à décider la non-terminaison d'un système de réécriture de termes. Etant donné une instance $P = ((u_i, v_i))$ de PCP, on va donc construire un système R de réécriture tel que PCP est satisfiable ssi R ne termine pas. En vrai, on montre même que PCP est satisfiable ssi R possède un cycle.

1) Très important, ne pas négliger cette étape ! Il faut bien expliquer comment encoder des mots sur un système de termes : pour chaque lettre $a \in A$, on associe un symbole de fonction unaire $\bar{a}$. On rajoute une constante ε qui code le mot vide, et un mot $w = w_1 \dots w_n$ sera encodé dans le système de termes par $\bar{w_1}(\bar{w_2}(\dots(\bar{w_n}(\varepsilon))))$, qu'on notera $\bar{w}$.
2) On peut maintenant définir notre signature : $\Sigma = \{f^3\} \cup \{\bar{a}, a \in A\}$
3) Le coeur de la réduction, le système de règles de réécriture :

$$\forall (u_i, v_i) \in P, \quad f(\bar{u_i}(x), \bar{v_i}(y), z) \to f(x, y, z) \tag{1}$$

La deuxième règle est $f(\varepsilon, \varepsilon, z) \to f(z, z, z)$ qui permet de boucler si on a réussi à matcher, mais pour éviter une boucle triviale de la forme $f(\varepsilon, \varepsilon, \varepsilon)$, on force z à être non vide, d'où les règles :

$$\forall a \in A, \quad f(\varepsilon, \varepsilon, \bar{a}(x)) \to f(\bar{a}(x), \bar{a}(x), \bar{a}(x)) \tag{2}$$

Maintenant que le système R de réécriture de termes est bien défini, il reste à montrer qu'il ne termine pas ssi PCP est satisfiable.

4) ($\Rightarrow$) Soit $t_1 \to t_2 \to \ldots \to t_n \to \ldots$ une suite de réduction infini de R. On montre d'abord que cette suite utilise une infinité de fois les réduction (2). En effet, si ce n'est pas le cas, on regarde la suite après la dernière réduction de type (2), et on peut définir une mesure sur les termes qui est décroissante pour les réductions (1) : $|f(x,y,z)| = |x| + |y|, |\bar{a}(x)| = 1 + |x|$. On ne peut donc pas avoir une suite infini de réductions (1).

   Il existe donc i_0 tel que $t_{i_0} \to^{(2)} t_{i_0+1}$, et donc il existe $z \in \overline{A^*}$ tel que $t_{i_0+1} = f(z,z,z)$. On regarde i_1 le prochain instant où on utilise une réduction (2), ce qui nous donne une suite d'indice k_i tel que $z = \overline{u_{k_1}}(\ldots(\overline{u_{k_m}}(\varepsilon))) = \overline{v_{k_1}}(\ldots(\overline{v_{k_m}}(\varepsilon)))$ puisqu'on utilise seulement des réductions (1) entre les deux instants. On a donc directement une solution au PCP qui est donné par la suite d'indice k_i et le mot correspondant dans A^*.

5) ($\Leftarrow$) Si on a une solution $z = u_{k_1} \ldots u_{k_m} = v_{k_1} \ldots v_{k_m}$ au PCP, on en déduit immédiatement le cycle :

$$f(z,z,z) \to f(\overline{u_{k_2}}(\ldots), \overline{v_{k_2}}(\ldots), z) \to^* f(\overline{u_{k_m}}(\varepsilon), \overline{v_{k_m}}(\varepsilon), z) \to f(\varepsilon,\varepsilon,z) \to f(z,z,z)$$

   Ce qui donne immédiatement la non-terminaison de R.

□

Theorème 3. *La confluence d'un système de réécriture de termes est indécidable.*

Démonstration. On utilise le même encodage pour les lettres et cette fois on a juste besoin d'un symbole de fonction binaire g pour tout faire marcher, ainsi que deux constantes S et T, qu'on peut voir comme un symbole initial et un symbole terminal. On va montrer que PCP a une solution ssi R est confluent.

1) On définit la signature $\Sigma = \{g^2, S, T\} \cup \{\bar{a}, a \in A\}$
2) On a trois types de règles :
   - Les premières permet d'essayer un matching :

$$\forall (u_i, v_i) \in P, \quad S \to g(\overline{u_i}, \overline{v_i})$$
$$\forall (u_i, v_i) \in P, \quad g(x,y) \to g(\overline{u_i}(x), \overline{v_i}(y))$$

   - La deuxième permet de repartir à zéro : $g(x,y) \to S$
   - La dernière permet de valider lorsqu'on trouve un matching : $g(x,x) \to T$
3) ($\Rightarrow$) Soit t un terme et $t \to^* t_1, t \to^* t_2$. Si $t_1, t_2 \neq T$, on a la confluence car $t_1 \to^* S, t_2 \to^* S$. Si maintenant $t_1 = T \neq t_2$, on a $t_2 \to^* S$ et il suffit de montrer que $S \to^* T$ pour conclure.

   Par hypothèse, PCP a une solution w correspondant aux indices $(k_i)_{1 \leq i \leq m}$ et on a alors :

$$S \to g(\overline{u_{k_m}}, \overline{v_{k_m}}) \to^* g(\overline{u_{k_1}}(\ldots), \overline{v_{k_1}}(\ldots)) = g(\overline{w}, \overline{w}) \to T$$

   D'où R est bien confluent.
4) ($\Leftarrow$) Si R est confluent, puisque $S \leftarrow g(\bar{a}, \bar{a}) \to T$, S et T sont joignables dans R. On prends le plus court chemin joignant S à T (qui ne repasse donc pas par S) qui nous donne immédiatement la suite d'indice qui construit la solution au PCP.

□