Projet

Le but de ce projet est d'implanter une version du critère de terminaison dit « étiquetage sémantique ».
Les présupposés du théorème sont les suivants :
- tous les termes sont dans une algèbre de termes $\mathscr{T}(\mathscr{F}, \mathscr{X})$ (donc bien formés) ;
- l'algèbre $\mathscr{T}(\mathscr{F}, \mathscr{X})$ contient un nombre infini (au moins dénombrable) de variables ;
- les règles des systèmes de récriture sont régulières, les membres à gauche ne sont pas réduits à des variables ;
- R est un système de récriture ;
- une $\mathscr{F}$-algèbre $\mathscr{M}$ est définie par un domaine M et une fonction $f_{\mathscr{M}} : M^n \to M$ pour tout $f \in \mathscr{F}$ d'arité n. L'ensemble des $f_{\mathscr{M}}$ constitue une *interprétation*.

Pour une valuation v, l'évaluation $[\![t]\!]_v$ d'un terme t est définie inductivement par :

$$[\![t]\!]_v = v(x) \text{ si } t = x \in \mathscr{X},$$
$$[\![f(t_1, \ldots, t_n)]\!]_v = f_{\mathscr{M}}([\![t_1]\!]_v, \ldots, [\![t_n]\!]_v) \text{ pour } f \in \mathscr{F}, t_1, \ldots, t_n \in \mathscr{T}(\mathscr{F}, \mathscr{X}).$$

Pour chaque $f \in \mathscr{F}$ on choisit un ensemble $E_f \neq \emptyset$ d'étiquettes. On note $\overline{\mathscr{F}} = \{f_e | f \in \mathscr{F}, e \in E_f\}$ où l'arité de chaque f_e est l'arité de f.

L'étiquetage d'un symbole dans un terme se fait en fonction de l'évaluation de ses arguments. Pour ce faire, on choisit pour chaque $f \in \mathscr{F}$ d'arité n, une fonction d'étiquetage de symbole $\pi_f : M^n \to E_f$. Pour une valuation v et une évaluation associée $[\![\_]\!]_v$, l'étiquetage d'un terme est défini inductivement par :

$$L_v(x) = x \in \mathscr{X}$$
$$L_v(f(t_1, \ldots, t_n)) = f_{\pi_f([\![t_1]\!]_v, \ldots, [\![t_n]\!]_v)}(L_v(t_1), \ldots, L_v(t_n)) \text{ pour } f \in \mathscr{F}, t_1, \ldots, t_n \in \mathscr{T}(\mathscr{F}, \mathscr{X}).$$

On dit que $\mathscr{M}$ est un *modèle* de R si $\forall l \to r \in R, [\![l]\!]_v = [\![r]\!]_v$ pour toute valuation v. En particulier dans ce cas on a que pour tout v, $s \to_R t \Rightarrow [\![s]\!]_v = [\![t]\!]_v$.

Pour une $\mathscr{F}$-algèbre $\mathscr{M}$ et les ensembles et fonctions correspondants E_f et π_f (pour tout $f \in \mathscr{F}$), on note $\overline{R} = \{L_v(l) \to L_v(r) | l \to r \in R \text{ et pour tout } v\}$. Si $\mathscr{M}$ est un modèle de R alors en particulier pour tout v, $s \to_R t$ entraîne que $L_v(s) \to_{\overline{R}} L_v(t)$.

Théorème 1 (Zantema) *Pour une $\mathscr{F}$-algèbre $\mathscr{M}$ constituant un modèle de R, pour des ensembles E_f et fonctions π_f choisis pour tout $f \in \mathscr{F}$ et définissant $\overline{R}$, les terminaison de R et $\overline{R}$ sont équivalentes.*

Prenons par exemple le système R sur $\mathscr{A} = \{a, b, c, d\}$ constitué des règles :

$$\begin{cases} a(a(c(x))) \to c(a(a(x))) & b(c(x)) \to b(a(d(x))) \\ d(a(a(x))) \to a(a(d(x))) & d(b(x)) \to c(b(x)) \end{cases}$$

En prenant $\mathscr{D} = \{\bot, \top\}$, on obtient des contraintes de la forme

$$\begin{aligned} &(\top = \top \wedge \bot = \bot \wedge x = x \wedge \text{NOT}\, x = \text{NOT}\, x) \\ &\wedge \neg(\top = \bot) \wedge \neg(\top = x) \wedge \cdots) \\ &\wedge (a(x) = x \vee a(x) = \text{NOT}\, x \vee a(x) = \top \vee a(x) = \bot) \\ &\wedge (b(x) = x \vee b(x) = \text{NOT}\, x \vee b(x) = \top \vee b(x) = \bot) \\ &\wedge \cdots \\ &\wedge ((d(x) = \top \wedge c(x) = \text{NOT}\, x \wedge b(x) = x) \Rightarrow \top = \text{NOT}\, x) \\ &\wedge ((d(x) = \top \wedge c(x) = \text{NOT}\, x \wedge b(x) = \bot) \Rightarrow \top = \top) \\ &\wedge \cdots \end{aligned}$$

Avec $f_a(x) = \text{NOT}\, x$, $f_b = f_c = f_d = \top$ on obtient un modèle de $\rightarrow_R$. Un système étiqueté (sur l'ensemble d'étiquettes $\{0, 1\}$) correspondant est

$$\begin{cases} a_0(a_1(c_0(x))) \rightarrow c_0(a_1(a_0(x))) & b_1(c_0(x)) \rightarrow b_0(a_1(d_0(x))) \\ a_0(a_1(c_1(x))) \rightarrow c_1(a_0(a_1(x))) & b_1(c_1(x)) \rightarrow b_0(a_1(d_1(x))) \\ d_0(a_1(a_0(x))) \rightarrow a_0(a_1(d_0(x))) & d_1(b_0(x)) \rightarrow c_1(b_0(x)) \\ d_1(a_0(a_1(x))) \rightarrow a_0(a_1(d_1(x))) & d_1(b_1(x)) \rightarrow c_1(b_1(x)) \end{cases}$$

Sa terminaison est aisément prouvée avec une interprétation polynomiale.
Prenons un autre exemple où les symboles ne sont pas tous unaires :

$$\begin{cases} x + 0 & \rightarrow & x & \quad x \times 0 & \rightarrow & 0 \\ x + s(y) & \rightarrow & s(x + y) & \quad x \times s(y) & \rightarrow & (x \times y) + x \end{cases}$$

Il devient alors moins évident de déterminer quand deux interprétations sont égales ou pas. La méthode suggérée est de passer par des tables de vérité :

1. déterminer les expressions booléennes utiles, en énumérant les interprétations possibles pour les symboles de fonction ;
2. calculer grâce aux tables de vérité celles qui sont équivalentes ou pas ;
3. fabriquer les contraintes en ajoutant aux contraintes de modèle, les égalités entre expressions equivalentes et les diségalités entre expressions non équivalentes.

En prenant $\mathscr{D} = \{\bot, \top\}$, on obtient des contraintes de la forme (avec mise en valeur des contraintes associées à la dernière règle du système) :

$\top = \top \wedge \bot = \bot \wedge x = x \wedge \text{NOT}\, x = \text{NOT}\, x \wedge x \text{ AND } y = x \text{ AND } y \wedge x \text{ OR } y = x \text{ OR } y \wedge$
$\ldots \wedge (x \text{ AND } \top = (x \text{ AND } y) \text{ OR } x) \wedge \ldots$
$(plus(x, y) = x \text{ OR } y \Rightarrow times(x, y) = x \text{ AND } y \Rightarrow Sx = \top \Rightarrow x \text{ AND } \top = (x \text{ AND } y) \text{ OR } x)$
$\ldots$
$\ldots \wedge \neg(x \text{ AND } y = (x \text{ AND } y) \text{ OR } x) \wedge \ldots$
$(plus(x, y) = x \text{ OR } y \Rightarrow times(x, y) = x \text{ AND } y \Rightarrow Sx = x \Rightarrow x \text{ AND } y = (x \text{ AND } y) \text{ OR } x) \wedge \ldots$

L'exercice consiste à chercher, sur la donnée d'un système de récriture R, une preuve de terminaison utilisant l'étiquetage sémantique.

1. Étant donnés un système, un domaine et une interprétation, fournissez une fonction vérifiant que l'interprétation construit un modèle du système.
2. On se donne comme domaine les booléens, comme fonctions d'interprétation les fonctions booléennes (XOR, AND, OR, NOT). Donnez une fonction (ou un programme) qui, sur la donnée d'un système, produit les contraintes sur les interprétations afin que celles-ci construisent un modèle. Ces contraintes seront émises dans un format SAT classique afin d'être envoyées à un SAT-solver (minisat, glucose, SatElite,...). On veillera à émettre également la liste des associations entre entiers codant les atomes dans le SAT et les atomes eux-mêmes.
3. Donner une fonction (ou un programme) qui sur la donnée d'un système, de la liste d'association et du résultat du SAT-solver, retourne les interprétations adéquates.
4. Étant donné un ensemble d'étiquettes fixé, proposer un programme qui, sur la donnée d'un système de récriture au format TRS (théorie libre, termes uniquement `http://www.lri.fr/~marche/tpdb/format.html`) et d'un modèle de ce système, retourne (au format TRS) un système étiqueté (par l'étiquetage sémantique) dont la terminaison est équivalente à la terminaison du système original. La fonction π_f devra être engendrée de façon à n'être pas constante sur toute la signature !
5. En utilisant un outil de preuve de terminaison de votre choix proposez une fonction qui sur la donnée d'un système (au format TRS) propose une preuve de terminaison du système étiqueté. Vous pourrez tester vos programmes sur la base de données de problèmes de terminaison TPDB (`http://www.lri.fr/~marche/tpdb`).

De façon facultative, vous pourrez essayer de considérer d'autres domaines, d'autres types de fonctions d'interprétation, etc.

Le projet est attendu en OCaML, toutefois si vous souhaitez le coder en un autre langage, demandez-nous notre accord au préalable.