Constellation comme machine à jetons

Support de travail

1 Présentation originale de la résolution stellaire

Voici un programme logique très simple. Les deux premières lignes permettent de calculer l'addition unaire. Un entier est représenté par un terme correspondant à une succession de symboles de fonction s par dessus une constante 0. Les symboles X, Y, Z, R sont des variables et s^n correspond à n répétitions du symbole s. On notera $\overline{n}$ la représentation d'un entier naturel n. La dernière ligne est une requête demandant la somme de n et m qui devra être stockée dans la variable R.

```
add(0, Y, Y).
add(s(X), Y, s(Z)) :- add(X, Y, Z).
?add(s^n(0), s^m(0), R).
```

L'idée du calcul est que la requête va tenter de se connecter à la première ligne. Si $n = 0$, cela va fonctionner et on aura $Y = s^m(0)$ et $R = Y = s^m(0)$ signifiant que $0 + m = m$. Si $n > 0$, alors, on se connectera plusieurs fois à la seconde ligne qui signifie que pour énoncer $(X + 1) + Y = Z + 1$, il faut passer par une justification de l'assertion $X + Y = Z$. En passant plusieurs fois par la seconde ligne, on terminera sur la première ligne qui donnera une valeur à R correspondant au résultat.

Ce programme peut être représenté par la constellation suivante :

$$\Phi_{\mathbf{N}}^{n,m} = [+add(\overline{0}, Y, Y)] + [-add(X, Y, Z), +add(s(X), Y, s(Z))] + [-add(\overline{n}, \overline{m}, R), R]$$

où les polarités représentent la distinction entre entrée et sortie, et la présence de rayons non polarisés permet d'avoir des sortes de sorties de type *affichage* qui ne participeront pas à l'interaction mais qui apparaîtront à la fin du calcul. La figure 1 présente

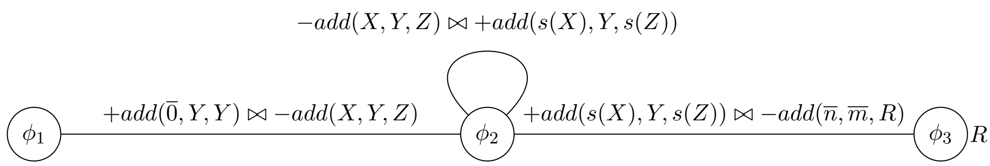
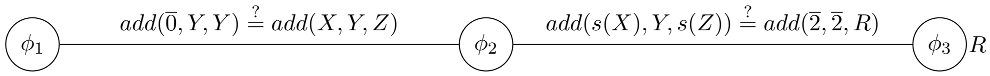
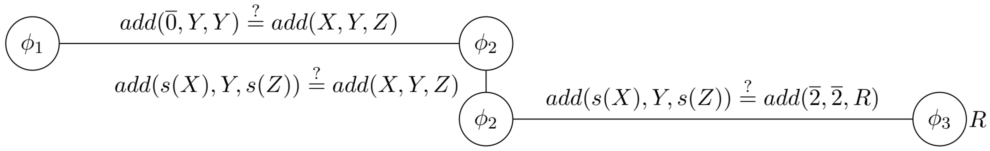


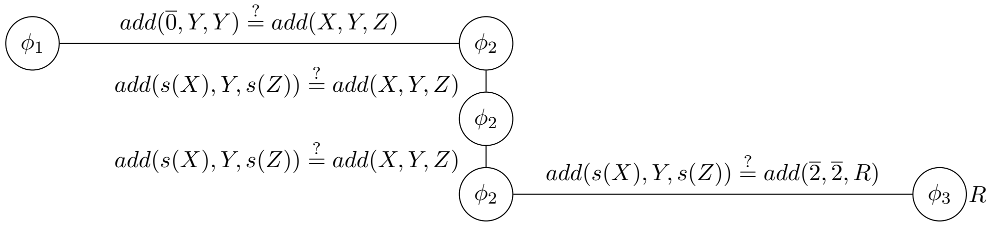
FIGURE 1 – Graphe de dépendance de $\Phi_{\mathbf{N}}^{n+m}$.



(a) Diagramme incorrect (0 récursion).



(b) Diagramme correct calculant $2 + 2$ (1 récursion).



(c) Diagramme incorrect (2 récursions).

FIGURE 2 – Exemples de diagrammes pour la constellation $\Phi_{\mathbf{N}}^{2+2}$.

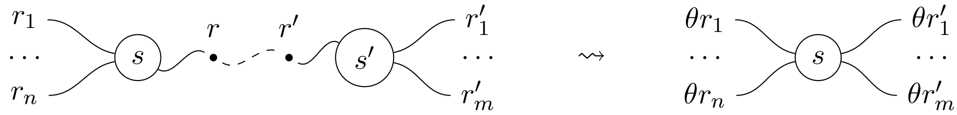


FIGURE 3 – Illustration d'une étape de fusion où θ est l'unificateur principal de l'équation associée à la paire de rayons (r, r'). La fusion des deux étoiles s et s' le long des rayons r et r' produit une nouvelle étoile s.

le graphe de dépendance qui correspond à toutes les liaisons possibles entre les étoiles le long de leurs rayons. La figure 2 présente des possibles diagrammes, c'est-à-dire des connexions concrètes entre les occurences d'étoiles (la réutilisation est possible) tel que chaque rayon est connecté à un unique autre rayon et que les connexions suivent le graphe de dépendance. En particulier, les diagrammes doivent être des graphes connexes et saturés (impossibles à étendre avec plus d'étoiles).

L'exécution de cette constellation consiste à construire tous les diagrammes saturés possibles. Chaque diagramme induit un problème d'unification où les variables sont locales/propres à leur étoile. Soit le problème d'unification ne peut pas êter résolu et dans ce cas, le diagramme est ignoré, soit il a une solution et dans ce cas, la solution est appliquée à l'étoile des rayons libres (non connectés) qui forme le résultat de l'évaluation du diagramme (c'est en fait une manière détournée de calculer la contraction de diagramme par la règle d'unification et de propagation de solution, comme décrit dans la figure 3). En réduisant tous ces diagrammes corrects, on obtient une nouvelle constellation appelée *forme normale* de la constellation de départ. Par exemple, si $\mathtt{Ex}(\Phi)$ désigne l'exécution d'une constellation Φ, alors on a $\Phi_{\mathbf{N}}^{2+2} = [\overline{4}] = [\overline{2+2}]$.

2 Vers une définition alternative d'exécution

Calculer tous les diagrammes d'une constellation peut être très coûteux, difficile à réaliser et redondant. C'est bien loin de ce qu'il se passe réellement lorsqu'un programme s'exécute en pratique. Au lieu de dupliquer explicitement une étoile (ce qui peut correspondre à une étape de récursion), il faudrait *repasser dessus*. Ce qui induit qu'un graphe de dépendance peut être parcouru comme un automate non-déterministe.

En effet, pour le graphe de dépendances de la figure 1 adapté au calcul de 2 + 2, on peut imaginer un jeton gérant un problème d'unification à chaque déplacement, qui commence au sommet ϕ_3 avec le problème d'unification vide $\emptyset$. On peut seulement se déplacer vers ϕ_2, ce qui ajoute l'équation associée à l'arête traversée qu'on essaie de résoudre pour obtenir les équations suivantes :

$$\{X \stackrel{?}{=} \overline{1}, Y \stackrel{?}{=} \overline{2}, R \stackrel{?}{=} s(Z)\}.$$

Le jeton a le choix entre aller vers ϕ_1 ou alors passer par la boucle sur ϕ_2. Le jeton va en fait se séparer en deux jetons indépendants qui vont chacun gérer une copie du problème d'unification qu'on avait :

— le premier jeton va aller vers ϕ_1, ajouter l'équation associée à l'arête entre ϕ_1 et ϕ_2 à son problème d'unification. Il y aura une erreur sur l'équation $\overline{0} \stackrel{?}{=} \overline{1}$. Le jeton meurt ;

— le second jeton va continuer dans la boucle et produire l'ensemble d'équations suivant (ne pas oublier que les variables sont locales aux étoiles ; on peut utiliser des renommages pour rendre ça explicite) :

$$\{X \stackrel{?}{=} \overline{0}, Y \stackrel{?}{=} \overline{2}, R \stackrel{?}{=} s(s(Z))\}$$

Ensuite, on refait la même chose avec le seul jeton survivant. Il se trouve qu'il ne pourra plus survivre dans la boucle et seulement aller vers ϕ_1 qui nous donnera l'équation $R \stackrel{?}{=} \overline{4}$. Les parcours survivants qui ne peuvent plus être étendus correspondent en fait à des diagrammes saturés corrects : exactement ce que l'on veut construire pour l'exécution. De plus, le problème d'unification géré par les jetons calculent en fait l'évaluation des diagrammes pas à pas.

De manière générale, les jetons travaillent en groupe sur un même problème d'unification puisqu'il y a plusieurs chemins qu'on peut parcourir en même temps. Lorsqu'il y a des choix non-déterministes,

3 Machine stellaire

Ces observations nous amènent à une définition de machine qui exécute une constellation vue comme un automate non-déterministe. De façon anecdotique, cela donne aussi une version calculatoire des systèmes de transitions étiquetés (labelled transition systems ou LTS) utilisés en model checking.

Une constellation Φ est représentée par son graphe de dépendances appelé *machine stellaire* de Φ. Cela correspond à un graphe d'état où les états sont les étoiles.

Question : est-ce que la machine peut commencer à partir de n'importe où ?

Un *jeton* j est défini par $j = (i, p)$ où p correspond à un sommet sur le graphe d'état, et i est un entier naturel correspondant à un identifiant de famille. À chaque identifiant de famille est associé un problème d'unification $\texttt{Prob}(i)$.

Une *processus* P de la machine stellaire est un ensemble de jetons de même famille (qui pourrait être représentée par une file). La machine stellaire est associée à une file $F = \langle P_1, ..., P_n \langle$ de processus.

Soit un jeton $j_0 = (i, p)$ placé n'importe où sur la machine stellaire. Le jeton peut effectuer les transitions suivantes :

— se diviser en jetons $(i, p_1), ..., (i, p_n)$ qui se propagent sur les positions $p_1, ..., p_n$ pour chaque arête adjacente qui est déterministe (il n'y a qu'un chemin possible pour chacune des rayons associés à ces arêtes à partir de l'étoile associée à i) et ensuite, on ajoute les équations correspondant à $p_1, ..., p_n$ au problème $\texttt{Prob}(i)$ qu'on essaie de résoudre. Si l'unification échoue, le processus de la famille i est retirée de la file F : tous les jetons meurent ;

— pour chaque choix non-déterministe (lorsqu'il y a plus d'une arête pour un même rayon de l'étoile i), tous les jetons de famille i sont dupliqués et produisent autant de nouvelles familles distinctes qu'il y a de choix non-déterministes. Cela représente le parcours de plusieurs diagrammes en parallèle ;

— Dans le cas du choix co-déterministe, lorsqu'un jeton va d'un rayon r de l'étoile i vers un rayon r' d'une étoile j qui est aussi connecté à un autre rayon r'' d'une étoile k, il faut réfléchir à comment faire. C'est un peu plus subtil.

Au début de l'exécution, on place un jeton par composante connexe puis on effectue les étapes décrites ci-dessus. Une famille s'arrête lorsqu'elle ne peut plus progresser dans le graphe sachant qu'il est impossible de retourner en arrière.