Implémentation de la β-réduction au sein d'une machine de Turing

Léo Gayral

2017-2018

ref : `https://www.irif.fr/~carton/Enseignement/Complexite/ENS/Redaction/2009-2010/pablo.rauzy.pdf`

Définition 1 (Indices de de Bruijn, représentation des λ-termes)**.** Une façon équivalente de représenter des λ-termes consiste à utiliser les indices de de Bruijn. On distingue au sein d'une formule deux types de variables : les variables *libres*, qu'on notera avec des $\{l_i, i \in \mathbb{N}^*\}$, et des variables *liées*, notées avec des $\{v_i, i \in \mathbb{N}^*\}$, où l'indice de v_i indique à quel λ cette variable est liée. Ainsi, la formule $\lambda ab.bac$ deviendra $\lambda\lambda v_1 v_2 l_1$.

Pour représenter les λ-termes sur un alphabet fini, on va considérer un codage unaire des entiers, ce qui donne l'ensemble de variables $\mathcal{V} = (v|l)1^+$. On va en outre utiliser une notation préfixe pour s'affranchir de l'ambiguïté et des parenthèses. En mettant de côté les considérations sur les variables libres ou liées, les λ-termes seront alors des mots de $\Lambda := \mathcal{V}|@\Lambda\Lambda|\lambda\Lambda$. Dans cette représentation, la formule $\lambda ab.bac$ s'écrit alors $\lambda\lambda@@v1v11l1$.

On travaillera donc sur l'alphabet $\Sigma = \{v, l, 1, \lambda, @\}$ par la suite.

Théorème 1. On peut construire une machine de Turing à 5 rubans semi-infinis qui, étant donné un λ-terme en entrée, détecte s'il est sous forme normale, et réduit le *redex* le plus à gauche dans le cas contraire.

Démonstration.
Le premier point à remarquer est que, dans notre représentation des λ-termes, un redex du type $(\lambda x.t_1)\, t_2$ se traduit par la présence de $@\lambda$ au début de son écriture.

On considère un ruban d'entrée E qui sera *lu de gauche à droite*, et un ruban de sortie S sur lequel on *écrit de gauche à droite*. Pour le cas où un

redex est détecté, on aura besoin d'un ruban F pour la *fonction* et d'un ruban A pour son *argument*, ainsi que d'un ruban M pour la *mémoire* de travail.

On va ainsi lire E, et le copier sur S tant qu'on ne détecte pas de @λ dans l'entrée. Si aucun @λ n'est détecté, alors l'entrée est déjà sous forme normale, et copiée telle quelle sur S.

Supposons donc qu'un @λ est lu sur l'entrée. Dans ce cas, on ne copie @ nulle part. On cherche alors à isoler les chaînes associées à t_1 et t_2 dans le redex $(\lambda x.t_1)\, t_2$. Pour ce faire, il suffit de comparer le nombre de @ et de variables rencontrées en lisant E.

En effet, si on regarde l'arbre syntaxique d'un λ-terme, chaque @ induit deux sous-arbres, et on peut alors montrer par induction que le nombre de feuilles, de variables, est égal au nombre de nœuds binaires, de @, plus un. En outre, en lisant l'encodage préfixe de cet arbre, cette propriété n'est vraie que sur l'expression complète, et tout préfixe de l'expression a au moins autant de @ que de variables. Cette propriété est analogue à celle sur le nombre de parenthèses ouvrantes et fermantes dans un bon parenthésage. Cette propriété est en particulier au sein des *sous-arbres* associés aux termes $\lambda x.t_1$ et t_2.

On copie ainsi E sur F en partant du λ, en ajoutant un 1 sur M à chaque @ rencontré et en effaçant un 1 à chaque v ou l rencontré. Si M est déjà vide, on finit d'écrire la variable sur F qui contient alors le code de $\lambda x.t_1$. On procède de même pour écrire le code de t_2 sur A. On laisse enfin le curseur de E sur la prochaine lettre à écrire sur S après réduction du redex.

Pour effectuer la réduction, il faut dans F détecter quelles variables correspondent bien au premier λ de l'expression. Pour ce faire, on va devoir garder une trace de notre position au sein de l'arbre syntaxique dans M, en ajoutant deux symboles $\{g, d\}$ à l'alphabet de travail. On va alors lire F de gauche à droite, et le copier sur S à l'exception du tout premier λ. À chaque @ rencontré, on ajoute g sur M, et à chaque λ on ajoute un 1.

Si on rencontre une variable, on commence par regarder si elle est liée au tout premier λ de F en comparant le nombre de 1 après v et dans la mémoire M. Si ces quantités sont égales, on écrit A en entier au lieu de la variable sur S, et sinon on copie la variable telle quelle. Ceci étant fait, on efface M jusqu'à trouver un g – ce qui signifie qu'on était dans le fils gauche d'un nœd de l'arbre syntaxique – qu'on remplace alors par un d.

Une fois la lecture de F terminée, et les occurrences de variables liées au

premier λ remplacées par l'argument A, il suffit enfin de reprendre la copie du ruban E sur S, sans chercher à détecter un autre redex. □