

README Ce fichier n'est qu'un brouillon de réécriture de l'article de Grandjean et Schwentick *Machine-independent characterizations and complete problems for deterministic linear time* pour voir si ces notions peuvent s'étendre au temps $\mathcal{O}(n^k)$.

Notations Par abus de notation découlant de la théorie des ensembles, j'écrirai n pour $\llbracket 0, n - 1 \rrbracket$ voire pour $\llbracket 1, n \rrbracket$ quand il n'y aura pas d'ambiguïté. De manière générale, si un terme ressemble à un entier naturel mais qu'il est mis à la place d'un ensemble, il faut le lire comme l'ensemble $\llbracket 0, n - 1 \rrbracket$.

Preliminaries (p.198)

RAM data structures (p.198)

Définition 1 (RAM data structures). Soit t un type, c'est-à-dire une signature fonctionnelle ne contenant que des symboles de constantes ou de fonctions unaires.

Une RAM-structure s de type t est un uplet constitué de :

- $n \in \mathbb{N}$ est la taille de la structure ;
- $C \in \mathbb{N}$ pour chaque symbole $C \in t$;
- $f : n \rightarrow \mathbb{N}$ pour chaque symbole $f \in t$.

On notera $s.n, s.C, s.f$ les composantes n, C, f de s (cette notation est à rapprocher de l'accès à un attribut ou à une fonction membre en programmation objet).

On dira que s est c -bornée pour $c \in \mathbb{N}$ lorsque $s.C, s.f(i) < cs.n$ pour tous $C, f \in t$ et $i \in n$.

Définition 2 (Fonction de RAM). Soient t_1, t_2 des types.

Une (t_1, t_2) -fonction de RAM Γ est une fonction telle qu'il existe $c_1, c_2 \in \mathbb{N}$, tels que Γ envoie les structures c_1 -bornées de type t_1 sur des structures c_2 -bornées de type t_2 .

On dit que Γ est polynomiale lorsque $\Gamma(s).n = \mathcal{O}((s.n)^k)$.

Machine RAM (p.200)

La machine RAM reste la même (heureusement). On va utiliser la $\{+\}$ -RAM ou des versions un brin plus puissantes comme la $\{+, -, \times, \div k\}$ -RAM pour un $k \in \mathbb{N}$ fixé.

Définition 3 (Temps polynomial). On définit $\text{DTIME}_{\text{RAM}}(\mathcal{O}(n^k))$ comme étant l'ensemble des fonctions calculables sur $\{+\}$ -RAM en temps $\mathcal{O}(n^k)$, telles que le nombre de registres utilisés, la longueur des nombres manipulés (y compris les adresses de registres) soient bornés par $\mathcal{O}(n^k)$.

Réductions affines (p.202)

On laisse inchangées les notions de *transformations affines* (définition 2.3), de *réductions affines* (définition 2.5), de *projections affines* (définition 2.7). Les théorèmes et lemmes suivants ou intermédiaires sont aussi inchangés. Ils permettront de définir des réductions qui *restent* dans $\text{DTIME}_{\text{RAM}}(\mathcal{O}(n^k))$ pour un k fixé.

Le framework algébrique (p.208)

LSRS (p.208)

Le LSRS en tant que tel n'a pas l'air collé à la définition du temps linéaire. On garde la définition pour le moment.

On doit refaire la définition 3.3.

Pour t un type, on note F_t l'ensemble de symboles de fonctions suivants : $\{1(-), n(-), id(-)\} \cup \{f_C | C \in t\} \cup \{f | f \in t\}$.

Remarque 1. Soient t un type et S un LSRS pour f_1, \dots, f_h sur F_t . L'entrée d'un LSRS peut être vue comme étant une RAM-structure s de type t , qu'il lit en interprétant les symboles de F_t de la façon suivante :

- $\forall f \in t_1 : f(i) = \begin{cases} s.f(i) & \text{si } i < s.n \\ 0 & \text{sinon} \end{cases}$
- $\forall C \in t_1 : f_C(i) = s.C$
- $1(i) = 1, n(i) = s.n, id(i) = i$

La sortie du LSRS peut aussi être vue comme une nouvelle structure $s' = S(s)$ de type $\{f_1, \dots, f_h\}$.

Ici, on a plusieurs possibilités pour adapter le concept de *linéairement représenté* (définition 3.3) à n^k .

Définition 4 (RAM n^k -représentée par LSRS - Proposition 1). Soient t_1, t_2 des types. Soit Γ une (t_1, t_2) -fonction de RAM.

Soit S un LSRS pour f_1, \dots, f_h sur F_{t_1} .

On dit que Γ est n^k -représentée par S lorsqu'il existe un entier c et une projection affine P tels que, pour chaque structure s c -bornée, S définit des fonctions $f_1, \dots, f_h : c(s.n)^k \rightarrow c(s.n)^k$ telles que $\Gamma(s) = P((s.n)^k, S(s))$ ($S(s)$ est la structure définie par le LSRS).

La modification réside dans le domaine de définition des fonctions et la taille de la sortie dans la projection.

Problèmes 1.

- Est-ce qu'on capture tout $DTIME_{RAM}(\mathcal{O}(n^k))$?
- Est-ce que $P((s.n)^k, S(s))$ capture toutes les structures de taille $\mathcal{O}(n^k)$?

Si Γ est n^k -représentée par un LSRS alors on dit que Γ est définissable par LSRS.

La définition 3.4 (définition par cas) et les lemmes 3.5 (la définition par cas ne change pas la puissance des LSRS) et 3.6 (composition de fonctions définissables par LSRS reste définissable par LSRS) restent les mêmes.

LRS (p.211)

La définition d'un terme récursif (non numérotée) et d'un LRS (définition 3.7) restent les mêmes. Le lemme 3.8 d'existence d'une solution unique au LRS aussi.

On doit adapter la définition de n^k -représentée par LRS :

Définition 5 (RAM n^k -représentée par LRS - Proposition 1). Soit Γ une fonction de RAM.

Soit E un LSRS $g(x) = \sigma(x)$.

On dit que Γ est n^k -représentée par E lorsqu'il existe un entier c et une projection affine P tels que, pour chaque structure s c -bornée, E définit une fonction $g : c(s.n)^k \rightarrow c(s.n)^k$ telle que $\Gamma(s) = P((s.n)^k, E(s))$ ($E(s)$ est la structure définie par le LRS, elle est de type $\{g\}$).

La modification réside dans le domaine de définition de g .

Et là, c'est la foire.

On doit vérifier si le théorème principal de l'article tient encore au temps polynomial.

Conjecture 1 (Adaptation du lemme 3.10 (p.212)). Toute fonction de RAM n^k -représentée par LSRS est aussi n^k -représentable par LRS.

Démonstration. Ici commence la relecture de la preuve.

□