Théorème de Savitch

Leçons 913, 915

Références : LFCC, Carton p. 219.

Théorème

Soit $s : \mathbb{N} \to \mathbb{R}_+$ telle que $s(n) \geq n$ pour n assez grand.
Toute machine de Turing non déterministe qui fonctionne en espace $S(n)$ est équivalente à une machine de Turing déterministe en espace $O(s(n)^2)$.

Résumé

I – Se ramener à une machine avec un unique état final.

II – Définition d'une fonction $\mathrm{ACCESS}(C, C', t, r)$, vraie s'il existe un calcul entre deux configurations C et C' de longueur au plus t, n'utilisant que des configurations intermédiaires de longueur au plus r.

III – Complexité de ACCESS

i – si $s(n)$ est calculable

ii – si $s(n)$ n'est pas calculable

IV – Utilisation de ACCESS pour conclure.

I • Si M fonctionne en espace $s(n)$,
On peut modifier M pour que, avant d'accepter, elle remplace tous les symboles de la bande par # et se place dans un état q_f.
→ ainsi, il n'y a qu'une configuration acceptante.

II • On définit la fonction ACCESS par récurrence, avec le cas de base :
si $t = 0$: est-ce que $C = C'$?
si $t = 1$: est-ce que $C = C'$ ou $C \to C'$ en une étape de calcul ?

• Sinon, en parcourant toutes les configurations C'' en vérifiant s'il existe un calcul $C \to C'$ avec C'' en position médiane.

ACCESS(C, C', t, r)
- si t = 0 : renvoyer C = C'
- sinon si t = 1 : renvoyer C = C' ou $C \to C'$.
- sinon
  - pour tout C'' de taille r :
    - si ACCESS(C, C'', $\lceil t/2 \rceil$, r) et ACCESS(C'', C', $\lfloor t/2 \rfloor$, r)
      - renvoyer VRAI
  - renvoyer FAUX

III.) • <u>Remarque</u>

→ il y a au plus $\log_2 t$ appels récursifs embriqués car t est divisé par 2 à chaque fois

→ chaque appel récursif utilise : • trois variables C, C' et C'', chacune de taille au plus r.
• un entier t

→ r ne change pas, il suffit de le stocker une fois

Conclusion : il faut un espace

$$O(\underbrace{\log r}_{\text{stocker r en binaire}} + \underbrace{(\log t + r)}_{\text{pour chaque appel embriqué}} \underbrace{\log t}_{\text{nombre d'appels embriqués}})$$

stocker r en binaire

stocker t en binaire

stocker C, C', C''

nombre d'appels embriqués

pour chaque appel embriqué

i Supposons que $s(n)$ est calculable.

- M fonctionne en espace $s(n)$, et elle effectue donc un nombre d'étapes $t_M(n) \leq 2^{Ks(n)}$ pour une certaine constante K.
- Un mot w de taille n est donc accepté par M si et seulement si $ACCESS(q_0 w, q_f, 2^{Ks(n)}, s(n))$

car il n'y a qu'une seule ~~état~~ configuration finale

ie $w \in L(M) \iff ACCESS(q_0 w, q_f, 2^{ks(n)}, s(n))$

- On peut alors définir une machine de Turing déterministe qui
  - calcule $s(n)$
  - utilise ACCESS avec $t = 2^{Ks(n)}$ et $r = s(n)$.
- L'espace utilisé est alors

$$O(K^2 s(n)^2) = O(s(n)^2).$$

car
$$\log r = O(\log(s(n))) = O(s(n))$$
$$\log t = O(\log(2^{Ks(n)})) = O(Ks(n)) = O(s(n))$$

ii • Si $s(n)$ n'est pas calculable, il va falloir ruser pour trouver un majorant de l'espace nécessaire.

- Posons n : la taille de l'entrée w

  m : la taille maximale d'une configuration accessible depuis $q_0 w$.
- Calculons m en espace $O(m^2) = O(s(n)^2)$, avec la propriété suivante : pour $k \geq n+1$

  ↔ soit N_k = le nombre de configurations de taille au plus k accessibles à partir de $q_0 w$.

  → $(N_k)_{k \geq n+1}$ est croissante !

  → et elle est bornée par le nombre de configurations de taille au plus $s(n)$.

• Donc $(N_k)_{k \geq n+1}$ converge, et on a :

$$m = \min\{ k \geq n+1 \mid N_k = N_{k+1} \}$$

Comme la taille d'une configuration ne varie que d'un à chaque étape du calcul, soit $N_{k+1} > N_k$ soit $N_{k'} = N_k$ pour $k' \geq k$.

→ m est donc aussi bien défini.

• On en déduit l'algorithme :

CALCUL $m(w)$ où $|w| = n$

- $N := -1$, $\Pi := 0$, $k := 0$
- Tant que $N \neq \Pi$
  - $k := k+1$, $N := \Pi$, $\Pi := 0$
  - Pour tout C de taille au plus k :
    - Si $\text{ACCESS}(q_0 w, C, 2^{Kk}, k)$: $\Pi := \Pi + 1$

cette constante peut être prise comme étant égale à $\log_2(1+|Q|+|\Gamma|)$

• On a toujours $k \leq m$ donc les appels à ACCESS sont en $O(m^2)$, et toutes les variables se stockent en $O(m)$.

→ CALCULm tourne donc en $O(m^2) = O(s(n)^2)$.

IV • Il suffit finalement de définir Π' qui calcule m avec CALCULm, puis qui utilise la fonction $\text{ACCESS}(q_0 w, q_f, 2^{Km}, m)$.

→ elle donne le résultat en $O(s(n)^2)$.