

TD 7

—

Automates à pile déterministes

Exercice 1

1. Montrer que D_n^* peut être accepté par sommet de pile par un automate D+TR+S.
2. Montrer que D_n^* est D+TR mais pas D+S.

Exercice 2

1. Montrer qu'un langage L est déterministe et préfixe ($L \cap L\Sigma^+ = \emptyset$) ssi il existe un automate déterministe qui accepte L par pile vide.
2. Montrer que pour les automates à pile déterministes, l'acceptation par pile vide est équivalente à l'acceptation par pile vide ET état final.

Exercice 3

1. Montrer que tout langage algébrique déterministe est non ambigu.
2. *Lemme d'itération déterministe*
Montrer que pour tout langage déterministe $L \subseteq \Sigma^*$, il existe un entier K tel que pour tout $w \in L$ avec au moins K lettres distinguées, w se factorise en $w = \alpha u \beta v \gamma$ avec
 - (a) $\forall n \in \mathbb{N}, \alpha u^n \beta v^n \gamma \in L$;
 - (b) soit α, u et β , soit β, v et γ contient chacun au moins une lettre distinguée;
 - (c) $u\beta v$ contient moins de k lettres distinguées;
 - (d) pour tout $m \in \Sigma^*$,

$$\exists p \in \mathbb{N}, \alpha u^p \beta v^p \gamma m \in L \Rightarrow \forall p \in \mathbb{N}, \alpha u^p \beta v^p \gamma m \in L$$

3. Montrer que le langage $L = \{a^n b^n \mid n > 0\} \cup \{a^n b^{2n} \mid n > 0\}$ est non-ambigu mais pas déterministe.
4. Montrer que la famille des langages algébriques déterministes n'est pas close par union, concaténation, étoile et morphisme.

Exercice 4

1. Soit $\mathcal{A} = (Q, \Sigma, Z, T, q_0, z_0, F, K)$ un automate à pile déterministe reconnaissant par sommet de pile et état final (une configuration (q, az) est acceptante si $(q, z) \in K \subseteq Q \times Z$). Montrer qu'on peut effectivement construire un automate à pile déterministe équivalent reconnaissant par état final.
2. Soit \mathcal{A} un automate à pile déterministe. Montrer qu'on peut effectivement construire un automate à pile qui reconnaît le même langage et dont les ε -transitions sont uniquement effaçantes : $(p, x) \xrightarrow{\varepsilon} (q, \varepsilon)$.