
Correction des exercices de TD

Table des matières

1	TD1	1
1.1	Exercice 1	1
1.2	Exercice 2	2

1 TD1

1.1 Exercice 1

1. Comment peut-on caractériser un langage rationnel (régulier) ?
2. Les langages de programmation (C, Java, Python) sont-ils réguliers ? Pourquoi ?
3. Comment peut-on caractériser un langage algébrique ?
4. Soit l'extrait suivant de grammaire EBNF de requête de consultation SQL :

```

SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  select_expr [, select_expr] ...
  [into_option]
  [FROM table_references
    [PARTITION partition_list]]
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
    [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | position}
    [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
;

```

Expliquer les différentes conventions de notation utilisées !

5. Les requêtes suivantes sont-elles correctes ?

```

select 5 ;
SELECT distinct nom from etudiant
select ref from articles order by rayon asc, ref asc;

```

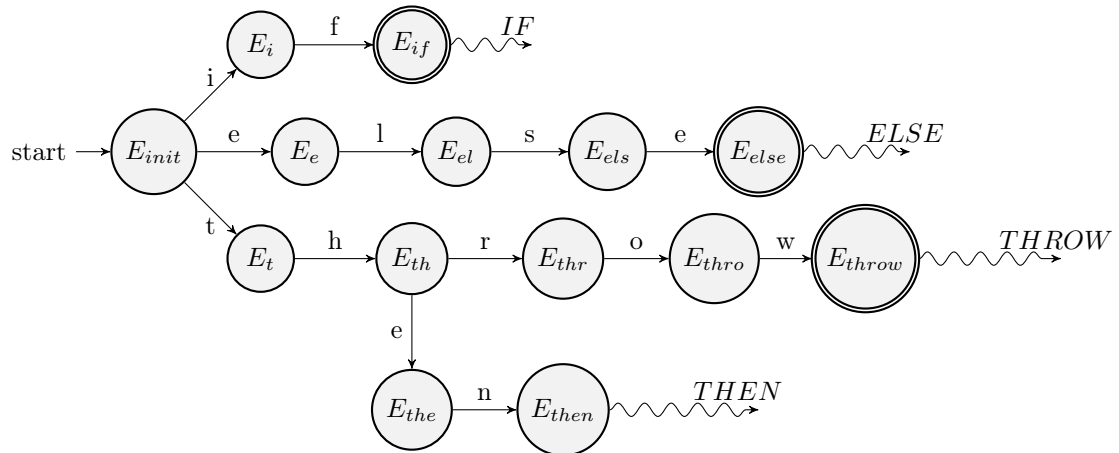
1. Tout langage reconnu par un automate d'état fini déterministe est un langage rationnel (régulier).
2. Les langages de programmation ne sont pas des langages réguliers car ils contiennent une structure emboîtée non bornée (aucun automate d'état fini déterministe ne peut être construit pour le langage $a^n b^n$).

3. Langages a association de symboles (structure emboîtée non bornée), reconnus par un automate à pile avec un arbre de dérivation constructible par induction.
4. EBNF = Extended Backus-Naur Form. Les $[]$ sont utilisés pour désigner une option facultative, $|$ pour le « ou », \dots pour l'opération de répétition de n caractères ($n \geq 0$) et $\{\}$ pour indiquer un choix non optionnel.
5. Seule la seconde requête n'est pas correcte (il manque le $;$).

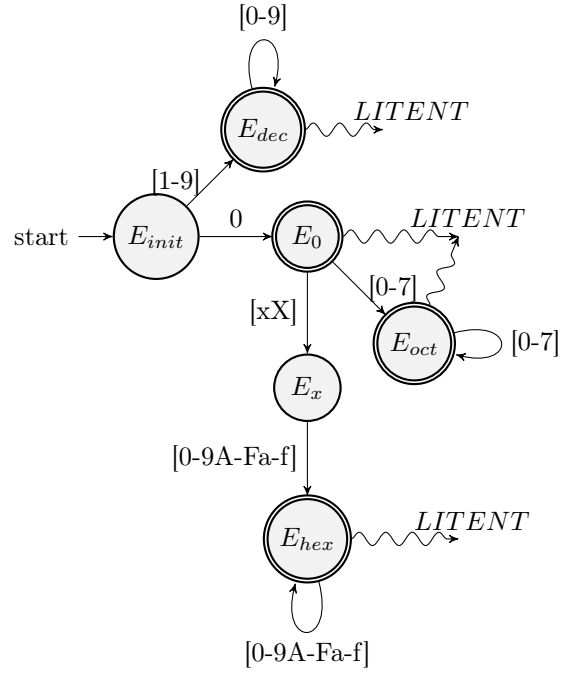
1.2 Exercice 2

- Dessiner un Automate à états Finis Déterministe (AFD) distinct pour chacun des langages suivants :
 1. Langage de certains mots-clés du C : $L_{key} = \{if, then, else, throw\}$ (sensible à la casse).
 2. Langage des littéraux numériques entiers du C (ou C++, ou Java), décimaux L_{c10} , octaux L_{c8} , hexadécimaux L_{c16} .
 3. Langage L_{id} des identificateurs composés d'une lettre au moins, éventuellement suivie de chiffres, de lettres et de “_”.
 4. Langage des littéraux numériques flottants décimaux L_f ; la suite de chiffres à gauche ou bien à droite du point décimal pouvant être vide; l'exposant entier n'est pas obligatoire. Exemples : 13., 1.7e23, .89E-34.
 5. Langage L_{sep} des séparateurs composés de blancs (espace, $\backslash t$, $\backslash n$), des commentaires à la C et à la C++.
- Dessiner un unique AFD à jeton reconnaissant une partie de ces langages. Vous reconnaîtrez notamment : le mot-clé if, les identificateurs, les entiers décimaux, les flottants sans exposant, les séparateurs. Utiliser des jetons négatifs pour les lexèmes à filtrer (séparateurs).

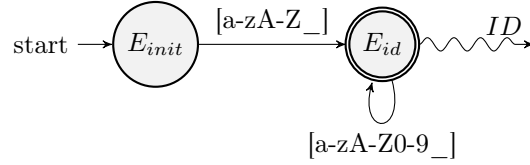
- 1. Automate fini déterministe L_{key} :



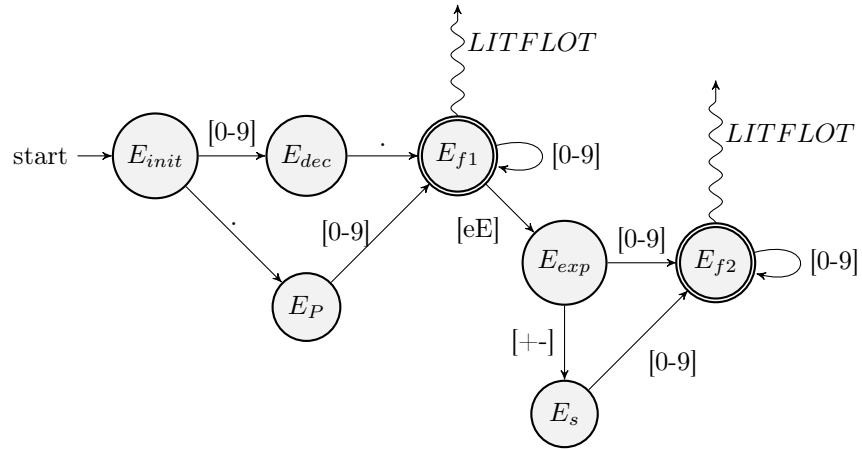
2. Automate fini déterministe de L_{c10} , L_{c8} et L_{c16} :



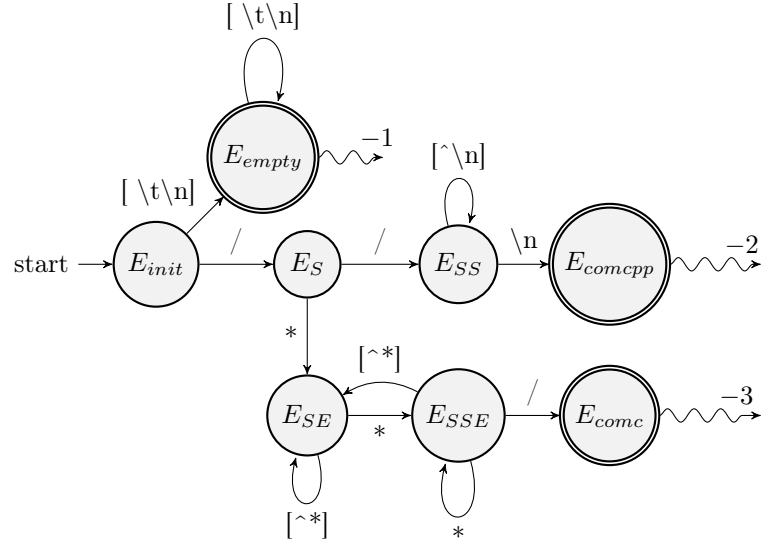
3. Automate fini déterministe de L_{id} :



4. Automate fini déterministe de L_f :



5. Automate fini déterministe de L_{sep} :



— AFD de L_{if} , L_{id} , L_{c10} , L_f et L_{sep} :

