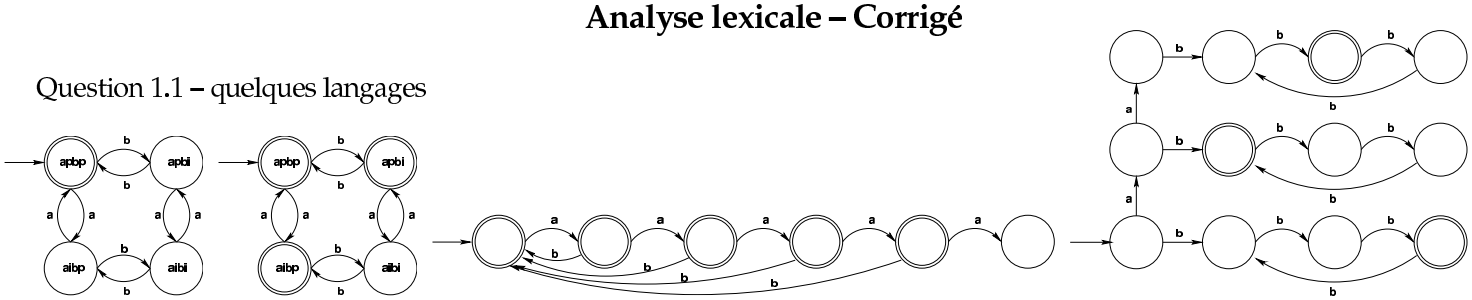
Compilation - TD 1

Analyse lexicale – Corrigé

Question 1.1 – quelques langages



Question 1.2 – expressions régulières

$(aa|bb|(ab|ba)((aa|bb))*(ab|ba))*$
...
$(bb*(a|aa|aaa|aaaa))*$
$(bbb*)|ab(bbb)*|aabb(bbb)*$

Question 1.3 – définitions régulières

$$d_0 = 0$$
$$d1 = (1d_0)*1|(d_01)*d_0$$
$$d2 = (2d_1)*2|(d_12)*d_1$$
...
$$d9 = (9d_8)*9|(d_89)*d_8$$

Question 1.4 – sur l'expressivité

La réponse est gardée par le Dragon.

Question 2.1

3

Question 2.2 – les commentaires

C'est très simple, il suffit de ne jamais mettre de commentaire dans son code.

Question 3.1

On sait construire les automates (triviaux) qui reconnaissent un caractère de Γ, ainsi que le . (wildcard). Si les expressions régulières R_1 et R_2 sont reconnues respectivement par les automates A_1 et A_2, alors

- Un automate reconnaissant R_1R_2 est construit comme la juxtaposition de A_1 et A_2. Il a pour entrée l'état d'entrée de A_1, pour états accepteurs les états accepteurs de A_2, et possède des ϵ-transitions de tous les états accepteurs A_1 vers l'entrée de A_2.
- Un automate non déterministe reconnaissant $R_1|R_2$ est construit de la manière suivante à partir des automates A_1 et A_2 : son état d'entrée est un nouvel état qui possède deux ϵ-transitions vers les états d'entrée de A_1 et A_2. Il possède un état accepteur, qui est un nouvel état recevant des ϵ-transitions de tous les états accepteurs de A_1 et A_2.
- On construit facilement de même un automate reconnaissant R_1* à partir de A_1 en reliant ses sorties à son entrée par des ϵ-transitions.

Question 3.2

Cet automate est clairement non déterministe.

Question 3.3

Le principe est de parcourir l'automate non déterministe et de construire un nouvel automate déterministe, dans lequel chaque état représente un ensemble d'états de l'AFN, l'ensemble des états accessibles après avoir lu les mêmes symboles d'entrée.

REMARQUE : la traduction inverse sera vue dans le cours de Marianne Delorme.