

TP2 : L'ANALYSE LEXICALE POUR LE LANGAGE A.

1. OBJECTIF

L'objectif de ce TP est de programmer un analyseur pour le langage A de description d'automates finis déterministe, Dans ce projet, l'analyseur lexicale sera produit automatiquement avec le logiciel Flex, à partir d'un fichier de spécification. Ce fichier décrit les unités lexicales du langage.

2. LE LANGAGE A DE DESCRIPTION D'AUTOMATES FINIS DETERMINISTE.

Le langage de description d'automates finis déterministe va permettre à ses utilisateurs d'écrire un automate de manière simple, structurée et lisible. Voici ces principales caractéristiques :

1. Mots clés réservés :

Nous allons définir des mots clés spécifiques que le programmeur ne peut pas utiliser comme noms d'états :

- automate
- alphabet
- etats
- initial
- finaux
- transitions
- verifier

Ces mots-clés nous permettront de structurer correctement l'analyse syntaxique.

2. Les commentaires :

Nous allons permettre aux utilisateurs du langage A d'utiliser des commentaires sur une seule ligne en commençant par le caractère spécial #.

3. Les identifiants :

Les identifiants (noms d'états, nom de l'automate) suivront une règle simple :

- Commencent par une lettre.
- Suivie de lettres ou chiffres.
- Pas d'accents ni caractères spéciaux.

Exemples valides : q0, Etat1, S3

4. L'alphabet :

L'alphabet sera généralement une liste de symboles, chacun composé d'un caractère simple {a, b, c...}.

- Nous allons limiter les symboles aux lettres de l'alphabet minuscule.

Exemple : `alphabet = {a, b, c}`

5. Les états :

Les états seront une liste des identifiants représentant les états de l'automate.

Exemple : `etats = {q0, q1, q2}`

6. Les transitions :

Les transitions seront une liste des transitions sous la forme :

`etat_source : symbole -> etat_destination;`

Exemple : `transitions = {
 q0 : a -> q1;
 q1 : b -> q2;
}`

7. Validation d'un mot :

La vérification d'un mot par un automate va se faire par l'instruction :

`verifier nomAutomate "mot" ;`

8. Les caractères spéciaux :

Nous allons utiliser :

- { } pour les blocs,
- ; pour terminer les instructions,
- = pour l'affectation,
- : et -> pour exprimer une transition.
- " pour le mot à vérifier par l'automate créer.

Voici un exemple d'un programme en A :

```
automate NomAutomate { #déclaration de l'automate  
    alphabet = {a, b, c}; #déclarations des alphabets  
    etats = {q0, q1, q2}; #déclaration des états  
    initial = q0; #spécification de l'état initial  
    finaux = {q2}; #spécification des états finaux  
    transitions = { #déclaration des transitions.  
        q0 : a -> q1; #première transition
```



```
    q1 : b -> q2; #deuxieme transition
    q2 : c -> q0; #troisieme transition
};
} #fin du bloc de déclaration de l'automate NomAutomate.
verifier NomAutomate "ab"; #vérification du mot « ab » par l'automate.
```

3. CE QU'IL FAUT FAIRE

1. Écrire l'analyseur lexical sous la forme d'un fichier .l.
2. Générer l'analyseur à l'aide du programme Flex.
3. Copier le programme exemple donné ci-dessus dans un fichier situé dans le même dossier que l'analyseur généré.
4. Exécuter l'analyseur sur ce fichier exemple.
5. Sur une feuille séparée, écrire votre nom et prénom, puis proposer des améliorations pour cette partie de l'analyse lexicale.