Titre de la note 06/10/20

Implémentation d'un analyseur lexical

A) Diagramme de transition

1) créer un nouveau projet Java.
2) Définir la classe Categorie

classe & categorie

final

INF → 60

SUP → 62

SUPEGAL → 256

INFEGAL → 257

NONEGAL → 258

EGAL → 61

3) Définition de la classe UniteLexicale

Attributs: categorie : entier
lexeme : chaîne de caractères

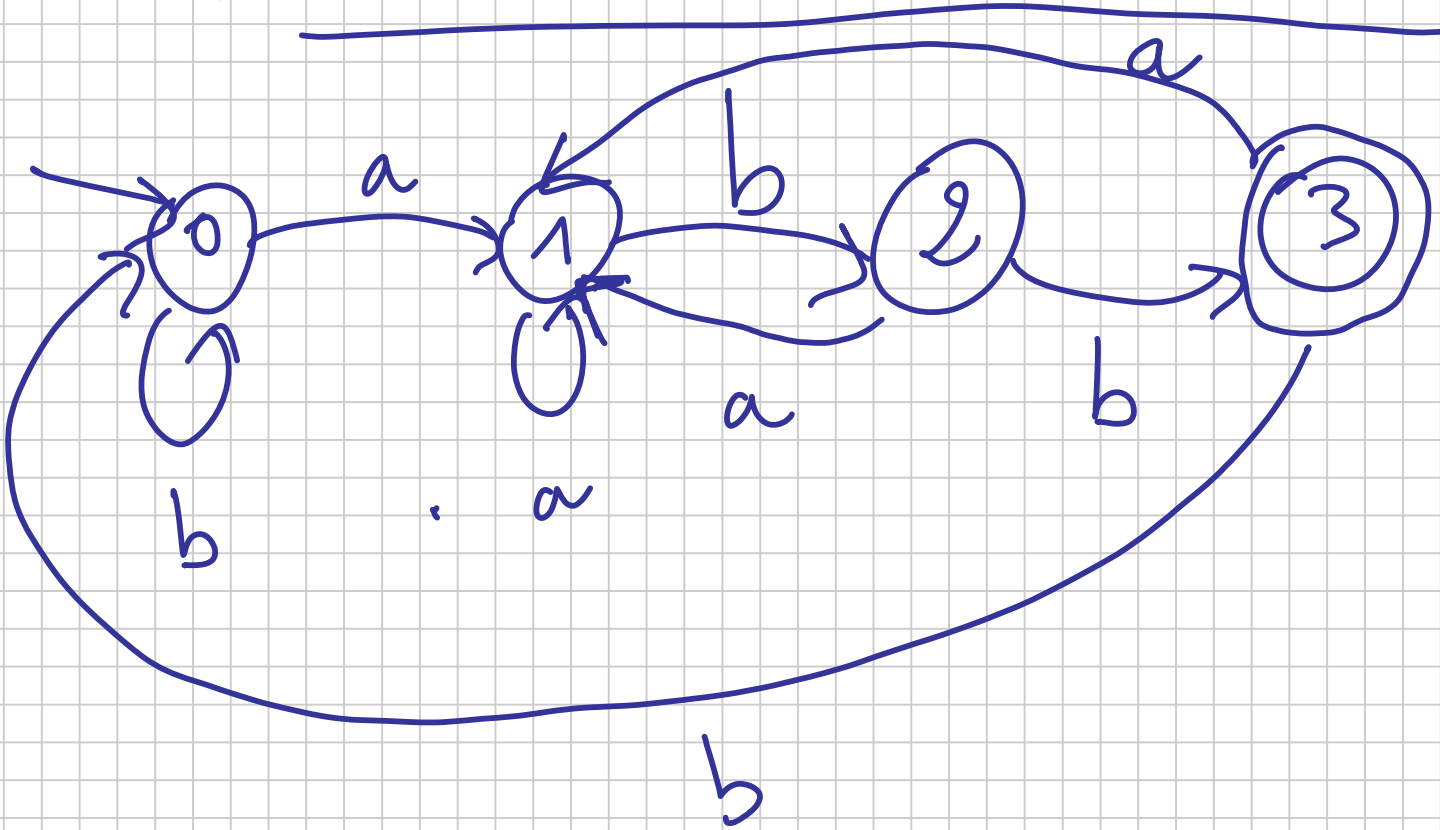
Méthodes : Constructeur
Redéfinition de toString()

4) Définition de la classe Scanner

Attributs :
- flux Caractères : tableau de caractères
- courant : entier

Méthodes :
- Constructeur (nom du fichier)
- caractèreSuivant ( ) : retourne le caractère suivant ou -1.
- reculer ( )
- obtenirOpRel ( ) : UniteLexicale
  ⇒ unité = nullité (echec ( ))

B) Automate Finii Deterministe AF:



a	17	17	17	17
b	0	3	16	0

Table de transition

1) Définir un fichier texte (AFD)

[alphabet]

0..9
+|-

a b

[etat]

0 1 2 3

0
0
3

[transition]

1 0
1 2
1 3
1 0

2) Définir la classe Symbole:

Attributs: Symbole : chaîne de caract

Méthodes:
- + constructeur
- + méthode d'accès get et set
- + redéfinition de toString()
- + redéfinition de equals()

3) Définir la classe Etat :

Attributs : label : chaîne de caractères
initial, final : boolean

Méthodes :
- Constructeur
- Méthodes d'accès get et set
- redéfinition de toString()
- redéfinition d'equals()

4) Définir la classe Automate

Attributs : alphabet : liste de Symbole.
états : liste d'état
transitions : liste de chaîne de caractères
fluxCaractères : tableau de caractères
courant : entier

Méthodes :
- constructeur
- chargerAutomate (nom du fichier)
  - lire le Symbole .....
  - lire les états .....
  - lire les transitions .....

- eof() : Si on a atteint la fin du flux.
- lireTexteEntrée(nom du fichier)
  remplir flux Caractères
- trans(e: Etat, s: Symbole): Etat
- reconnaitre(): booléen
  (Algorithme du transparent 2)