

Compilation (#1b) : Analyse lexicale

C. Deleuze & L. Gonnord

Grenoble INP/Esisar

2022-2023

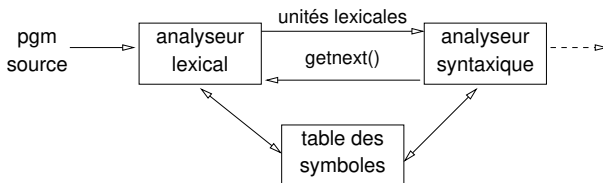


Plan

- 1 Rôle de l'analyseur lexical
- 2 Spécification des unités lexicales
- 3 Reconnaissance des unités lexicales
- 4 Outils de construction d'analyseurs lexicaux

- 1 Rôle de l'analyseur lexical
- 2 Spécification des unités lexicales
- 3 Reconnaissance des unités lexicales
- 4 Outils de construction d'analyseurs lexicaux

Place de l'analyseur lexical



UNITÉ LEXICALE	LEXÈMES	DESCRIPTION INFORMELLE DES MODÈLES
if	if	if
then	then	then
assign	:=	:=
oprel	< <= > >= = <>	< <= > >= = <>
id	position vitesse ...	lettre suivie de lettres ou chiffres
nb	3.1416 0 6.02E23 ...	toute constante numérique
chaîne	"core dumped" ...	tous caractères entre " et " sauf "

Figure: Exemples d'unités lexicales

Attributs des unités lexicales

...

- 1 Rôle de l'analyseur lexical
- 2 Spécification des unités lexicales
 - Chaines et langages
 - Expressions régulières
- 3 Reconnaissance des unités lexicales
- 4 Outils de construction d'analyseurs lexicaux

2 Spécification des unités lexicales

- Chaines et langages
- Expressions régulières

- alphabet, mot (on dira chaîne)

- alphabet, mot (on dira chaîne)
- langage

- alphabet, mot (on dira chaîne)
- langage
 - opérations sur les langages

- alphabet, mot (on dira chaîne)
- langage
 - opérations sur les langages

OPÉRATION	NOTATION	DÉFINITION
union de L et M	$L \cup M$	$\{s \mid s \in L \text{ ou } s \in M\}$
concaténation de L et M	LM	$\{st \mid s \in L \text{ et } t \in M\}$
fermeture positive	L^+	$\bigcup_{i=1}^{\infty} L^i$
fermeture de Kleene	L^*	$\bigcup_{i=0}^{\infty} L^i$

Figure: Définitions des opérations sur langages

2 Spécification des unités lexicales

- Chaines et langages
- Expressions régulières

Règles de définition des expressions régulières sur un alphabet Σ

- chapitre 1 du cours CS322
 - on notera $|$ au lieu de $+$

Règles de définition des expressions régulières sur un alphabet Σ

- chapitre 1 du cours CS322

- on notera $|$ au lieu de $+$

❶ ϵ est l'expression régulière qui dénote $\{\epsilon\}$

❷ $a \in \Sigma$ a = expression régulière qui dénote $\{a\}$

❸ r et s expressions régulières qui dénotent $L(r)$ et $L(s)$

$r | s$ dénote $L(r) \cup L(s)$

rs dénote $L(r)L(s)$

r^* dénote $(L(r))^*$

- 1 Rôle de l'analyseur lexical
- 2 Spécification des unités lexicales
- 3 Reconnaissance des unités lexicales
 - Automates finis
 - Construction d'un analyseur lexical
- 4 Outils de construction d'analyseurs lexicaux

Un reconnaisseur pour un langage est un programme qui prend en entrée une chaîne x et répond si oui ou non x est une chaîne du langage.

3 Reconnaissance des unités lexicales

- Automates finis
- Construction d'un analyseur lexical

Automate fini non déterministe

C'est un modèle mathématique qui consiste en :

- 1 ensemble d'états E
- 2 ensemble de symboles d'entrée Σ (alphabet des symboles d'entrée)
- 3 fonction de transition $\text{Transiter} : (\text{état}, \text{symbole}) \rightarrow \{ \text{états} \}$ (avec $\text{symbole} \in \Sigma \cup \{\epsilon\}$)
- 4 état distingué "initial"
- 5 ensemble d'états distingués "finaux"

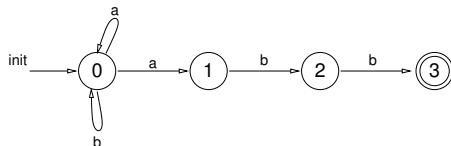
Automate fini non déterministe

C'est un modèle mathématique qui consiste en :

- 1 ensemble d'états E
- 2 ensemble de symboles d'entrée Σ (alphabet des symboles d'entrée)
- 3 fonction de transition $\text{Transiter} : (\text{état}, \text{symbole}) \rightarrow \{ \text{états} \}$ (avec $\text{symbole} \in \Sigma \cup \{\epsilon\}$)
- 4 état distingué "initial"
- 5 ensemble d'états distingués "finaux"

Un AFN accepte une chaîne x s'il existe un chemin dans le graphe entre l'état initial et un état final tel que les étiquettes d'arcs le long de ce chemin épellent le mot x .

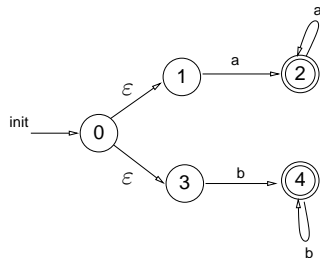
AFN reconnaissant le langage $(a|b)^*abb$



état	symbole d'entrée		
	a	b	ε
0	{ 0, 1 }	{ 0 }	-
1	-	{ 2 }	-
2	-	{ 3 }	-

Figure: Table de transition de l'automate précédent

Un autre AFN

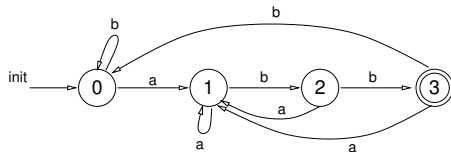


Automates finis déterministes (AFD)

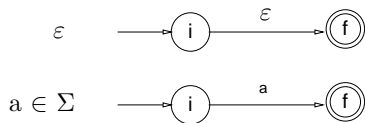
Cas particulier d'AFN.

- 1 aucune ϵ -transition,
- 2 pour chaque état e et symbole a , au plus un arc étiqueté a quittant e .

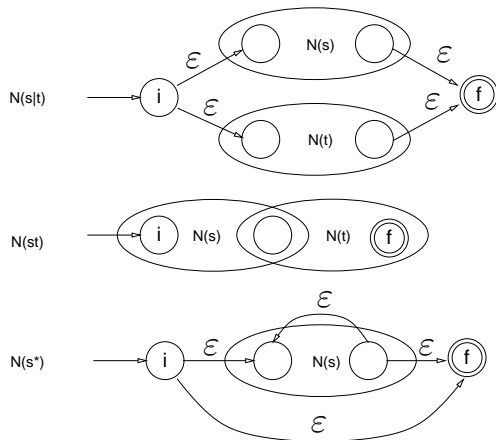
AFD pour le langage $(a|b)^*abb$



Construction de Thompson – 1



Construction de Thompson – 2



3 Reconnaissance des unités lexicales

- Automates finis
- Construction d'un analyseur lexical

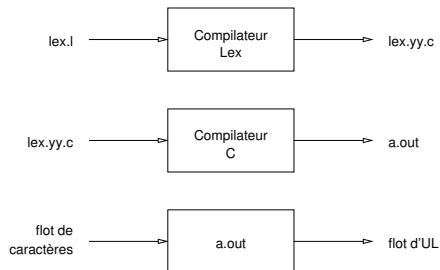
- construire un automate pour chaque modèle
- les fusionner (en distinguant les états finaux)
- rendre déterministe/minimiser
- implémenter/exécuter l'automate

- construire un automate pour chaque modèle
- les fusionner (en distinguant les états finaux)
- rendre déterministe/minimiser
- implémenter/exécuter l'automate

une petite subtilité (voir postparation)

- 1 Rôle de l'analyseur lexical
- 2 Spécification des unités lexicales
- 3 Reconnaissance des unités lexicales
- 4 Outils de construction d'analyseurs lexicaux

Utilisation de lex



Spécification lex

en trois parties

déclarations

eg macros pour les regexps

%%

règles de traduction

m_1 { action_1 }

m_2 { action_2 }

...

m_n { action_n }

%%

procédures auxiliaires

utilisées dans les actions

m_i = modèle

action_i = code à exécuter quand un lexème concorde avec le modèle

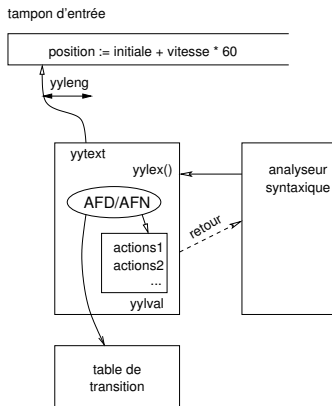
```
/* table de transition
   insérée ici ... */

int yylex() {
    while(1) {
        simule automate jusqu'à match

        fixe yytext    /* le lexème */
            yyleng     /* trouvé      */

        if m_1 actions_1 /* typique : return(UL) */
        if m_2 actions_2 /* peut fixer yylval pour attribut
            ...
    }
}
```

Fonctionnement du code généré par lex



Postparation

Soient les unités lexicales et modèles suivants :

id	lettre suivie de lettres ou chiffres
assign	=
oprel	< <= > >=

On a construit un automate reconnaissant chacun des trois modèles, on les a fusionnés et on analyse lexicalement le texte :

a <= b

Quelles unités lexicales vont être reconnues ? Comment résoudre le problème ?

Bilan

- 1 Rôle de l'analyseur lexical
- 2 Spécification des unités lexicales
 - Chaines et langages
 - Expressions régulières
- 3 Reconnaissance des unités lexicales
 - Automates finis
 - Construction d'un analyseur lexical
- 4 Outils de construction d'analyseurs lexicaux