

# Analyseurs syntaxiques

Damien Nouvel



# Plan

1. Automates à pile
2. Analyse syntaxique LL
3. Analyse syntaxique LR

# Machine de Turing



Alan Turing (UK, 1912 - 1954)

- ▶ **Machine de Turing**
  - Ruban : suite de cases mémoire
  - Tête de lecture / écriture
  - Ensemble d'états
  - Configurations et actions

# Machine de Turing

- ▶ Formellement, heptuplet :  $(Q, \Sigma, \Gamma, \$, \delta, q_0, F)$ 
  - $Q$  : ensemble des états
  - $\Sigma \subseteq \Gamma \setminus \{\$\}$  : symboles terminaux
  - $\Gamma$  : symboles terminaux et non-terminaux
  - $\$ \in \Gamma$  : symbole de fin de chaîne (blanc)
  - $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{G, D\}$  : fonction de transition
  - $q_0 \in Q$  : état initial
  - $F \subseteq Q$  : ensemble de états finaux

⇒ Automate avec non-terminaux, écriture et déplacement

- ▶ **Configuration**

- Symboles sur le ruban
- Position de la tête de lecture
- État courant

# Automate à pile

- ▶ À mi-chemin entre l'**automate** et la **machine de Turing**
- ▶ Formellement, heptuplet :  $(Q, \Sigma, \Gamma, \perp, \delta, q_0, F)$ 
  - Par défaut, mêmes symboles que pour la machine de Turing
  - $\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$  : fonction de transition
  - $\perp \in \Gamma$  : symbole de fond de pile

⇒ Reconnaissance de grammaires hors-contexte

- ▶ **Configuration**

- Symboles sur le ruban à analyser
- Symboles sur la pile
- État courant

# Reconnaissance de langage

## ► Transition

- Entrée
  - $q \in Q$  : état courant
  - $a \in \Sigma$  : symbole sur le ruban
  - $\gamma \in \Gamma$  : symbole à dépiler (ou  $\epsilon$ )
- Action
  - $q \in Q$  : état dans lequel basculer
  - $\gamma \in \Gamma$  : symbole à empiler (ou  $\epsilon$ )
  - Consommer un symbole (déplacer la tête de lecture à droite)

## ► Une transition change la configuration

- Transition :  $\delta(q_i, a, \gamma_i) = (q_j, \gamma_j)$
- Symbole  $a$  sur le ruban
- Dérivation :  $(q_i, aw, \gamma_i\alpha) \rightarrow (q_j, w, \gamma_j\alpha)$

⇒ Objectif : **état final** et **pile vide** et **entrée vide**

# Exemple

- ▶ Reconnaissance de  $\{a^n b^n, n > 0\}$ 
  - $Q = \{q_0, q_1, q_2\}$
  - $\Gamma = \{P\}$
  - $\Sigma = \{a, b\}$
  - $F = \{q_2\}$
  - $\delta$  définie par la table

$Q$	$\Gamma$	$\Sigma$	$Q$	$\Gamma$
$q_0$	$\epsilon$	$a$	$q_1$	$P$
$q_1$	$\epsilon$	$a$	$q_1$	$P$
$q_1$	$P$	$b$	$q_2$	$\epsilon$
$q_2$	$P$	$b$	$q_2$	$\epsilon$

# Exemple

- ▶ Reconnaissance des imbrications correctement formées
  - Uniquement parenthèses et crochets, par exemple  $[(\llbracket () \rrbracket)]()$
  - Définition formelle de l'automate
  - Configurations pour le mot  $()[(\llbracket () \rrbracket)]$
  - $Q = \{q_0\}$
  - $\Gamma = \{P, C\}$
  - $\Sigma = \{(\, , \, ), \, [\, , \, ]\}$
  - $F = \{q_0\}$
  - $\delta$  définie par la table

$Q$	$\Gamma$	$\Sigma$	$Q$	$\Gamma$
$q_0$	$\epsilon$	$($	$q_0$	$P$
$q_0$	$\epsilon$	$[$	$q_0$	$C$
$q_0$	$P$	$)$	$q_0$	$\epsilon$
$q_0$	$C$	$]$	$q_0$	$\epsilon$



# Plan

1. Automates à pile
2. Analyse syntaxique LL
3. Analyse syntaxique LR

# Analyseur LL : Premiers et Suivants

- ⇒ Analyseur **descendant** (ou prédictif)
- ⇒ **LL** Left-to-right Leftmost derivation
  - ▶ Détermination des non-terminaux annulables
  - ▶ Détermination de  $Premiers(X) = \{a \in \Sigma \mid X \xrightarrow{*} a\alpha\}$ 
    - Initialiser pour tout terminal  $Premiers(a) = \{a\}$
    - Pour chaque règle  $X \rightarrow X_1 X_2 \dots X_n$
    - Ajout  $Premiers(X_i)$  à  $Premiers(X)$  si  $X_1 \dots X_{i-1}$  annulables
  - ▶ Détermination de  $Suivants(X) = \{a \in \Sigma \mid S \xrightarrow{*} \alpha X a \beta\}$ 
    - Initialiser  $Suivants(S) = \{\$ \}$
    - Pour chaque règle  $X \rightarrow X_1 X_2 \dots X_n$
    - Ajout  $Premiers(X_j)$  à  $Suivants(X_i)$  si  $X_{i+1} \dots X_{j-1}$  annul.
    - Ajout  $Suivants(X)$  à  $Suivants(X_i)$  si  $X_{i+1} \dots X_n$  annulables

# Analyseur LL : table et analyse

- ▶ Création d'une table  $T$  dans  $\Gamma \times \Sigma$ 
  - Pour chaque règle  $X \rightarrow \alpha$ 
    - Pour chaque  $a \in Premiers(\alpha)$  ajouter  $X \rightarrow \alpha$  à  $T[X, a]$
    - Si  $\alpha$  annulable, pour  $a \in Suivants(X)$  ajouter  $X \rightarrow \alpha$  à  $T[X, a]$
- ▶ Fonctionnement de l'automate à pile
  - Au départ : la pile contient  $S\perp$ , ajouter  $\$$  à la chaîne
  - Pour chaque configuration avec  $a$  sur le ruban et  $X$  sur la pile

⇒ Si  $X$  est un terminal et que  $X = a$

  - Dépiler  $X$  et avancer sur l'entrée
  - Si  $a = \perp$  l'analyse est terminée

⇒ Si  $X$  est un non-terminal

  - Trouver  $T[X, a] = X \rightarrow \alpha$
  - Dépiler  $X$
  - Empiler  $\alpha$  (dont le premier élément est en haut de pile)

# Exercice

- ▶ Grammaire des fonctions mathématiques
  - $S \rightarrow N(P)$
  - $N \rightarrow f|g$
  - $P \rightarrow VA$
  - $V \rightarrow x|y$
  - $A \rightarrow ,|P|\epsilon$
- ▶ Annulables :  $\{A\}$
- ▶ Premiers et suivants (le terminal  $,$  est noté *virg*)
  - $Premiers(S) = \{f, g\}, Suivants(S) = \{\$\}$
  - $Premiers(N) = \{f, g\}, Suivants(N) = \{(\}$
  - $Premiers(P) = \{x, y\}, Suivants(P) = \{)\}$
  - $Premiers(V) = \{x, y\}, Suivants(V) = \{virg, )\}$
  - $Premiers(A) = \{virg\}, Suivants(A) = \{)\}$

# Exercice

► Table LL

	(	)	$f$	$g$	$x$	$y$	,	\$
$S$			$S \rightarrow N(P)$	$S \rightarrow N(P)$				
$N$			$N \rightarrow f$	$N \rightarrow g$				
$P$					$P \rightarrow VA$	$P \rightarrow VA$		
$V$					$V \rightarrow x$	$V \rightarrow y$		
$A$		$A \rightarrow \epsilon$					$A \rightarrow , P$	

► Faire les analyses pour

- $f(x)$
- $f(x, y, x)$

# Exercice

- ▶ Chiffres binaires positifs ou négatifs à virgule
  - Grammaire
    - $S \rightarrow TCNV$
    - $T \rightarrow + | - | \epsilon$
    - $N \rightarrow CN | \epsilon$
    - $C \rightarrow 0 | 1$
    - $V \rightarrow .N | \epsilon$
  - Annulables :  $\{T, N, V\}$
  - Premiers et suivants
    - $Premiers(S) = \{+, -, 0, 1\}$ ,  $Suivants(S) = \{\$\}$
    - $Premiers(T) = \{+, -\}$ ,  $Suivants(T) = \{0, 1\}$
    - $Premiers(N) = \{0, 1\}$ ,  $Suivants(N) = \{., \$\}$
    - $Premiers(C) = \{0, 1\}$ ,  $Suivants(C) = \{0, 1, ., \$\}$
    - $Premiers(V) = \{.\}$ ,  $Suivants(V) = \{\$\}$

# Exercice

► Table LL

	+	-	0	1	.	\$
<i>S</i>	$S \rightarrow TCNV$	$S \rightarrow TCNV$	$S \rightarrow TCNV$	$S \rightarrow TCNV$		
<i>T</i>	$T \rightarrow +$	$T \rightarrow -$	$T \rightarrow \epsilon$	$T \rightarrow \epsilon$		
<i>N</i>			$N \rightarrow CN$	$N \rightarrow CN$	$N \rightarrow \epsilon$	$N \rightarrow \epsilon$
<i>C</i>			$C \rightarrow 0$	$C \rightarrow 1$		
<i>V</i>					$V \rightarrow .N$	$V \rightarrow \epsilon$

► Faire les analyses pour

- -1001
- 110.110
- +0.01
- 1.001

# Inconvénients de l'analyse LL

- ▶ Déterministe
    - Selon la pile et le ruban, un ou plusieurs choix
    - Comment choisir une règle s'il y a plusieurs choix?
- ⇒ Analyses LL( $k$ ) pour déterminer selon  $k$  terminaux



# Plan

1. Automates à pile
2. Analyse syntaxique LL
3. Analyse syntaxique LR

# Principes généraux

- ⇒ Analyseur **ascendant**
- ⇒ **LR** Left-to-right Rightmost derivation
  - **États** de l'automate
    - Avancement dans l'analyse
    - Simultané pour plusieurs règles
  - **Actions** possibles
    - Shift (décalage) *s*
    - ⇒ Un symbole d'entrée est consommé
    - Reduce (réduction) *r*
    - ⇒ Une règle de grammaire a été utilisée
    - Accepter la chaîne en entrée *acc*

# Items et ensembles d'items

- ▶ **Augmentation** de grammaire (nouveau symbole de départ)
  - ▶ **Item**
    - Une règle et l'état d'avancement pour cette règle
    - Utilisation du point  $\cdot$
    - Exemple :  $X \rightarrow \alpha \cdot \beta$
  - ▶ Détermination d' **ensemble d'items**
    - **Noyau** : ce que l'analyseur doit consommer
    - Exemple :  $X \rightarrow \alpha \cdot Y\beta$
    - **Fermeture** : règles commençant par  $Y$  (non-terminal)
    - Exemple :  $Y \rightarrow \cdot \gamma$
    - **Transition** : avancée dans l'analyse
    - Exemple :  $Transition(X \rightarrow \alpha \cdot Y\beta, Y) = X \rightarrow \alpha Y \cdot \beta$
    - ⇒ Numérotation des ensembles :  $Transition(i, Y) = j$
- ⇒ Calcul de toutes les possibilités par fermeture et transition

# Exemple : fermeture et transition

- ▶ Fonctions récursives ( $n$  un nom de fonction,  $c$  un chiffre)
  - $F \rightarrow n(E)$
  - $E \rightarrow E + c | F | c$
- ▶ Augmentation ( $S \rightarrow F$ )
- ▶ Ensembles d'items
  - 0 :  $\{S \rightarrow \cdot F\} \cup \{F \rightarrow \cdot n(E)\}$
  - 1 :  $\{S \rightarrow F \cdot\}$
  - 2 :  $\{F \rightarrow n \cdot (E)\}$
  - 3 :  $\{F \rightarrow n(\cdot E)\} \cup \{E \rightarrow \cdot E + c, E \rightarrow \cdot F, F \rightarrow \cdot n(E), E \rightarrow \cdot c\}$
  - 4 :  $\{F \rightarrow n(E \cdot), E \rightarrow E \cdot + c\}$
  - 5 :  $\{E \rightarrow F \cdot\}$
  - 6 :  $\{E \rightarrow c \cdot\}$
  - 7 :  $\{F \rightarrow n(E) \cdot\}$
  - 8 :  $\{E \rightarrow E + \cdot c\}$
  - 9 :  $\{E \rightarrow E + c \cdot\}$

# Construction de la table LR

- ▶ Attribution d'un état (entier) pour chaque ensemble d'items
- ▶ Création d'une table *Action* dans  $C \times \Sigma$
- ▶ Pour chaque item de l'ensemble d'items  $i$ 
  - Si l'item est de la forme  $X \rightarrow \alpha \cdot Y\beta$  (avec  $Y$  terminal ou non)
  - et que  $Transition(i, Y) = j$

⇒ Ajouter  $s : j$  dans la table  $Action[i, a]$

  - Si l'item est de la forme  $X \rightarrow \alpha \cdot$  de la règle  $k$

⇒ Ajouter  $r : R$  sur la ligne  $i$  (avec  $R$  la règle)

  - Si l'item est de la forme  $S \rightarrow \alpha \cdot$

⇒ Ajouter  $acc$  dans  $Action[i, \$]$

## Exemple : construction de la table

## ► Table LR

	$n$	(	)	+	$c$	$\$$	$E$	$F$
0	$s : 2$							1
1	$r : S \rightarrow F$					$acc$		
2		$s : 3$						
3	$s : 2$				$s : 6$		4	5
4			$s : 7$	$s : 8$				
5	$r : E \rightarrow F$							
6	$r : E \rightarrow c$							
7	$r : F \rightarrow n(E)$							
8					$s : 9$			
9	$r : E \rightarrow E + c$							

# Déroulement de l'analyse

## ► Initialisation

- Calculer les ensembles d'items
- Construire la table LR
- Ajouter \$ à la fin de la chaîne sur le ruban
- Initialiser l'état à celui de  $S \rightarrow \cdot \alpha$

## ► Analyse selon l'état courant et le symbole sur le ruban

- $s : i$ 
  - Empiler l'état courant
  - Avancer dans le ruban
  - Changer l'état pour  $Action[i, a]$  ( $a$  le symbole sur le ruban)
- $r : R$ 
  - Dépiler le nombre de symboles de la partie droite de  $R$
  - Changer l'état pour  $Action[i, X]$ , avec  $i$  l'état en haut de la pile
- $acc$ 
  - Accepter la chaîne

## Exemple : analyse

- ▶ Entrée :  $n(c + c)$

État	Pile	Entrée
0	$\perp 0$	$n(c + c)\$$
2	$\perp 02$	$(c + c)\$$
3	$\perp 023$	$c + c)\$$
6	$\perp 0236$	$+c)\$$
4	$\perp 0234$	$+c)\$$
8	$\perp 02348$	$c)\$$
9	$\perp 023489$	$)\$$
4	$\perp 0234$	$)\$$
7	$\perp 02347$	$\$$
1	$\perp 01$	$\$$



# Exercice

- ▶ Manche de tennis : jeux, services, échanges, points
  - $M \rightarrow JM|Jm$
  - $J \rightarrow sEj$
  - $E \rightarrow EP|P$
  - $P \rightarrow c|letP$
- ▶ Calculer les ensembles d'items
- ▶ Construire la table LR
- ▶ Analyser les expressions :  $scjm$ ,  $sletccjscjm$  ...

# Exercice : ensembles d'items

## ► Ensembles

- 0  $\{S \rightarrow \cdot M\} \cup \{M \rightarrow \cdot JM, M \rightarrow \cdot Jm, J \rightarrow \cdot sEj\}$
- 1  $\{S \rightarrow M\cdot\}$
- 2  $\{M \rightarrow J\cdot M, M \rightarrow J\cdot m\} \cup \{M \rightarrow \cdot JM, J \rightarrow \cdot sEj, M \rightarrow \cdot Jm\}$
- 3  $\{J \rightarrow s\cdot Ej\} \cup \{E \rightarrow \cdot EP, E \rightarrow \cdot P, P \rightarrow \cdot c, P \rightarrow \cdot letP\}$
- 4  $\{M \rightarrow JM\cdot\}$
- 5  $\{M \rightarrow Jm\cdot\}$
- 6  $\{J \rightarrow sE\cdot j, E \rightarrow E\cdot P\} \cup \{P \rightarrow \cdot c, P \rightarrow \cdot letP\}$
- 7  $\{E \rightarrow P\cdot\}$
- 8  $\{P \rightarrow c\cdot\}$
- 9  $\{P \rightarrow let\cdot P\} \cup \{P \rightarrow \cdot c, P \rightarrow \cdot letP\}$
- 10  $\{J \rightarrow sEj\cdot\}$
- 11  $\{E \rightarrow EP\cdot\}$
- 12  $\{P \rightarrow letP\cdot\}$

## Exercice : table LR

## ► Table LR

	<i>m</i>	<i>s</i>	<i>j</i>	<i>c</i>	<i>let</i>	<i>\$</i>	<i>M</i>	<i>J</i>	<i>E</i>	<i>P</i>
0		<i>s</i> : 3					1	2		
1	<i>r</i> : <i>S</i> → <i>M</i>					<i>acc</i>				
2	<i>s</i> : 5	<i>s</i> : 3					4	2		
3				<i>s</i> : 8	<i>s</i> : 9				6	7
4	<i>r</i> : <i>M</i> → <i>JM</i>									
5	<i>r</i> : <i>M</i> → <i>Jm</i>									
6			<i>s</i> : 10	<i>s</i> : 8	<i>s</i> : 9					11
7	<i>r</i> : <i>E</i> → <i>P</i>									
8	<i>r</i> : <i>P</i> → <i>c</i>									
9				<i>s</i> : 8	<i>s</i> : 9					12
10	<i>r</i> : <i>J</i> → <i>sEj</i>									
11	<i>r</i> : <i>E</i> → <i>EP</i>									
12	<i>r</i> : <i>P</i> → <i>letP</i>									

# Exercice : analyse

- ▶ Entrée : *scjm*

État	Pile	Entrée
0	⊥0	<i>scjm</i> \$
3	⊥03	<i>cjm</i> \$
8	⊥038	<i>jm</i> \$
7	⊥037	<i>jm</i> \$
6	⊥036	<i>jm</i> \$
10	⊥03610	<i>m</i> \$
2	⊥02	<i>m</i> \$
5	⊥025	\$
1	⊥01	\$