



Les méthodes d'analyse syntaxique Ascendante Déterministe



PresenterMedia



Méthodes d'analyse syntaxique

Déterministes

**Descendantes
Non
Déterministes**

**Descendantes
LL(1), LL(k)**

**Ascendantes
LR(1), SLR(1),
LALR(1), LR(k)**

**Descendante
parallèle**

**prédictive
avec retour
arrière**

8. Analyse ascendante déterministe



8.1 Analyse Ascendante

- ➡ L'*analyse ascendante*, contrairement à la *descendante*, consiste à **réduire la chaîne d'entrée** en un symbole initial, qui est l'**axiome**.
- ➡ Il s'agit de la construction de l'**arbre syntaxique** en partant des éléments de la phrase à analyser (les terminaux) qui sont les **feuilles de l'arbre**, vers la **racine** qui correspond à l'**axiome** → **Construction du bas vers le haut**

8. Analyse ascendante déterministe



Remarques



*Ce processus de construction de l'arbre est **une réduction**, qui se traduit par le **remplacement** de la partie droite des productions par la partie gauche.*



*Les grammaires **LR** (**ascendantes**) sont **plus générales** que les grammaires **LL** (**descendantes**) mais aussi plus compliquées.*



*l'analyseur Ascendant LR va procéder par des **réductions** et par des **décalages** du tampon d'entrée.*

8. Analyse ascendante déterministe



8.1 Analyse Ascendante



Bien que la chaîne d'entrée soit parcourue de gauche à droite, les réductions quant à elles se font de droite à gauche

Soit la grammaire:

S	→	aABe
A	→	Abc b
B	→	d

*Analysons la chaîne **abbcde**.*

8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)

entrée:

a	b	b	c	d	e	\$
---	---	---	---	---	---	----

Production
$S \rightarrow aABe$
$A \rightarrow Abc$
$A \rightarrow b$
$B \rightarrow d$

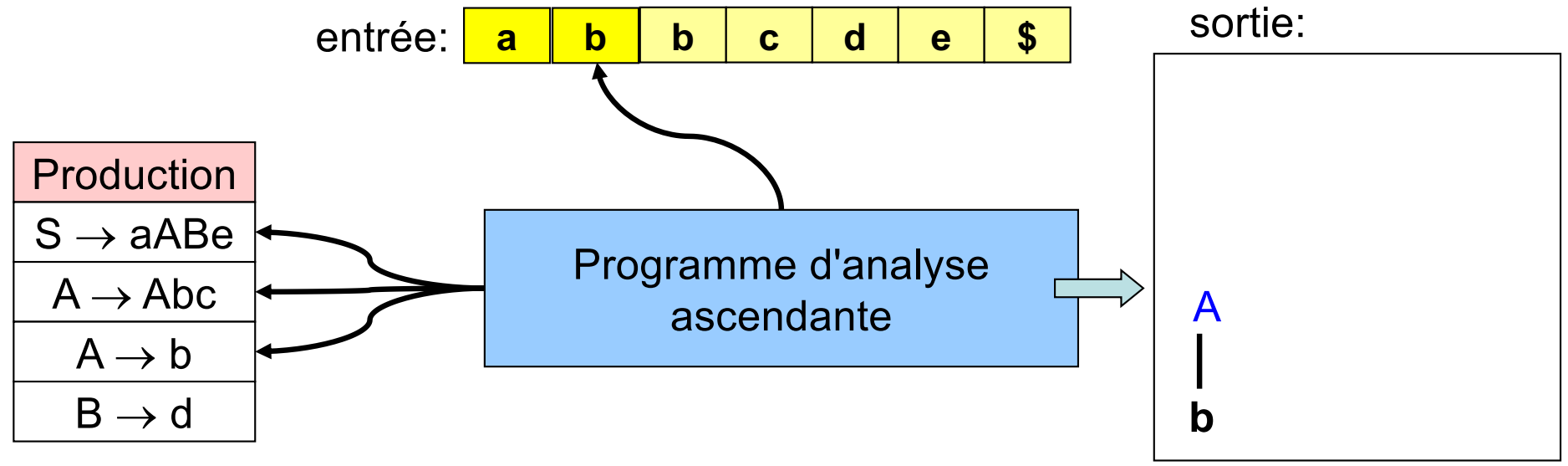
Programme d'analyse ascendante

sortie:

8. Analyse ascendante déterministe



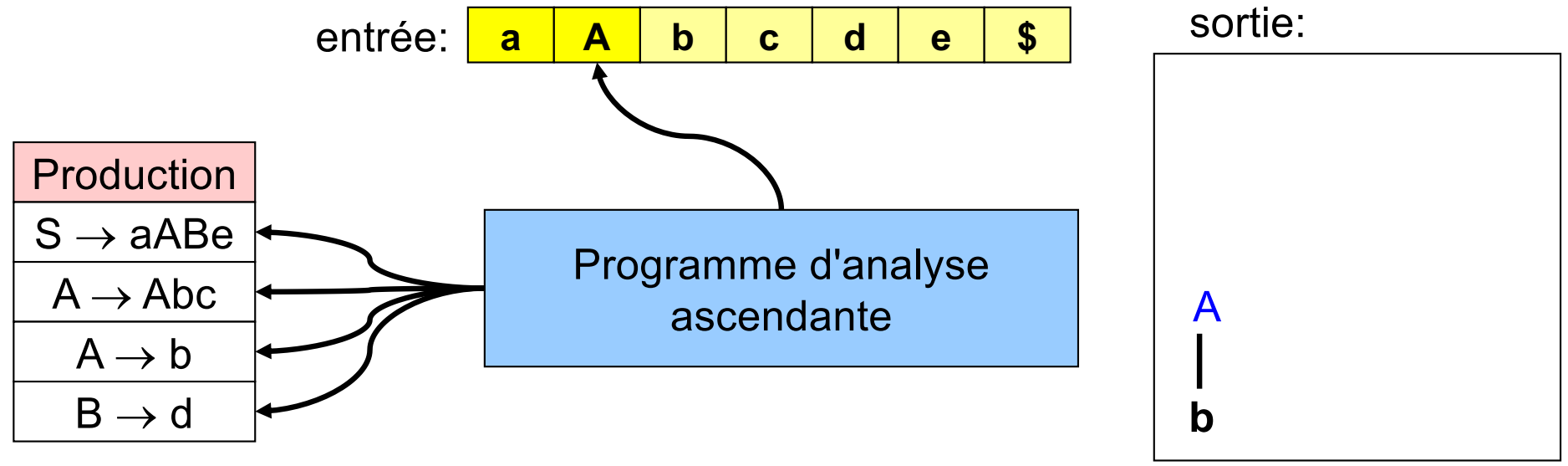
Exemple (Analyse Ascendante)



8. Analyse ascendante déterministe



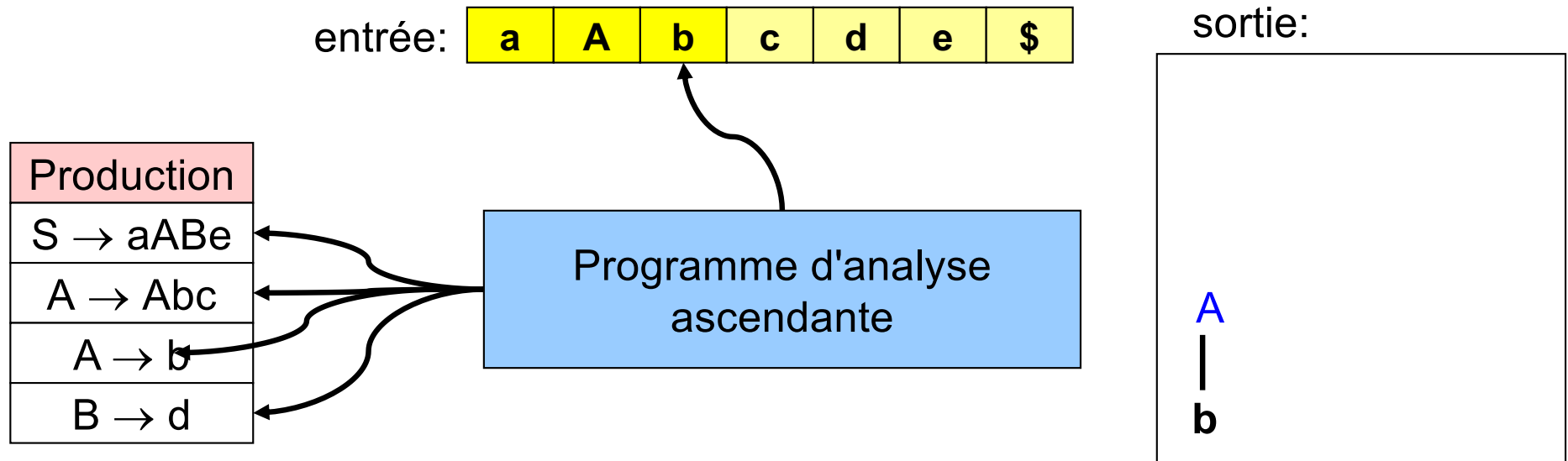
Exemple (Analyse Ascendante)



8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)



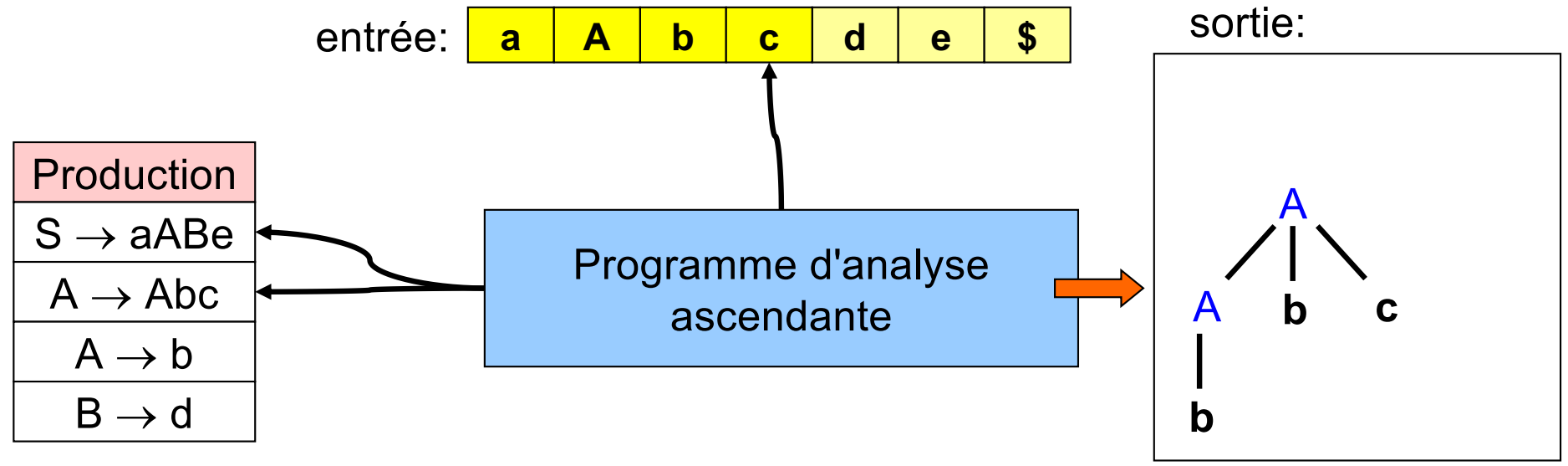
Nous ne faisons pas de réduction dans ce cas.

Un analyseur réduirait bien, mais se trouvant bloqué, ferait un retour arrière !

8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)



8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)

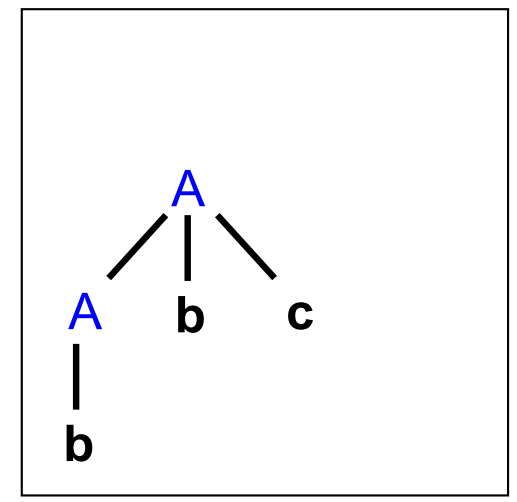
entrée:

a	A	d	e	\$
---	---	---	---	----

Production
$S \rightarrow aABe$
$A \rightarrow Abc$
$A \rightarrow b$
$B \rightarrow d$

Programme d'analyse ascendante

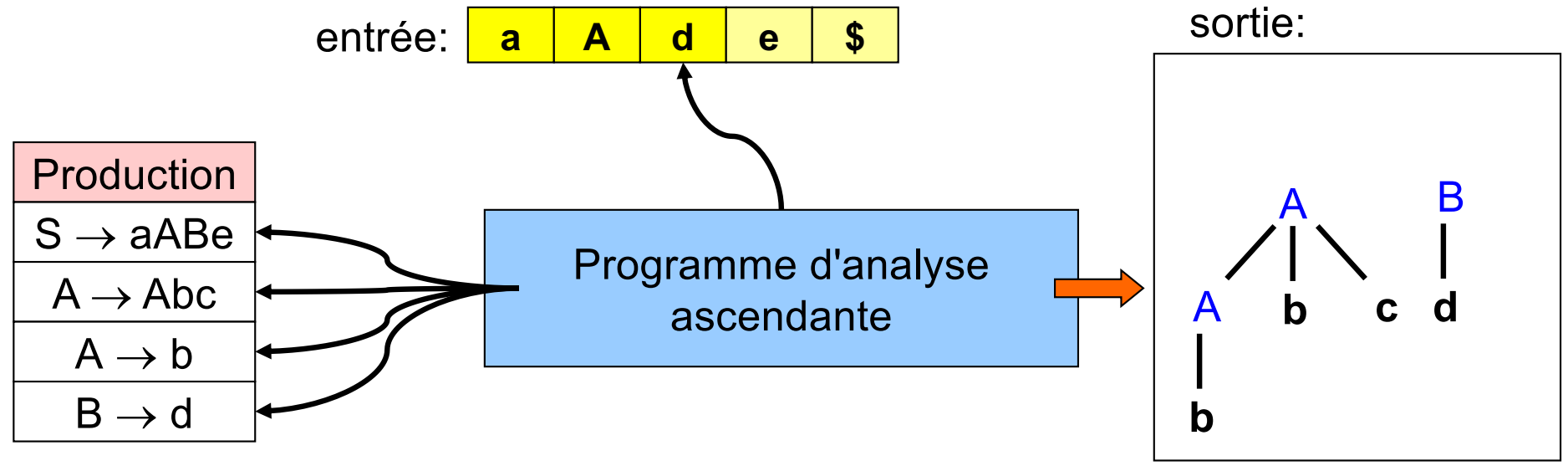
sortie:



8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)



8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)

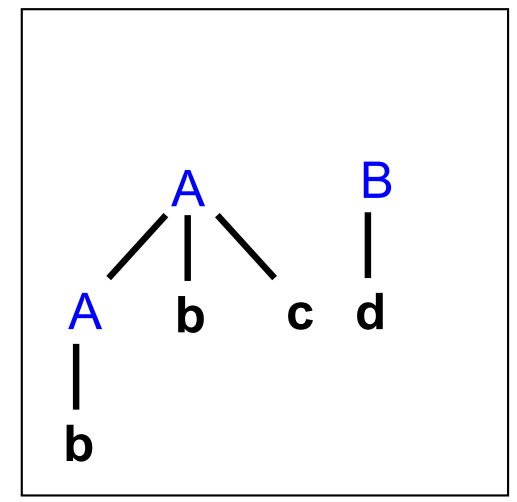
entrée:

a	A	B	e	\$
---	---	---	---	----

Production
$S \rightarrow aABe$
$A \rightarrow Abc$
$A \rightarrow b$
$B \rightarrow d$

Programme d'analyse ascendante

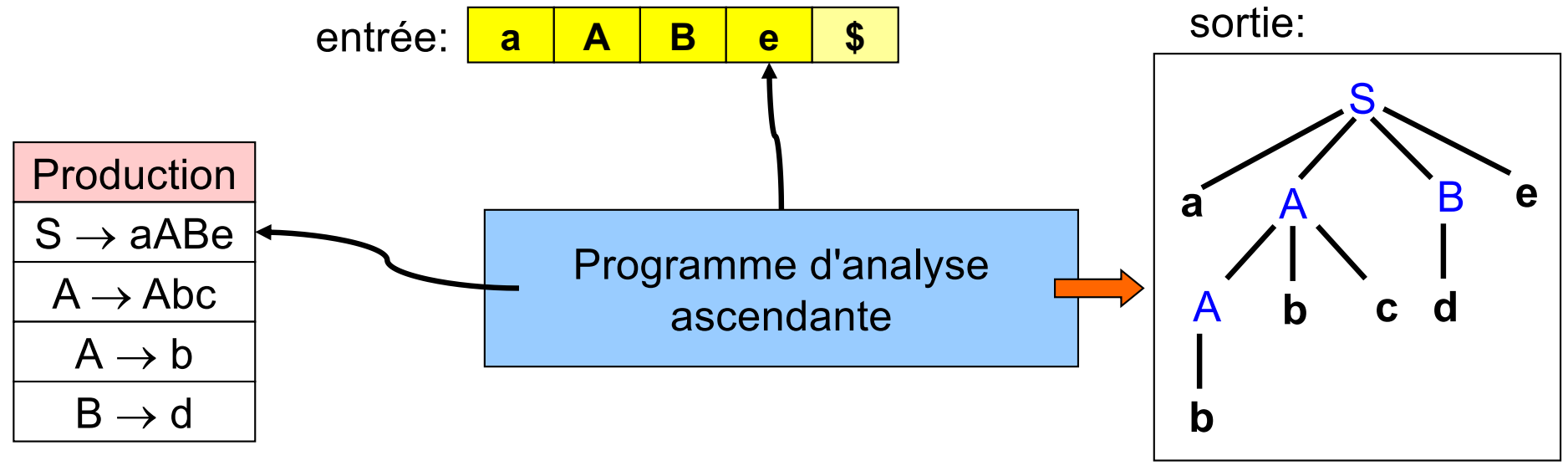
sortie:



8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)



8. Analyse ascendante déterministe



Exemple (Analyse Ascendante)

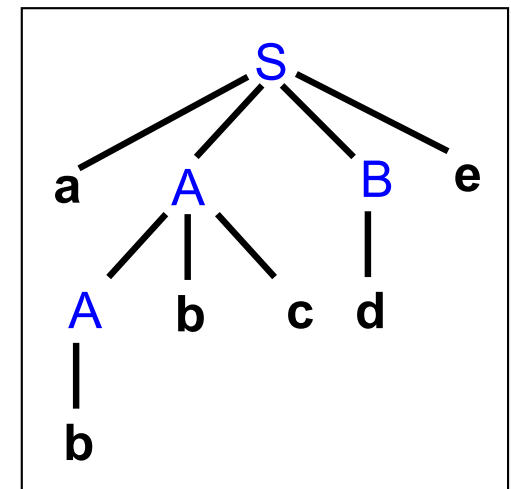
entrée:

S	\$
---	----

Production
$S \rightarrow aABe$
$A \rightarrow Abc$
$A \rightarrow b$
$B \rightarrow d$

Programme d'analyse ascendante

sortie:



Cette analyse est appelée **une analyse LR**. Elle parcourt la chaîne de gauche à droite (**Left to right**), et fait des réductions de droite à gauche (**Rightmost derivation**).

8. Analyse ascendante déterministe



Remarque

Scanner les productions pour rechercher une correspondance avec une chaîne d'entrée, et faire éventuellement des retours arrières, rend très inefficace la méthode employée dans l'exemple précédent.

Pouvons nous faire mieux





Les Méthodes d'Analyse LR



8. Analyse ascendante déterministe



8.2 Analyseur

LR :

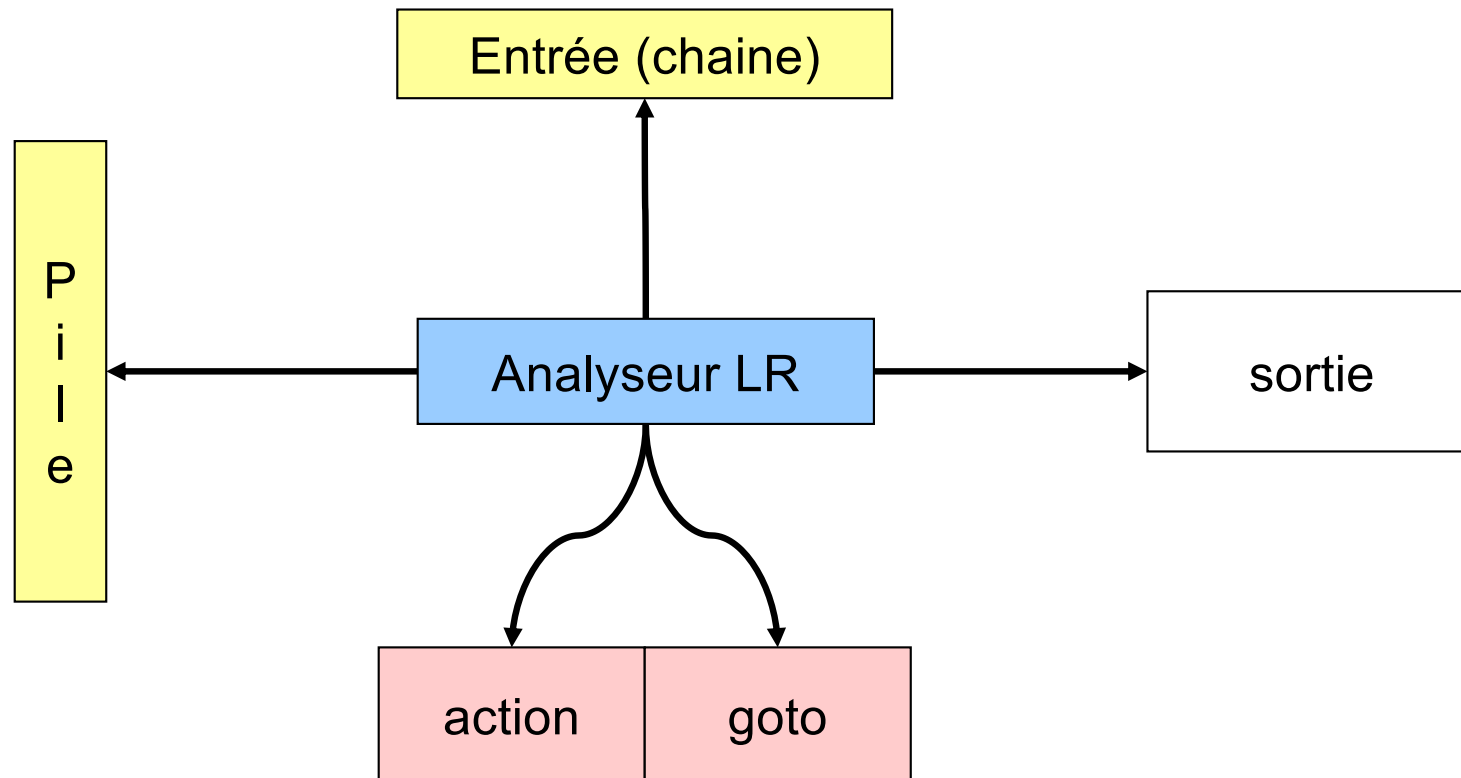
L'analyse LR est une méthode d'analyse efficace déterministe. C'est une méthode d'analyse ascendante de gauche à droite. Elle utilise une pile et une table d'analyse LR qui permet de décider de l'action à faire (décaler ou réduire).

8. Analyse ascendante déterministe

8.2 Analyseur LR :



LR :

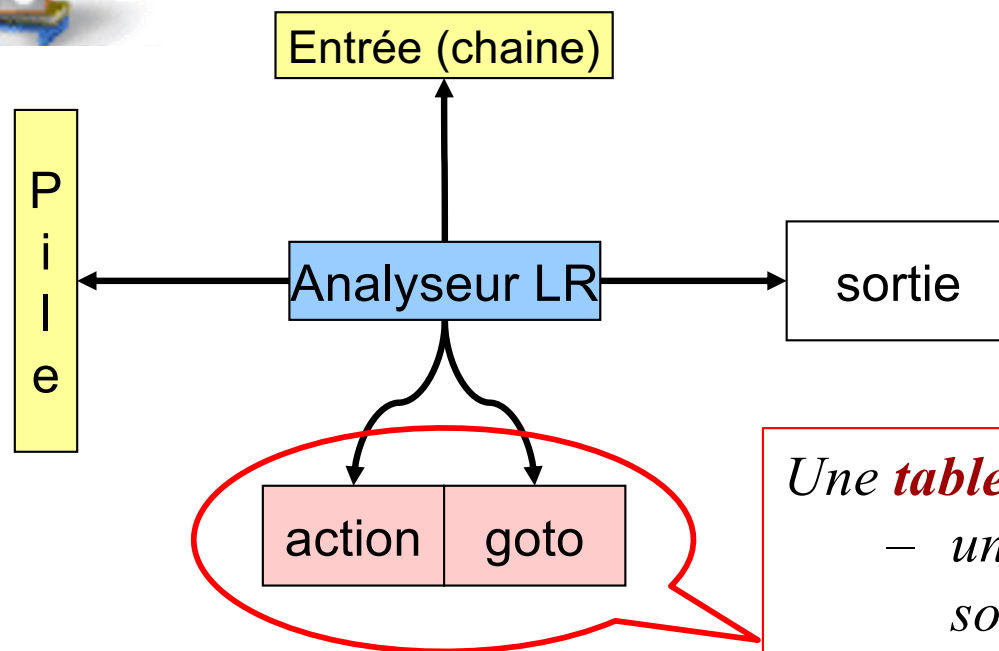


Fonctionnement des Analyseurs LR

8. Analyse ascendante déterministe

8.2 Analyseur

LR :



Une **table d'analyse** composée de deux tables:

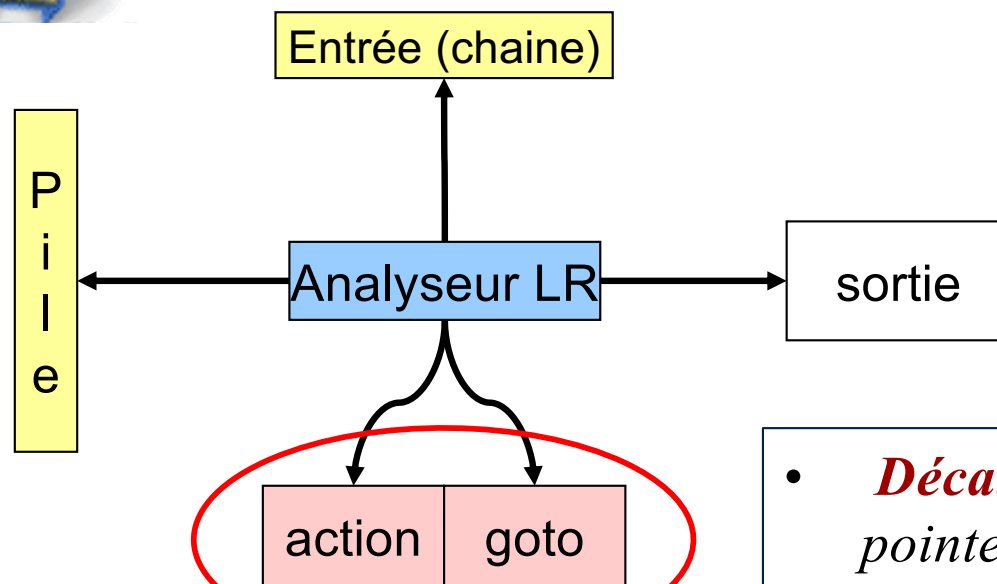
- une table **Action** qui en fonction du sommet de la pile et de la prévision, détermine s'il faut faire un **Décalage** ou une **Réduction**
- une table **Successeur (Goto)** qui détermine ce qui va être mis sur la pile

Fonctionnement des Analyseurs LR

8. Analyse ascendante déterministe

8.2 Analyseur

LR :



- **Décalage (shift)** : décaler d'une lettre le pointeur sur le mot en entrée
- **Réduction (reduce)** : réduire une chaîne (suite consécutive de terminaux et non terminaux à gauche du pointeur sur le mot en entrée) par un non-terminal en utilisant une des règles de production.

Fonctionnement des Analyseurs LR



8.2 Analyseur

LR :

8. Analyse ascendante déterministe

Table d'analyse LR (Action, Goto)

C'est une sorte d'*automate à pile* où:

- La **table Action** va nous dire ce qu'il faut faire quand on lit une lettre **a** (*terminal*) et qu'on est dans un état **i**:
 - Soit on **décale**. Dans ce cas, on empile la lettre lue et on va dans un autre état **j**. On note **Dj**
 - Soit on **réduit** par la règle de production numéro **p**, c à d qu'on remplace la chaîne en sommet de pile par le non-terminal de la partie gauche de la règle de production, et on va dans l'état **j** qui dépend du non-terminal en question. On note ça **Rj**.
 - Soit on **accepte** le mot. Ce qui sera noté **ACC**.
 - Soit c'est une **erreur**, si Case vide dans la table d'analyse.
- La **table GOTO** indique simplement à quel état aller lorsqu'un non-terminal donné a été reconnu.

8. Analyse ascendante déterministe

8.2 Analyseur

LR :

*Nous avons dans cette méthode d'analyse ascendante **2 types** de conflits:*

*1- Un conflit entre un **décalage** et une **réduction (1)**, nous avons choisi dans cet exemple de favoriser la réduction par rapport au décalage.*

*2- Un conflit entre **deux réductions (2)**, nous avons choisi, la réduction du membre droit avec le plus grand nombre de symboles.*

Si aucun de ces conflits ne risque de se produire alors la grammaire est LR(0).

8. Analyse ascendante déterministe



8.2 Analyseur LR :

3 méthodes dérivent de la méthode générale:

- *SLR (Simple LR)*: facile à implanter et non ambiguë mais adaptée à une classe limitée de grammaires,
- *LR canonique*: est la plus efficace car la plus générale mais la plus coûteuse en temps et en mémoire,
- *LALR (Look Ahead)* est intermédiaire entre les 2 méthodes précédentes et prédictive,

La dernière méthode a donné naissance à l'outil **YACC**



Construction de la Table d'analyse par La **Méthode** **des Items**



8. Analyse ascendante déterministe



*Pour construire une **table d'analyse LR** , on dispose de deux méthodes :*

- *Méthode des **contextes**.*
- *Méthode des **items**. .*

*Dans la suite du cours, nous allons nous intéresser à la méthode des **Items**.*

8. Analyse ascendante déterministe

8.3 La méthode des items



Méthode des Items: Une méthode pratique d'analyse LR basée sur la notion d'Items.

➔ Un **item** est une **production pointée** $A \rightarrow u \bullet v$, où le «•» montre l'avancement dans la reconnaissance. La partie à gauche du point **u** représente le sous-arbre de la grammaire ayant été déjà reconnu et réduit, et la partie à droite **v** représente ce qui reste à analyser.

Exemple:

L'**item** représentant le fait que l'analyseur a lu les trois premiers symboles de la partie droite de la production $A \rightarrow \text{POUET}$ s'écrit $A \rightarrow \text{POU} \bullet \text{ET}$.

8. Analyse ascendante déterministe



8.3 La méthode des items

***Méthode des Items:** Une méthode pratique d'analyse LR basée sur la notion d'Items.*

➔ Un **item** est une **production pointée** $A \rightarrow u \bullet v$, où le «•» montre l'avancement dans la reconnaissance. La partie à gauche du point **u** représente le sous-arbre de la grammaire ayant été déjà reconnu et réduit, et la partie à droite **v** représente ce qui reste à analyser.

Exemple2:

la production $A \rightarrow BCD$ peut fournir les quatre items suivants :

➤ $A \rightarrow \bullet BCD$

➤ $A \rightarrow B \bullet CD$

➤ $A \rightarrow BC \bullet D$

➤ $A \rightarrow BCD \bullet$ (dans ce cas, si pas de conflit, on peut Réduire par cette règle)

8. Analyse ascendante déterministe



8.3 La méthode des items

Notion d'items

- Pour la règle $A \rightarrow \alpha : A \rightarrow \alpha_1 \bullet \alpha_2$ avec $\alpha = \alpha_1 \alpha_2$ signifie qu'il est possible que l'on soit
 - en cours d'analyse d'une règle $A \rightarrow \alpha$,
 - après l'analyse de α_1 (α_1 est sur la pile)
 - avant l'analyse de α_2

2 types d'items sont particuliers :

- $A \rightarrow \bullet \alpha$ qui prédit que l'on peut débuter une analyse utilisant la règle $A \rightarrow \alpha$
- $A \rightarrow \alpha \bullet$ qui reconnaît que l'on a terminé l'analyse d'une règle $A \rightarrow \alpha$ (et détermine, s'il n'y a pas de conflit, que l'on peut faire le réducteur correspondant)

8. Analyse ascendante déterministe



8.3 La méthode des items

➡ *La méthode des items repose sur la construction des **ensembles des items de la grammaire.***

➡ *Pour la construction des items d'une grammaire, nous avons besoin de deux fonctions :*

❑ ***La fermeture** d'un ensemble d'items: opération de base permettant de construire un ensemble d'items*

❑ *La fonction **GOTO.***

8. Analyse ascendante déterministe

8.3 .1 La fermeture d'un ensemble d'items



La fermeture d'un ensemble d'items est constituée de l'union de la fermeture de chacun de ses items. L'opération de fermeture permet d'obtenir l'ensemble des possibilités sur l'état de l'analyse .

Fermeture d'un ensemble d'items I :

- 1- Mettre chaque item de I dans Fermeture(I)
- 2- Pour chaque item i de Fermeture(I) de la forme $A \rightarrow \alpha \cdot B \beta$
Pour chaque production $B \rightarrow \gamma$
rajouter l'item $B \rightarrow \cdot \gamma$ dans Fermeture (I)
- 3- Recommencer 2 jusqu'à ce qu'on n'ajoute rien de nouveau

8. Analyse ascendante déterministe



Exemple (Fermeture)

Exemple1: considérons la grammaire augmentée des expressions :

$$\left\{ \begin{array}{l} E' \rightarrow E \\ E \rightarrow E+T / T \\ T \rightarrow T*F / F \\ F \rightarrow (E) / id \end{array} \right.$$

I_0 est le premier ensemble d'items formé de l'unique Item $[E' \rightarrow \bullet E]$ (où E' est le nouvel axiome de la grammaire augmentée) ainsi que sa fermeture:

I_0 :

$$\left. \begin{array}{l} E' \rightarrow \bullet E \# \\ E \rightarrow \bullet E+T / \bullet T \\ T \rightarrow \bullet T*F / \bullet F \\ F \rightarrow \bullet (E) / \bullet id \end{array} \right\} \begin{array}{l} \text{Noyau} \\ \text{Fermeture de } [E' \rightarrow \bullet E] \end{array}$$

8. Analyse ascendante déterministe



Exemple (Fermeture)

→ Trouver les *autres états* (ensembles d'items accessibles) et les *transitions* entre eux

- En partant de l'état initial (I_0), on déterminera tous les états qui peuvent être atteints en une étape.
- Les *transitions possibles* pour un ensemble d'items (un état) peuvent être trouvés en regardant les *symboles* (terminaux et non-terminaux) qui se trouvent *juste après le point*.

8. Analyse ascendante déterministe



Exemple (Fermeture)

- Dans le cas de l'ensemble d'items I_0 de l'exemple précédent , ce sont les terminaux '**id**' et '**(**' et les non-terminaux **E**, **T** et **F** qui se trouvent juste après le point.

$E' \rightarrow \bullet E$
 $E \rightarrow \bullet E+T / \bullet T$
 $T \rightarrow \bullet T^* F / \bullet F$
 $F \rightarrow \bullet (E) / \bullet id$

8. Analyse ascendante déterministe

8.3.2 La Fonction GOTO



➡ L'opération transition : Pour trouver l'ensemble d'items auxquels conduit un symbole X , on suit la fonction $GOTO(I, X)$ où I est un ensemble d'items et X est un symbole de la grammaire:

1. Prendre l'ensemble S de tous les items appartenant à I où il y a un point en face de ce symbole X
2. Pour chaque item dans S , déplacer le point à la droite de X .
3. Compléter par la *fermeture* de l'ensemble des items obtenus.

8. Analyse ascendante déterministe

8.3.2 La Fonction GOTO



➔ $GOTO(I, X)$ est définie comme la *fermeture* de l'ensemble de tous les items $A \rightarrow \alpha X \bullet \beta$ tels que $A \rightarrow \alpha \bullet X \beta$ appartienne à I

$$GOTO(I, X) = fermeture([A \rightarrow \alpha X \bullet \beta])$$

Exemple2:

$E' \rightarrow E$

$E \rightarrow E + T / T$

$T \rightarrow T * F / F$

$F \rightarrow (E) / id$

Si I est l'ensemble des deux Items

$[E' \rightarrow E \bullet]$

$[E \rightarrow E \bullet + T]$

alors

$$GOTO(I, +) = I'$$

$E \rightarrow E + \bullet T$

$T \rightarrow \bullet T * F$

$T \rightarrow \bullet F$

$F \rightarrow \bullet (E)$

$F \rightarrow \bullet id$

8. Analyse ascendante déterministe

8.3.3 Construction des ensembles d'items

➔ Construction des ensembles d'items:

L'algorithme décrit ci-dessous représente la méthode de construction de la **collection canonique d'ensembles d'items LR(0)** pour une grammaire augmentée G' :

Procédure Items(G') ;

Début $C := \{ \text{fermeture} (\{[S' \rightarrow S]\}) \}$;

Répéter Pour chaque ensemble d'items I de C et pour chaque symbole de la grammaire X tel que $GOTO(I, X)$ soit non vide et non encore dans C **faire**

Ajouter $GOTO(I, X)$ à C **Jusqu'à** ce qu'aucun nouvel ensemble d'items ne puisse plus être ajouté à C

Fin



Les analyseurs LR(0)



8. Analyse ascendante déterministe

8.4 Analyseurs LR(0)



→ Dans les analyseurs LR(0), aucune prévision n'est nécessaire.

Exemple (Constructions des items LR(0))

Soit la grammaire G augmentée suivante:

$E' \rightarrow E$

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

*Déterminer la collection canonique
d'ensembles d'items LR(0)
correspondante ?*

8. Analyse ascendante déterministe



Exemple (Constructions des items LR(0))

I_0 :
 $E' \rightarrow \cdot E$
 $E \rightarrow \cdot E+T / \cdot T$
 $T \rightarrow \cdot T^* F / \cdot F$
 $F \rightarrow \cdot (E) / \cdot id$

$I_1 : GOTO(I_0, E)$
 $E' \rightarrow E \cdot$
 $E \rightarrow E \cdot +T$

$I_2 : GOTO(I_0, T)$
 $E \rightarrow T \cdot$
 $T \rightarrow T \cdot * F$

$I_3 : GOTO(I_0, F)$
 $T \rightarrow F \cdot$

$I_4 : GOTO(I_0, ()$
 $F \rightarrow (\cdot E)$
 $E \rightarrow \cdot E+T / \cdot T$
 $T \rightarrow \cdot T^* F / \cdot F$
 $F \rightarrow \cdot (E) / \cdot id$

$I_5 : GOTO(I_0, id)$
 $F \rightarrow id \cdot$

$I_6 : GOTO(I_1, +)$
 $E \rightarrow E+ \cdot T$
 $T \rightarrow \cdot T^* F / \cdot F$
 $F \rightarrow \cdot (E) / \cdot id$

$I_7 : GOTO(I_2, *)$
 $T \rightarrow T^* \cdot F$
 $F \rightarrow \cdot (E) / \cdot id$

$I_8 : GOTO(I_4, E)$
 $F \rightarrow (E \cdot)$
 $E \rightarrow E \cdot +T$

$GOTO(I_4, T) = I_2$
 $GOTO(I_4, F) = I_3$
 $GOTO(I_4, () = I_4$
 $GOTO(I_4, id) = I_5$

$I_9 : GOTO(I_6, T)$
 $E \rightarrow E+T \cdot$
 $T \rightarrow T \cdot * F$

$GOTO(I_6, F) = I_3$
 $GOTO(I_6, () = I_4$
 $GOTO(I_6, id) = I_5$

$I_{10} : GOTO(I_7, F)$
 $T \rightarrow T^* F \cdot$

$GOTO(I_7, () = I_4$
 $GOTO(I_7, id) = I_5$

$I_{11} : GOTO(I_8,))$
 $F \rightarrow (E) \cdot$

$GOTO(I_8, +) = I_6$
 $GOTO(I_9, *) = I_7$



Construction de l'automate caractéristique LR(0)



8. Analyse ascendante déterministe

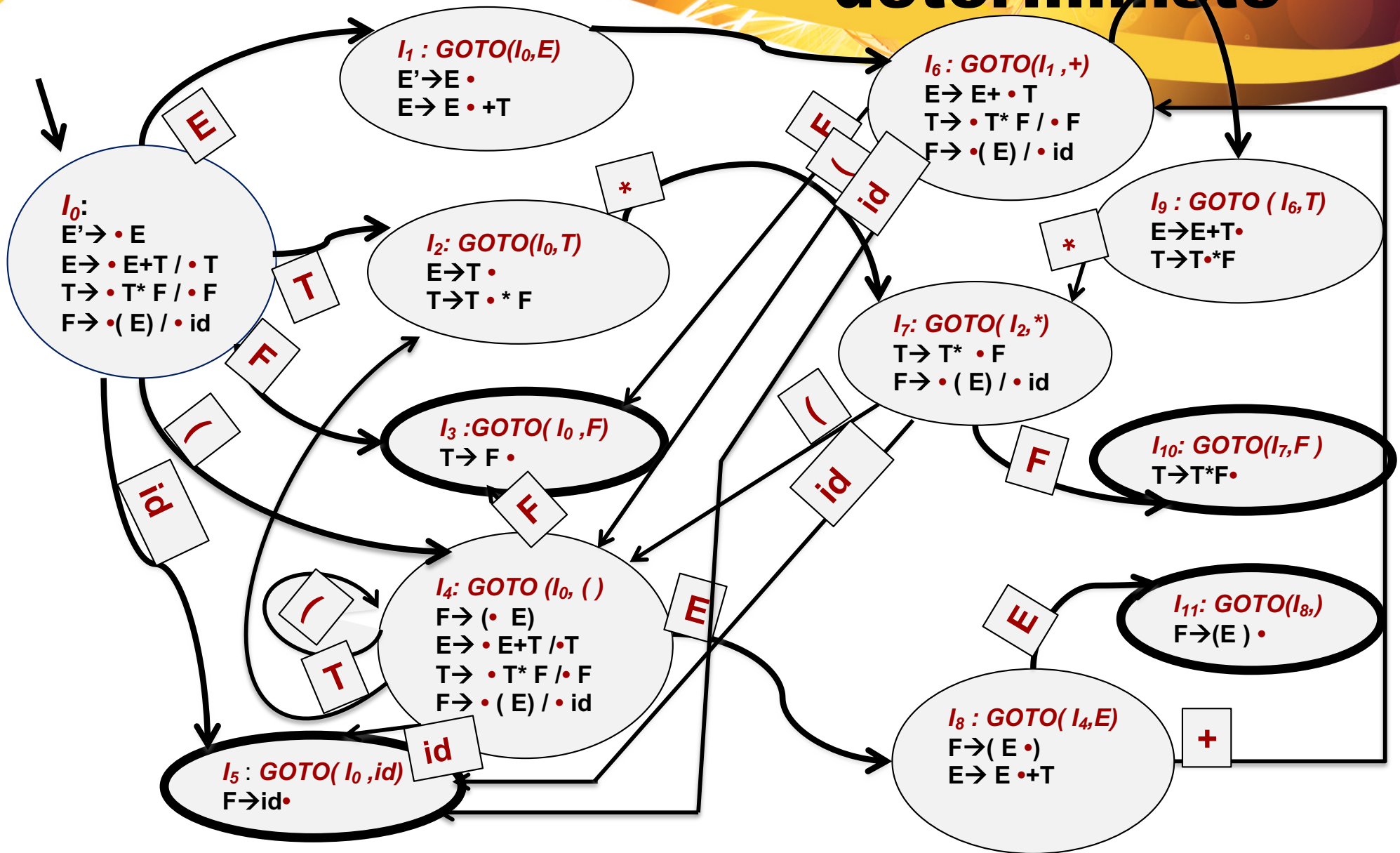


8.4.1 Construction de l'automate caractéristique LR(0)

- Chaque *ensemble d'items* représente un *état* de l'automate.
- I_0 est l'*état initial* de l'automate
- *Les transitions* correspondent à tout symbole (terminal ou non) se situant directement à droite de \bullet dans un ensemble d'items. C'est la fonction $GOTO(I, X)$.

Exemple (Automate LR(0))

8. Analyse ascendante déterministe





Exemple (Automate LR(0))

8. Analyse ascendante déterministe

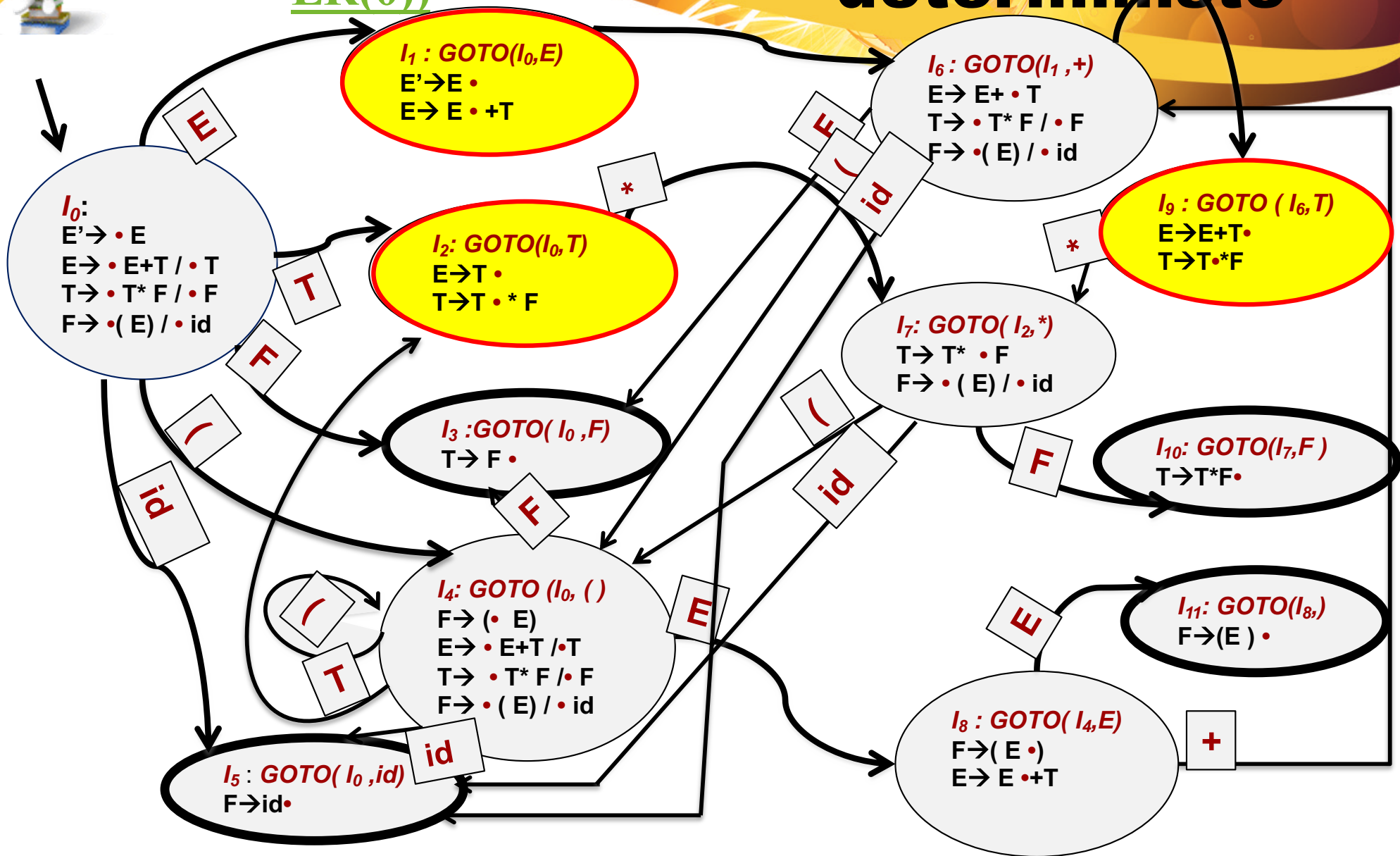
D'après l'automate précédent, Est-ce que la grammaire correspondante est une grammaire LR(0)?

Non, elle n'est pas LR(0) car il y a des conflits Décalage/Réduction dans les ensembles d'items: I_1 , I_2 et I_9 .

Remarque: Les états I_3 , I_5 , I_{10} et I_{11} sont des états finaux. Ils correspondent à des réductions sans conflits.

8. Analyse ascendante déterministe

Exemple (Automate LR(0))



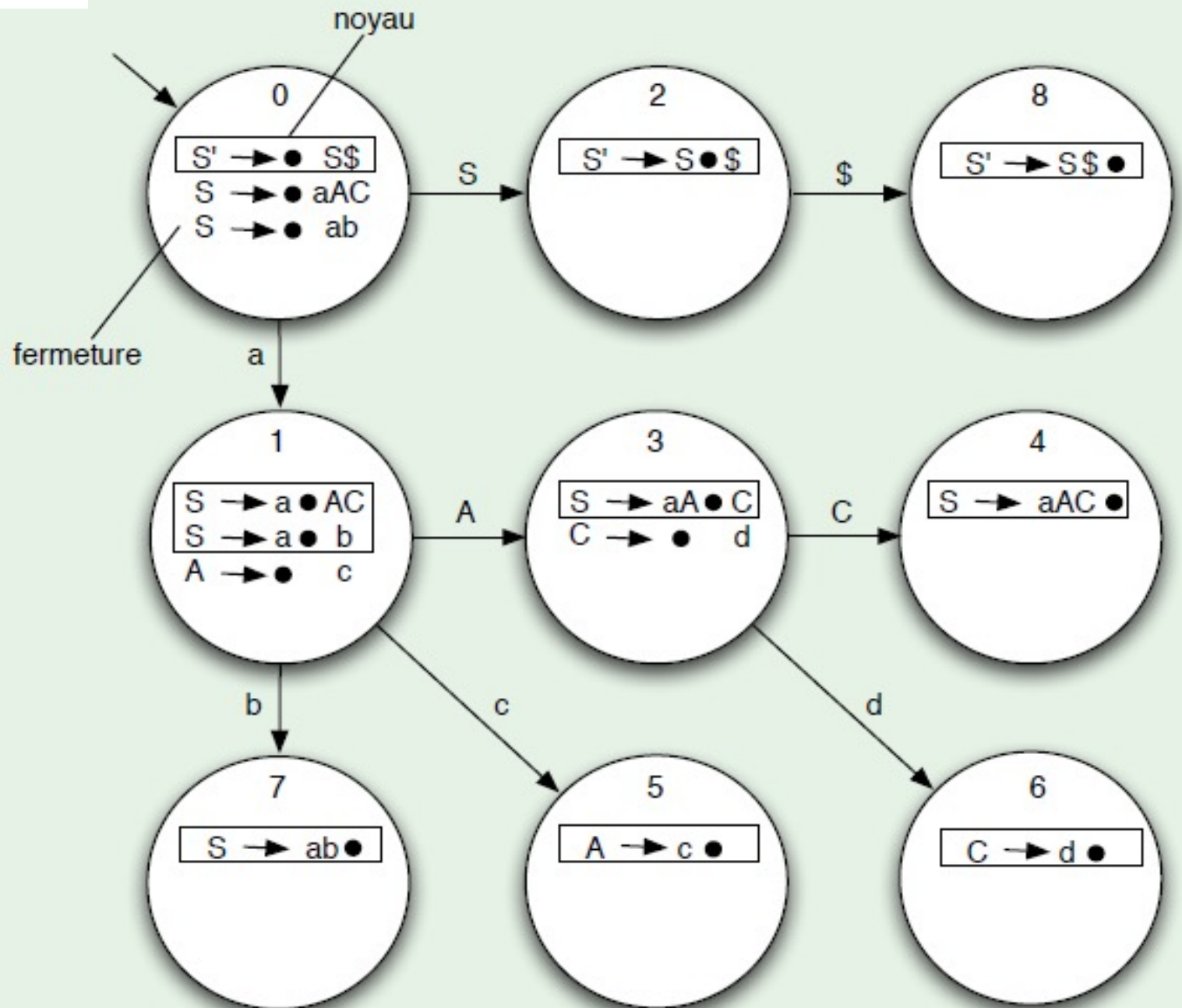
Exercice d'application

Construire l'automate caractéristique LR(0) de la grammaire suivante:

S'	\rightarrow	$S\$$	(0)
S	\rightarrow	aAC	(1)
S	\rightarrow	ab	(2)
A	\rightarrow	c	(3)
C	\rightarrow	d	(4)



Exercice d'application : Automate LR(0)





Construction de la table d'analyse LR(0)





8.4.2 Table d'Analyse LR(0)

8. Analyse ascendante déterministe

La table d'analyse est composée de deux parties **ACTION** (les colonnes sont les *terminaux* en plus de #) et **GOTO** (les colonnes sont les *non-terminaux*).

→ Construction de la table Action:

1. Construire $\mathbf{C} = \{ I_0, I_1, \dots, I_n \}$ la collection des ensembles d'items LR(0)
2. $\forall \mathbf{I}_i \in \mathbf{C}$
 - si $[A \rightarrow \alpha \bullet] \in \mathbf{I}_i$ alors : Action[\mathbf{I}_i] == Reduction _{i}
/* où $A \rightarrow \alpha$ est la règle i */
 - si $[A \rightarrow \alpha \bullet a\beta] \in \mathbf{I}_i$ alors : Action[\mathbf{I}_i] == Decalage
 - si $[S' \rightarrow S\# \bullet] \in \mathbf{I}_i$ alors : Action[\mathbf{I}_i] == Accept



8.4.2 Table d'Analyse LR(0)

8. Analyse ascendante déterministe

→ Construction de la table GOTO

Pour Chaque $I_j = GOTO(I_i, X)$ où X est un non terminal, alors il s'agit d'un changement d'état de I_i vers I_j après la reconnaissance de X : $T[I_i, X] \leftarrow I_j$.



Automate caractéristique LR(0)

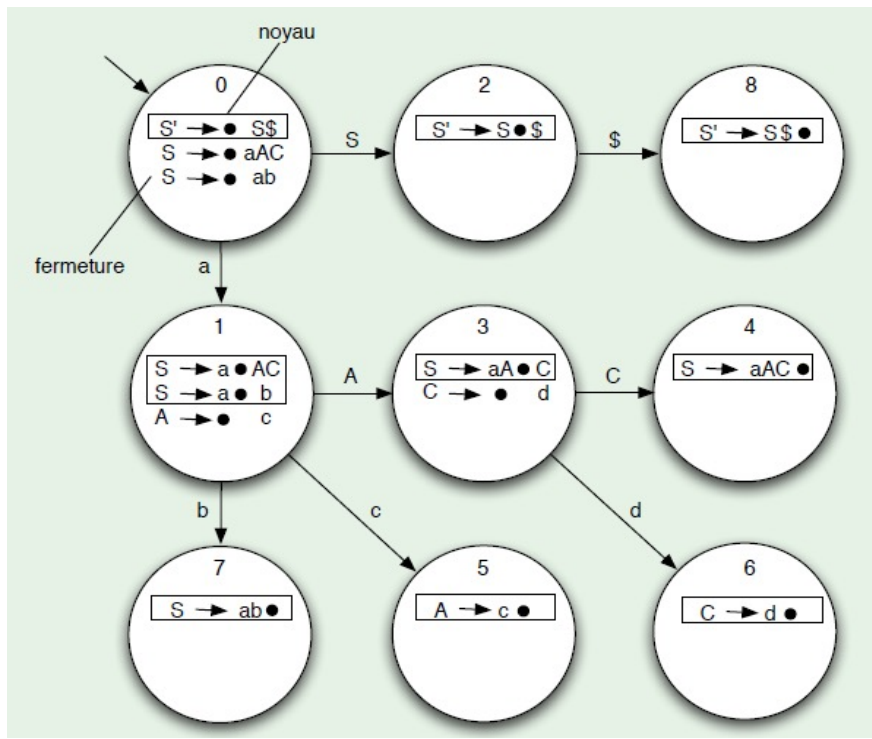


Table d'Analyse LR(0)

	a	b	c	d	\$	A	C	S
0	D1							2
1		D7	D5			3		
2					Acc			
3				D6			4	
4	R1	R1	R1	R1	R1			
5	R3	R3	R3	R3	R3			
6	R4	R4	R4	R4	R4			
7	R2	R2	R2	R2	R2			

Remarque

	a	b	c	d	\$	A	C	S
0	D1							2
1		D7	D5			3		
2					Acc			
3				D6			4	
4	R1	R1	R1	R1	R1			
5	R3	R3	R3	R3	R3			
6	R4	R4	R4	R4	R4			
7	R2	R2	R2	R2	R2			

→ Les réductions arrivent **indépendamment du symbole** suivant du tampon d'entrée. C'est la raison pour laquelle ce sont les **tables d'analyse syntaxique LR(0)** : elles ne font aucune anticipation (c'est-à-dire qu'elles ne regardent aucun symbole d'avance) pour décider quelle réduction faire.



Exercice d'application : Analyse de la chaîne acd\$ de l'exercice précédent

S'	\rightarrow	$S\$$	(0)
S	\rightarrow	aAC	(1)
S	\rightarrow	ab	(2)
A	\rightarrow	c	(3)
C	\rightarrow	d	(4)

	a	b	c	d	\$	A	C	S
0	D1							2
1		D7	D5			3		
2					Acc			
3				D6			4	
4	R1	R1	R1	R1	R1			
5	R3	R3	R3	R3	R3			
6	R4	R4	R4	R4	R4			
7	R2	R2	R2	R2	R2			

File	Chaine	Action
0	acd\$	D1
0a1	cd\$	D5
0a1 <u>c</u> 5	d\$	R3
0a(1A)3	d\$	D6
0a1A3 <u>d</u> 6	\$	R4
0a1A(3C)4	\$	R1
(0S)2	\$	Acceptée