Chapitre 6 : Analyse ascendante

6.2 Analyse LR(0)

Définition. Un item LR(0) (item en abrégé) d'une grammaire G est une production de G avec un point repérant une position de sa partie droite.

Exemple

A → XYZ ⇒
A → •XYZ
A → X•YZ
A → XY•Z
A → XYZ•
La production A → ε fournit uniquement l'item A → •

Intuitivement, un item indique la « quantité » de partie droite qui a été reconnue, à un moment donné, au cours du processus d'analyse.

Exemple

A → XYZ indique qu'on espère avoir en entrée une chaîne dérivable depuis XYZ

A → X•YZ indique que nous venons de voir en entrée une chaîne dérivée de X et que nous espérons maintenant voir une chaîne dérivée de YZ.

Exemple

(1) E → E+T ⇒ E → •E+T, E → E•+T, E → E+•T, **E → E+T•** (manche reconnu)
(2) E → T ⇒ E → •T , **E → T•** (manche reconnu)
(3) T→ T * F ⇒ T→ •T * F, T→ T• * F, T→ T *• F, **T→ T * F•** (manche reconnu)
(4) T → F ⇒ T →• F , **T → F•** (manche reconnu)
(5) F → (E) ⇒ F → • (E) , F → (•E) , F → (E•) , **F → (E)** • (manche reconnu)
(6) F → id ⇒ F → • id , **F → id•** (manche reconnu)

L'idée centrale de la méthode SLR est de construire, tout d'abord, à partir de la grammaire, un automate fini déterministe pour reconnaître les préfixes viables.

- Les items sont regroupés en ensembles qui constituent les états de l'analyseur SLR

- Les items peuvent être vus comme les états d'un NFA reconnaissant les préfixes viables, et le "regroupement" est exactement la construction des sous ensembles présentée lors de la conversion NFA → DFA.

- une collection d'ensembles d'items LR(0), que nous appellerons collection LR(0) canonique fournit la base de la construction des analyseurs SLR.

- Pour construire la collection LR(0) canonique pour une grammaire, nous définissons une grammaire augmentée et deux fonctions Fermeture et Transition.

Définition: Si G est une grammaire d'axiome S, alors G', la grammaire augmentée de G, est G avec un nouvel axiome S' et une nouvelle production S'→ S.

But: indiquer à l'analyseur quand il doit arrêter l'analyse et annoncer que la chaîne d'entrée a été acceptée.

L'opération fermeture.

Si I est un ensemble d'items pour une grammaire G, Fermeture de I est l'ensemble d'items construit à partir de I par les deux règles:

1. Initialement, placer chaque item de I dans Fermeture (I)

2. Si $A \rightarrow \alpha \bullet B\beta$ est dans fermeture (I), et $B \rightarrow \gamma$ est une production, ajouter l'item $B \rightarrow \bullet\gamma$ à Fermeture (I), s'il ne s'y trouve pas déjà. Nous appliquons cette règle jusqu'à ce qu'aucun nouvel item ne puisse plus être ajouté à Fermeture de I.

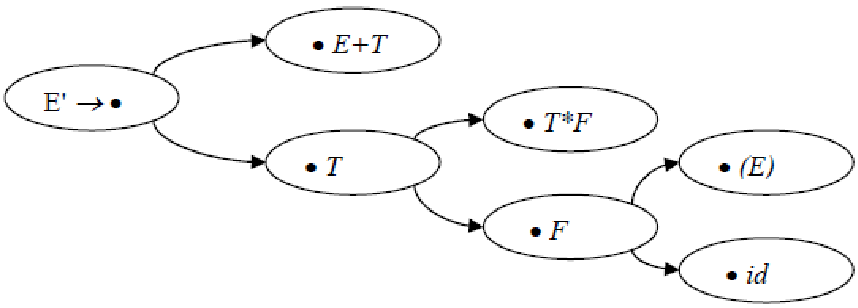
Intuitivement, si $A \rightarrow \alpha \bullet B\beta$ est dans Fermeture (I) cela indique que à un certain point du processus d'analyse, nous pourrions voir se présenter dans l'entrée, une sous-chaîne dérivable depuis $B\beta$ comme entrée. Si $B \rightarrow \gamma$ est une production, nous supposons que nous pourrions également voir, à ce moment là une chaîne dérivable depuis γ.

Autrement : soit $A \rightarrow \alpha \bullet B\beta$ avec B un non terminal, espérer avoir une chaîne dérivable depuis B c'est espérer avoir toute chaîne dérivable à partir de toute partie droite $B \rightarrow \gamma$.

Exemple

Considérons la grammaire augmentée des expressions:

E' → E
E → E+T |T
T → T*F|F (5)
F → (E) |id
Si I est l'ensemble formé de l'unique item {[E'→ • E]}, alors Fermeture (I) contient les items:



on a donc

E' → •E E → •E+T E → •T T → •T*F	ici E' → •E est placé dans Fermeture (I) par la règle (1). Comme il y'a un E immédiatement à droite d'un
T → •F F → • (E) F → • id	point, par la règle (2) nous ajoutons les E-productions avec des points aux extrémités gauches. C'est -à dire E→•E+T et E → •T, nous avons maintenant un T immédiatement à droite d'un point, nous ajoutons donc T → •T*F et T → •F puis le F à droite d'un point implique l'ajout de F→•(E) et F → •id. On ne peut plus alors ajouter aucun autre item à Fermeture (I) par la règle (2).

L'opération Transition.

Transition (I,X) où I est un ensemble d'items et X est un symbole de la grammaire.

Transition(I,X) est définie comme la fermeture de l'ensemble de tous les items [A→αX•β] tel que [[A→α•Xβ] appartienne I. Intuitivement, si I est l'ensemble d'items qui sont valides pour un préfixe viable donné γ, alors Transition(I,X) est l'ensemble des items qui sont valides pour le préfixe viable γX.

Exemple

Si I est l'ensemble des deux items {[E'→E•], [E→ E•+T], alors Transition(I,+) consiste en:

E → E+•T *T → •T*F* *T → •F* *F → • (E)* *F → • id*	Nous calculons Transition(I,+) en recherchant dans I, les items ayant + immédiatement à droite du point, E'→ E• ne convient pas, mais E →E•+T répond au critère. Nous faisons franchir au point + afin d'obtenir {[E→ E+•T]}, puis nous calculons Fermeture sur cet ensemble.

Algorithme Construction des ensembles d'items

Voici l'algorithme pour construire C, **la collection canonique d'ensembles d'items LR(0)** pour la grammaire augmentée G'.

Procedure Items (G');
début
C := {fermeture ({[S' → •S]})}
répéter
pour chaque ensemble d'items I de C et pour chaque symbole de la grammaire X tel que Transition(I,X) soit non vide et non encore dans C
ajouter Transition (I,X) à C
fin pour
jusqu'à ce qu'aucun nouvel ensemble d'items ne puisse être plus ajouté à C

fin

Exemple

La collection canonique d'ensembles d'items LR(0) pour la grammaire augmentée : G' (V, Σ, R, S), avec V = {E', E, T, F, +,*,(,), id}, Σ= {+,*,(,),id}, S = E' et R définit par les règles de production suivantes:

E' → E

E → E+T |T (5)

T → T*F|F

F → (E) |id

est présentée ci dessous:

I0 : {fermeture ({[E'→ •E]})}

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

I1 := transition(I0,E)

$E' \rightarrow E\bullet$
$E \rightarrow E\bullet +T$

I2 := transition(I0,T)

$E \rightarrow T\bullet$
$T \rightarrow T\bullet *F$

I3 := Transition (I0,F)

$T \rightarrow F\bullet$

I4 := Transition (I0, ( )

$F \rightarrow (\bullet E)$
$E \rightarrow \bullet E+T$
$E \rightarrow \bullet T$
$T \rightarrow \bullet T*F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

Transition (I0,+) = ∅
Transition (I0,*) =∅
Transition (I0,)) = ∅

I5 := Transition (I0, id)
$F \rightarrow id\bullet$
I6 := transition (I1,+)

$E \rightarrow E+\bullet T$
$T \rightarrow \bullet T*F$
$T \rightarrow \bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I7 := Transition (I2, *)

$T \rightarrow T*\bullet F$
$F \rightarrow \bullet (E)$
$F \rightarrow \bullet id$

I8 := Transition (I4, E)

$F \rightarrow (E\bullet)$
$E \rightarrow E\bullet +T$

I9 := transition (I6,T)

$E \rightarrow E+T\bullet$
$T \rightarrow T\bullet *F$

Transition (I4,F) = I3
Transition (I4, ( ) = I5

I10 = Transition (I7, F)

$T \rightarrow T*F\bullet$

I11 = Transition (I8,))

$F \rightarrow (E)\bullet$

La fonction de Transition pour cet ensemble est donnée sous la forme d'un diagramme de transition d'un automate fini déterministe D (voir figure 6.3).

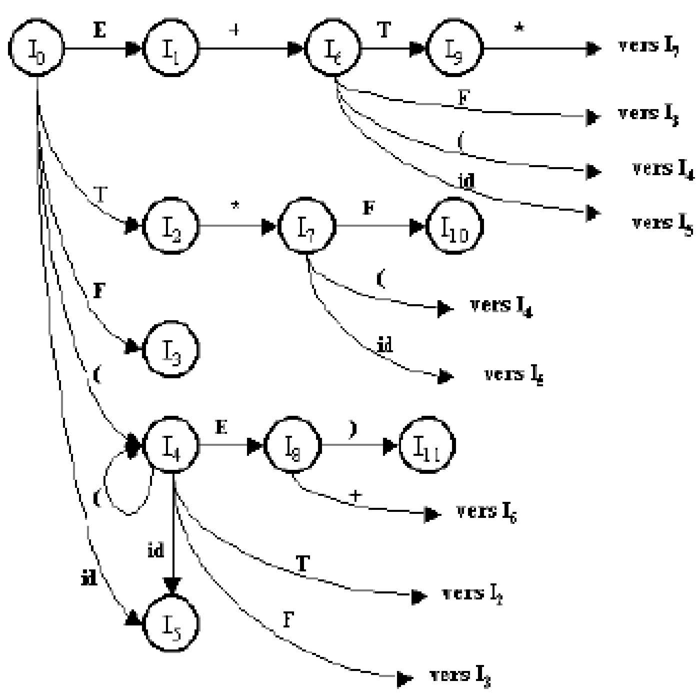


Figure 6.3 : Diagramme de transition de l'automate fini déterministe D reconnaissant les préfixes viables.