

## Chapitre 7 : Traduction dirigée par la syntaxe

### 7.1.2 Attributs hérités

#### Exemple

Soit une grammaire permettant de déclarer des entiers et des réels :

$D \rightarrow T L$   
 $T \rightarrow \text{entier} \mid \text{réel}$   
 $L \rightarrow L, \text{id} \mid \text{id}$

Dans la DDS suivante, on utilise un attribut hérité pour distribuer l'information de typage aux différents identificateurs d'une déclaration.

Productions	Règles sémantiques
$D \rightarrow T L$	$L.Type\ h := T.type$
$T \rightarrow \text{entier}$	$T.Type := \text{entier}$
$T \rightarrow \text{réel}$	$T.Type := \text{réel}$
$L \rightarrow L_1, \text{id}$	$L_1.Type\ h := L.Type\ h \quad \text{ajouterType}(\text{id.entrée}, L.Type\ h)$
$L \rightarrow \text{id}$	$\text{ajouterType}(\text{id.entrée}, L.Type\ h)$

La première règle sémantique  $L.Type\ h := T.Type$ , associée à la production  $D \rightarrow T L$ , donne à l'attribut hérité  $L.Type\ h$  la valeur de type de la déclaration (entier ou réel). Les autres règles sémantiques font descendre ce type le long de l'arbre par l'intermédiaire de l'attribut hérité  $L.Type\ h$ . La procédure `ajouterType()` est appelée pour permettre d'ajouter le type de chaque identificateur au niveau de son entrée dans la table des symboles.

En utilisant la DDS précédente, donner l'arbre syntaxique décoré de la déclaration : entier a, b, c. Avant la décoration, nous avons l'arbre syntaxique de la figure 7.3.

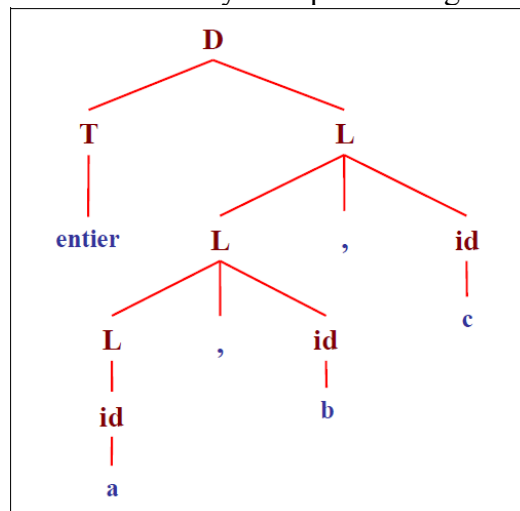


Figure 7.3. Arbre syntaxique de : entier a, b, c

Après la décoration, nous obtenons l'arbre syntaxique décoré de la figure 7.4.

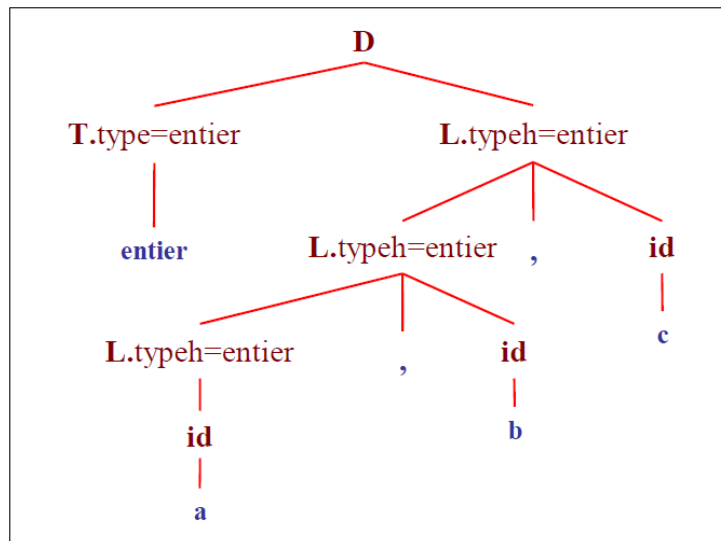


Figure 7.4. Arbre syntaxique décoré de : entier a, b, c