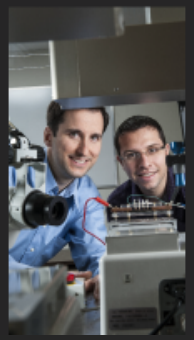





POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

LOG3210
Éléments de langage et compilateurs

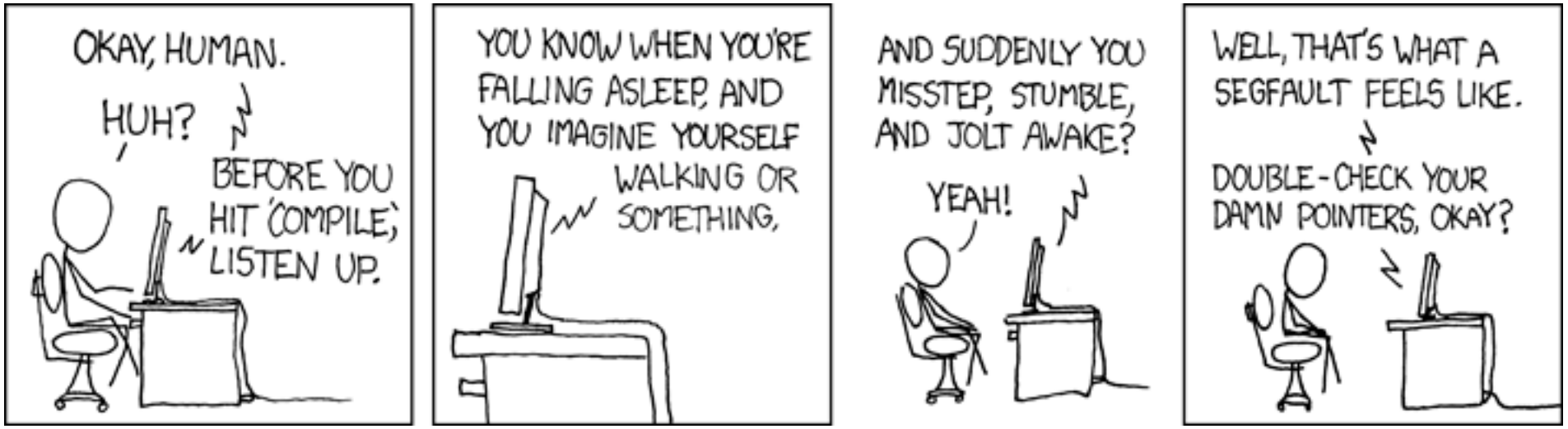
Traduction dirigée par la syntaxe

Semaine 5

PLAN

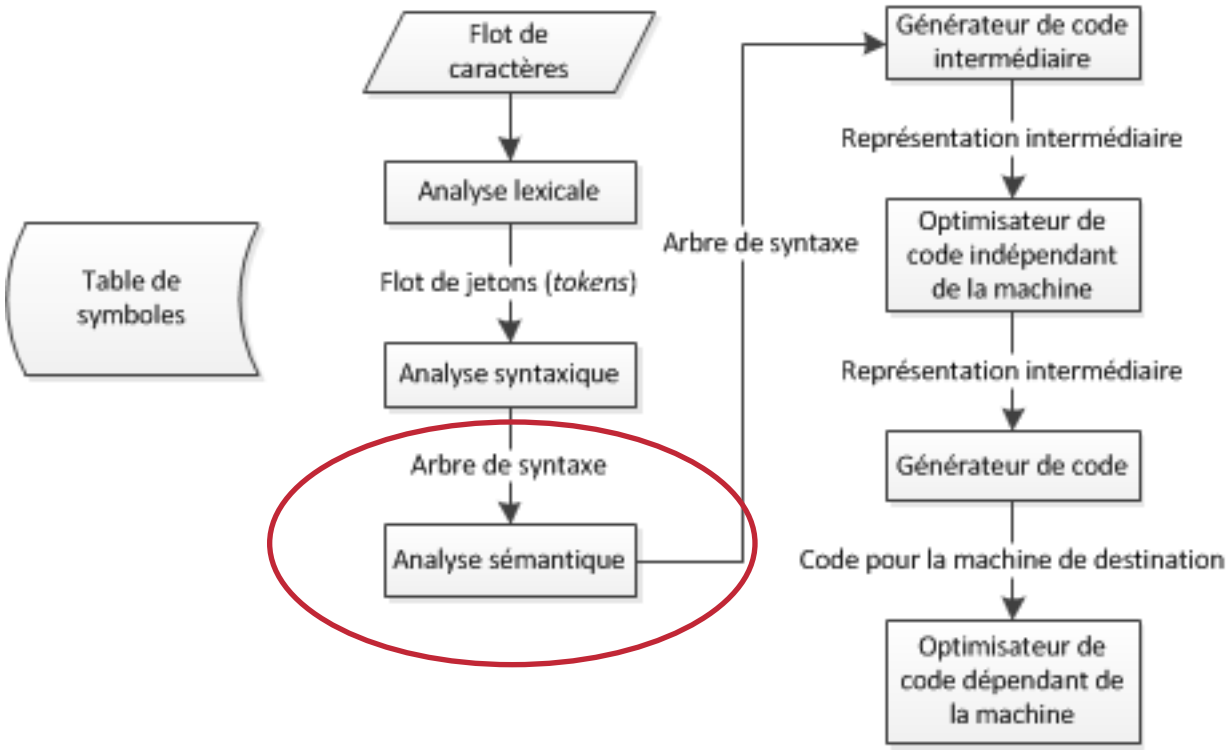
1. Chapitre 5 – Traduction dirigée par la syntaxe

Compiler Complaint



 https://xkcd.com/371/

PHASES D'UN COMPILATEUR



Analyse sémantique : Vérification ou transformation sur l'arbre de syntaxe.

ÉCOLE
POLYTECHNIQUE
MONTRÉAL

LOG3210
Cours 5

Traduction dirigée par la syntaxe

Arbres de syntaxe abstraits (AST)

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Très importants pour les environnements intégrés de développement (IDE)
- Permettent de séparer la traduction de l'analyse syntaxique
- Représentation:
  - Forme condensée d'un arbre de parsage
    - Les opérateurs et les mots clefs sont plutôt associés au nœud racine
    - Les séquences de production avec seulement un symbole non terminal à droite peuvent être compressés

AST vs arbres de parsage

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Un AST ne représente pas nécessairement chaque détail du langage
  - Par exemple, les parenthèses sont implicites
  - Un if-else pourrait être représenté différemment
- L'AST est, à toutes fins pratiques, insensible à la grammaire, ce qui n'est pas le cas de l'arbre de parsage.
- Dans un AST, les nœuds internes sont des structures du langage de programmation.
- Dans un arbre de parsage, les nœuds internes sont des non terminaux
  - Certains non terminaux sont des *helpers* qui ne seront pas dans un AST

Arbres syntaxiques abstraits Avantages

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Une grammaire qui est adéquate pour l'analyse syntaxique pourrait ne pas refléter la structure hiérarchique d'un langage de programmation

- Les méthodes d'analyses syntaxique imposent un ordre dans la construction de l'arbre de parsage. Cet ordre pourrait ne pas être l'ordre dans lequel les informations deviennent disponibles dans les structures du langage

Exemple

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Exemple avec la grammaire
  - *expr* → *expr* + *term* | *expr* – *term* | *term*
  - *Term* → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Définitions dirigées par la syntaxe

ÉCOLE POLYTECHNIQUE MONTRÉAL

- SDD (Définitions Dirigées par la Syntaxe)
  - Spécifications de haut niveau pour la traduction
  - Spécifie la valeur d'attributs en associant des règles sémantiques aux productions de la grammaire.

- Exemple:

PRODUCTION	SEMANTIC RULE	
$E \rightarrow E_1 + T$	$E.code = E_1.code \parallel T.code \parallel '+'$	(5.1)

SDD

ÉCOLE POLYTECHNIQUE MONTRÉAL

- **Définition**
  - Grammaire libre de contexte avec un ensemble d'attributs et de règles
- Les attributs sont associés à des symboles de la grammaires
  - Par exemple, *X.a* , où *a* est un attribut du symbole *X*
- Les règles sont associées à des productions

SDD vs SDT

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Un schéma de traduction dirigée par la syntaxe (SDT) spécifie l'ordre de l'évaluation des règles sémantiques.
- Une SDD peut être implémentée par une SDT.
- Exemple:

$$E \rightarrow E_1 + T \quad \{ \text{ print } '+' \ \} \qquad (5.2)$$

SDD / SDT - Applications

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Exemples d'applications:
  - Génération d'AST
  - Inférence de types
  - Génération de code
  - Construction de graphe de flux de contrôle
  - Analyse de programme:
    - Détection de clones logiciels
    - Analyses de sécurité (SQLi, XSS, etc.)
  - IDE (Eclipse, NetBeans, etc.)
  - Traduction de programmes
  - Etc.

SDD - Attributs

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Deux types d'attributs

- Synthétisés
  - Pour un non terminal *A*, représenté par un nœud *N*, l'attribut est défini par une règle sémantique associée à une production qui a *N* en tête.
  - Attribut défini en fonction des valeurs des attributs des enfants de *N* et de *N* lui-même.

- Hérités
  - Pour un non terminal *B*, représenté par un nœud *N*, l'attribut est défini par une règle sémantique associée à une production qui a le parent de *N* en tête et où *N* appartient à la partie droite de la règle.
  - Attribut défini en fonction des valeurs des attributs du parent de *N*, de *N* lui-même et des frères de *N*.

SDD - Attributs

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Les symboles terminaux possèdent seulement des attributs synthétisés par l'analyseur lexical.
- Le symbole START ne possède pas d'attribut hérité.
- Il est toujours possible de réécrire une SDD afin d'utiliser seulement des attributs synthétisés (mais il est souvent plus naturel d'utiliser des attributs hérités)

Attributs synthétisés

ÉCOLE POLYTECHNIQUE MONTRÉAL

- SDD *S-attributed*
  - Il s'agit d'une SDD qui n'implique que des attributs synthétisés
- Une SDD S-attributed s'implémente naturellement avec un parseur LR
- Un arbre de parsage peut TOUJOURS être annoté par une évaluation *bottom-up* des règles sémantiques d'une SDD qui est *S-attributed*

SDD - Exemple

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Utilisons une SDD S-*attributed* pour évaluer la valeur des expressions arithmétiques:
- On considère ici que le terminal **n** signale la fin d'une expression.

	PRODUCTION	SEMANTIC RULES
1)	$L \rightarrow E$ **n**	
2)	$E \rightarrow E_1 + T$	
3)	$E \rightarrow T$	
4)	$T \rightarrow T_1 * F$	
5)	$T \rightarrow F$	
6)	$F \rightarrow (E)$	
7)	$F \rightarrow$ **digit**	

SDD - Exemple

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Solution:

	PRODUCTION	SEMANTIC RULES
1)	$L \rightarrow E$ **n**	$L.val = E.val$
2)	$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3)	$E \rightarrow T$	$E.val = T.val$
4)	$T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5)	$T \rightarrow F$	$T.val = F.val$
6)	$F \rightarrow (E)$	$F.val = E.val$
7)	$F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

SDD – Exemple 2

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Utilisons la SDD pour transformer une expression infixe en expression post-fixe

PRODUCTION	SEMANTIC RULES
$expr \rightarrow expr_1 + term$	
$expr \rightarrow expr_1 - term$	
$expr \rightarrow term$	
$term \rightarrow 0$	
$term \rightarrow 1$	
...	
$term \rightarrow 9$	

SDD – Exemple 2

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Solution

PRODUCTION	SEMANTIC RULES
$expr \rightarrow expr_1 + term$	$expr.t = expr_1.t \,\|\|\, term.t \,\|\|\, '+'$
$expr \rightarrow expr_1 - term$	$expr.t = expr_1.t \,\|\|\, term.t \,\|\|\, '-'$
$expr \rightarrow term$	$expr.t = term.t$
$term \rightarrow 0$	$term.t = '0'$
$term \rightarrow 1$	$term.t = '1'$
...	...
$term \rightarrow 9$	$term.t = '9'$

SDD – Exemple 3

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Exemple d'utilisation d'attributs hérités et synthétisés
  - On propage les résultats partiels vers la droite (noeuds frères) et vers le bas (noeuds enfants), ce sont des attributs hérités.
  - On propage les résultats finaux vers le haut (noeud parent), ce sont des attributs synthétisés

	PRODUCTION	SEMANTIC RULES
1)	$T \rightarrow F\ T'$	$T'.inh = F.val$ $T.val = T'.syn$
2)	$T' \rightarrow *\ F\ T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3)	$T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4)	$F \rightarrow \mathbf{digit}$	$F.val = \mathbf{digit}.lexval$

Exercice 1

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Écrire une SDT qui calcule le nombre d'instructions (représentées par le non terminal *S*) dans un programme et qui affiche ce nombre une seule fois (via la fonction *print* qui prend un paramètre).

  - $P' \rightarrow P$
  - $P \rightarrow SP_1$
  - $P \rightarrow S$
  - $S \rightarrow \{ A \}$
  - $A \rightarrow$ **p s**

Exercice 1 (solution)

ÉCOLE POLYTECHNIQUE MONTRÉAL

Règles de production	Actions sémantiques
(0) $P' \rightarrow P$	$print(P.s)$
(1) $P \rightarrow S\ P_1$	$P.s = 1 + P_1.s$
(2) $P \rightarrow S$	$P.s = 1$
(3) $S \rightarrow \{\ A\ \}$	
(4) $A \rightarrow$ **p s**	

Arbre de parsage annoté

- Un arbre de parsage annoté montre les valeurs des attributs de chacun des nœuds.

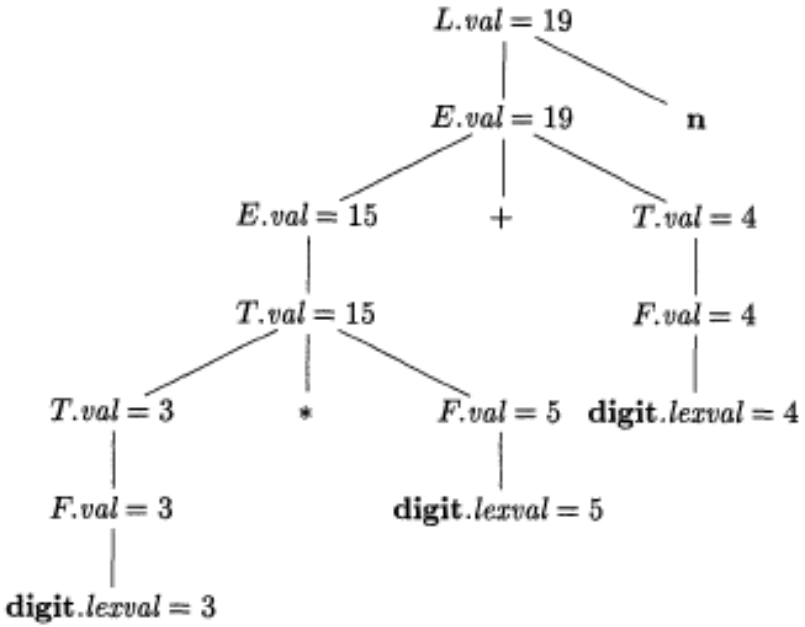


Figure 5.3: Annotated parse tree for $3 * 5 + 4$ **n**

Arbre de parsage annoté

- Construisons l'arbre de parsage annoté pour l'expression suivante:
  - 1*2*3*(4+5) **n**

	PRODUCTION	SEMANTIC RULES
1)	$L \rightarrow E$ **n**	$L.val = E.val$
2)	$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3)	$E \rightarrow T$	$E.val = T.val$
4)	$T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5)	$T \rightarrow F$	$T.val = F.val$
6)	$F \rightarrow (E)$	$F.val = E.val$
7)	$F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

Graphe de dépendances

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Un graphe de dépendances indique le flux d'informations entre les différents attributs.
  - Un arc d'un attribut à un autre signifie que la valeur du premier attribut est nécessaire pour calculer le deuxième.

- Le graphe de dépendances possède un nœud pour chaque attribut de chacun des nœuds dans l'arbre de parsage.

Graphe des dépendances

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Exemple:
  - Les traits pointillés représentent les arcs de l'arbre de parsage.
  - Les traits pleins représentent les arrêtes du graphe de dépendances.

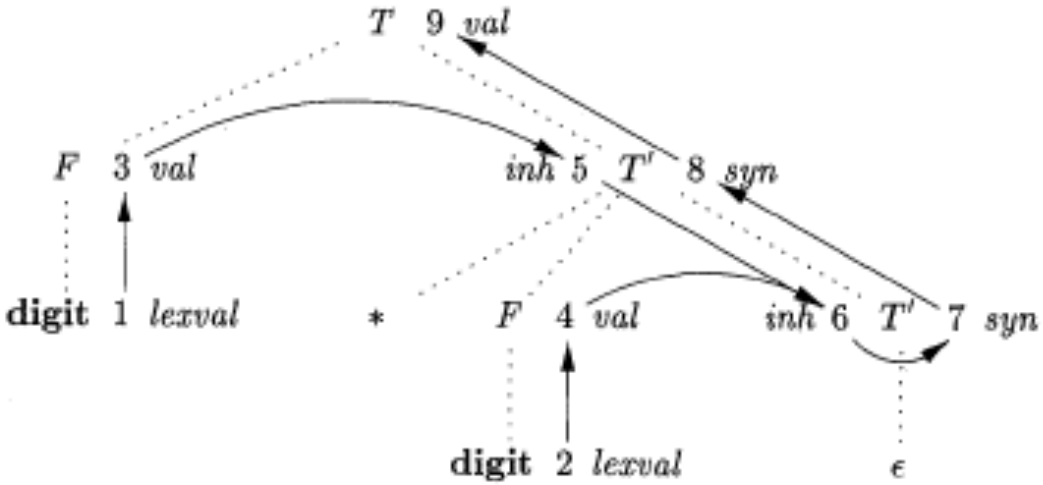


Figure 5.7: Dependency graph for the annotated parse tree of Fig. 5.5

Construction du graphe de dépendance

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Créer un nœud pour chaque attribut de chaque nœud de l'arbre de parsage.
- Si un attribut synthétisé A.*b* est défini en fonction de l'attribut X.*c*, ajouter un arc du nœud représentant X.*c* vers le nœud représentant A.*b*.
- Si un attribut hérité B.*c* est défini en fonction de l'attribut X.*a*, ajouter un arc du nœud représentant X.*a* vers le nœud représentant B.*c*.

Ordre topologique

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Un ordre topologique d'un graphe dirigé sans cycle est un ordre sur les nœuds du graphe tel que les nœuds de départ des arcs précèdent les nœuds d'arrivée.

- Tous les ordres topologiques d'un graphe de dépendances représentent un ordre valide d'évaluation des règles sémantiques.

- Exercice: identifier deux ordre topologiques distincts dans le graphe de la diapositive précédente

Classes de SDD

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Certaines classes de SDD sont définies afin de garantir l'existence d'une ordre topologique dans le graphe des dépendances (en empêchant la présence d'un cycle).

- S-Attributed
  - Tous les attributs sont synthétisés.
  - Attributs évalués en ordre *bottom-up*.
  - Traversement de l'arbre de parsage en postordre.

- Implantation par parseur LR
  - Les règles de traduction sont éxécutées quand le parseur fait un *reduce*.

Classes de SDD

ÉCOLE POLYTECHNIQUE MONTRÉAL

- L-attributed
  - Concept de base: les arcs du graphe des dépendances peuvent aller de gauche à droite, mais pas de droite à gauche
  - Les attributs peuvent toujours être évalués en traversant l'arbre de parsage en pré-ordre

- Propriété
  - Les SDD basées sur une grammaire LL(1) appartiennent aux définitions *L-attributed*

Définitions L-attributed

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Plus formellement, chaque attribut doit être soit:
  - Synthétisé
  - Hérité, avec les limitations suivantes. Soit une production $A \rightarrow X_1X_2\ldots X_n$ et un attribut hérité $X_i.a$. La règle sémantique ne peut utiliser que
    - Des attributs hérités associés à A
    - Des attributs hérités ou synthétisés associés aux symboles $X_1X_2\ldots X_{i-1}$ situés à gauche de X_i
    - Des attributs hérités ou synthétisés associés à X_i, mais seulement de façon à ce qu'il n'y ait pas de cycles dans le graphe des dépendances formé par les attributs de X_i

Implantation SDD L-attributed

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Généralement implantées dans un parseur LL.

- Les actions qui calculent des attributs hérités pour un non terminal *A* sont exécutées immédiatement avant le traitement de *A*. S'il existe des dépendances entre plusieurs attributs, les ordonner correctement.

- Les actions qui calculent des attributs synthétisés de la tête de la production sont exécutées après avoir traité tous les nœuds de la production.

Classes de SDD - Exemples

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Les SDD suivantes sont-elles L-attributed?

PRODUCTION	SEMANTIC RULE
$T \rightarrow F\ T'$	$T'.inh = F.val$
$T' \rightarrow *\ F\ T'_1$	$T'_1.inh = T'.inh \times F.val$

PRODUCTION	SEMANTIC RULES
$A \rightarrow B\ C$	$A.s = B.b;$
	$B.i = f(C.c, A.s)$

SDD – Effets de bord

- Effets de bord
  - Impression
  - Références globales
- Calcul des valeurs des attributs sur les nœuds de l'arbre de parsage
  - Annotation ou décoration de l'arbre de parsage

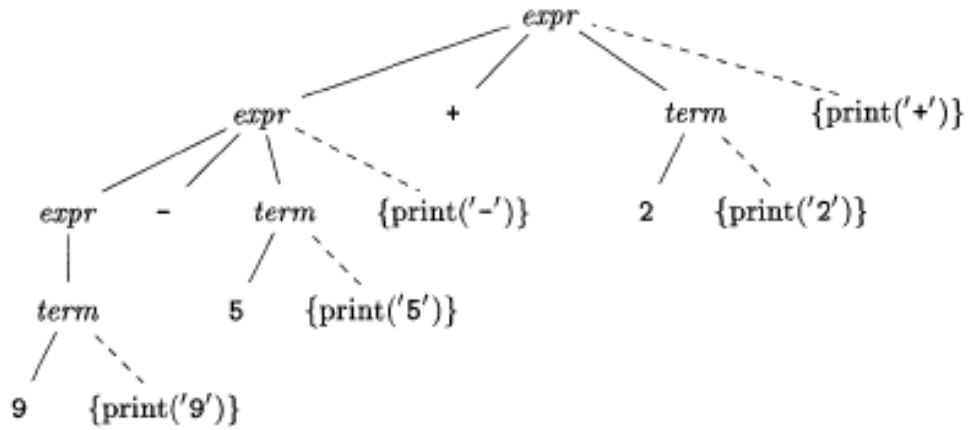


Figure 2.14: Actions translating 9-5+2 into 95-2+

Exercice 2

ÉCOLE POLYTECHNIQUE MONTRÉAL

1. En utilisant un effet de bord par appel de fonction, écrire une SDD pour ajouter chaque variable du système dans la table de symboles.
   - Supposez l'existence d'une fonction *addType*(varName, varType) qui ajoute la variable varName, de type varType, dans la table de symboles

2. Dessiner le graphe des dépendances pour la déclaration suivante:
   float id_1, id_2, id_3

	PRODUCTION	SEMANTIC RULES
1)	$D \rightarrow T\ L$	
2)	$T \rightarrow$ **int**	
3)	$T \rightarrow$ **float**	
4)	$L \rightarrow L_1$, **id**	
5)	$L \rightarrow$ **id**	

Exercice 2 - Solution

ÉCOLE POLYTECHNIQUE MONTRÉAL

	PRODUCTION	SEMANTIC RULES
1)	$D \rightarrow T\ L$	$L.inh = T.type$
2)	$T \rightarrow$ **int**	$T.type =$ integer
3)	$T \rightarrow$ **float**	$T.type =$ float
4)	$L \rightarrow L_1$, **id**	$L_1.inh = L.inh$
		$addType($**id**$.entry, L.inh)$
5)	$L \rightarrow$ **id**	$addType($**id**$.entry, L.inh)$

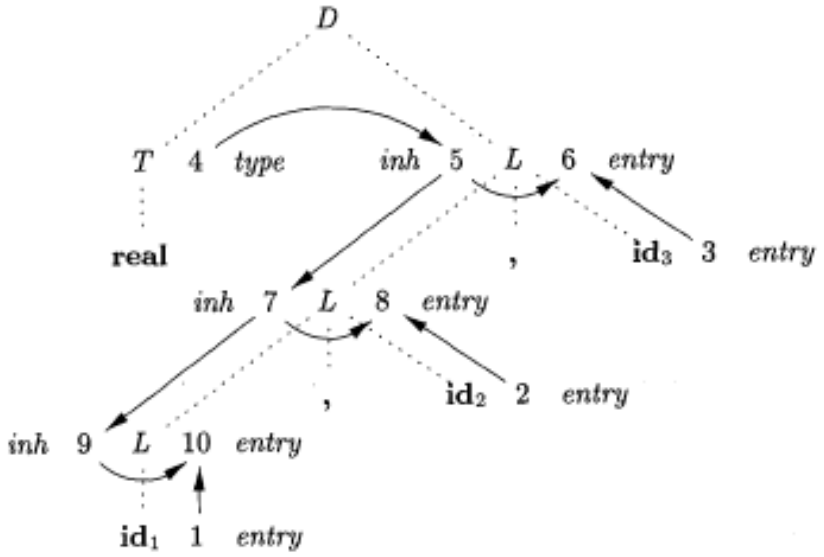


Figure 5.9: Dependency graph for a declaration **float** $\mathbf{id}_1$, $\mathbf{id}_2$, $\mathbf{id}_3$

Applications de la SDD Exemple

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Il est possible de construire un AST en utilisant la SDD

- Une technique similaire peut être utilisée pour générer le code intermédiaire et faire de la vérification de types

	PRODUCTION	SEMANTIC RULES
1)	$E \rightarrow E_1 + T$	$E.node = \textbf{new}\ Node('+', E_1.node, T.node)$
2)	$E \rightarrow E_1 - T$	$E.node = \textbf{new}\ Node('-', E_1.node, T.node)$
3)	$E \rightarrow T$	$E.node = T.node$
4)	$T \rightarrow (E)$	$T.node = E.node$
5)	$T \rightarrow \textbf{id}$	$T.node = \textbf{new}\ Leaf(\textbf{id}, \textbf{id}.entry)$
6)	$T \rightarrow \textbf{num}$	$T.node = \textbf{new}\ Leaf(\textbf{num}, \textbf{num}.val)$

Figure 5.10: Constructing syntax trees for simple expressions

Applications de la SDD Exemple

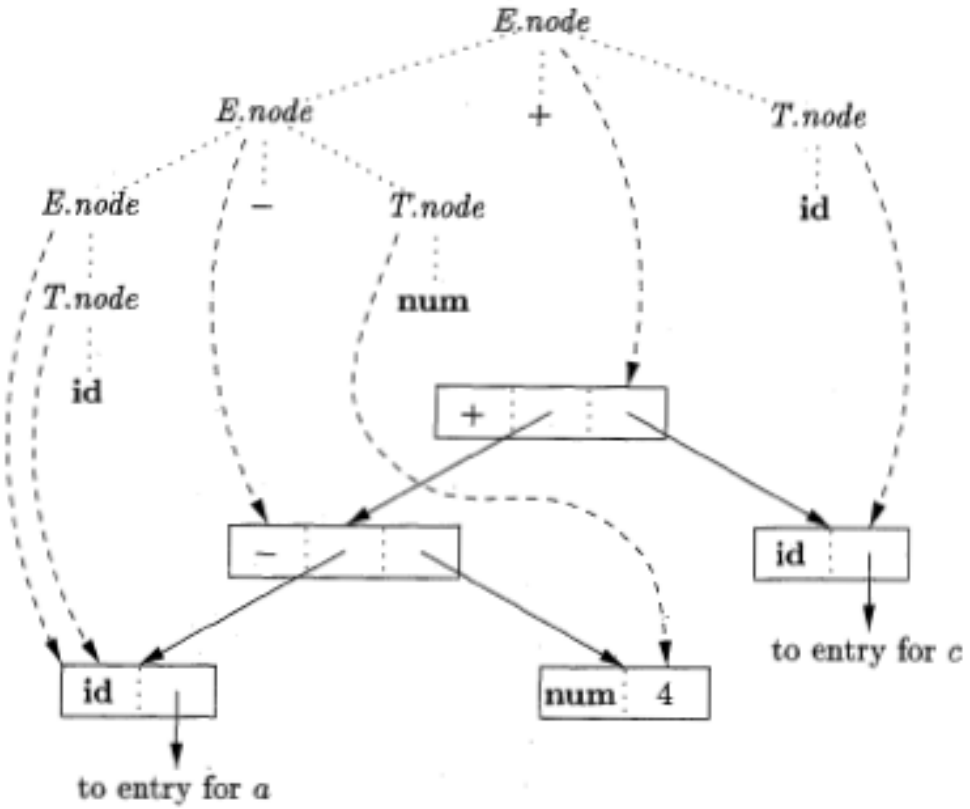


Figure 5.11: Syntax tree for $a - 4 + c$

Schémas de traduction

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Le but est de convertir les SDD en SDT afin d'exécuter les actions définies par la SDD au bon moment.

- Cas le plus simple:
  - SDD S-attributed avec un parseur *bottom-up*
  - On peut alors simplement effectuer les actions en même temps que les réductions du parseur LR
  - On maintiendra une pile d'attributs synthétisés en plus de la pile d'états habituelle.

- Exemple sur la grammaire des expressions du cours précédent.

Schéma de traduction

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Si la SDD est L-attributed, une implantation par parseur LL est généralement plus appropriée.
- Une fonction par non-terminal.
- La fonction reçoit les attributs hérités en paramètre et retourne les attributs synthétisés.
- Exemple sur la grammaire JSON de l'intra H13.

Perspective générale

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Les activités de rétro-ingénierie peuvent être considérées comme des problèmes de Traduction Dirigée par la Syntaxe (SDT) où
  - Le langage de départ représente un certain niveau d'abstraction
  - Le langage cible est à un niveau d'abstraction supérieur
- La représentation d'un niveau d'abstraction revient à la conception de langages
- Les problèmes de rétro-ingénierie reviennent à trouver une SDT adéquate

Outils de traduction

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Outils pour implanter des SDD
  - YACC
    - Méthodes négligentes, approche ascendante
  - TXL
    - Méthodes récursives, strictement fonctionnel
  - Méthodes hybrides
    - Sauvegardent le AST comme une structure de données et utilisent un paradigme pour effectuer les transformations désirées
  - JavaCC
    - Méthodes récursives (actions dans les règles de production)
    - Utilisation de visiteurs (action sur l'arbre *a posteriori*)

Exercice supplémentaire

ÉCOLE POLYTECHNIQUE MONTRÉAL

- Soit la grammaire suivante.

$$E \rightarrow E + T \mid T$$
$$T \rightarrow \textbf{num} \ . \ \textbf{num} \mid \textbf{num}$$

1. Donnez un SDD permettant de déterminer le type de chaque terme *T* et expression *E*

2. Étendez votre SDD pour traduire les expressions dans des expressions en notation postfixe.