

THÈSE de DOCTORAT

de l'UNIVERSITÉ PARIS 6

Spécialité :
Informatique

présentée par
M Gilles Roussel

pour obtenir le grade de
DOCTEUR de l'UNIVERSITÉ PARIS 6

sujet de thèse :

**Algorithmes de base pour
la Modularité et la Réutilisabilité
des Grammaires Attribuées**

soutenue le 1^{er} Mars 1994 devant le jury composé de :

M.	B. Lorho	Président
MM.	G. Cousineau P. Deransart	Rapporteurs
MM.	J. Berstel P. Franchi-Zanettacci M. Jourdan	Examineurs

Résumé

Contrairement à beaucoup d'autres langages informatiques, les grammaires attribuées (GA) classiques n'offrent pas encore la possibilité d'écrire une application de façon modulaire. Elles rendent également difficile la réutilisation d'algorithmes déjà spécifiés. En effet toute spécification est fortement liée à une syntaxe particulière.

Notre but a été d'introduire ces notions afin de rendre la spécification par GAs plus attrayante.

Il y a deux types complémentaires d'écriture modulaire pour les GAs.

Les grammaires attribuées fonctionnelles permettent de décomposer une application en une séquence de GAs. Chacune de ces GAs est une fonction qui prend en entrée un arbre et calcule un nouvel arbre. Celui-ci est l'entrée de la GA suivante dans la séquence.

- La méta-composition introduite par Ganzinger et Giegerich construit une GA qui a la même sémantique que la séquence de GAs. Afin d'augmenter l'applicabilité de la méta-composition, nous avons augmenté la classe des GAs acceptées et transformé les GAs générées afin d'éliminer les règles de copie inutiles introduites par la méta-composition.
- Une alternative à la méta-composition est la construction directe d'un évaluateur de la composée. En nous plaçant dans le cadre des évaluateurs par séquences de visites, nous pouvons à partir des différentes spécifications de construire un évaluateur de l'application totale sans construire la grammaire attribuée globale.

La deuxième méthode nous assure que l'on est capable de construire un évaluateur par séquences de visites s'il est possible de construire un tel évaluateur pour chacune des GAs d'entrée. Cette méthode a également l'avantage de permettre une compilation séparée avec l'introduction des filtres de couplage.

Les grammaires attribuées incomplètes permettent de décomposer une application en différentes unités de sens. Dans chacune d'elles on spécifie un problème particulier sur la syntaxe globale et on communique avec les autres unités de l'application grâce à des attributs d'entrée et de sortie. Il est ainsi possible de séparer des parties de spécifications qui peuvent être interdépendantes. Comme dans le cas précédent on a deux approches :

- une union au niveau des spécifications, la méta-union. On peut déduire certaines propriétés de la GA résultante à partir des GAs d'entrée.
- une union au niveau des évaluateurs, qui permet une véritable compilation séparée des différentes parties de l'application.

La deuxième notion que nous avons introduite est celle de réutilisabilité. Elle est basée sur une première technique qui permet de construire automatiquement une grammaire attribuée fonctionnelle spécifiant la projection d'une GA dite "gène" sur une syntaxe "réelle". Nous avons ensuite introduit un algorithme qui permet d'extraire automatiquement un gène d'une spécification.

Table des matières

1	Introduction	3
2	Généralités sur les grammaires attribuées	13
2.1	Définitions et Notations	13
2.2	Les méthodes d'évaluation	18
3	Les grammaires attribuées fonctionnelles et leur méta-composition	33
3.1	Présentation et définitions	33
3.2	Les grammaires attribuées fonctionnelles	33
3.3	Composition des spécifications : la méta-composition	36
3.3.1	Description de l'algorithme original	37
3.3.2	Interprétation personnelle des résultat de l'algorithme original . . .	39
3.3.3	Clôture des classes vis-à-vis de la méta-composition	42
4	Optimisation et transformation de grammaires attribuées	45
4.1	Introduction	45
4.2	Élimination de règles de copie conduisant à une constante	48
4.2.1	L'algorithme	49
4.3	Élimination des boucles de règles de copie	51
4.3.1	Repérage des attributs "égaux"	52
4.3.2	Transformation	61
4.3.3	Remarques générales	62
4.4	Conclusion	64
5	Implantation	65
5.1	Le système FNC-2	65
5.2	Le coupleur	66
5.3	Traitement des constructions sur les productions d'arité variable	70
5.3.1	Les attributs importés et les attributs exportés	75
5.3.2	Conclusion	75
6	Composition des évaluateurs	77
6.1	Une première approche	77

6.1.1	Introduction	77
6.1.2	Un évaluateur par séquences de visites pour une grammaire attribuée résultant de la méta-composition	78
6.1.3	Algorithme de construction de l'évaluateur	81
6.2	Ajout des conditionnelles	89
6.2.1	Constructions imbriquées et attributs locaux	91
6.2.2	La partie sémantique	93
6.3	Les Filtres de Couplage	94
6.4	Acceptation des dags dans les grammaires attribuées fonctionnelles	98
6.5	Optimisation de la fonction Σ pour diminuer les règles de copie	102
6.6	Comparaison avec d'autres travaux	104
6.7	Conclusion	104
7	Les Grammaires attribuées incomplètes	107
7.1	Présentation et définitions	107
7.2	Union des spécifications	113
7.2.1	La Méta-Union	114
7.3	Liaison des évaluateurs	116
7.3.1	Décomposition de la grammaire attribuée incomplète	122
8	De nouvelles classes de grammaires attribuées	129
8.1	Les nouvelles classes de grammaires attribuées	129
8.2	Le nouveau test DNC	129
8.3	Les classes FNC+ et DNC+	131
9	Réutilisabilité	137
9.1	La généricité dans les grammaires attribuée	137
9.1.1	Génération Automatique de la Grammaire Attribuée de Couplage	142
9.1.2	Extensions de l'algorithme de base	146
9.1.3	Remarques	147
9.2	L'extraction automatique de gènes	148
9.2.1	Introduction	148
9.2.2	Simplification de la sémantique	148
9.2.3	Simplification de la syntaxe	149
10	Conclusion	153