
PCL : Rapport d'activité 3

TDS et contrôles sémantiques

Version finale

Guillaume BOURGEON, Louen GRÉAU, Paul
NICOLAS

TELECOM Nancy

16 janvier 2023

Table des matières

1	Construction de la TDS	3
1.1	Réalisation de la TDS	3
2	Contrôles sémantiques	4
2.1	Définition des contrôles sémantiques	4
2.2	Implémentation des contrôles sémantiques	4
2.3	Difficultés rencontrées	5
3	Gestion de projet	6
3.1	Répartiton des tâches	6
3.2	Matrice SWOT	6
4	Annexe	7
4.1	Exemple de TDS	8
4.2	Exemple de contrôles sémantiques	9

1 Construction de la TDS

1.1 Réalisation de la TDS

Pour construire la TDS (Table Des Symboles), nous avons implémenté plusieurs classes Java pour le corps de la table. Nous avons donc fait une classe qui contient la tds et des classes pour chaque élément stocké dans une tds (argument de fonction, variable, fonction, etc.) Ces objets sont utilisés pour remplir la TDS lors de la construction de l'AST et du parcours récursif de celui-ci. La TDS globale contient donc les TDS des différents blocs du code. Chaque TDS contient les déclarations du bloc concerné avec les types, les noms, les arguments de fonctions, etc. On affiche alors la TDS du fichier complet à la fin de la construction de l'AST, celle-ci affiche à son tour les TDS des blocs qu'elle englobe.

```
=====
TDS_Let_1_1
Parent : TDS_root_0_0

-----
> Variable : N | Integer
> Variable : M | String
> Type : intArray | array of
> Variable : row | intArray
> Variable : col | intArray
> Variable : diag1 | intArray
> Variable : diag2 | intArray
> Procedure : printboard | 0 :
> Procedure : try | 1 : int

=====
TDS_printboard_2_2
Parent : TDS_Let_1_1

-----
> Variable : i | Integer
> Variable : j | Integer

=====
TDS_try_3_2
Parent : TDS_Let_1_1

-----
> Argument : c | int
> Variable : r | Integer
```

FIGURE 1.1 – Exemple de TDS pour le code sujet_8queens

2 Contrôles sémantiques

2.1 Définition des contrôles sémantiques

Afin de définir les contrôles sémantiques à implémenter, nous avons commencé par parcourir à nouveau les deux documents du langage Tiger (Manuel et Spécifications). Nous les avons déjà parcouru pour la partie syntaxique et la rédaction de la grammaire, mais nous avons cette fois-ci besoin de différents éléments relatifs à la sémantique.

Nous les avons ensuite triés par catégorie (portée des variables, déclaration des variables, appels de fonction,...) et avons listé pour chacun dans quelle règle il fallait l'implémenter.

2.2 Implémentation des contrôles sémantiques

Les contrôles sémantiques ne doivent pas influencer sur l'exécution du code, mais uniquement prévenir des différentes erreurs sémantiques.

Nous avons décidé de stocker dans un dictionnaire les erreurs, avec en clé la règle de grammaire provoquant l'erreur et en valeur l'erreur en question (double déclaration d'une variable, paramètre d'une fonction manquant,...).

Pour réaliser les différents contrôles, nous utilisons la TDS pour voir quels éléments se trouvent dedans à chaque instant, ainsi que le parcours de l'AST pour accéder aux différents noeuds et feuilles.

Voici la liste des contrôles sémantiques actuellement mis en place :

1. Double déclaration des variables ;
2. Double déclaration des types ;
3. Double déclaration des fonctions ;
4. Double déclaration des procédures ;
5. Déclaration avant utilisation des variables ;
6. Déclaration avant utilisation des types ;
7. Déclaration avant utilisation des fonctions ;
8. Déclaration avant utilisation des procédures ;
9. Vérifications du bon nombre de paramètres fonctions ;
10. Vérifications du bon nombre de paramètres procédures ;
11. Vérifications des noms de paramètres procédures ;
12. Vérifications des noms de paramètres fonctions ;

13. Vérifications du type pour les opérations arithmétiques ;
14. Vérifications du type pour les affectations ;

Nous souhaitons si possible en écrire d'autres avant la soutenance qui aura lieu ce jeudi 19 (division par 0, accès aux tableaux,...).

2.3 Difficultés rencontrées

Nous avons rencontré des difficultés pour certains contrôles sémantique dans l'utilisation du parcours récursif de l'AST. Pour répondre à ces problèmes, nous avons beaucoup plus travaillé en groupe que pour les parties précédentes où nous travaillions sur des tâches réparties à l'avance. L'avancement était donc plus lent mais nous avons tout de même réussi à implémenter les contrôles sémantiques.

3 Gestion de projet

3.1 Répartition des tâches

Étapes	Guillaume	Paul	Louen
Rédaction de la grammaire 1 (récursive gauche)	5h	5h	5h
Rédaction des tests	8h	3h	3h
Rédaction de la grammaire 2	17h	3h	25h
Labellisation de la grammaire	2h	1h	5h
Mise en place de l'AST	4h	4h	2h
Rédaction des règles de l'AST	2h	12h	30h
Construction de la TDS	5h	2h	5h
Remplissage de la TDS	3h	2h	5h
Liste des contrôles à implémenter	10h	10h	2h
Implémentation des contrôles sémantiques	20h	23h	4h
Rédaction des rapports d'activité	5h	5h	1h
Total	82h	71h	87h

3.2 Matrice SWOT

Forces : <ul style="list-style-type: none"> - Bonne cohésion du groupe - Bonne maîtrise d'Antlr - Louen a fait le projet l'an dernier 	Faiblesses : <ul style="list-style-type: none"> - Rendre la grammaire non réursive gauche - Mauvaise compréhension de l'utilisation de la TDS - Difficultés à implémenter les contrôles sémantiques
Opportunités : <ul style="list-style-type: none"> - Date de rendu repoussée - Vacances de Noël pour travailler - TD de PCL pour mieux comprendre les différents concepts 	Menaces : <ul style="list-style-type: none"> - Partiels de fin de semestre : la gestion du temps pourra être problématique.

4 Annexe

4.1 Exemple de TDS

```
=====
TDS_Let_1_1
Parent : TDS_root_0_0

-----
> Variable : N | Integer
> Variable : M | String
> Type : intArray | array of
> Variable : row | intArray
> Variable : col | intArray
> Variable : diag1 | intArray
> Variable : diag2 | intArray
> Procedure : printboard | 0 :
> Procedure : try | 1 : int

=====
TDS_printboard_2_2
Parent : TDS_Let_1_1

-----
> Variable : i | Integer
> Variable : j | Integer

=====
TDS_try_3_2
Parent : TDS_Let_1_1

-----
> Argument : c | int
> Variable : r | Integer
```

FIGURE 4.1 – Exemple de TDS pour le code sujet_8queens

4.2 Exemple de contrôles sémantiques

Ici figure seulement la partie du code produisant des erreurs sémantiques. Il y'a d'autres lignes de code avant et après.

```
var diag2 := intArray [ N+N-1 ] of 0
var diag2 := intArray [ N+N-1 ] of 0

function printboard() =
  (M := M + N;
   for i := 0 to N-1
   do (for j := 0 to N-1
       do print(if col[i]=j then " 0" else " .");
       print("\n"));
   print("\n"))

function try(c:int) =
  if c=N then (M:=N ; printboard())
  else for r := 0 to N-1
       do if row[r]=0 &
           diag1[r+c]=0 & diag2[r+7-c]=0
           then (row[r] := 1; diag1[r+c] := 1;
                diag2[r+7-c] := 1; col[c] := r;
                try(c+1);
                a := 1;
                let var T : StringArray := 1 in end;
                let var Z := 1 in end;
                row[r] := 0; diag1[r+c] := 0;
                diag2[r+7-c] := 0)

function try() = print("\n")

in try() end
```

FIGURE 4.2 – Code produisant des erreurs sémantiques

```
# Decl Var Simple ~ Double déclaration de : diag2
# Decl Var Type Absent ~ Type non déclaré avant utilisation : StringArray
# lvalue_id Var ~ Variable non déclarée avant utilisation : a
# Decl Procedure ~ Double déclaration de procédure pour : try
```

FIGURE 4.3 – Contrôles sémantiques affichés