Compte rendu d'activités

Wague, Mady
mady.wague@telecomnancy.eu

Tehe, Yohou
yohou.tehe@telecomnancy.eu

Marsou, Soulaïman*
soulaiman.marsou@telecomnancy.eu
*chef de projet

Groupe 14

16 janvier 2023

Table des matières

1 Activités **2**
1.1 TDS . . . 2
1.1.1 Variables . . . 2
1.1.2 Types . . . 2
1.1.3 Fonctions . . . 3
1.1.4 Paramètres . . . 3
1.2 Contrôles sémantiques . . . 4
1.3 Tests . . . 4

2 Gestion de projet **5**
2.1 Grammaire . . . 5
2.2 Construction de l'AST . . . 5
2.3 TDS et contrôles sémantiques . . . 6

1 Activités

Pour cette partie du projet, nous avons utilisé un visiteur tdsVisitor.java pour parcourir l'ensemble de l'arbre abstrait et réalisés les contrôles sémantiques.

1.1 TDS

Nous avons construit les tables des symboles pour chaque bloc visité lors du parcours de l'arbre abstrait. On considère comme bloc les boucles for, les boules while, les déclarations de fonction et les blocs "let in end". Chaque table de symbole a un numéro de région, un numéro d'imbrication et une liste d'entrées. Les entrées possibles sont les variables, les types, les fonctions et les paramètres.

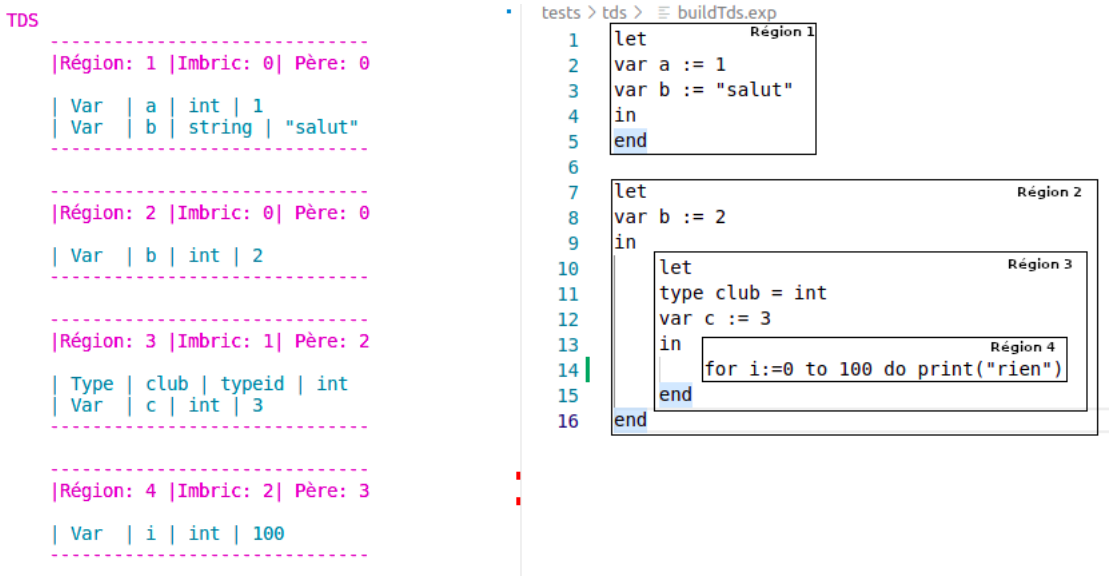

FIGURE 1 – Exemple de construction de la TDS d'un programme CHAOS.

Détaillons l'ensemble des entrées dans les paragraphes suivants.

1.1.1 Variables

Une variable est identifiée par son nom, son type et son contenu. Ces informations sont conservées dans la table des symboles.

1.1.2 Types

Un type est déterminé par son identifiant et son type dans le langage CHAOS. Ce type qui peut-être soit un int, un string, un array ou un record. Ces informations sont également conservées dans la table des symboles.

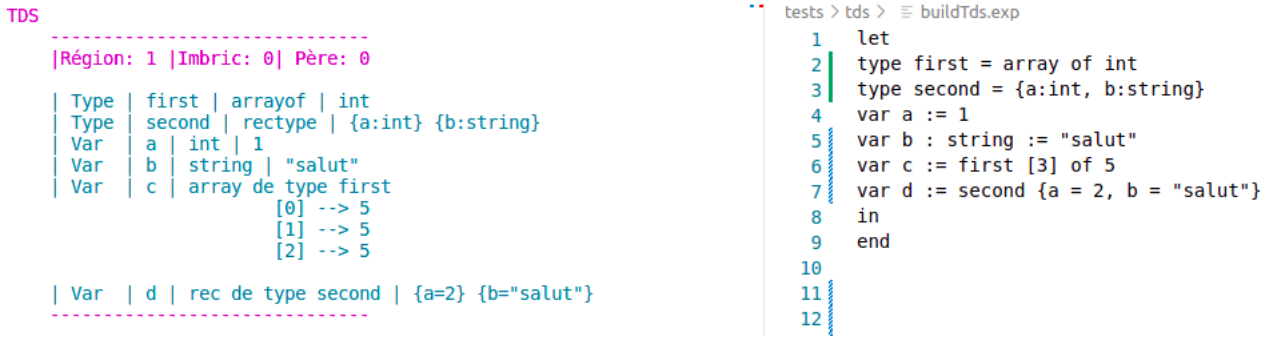

FIGURE 2 – Les types et les variables dans la TDS d'un exemple de programme.

1.1.3 Fonctions

Une fonction est définie par son identifiant, son type et sa propre table des symboles. En effet lorsque nous déclarons une fonction, nous ajoutons ses caractéristiques dans la table des symboles courante et nous créons une table des symboles pour cette fonction. Dans cette dernière table des symboles, nous ajoutons les paramètres de cette fonction et les déclarations de variables, de types ou de fonctions que nous rencontrons dans le bloc de cette fonction lors du parcours de l'arbre abstrait.

1.1.4 Paramètres

Par "Paramètres" nous désigons les paramètres de fonctions. Nous avons considéré les paramètres comme des variables sans valeur attribuée. Lors de l'affichage des tables de symbole, un mot clé "paramètre" distingue les paramètres des variables pour une meilleur compréhension. Ces paramètres sont ajoutés dans la table de symbole de la fonction qui les déclarent.

```
TDS
    ------------------------------
    |Région: 1 |Imbric: 0| Père: 0

    | Function | add | int | 3 paramètres
    ------------------------------

    ------------------------------
    |Région: 2 |Imbric: 1| Père: 1

    | paramètre  | p | string
    | paramètre  | a | int
    | paramètre  | b | int
    ------------------------------
```

tests > tds > func.exp

```
1  let
2
3    function add(a : int,p:string,b:int) : int = 1
4
5  in
6    add("MAIN",2)
7
8  end
```

FIGURE 3 – Les fonctions et les paramètres dans la TDS d'un exemple de programme.

1.2 Contrôles sémantiques

Réunies par thème dans une carte mentale, vous trouverez dans la figure suivante l'ensemble des contrôles sémantiques que l'on a identifié dans le document "Tiger_Specification.pdf" envoyé par email. Y sont indiqués les contrôles sémantiques réalisés ainsi que ceux que l'on n'a pas réalisés.

FIGURE 4 – Contrôles sémantiques réalisés

1.3 Tests

Pour tester nos contrôles sémantiques, nous avons utilisé les fichiers suivants : buildTds.exp, decVarError.exp, if-then.exp, main2.exp, buildVar.exp, for.exp, lvalue.exp, cs1.exp, ifthenelse.exp, main1.exp, opTdserror.exp.

Pour la présentation, nous utiliserons le fichier tests/tds/main.exp.

2 Gestion de projet

2.1 Grammaire

Répartition des tâches

Tâches	Serge Téhé	Soulaiman Marsou	Mady Wagué
Rédaction grammaire	R	R	R
Suppression récursivités gauche	R,A	I	I
Priorités opérateur	I	R,A	I
Vérification ambiguité	I	I	R,A
Factorisation de certaines règles	R,A	A	I
Tests	R	R,A	R

Réunion 1

Mise en place de l'organisation : Comme tous les membres du groupe n'avaient pas totalement terminé le tp1 de traduction, nous nous sommes consacrés à celui-ci dans le but de bien maîtriser l'outil ANTLR et de ne pas rencontrer de problème par rapport à cela durant le projet.

Réunion 2

Rédaction de la grammaire : Nous avons commencé à écrire la grammaire, mais avons rencontré des problèmes d'ambigüité, de récursivité gauche ou encore d'associativité et de priorité.

Réunion 3

Rédaction de la grammaire : Nous avons effectué différents tests pour vérifier si la grammaire était vraiment fonctionnelle et le cas échéant nous la modifions au fur et à mesure.

Réunion 4

Finalisation de la grammaire : Nous n'avons pas eu le temps d'effectuer toutes les tâches de la réunion précédente et n'avons pas eu le temps de rendre la grammaire LL(1). On s'est focalisé sur la vérification et résolution de la récursivité gauche, de la priorité, de l'associativité et les problèmes d'ambigüité.

2.2 Construction de l'AST

Répartition des tâches

Tâches	Serge Téhé	Soulaiman Marsou	Mady Wagué
Règles récursives/listes	I	R,A	I
Opérations	R,A	I	I
Toutes les autres règles	I	I	R,A

Réunion 1

Mise en place de l'organisation : Nous nous sommes départagé les classes à rédiger. Soulaïman devra initier toute l'arborescence des fichiers pour la construction de l'AST. Lorsque ce sera fini, chacun complétera les classes et méthodes de chacun.

Réunion 2

Rédaction du rapport : Chacun a fini de rédiger ses classes dans le dossier AST et les méthodes de AstCreator. Il faut donc terminer les méthodes de GraphVizVisitor.

2.3 TDS et contrôles sémantiques

Répartition des tâches

Tâches	Serge Téhé	Soulaiman Marsou	Mady Wagué
Mot clé Break	I	I	R,A
Mot clé Nil	X	X	X
Création de Tableau	I	R,A	I
Création d'instance immuable(record)	I	R,A	I
Lvalue	I	R,A	I
Toutes les opérations	R,A	I	I
Déclarations(variable et type)	R,A	I	I
Boucle	I	I	R,A
Condition	I	I	R,A
Appel et déclaration de fonction	R,A	I	I

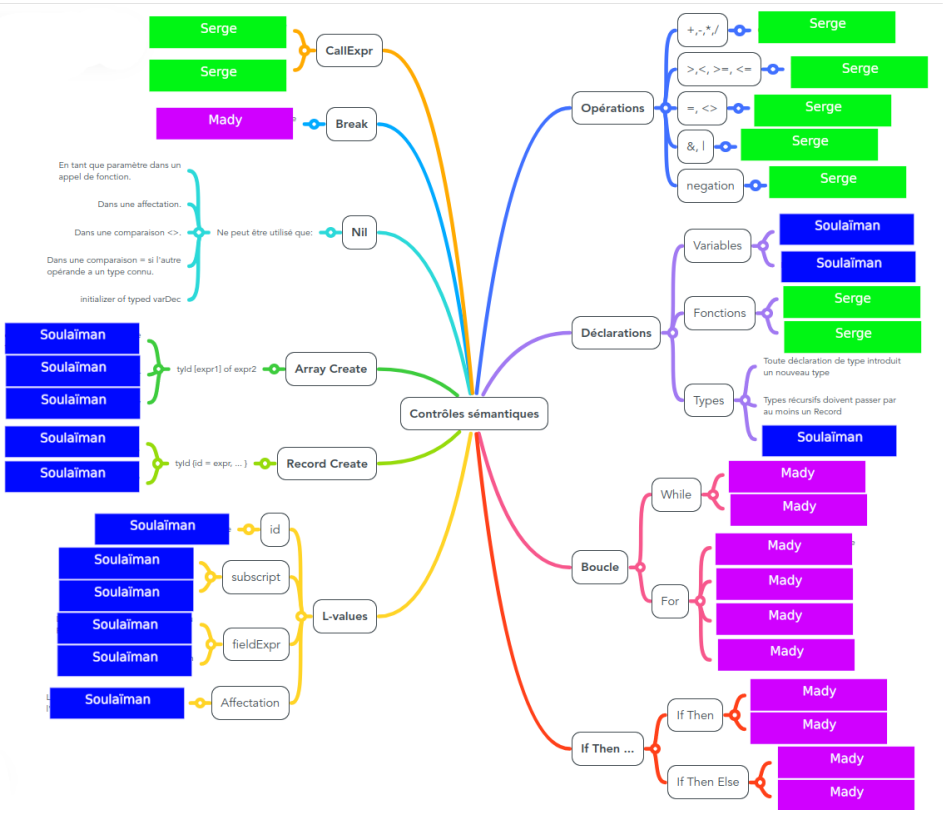

FIGURE 5 – Répartition des contrôles sémantiques.

Réunion 3

Mise en place de l'organisation (pour la TDS) : Nous avons initié les classes nécessaire à la création de la TDS et avons lister tous les contrôles sémantiques possibles afin de se répartir le travail.

Réunion 4

Finalisation des contrôles sémantiques : Durant cette réunion nous avions fait le point sur l'avancement des contrôles sémantiques et nous nous sommes donné pour consigne d'effectuer toute sorte de test pour la prochaine réunion.

Réunion 5

Rédaction du rapport + dernières vérifications : Nous avons vérifié une dernière fois les résultats sur les tests obtenue par chacun de nous pour les contrôles sémantiques et avons rédigé le rapport.