



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

LOG3210 – Élément de langages et compilateur

Hiver 2021

Intra

Sami Belaidi - 1900787

Soumis à: Doriane Olewicki

15 mars 2021

Grammaire

Le niveau le plus bas de mon parseur correspond aux éléments suivants :

- Une chaîne de caractères (StringValue)
- Une valeur booléenne (Boolvalue)
- Une suite de lettres et chiffres qui peut correspondre à une variable ou un nom de fonction (Identifier)
- Valeur entière (intValue)
- Valeur réelle (RealValue)

Tous ces nœuds sont utilisés à la base d'autres nœuds qui correspondent à des interactions avec d'autres nœuds.

Pour respecter la priorité des opérations, je me suis arrangé pour que les nœuds les plus prioritaires soient plus bas dans l'arbre de parsing. L'ordre des opérations est le suivant (en ordre descendant) : parenthèses, négation, négatif, multiplication, addition, comparaison et enfin une comparaison booléenne.

Pour la déclaration de fonctions, il faut tenir compte du fait qu'il y a possiblement des valeurs de retour, et ce seulement si c'est précisé. Pour ce faire, j'ai implémenté deux déclarations de fonction différentes.

Pour ce qui est des déclarations, des conditions *if*, des *while* et des assignations, cela était direct, car c'étaient principalement des éléments terminaux alignés.

Visiteur

Pour commencer, tous les nœuds bas niveaux impriment leur valeur respective, qui a été assigné au niveau de la grammaire. Ensuite les fonctions qui gèrent les opérations impriment le signe opératoire et gère également les espacements. Pour gérer les indentations, cela est fait grâce à la variable globale `indentationLevel`. En fait, avant d'imprimer un *statement*, j'imprime autant de caractères de tabulation que la valeur d'`indentationLevel`. Après avoir rencontré les jetons

terminaux *if*, *while* ou *fn*, nous incrémentons la valeur d'indentationLevel, et la décrémentation à la fin du bloc.

Une difficulté que j'ai rencontrée était de gérer l'impression du « pass » lorsqu'un bloc était vide. En fait, il fallait l'imprimer au début de chaque bloc si ce dernier était vide, et que le parent de ce bloc correspond à un nœud *if* ou *while*. Pour gérer le cas d'une fonction vide, je vérifie que le block est vide, et ensuite que le nombre d'enfants est strictement égal à 3 (le premier étant le nom de la fonction, le deuxième la paramètres et en troisième le bloc vide). Si le nombre d'enfants est égal à 4, cela veut dire qu'il y a une valeur de retour à la fonction, et qu'il n'est pas nécessaire d'imprimer « pass ».