

Travail pratique 2

Exercices pratiques et théoriques sur l’analyse grammaticale

Par  
Frédérik Boutin

Travail présenté à Mme Dorra Lamouchi

Dans le cadre du cours CompilationINF37207

21 avril 2024

# 2.1 - Exercice 1

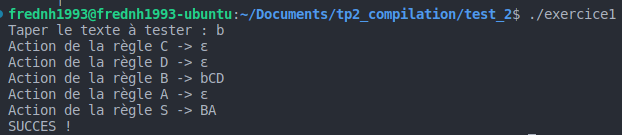


Figure 1 - Fichier exercice1.y

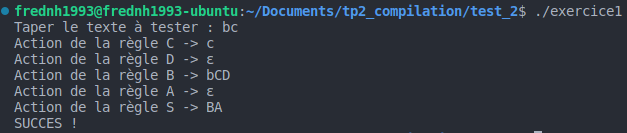
NOTE IMPORTANTE : Pour le code, voir les fichiers dans le sous-dossier **« code\_exercice1 »**, inclus dans le dossier compressé de la remise.

## Analyse des mots d’entrée

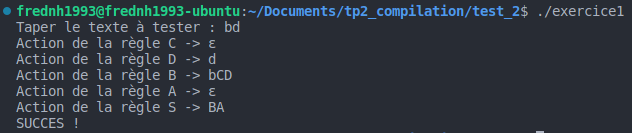
### Mot "b"



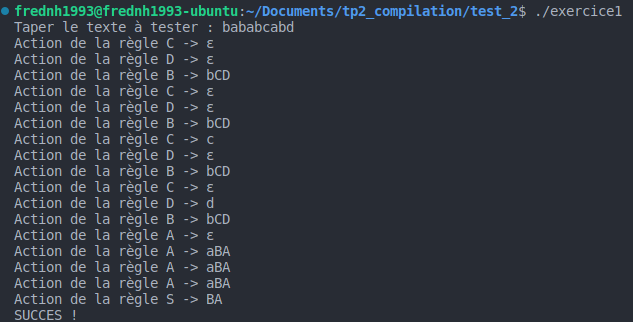
### Mot "bc"



### Mot "bd"



### Mot "bababcabd"



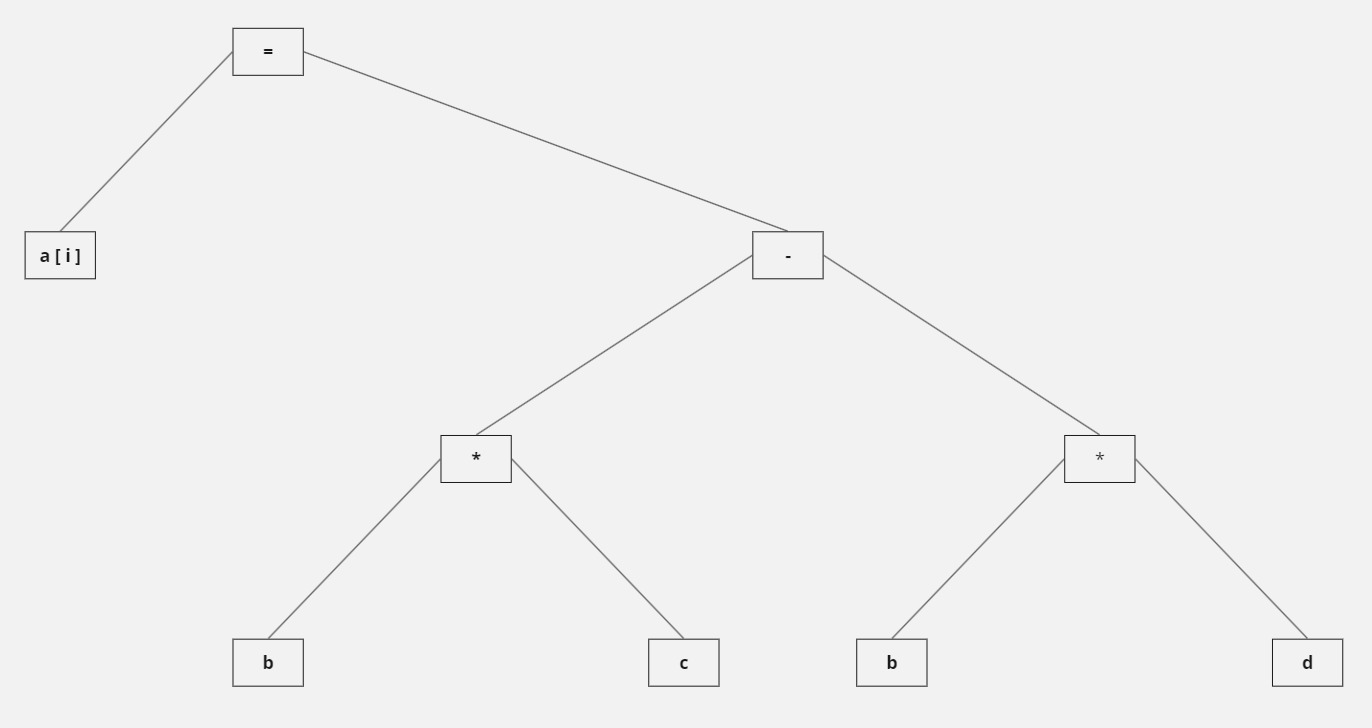
# 2.2 - Exercice 2

## Le code à 3 adresses

NOTE IMPORTANTE : Le ***"BeginFunc N"*** réserve la place pour les variables locales et temporaires du ***"main"***, ces variables sont ici ***"x", "y", "m2", "t1", "t2", "t3" et "t4"***. Puisque ces dernières sont toutes de type ***"int"* (4 octets),** 7 variables \* 4 octets = 28 octets qui représentent l’espace mémoire à réserver.

# 2.3 - Exercice 3

## Arbre syntaxique



## Quadruples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Op | Arg1 | Arg2 | result |
| (0) | \* | b | d | t1 |
| (1) | \* | b | c | t2 |
| (2) | - | t2 | t1 | t3 |
| (3) | := | t3 |  | a[ i ] |

## Triplets

|  |  |  |  |
| --- | --- | --- | --- |
|  | Op | Arg1 | Arg2 |
| (0) | \* | b | d |
| (1) | \* | b | c |
| (2) | - | (1) | (0) |
| (3) | := | a[ i ] | (2) |

# 2.4 - Exercice 4

## a) Code à 3 adresses

## b) Règles sémantiques de la production S -> for (S1 ; B ; S2) S3

# Retour sur le travail pratique

## Les éléments du travail plus problématiques

D’abord, pour ce qui est de l’exercice 1, j’ai eu un problème de fonctionnement dès le départ avec le modèle de code qui était donné par l’énoncé. Puisque ce modèle permettait d’éviter l’ajout d’un fichier lex (par sa fonction yylex() ) et puisque j’avais un problème de fonctionnement, j’ai décidé d’inclure un fichier lex (voir la section ***Annexe A***, en annexe, pour mon code initial) afin de pouvoir faire fonctionner l'analyseur. Les résultats de fonctionnement étaient bons, mais l'analyseur se terminait toujours avec le message *d'erreur* dans le terminal, à la suite des bonnes sorties. Ceci était occasionné par le fait d’appuyer sur la touche clavier *Enter*, lors de l’envoi du mot à l'analyseur. Cette action était mal gérée entre les deux fichiers lex et yacc. À la toute fin, j’ai copié mon code yac dans le fichier de code du modèle initial et tout fonctionnait ! Il a donc été possible de valider mon processus d’analyse de la grammaire, donc que les charactères de l’entrée passent chacun à travers les règles syntaxiques afin de retrouver la source (que j’ai spécifié avec ***%start S***), pour pouvoir valider le mot complet. Il s’agit donc d’une analyse syntaxique ascendante ici.

Deuxièmement, l’exercice 3 a) m’a fait hésiter pour ce qui est de l’assignation d’une valeur à un tableau, donc à une position du tableau. J’ai finalement choisi la représentation actuelle, après avoir regardé la diapositive 16 des notes de cours sur le « Code intermédiaire ».

# Annexe

## Annexe A – Première version de l’exercice 1 (À NE PAS UTILISER POUR LA CORRECTION)



Figure 2 - Fichier exercice1.lex (code initial)



Figure 3 - Fichier exercice1.yac (code initial)