

Projet d'Algorithmes pour la compilation

Devan SOHIER

2023-2024

Le but de ce projet est d'écrire en C une calculatrice effectuant sommes et produits.

1 Le programme que vous devez écrire

Vous devrez écrire un programme qui, une fois lancé, lit des opérations contenant sommes et produits de nombres flottants, et donne la valeur du résultat. Ce programme doit gérer les parenthésages dans ces opérations. La commande « q » doit permettre de quitter cette calculatrice.

Les nombres flottants peuvent être représentés comme des séquences de chiffres avec un éventuel point décimal ou en notation scientifique (un nombre flottant, le caractère « e », puis un nombre entier représentant l'exposant). On ne considère que des représentations en base 10, et que des nombres positifs. Les espaces sont ignorés.

2 Mode de fonctionnement

Cette calculatrice doit fonctionner comme un interpréteur, comprenant un analyseur lexical et un analyseur syntaxique LR. L'analyse lexicale permet de reconnaître les flottants, tandis que l'analyse syntaxique doit permettre de reconnaître et d'interpréter les opérations effectuées. Autant je demande que l'analyseur syntaxique soit LR, autant je vous laisse libre de la manière d'implémenter l'analyse lexicale (cela peut être un automate fini déterministe, mais elle peut aussi être fondue dans l'analyse lexicale).

Vous trouverez en figure 1 un exemple du comportement attendu de votre projet.

3 Les conditions du rendu de ce projet

Vous me remettrez un projet par groupe de un ou deux étudiants, par retour de mail, le 13 mars 2024 au plus tard. Ce mail contiendra deux pièces jointes portant les noms des étudiants du groupe :

- l'une, au format tgz, contenant un répertoire portant le même nom, le code source et un Makefile ;
- l'autre, au format pdf, lisible par Adobe Reader, contiendra votre rapport.

```
$> tar xvfz nom1_nom2.tgz
$> cd nom1_nom2
$> make
$> ./calc
> 5+4*3
17
> (5+4)*3
27
> 5+4*3+6*7
59
> 5.6 + 1.4e-1
5.74
> q
$>
```

FIGURE 1 – Tout rendu doit pouvoir être manipulé ainsi sous peine de ne pas être corrigé (nom1 et nom2 doivent bien sûr être remplacés par les noms des étudiants du groupe)

L'invocation de `make` doit produire un exécutable appelé `calc`. Chaque fichier source doit également contenir en commentaire le nom de ses auteurs. Votre code ne doit pas faire appel à des bibliothèques externes non-standard et doit compiler sous Gcc ou clang. Un exemple d'utilisation de votre projet est montré à la figure 1.

Un rendu ne satisfaisant pas ces conditions encourt le risque de ne pas être corrigé.

4 La notation de ce projet

L'élément essentiel de notation est le rapport. Il devra donc être soigné. Son introduction et sa conclusion devront contextualiser le projet dans le cadre de la chaîne de compilation et plus particulièrement de l'analyse lexicale et de l'analyse syntaxique, et non de votre situation personnelle¹.

Votre rapport ne doit pas être un manuel d'utilisation, et devra au contraire expliquer les choix que vous aurez opérés : vous devrez expliquer et argumenter les choix que vous avez faits, ainsi que tout autre parti pris d'implémentation. Votre rapport ne peut comporter d'éléments de code que de façon très limitée, comme support à vos explications.

En particulier, votre rapport devra présenter de façon détaillée et argumentée les structures de données utilisées, notamment les piles utilisées pour l'analyse LR : on peut en utiliser une stockant des structures de données (caractère, valeur associée, état de l'automate), ou plusieurs (une pour chacun de ces trois champs). Il devra également détailler les calculs préalables effectués pour construire les algorithmes utilisés (écriture d'expressions régulières, construction puis déterminisation de l'automate fini utilisé pour l'analyse lexicale si vous avez fait le choix d'en utiliser un ; grammaire utilisée, calcul de l'automate SLR, résolution d'éventuels conflits décalage-réduction dus à l'ambiguïté de la grammaire).

Les algorithmes ont été vus en cours, et je les connais ; l'élément techniquement intéressant (et donc noté) est la manière dont vous manipulez les structures de données nécessaires à les mettre en œuvre, et je ne lirai pas votre code pour comprendre cela.

L'utilisation de sources bien référencées est encouragée. Le plagiat (c'est-à-dire, vous attribuer le travail d'un autre, en l'utilisant sans le mentionner et sans le modifier substantiellement) est par contre proscrit, tant en matière de rapport que de code : il vous vaudra un 0, sans préjuger de procédures disciplinaires éventuelles qui s'ensuivraient et pourraient vous coûter jusqu'à une exclusion pendant 5 ans de l'enseignement supérieur.

La fonctionnalité de votre code sera bien sûr prise en compte (voir la section 3), ainsi que votre code, qui devra donc être lisible et structuré.

Sur la base d'un projet fonctionnel, il existe de nombreuses pistes d'amélioration (la liste ci-dessous n'est pas limitative) :

- gérer également les différences et division, ainsi que le moins unaire ;
- gérer diverses opérations supplémentaires telles que racines carrées, logarithmes, exponentielles...
- gérer des variables (affectation et utilisation).

Sans que cela soit impératif, il sera apprécié que vous en exploriez certaines (mais, j'insiste, sur la base d'un projet déjà fonctionnel).

Annexe : un analyseur SLR pour sommes et produits

$E \rightarrow$	$E + E$	$\{E.val = E^{(1)}.val + E^{(2)}.val\}$
	$E * E$	$\{E.val = E^{(1)}.val * E^{(2)}.val\}$
	(E)	$\{E.place = E^{(1)}.val\}$
	val	$\{E.place = \mathbf{val}.lex\}$

1. « Dans le cadre du cours formidable de M. Sohier que je suis parce que je veux devenir ingénieur... » : toute flatteuse qu'elle soit, et que je croie ou non à sa sincérité, ce n'est pas une bonne introduction. Je sais déjà que c'est dans le cadre d'un cours d'ingénierie, je peux m'intéresser à vos ambitions, mais pas dans le cadre d'un rapport. Ce qui m'intéresse c'est la manière dont vous replacez votre travail dans un contexte scientifique et technique. De même pour la conclusion en « *J'ai beaucoup appris de ce projet* » : outre que c'est mon rôle d'en juger, elle vous éloigne du problème, qui est de savoir comment on peut aller plus loin dans la thématique de ce projet.

	+	*	()	val	¬	<i>E</i>
0			d2		d3		1
1	d4	d5				acc	
2			d2		d3		6
3	r4	r4		r4		r4	
4			d2		d3		7
5			d2		d3		8
6	d4	d5		d9			
7	r1	d5		r1		r1	
8	r2	r2		r2		r2	
9	r3	r3		r3		r3	