

## Compte rendu : Recherche sur le Parsers

### Pouquoi les parser ?

Les parsers permettent de vérifier qu'une certaine expression suit une grammaire définie au préalable.

Cela nous sera très utile pour analyser les requêtes SQL et ne laisser passer que celles qui suivent la grammaire imposée par MySQL. Ainsi nous pourrions déjà droper certaines requêtes rien qu'en les lisant.

### Résultat des recherches

Il existe différents types de grammaires. Nous nous sommes donc intéressés à ces différents types et en avons conclu que nous devrions utiliser un parser pour context-free grammars.

Ils existent différents algorithmes pour parser ce type de langage dont le LL-parsing, le LR(1)-parsing et LALR(1)-parsing. Cependant aucun de ces algorithmes ne permettent de parser tous les context-free language. Il faudra donc s'assurer que la grammaire en entrée n'est pas ambiguë et ne possède pas de conflit interne (shift/reduce conflicts et/ou shift/shift conflicts).

### Outils utilisés

Il est compliqué de programmer à la main un parser. C'est pourquoi, nous utiliserons Lex & Yacc pour générer le parser. Yacc gère les LALR(1) grammars et génère en sortie du code C. Il est également possible de l'utiliser pour créer un parser en C++.

Lex n'est pas open source mais certaines de ses versions le sont, notamment flex que nous utiliserons. Yacc est quant à lui un outil open source.