

Pour en finir avec l'analyse syntaxique

1 Reprise sur erreur en général

Question 1-1 Caractérissez précisément la situation d'un analyseur LL en cas d'erreur de syntaxe. Discutez, en fonction des erreurs possibles, de ce qui est juste (il faudrait le garder) et faux (il faudrait le jeter) dans cette situation.

Question 1-2 Même question pour un analyseur LR.

Question 1-3 Donnez un exemple (parmi plein) de tentative de reprise sur erreur qui boucle. Comment garantir que cette situation n'arrivera pas ?

2 La reprise sur erreur en yacc/bison

Yacc fournit le token `error`, que l'on peut placer n'importe où dans une grammaire, typiquement suivant l'exemple suivant (éhontément coupé de la doc de bison) :

```
stmts: /* empty string */
      | stmts '\n'
      | stmts exp '\n'
      | stmts error '\n'
```

Sa sémantique est en gros : je m'attends à des erreurs dans `exp`, dans ce cas c'est pas grave de perdre `exp` (que le token `error` remplace) mais essaye de reconnaître `stmts` tout de même pour pouvoir passer à la suite.

Question 2-1 Essayez de formaliser le fonctionnement de l'analyseur qui permet ce comportement. Naturellement il faudra tenir compte de ce qu'il peut y avoir des `error` un peu partout.

Question 2-2 Quelles sont les stratégies à mettre en œuvre par le rédacteur de la grammaire pour une maximiser les chances de reprise sur erreur et leur pertinence dans un langage de type Pascal ?

Question 2-3 Cela colle-t-il avec votre expériences des erreurs en `gcc` ?