

Compiladores II

Fabio Mascarenhas - 2014.2

<http://www.dcc.ufrj.br/~fabiom/comp2>

SmallLua

```
bloco <- stat* (ret / '')
stat <- "while" exp "do" bloco "end" / "local" "id" "=" exp /
      "id" "=" exp / "function" "id" "(" (ids / '') ")" bloco "end" /
      "if" exp "then" bloco ("else" bloco / '') "end" /
      pexp "(" (exps / '') ")"
ret <- "return" exp
ids <- "id" ("," "id")*
exps <- exp ("," exp)*
exp <- lexp ("or" lexp)*
lexp <- rexp ("and" rexp)*
rexp <- cexp (rop cexp)*
cexp <- aexp ".." cexp / aexp
aexp <- mexp (aop mexp)*
mexp <- sexp (mop sexp)*
sexp <- "-" sexp / "not" sexp / "false" / "true" / "number" /
      "string" / lmb / pexp
lmb <- "function" "(" (ids / '') ")" bloco "end"
pexp <- "(" (" exp ")" / "id") "(" (" (exps / '') ")" )"*
rop <- "<" / "==" / "~="
aop <- "+" / "-"
mop <- "*" / "/"
```

Dojo

- Completem o parser de SmallLua que está no arquivo sml.lua, adicionando as regras que faltam na gramática
- O resultado do parser deve ser uma árvore sintática abstrata, incluindo informação de posição (para mensagens de erro)