

Slogo

Escrever, utilizando as ferramentas *flex* e *bison*, um interpretador para a linguagem de programação gráfica *SLogo* [1, 2] (“Simple Logo”) que passamos a descrever (produções):

entrada	→	ε entrada linha
linha	→	'\n' comandos '\n'
comandos	→	comando-simples comando-estruturado
comando-simples	→	movimento posicionamento ambiente
comando-estruturado	→	REPETE NAT '(' lista-comandos ')'
lista-comandos	→	comando-simples comando-simples ',' lista-comandos
movimento	→	cmov NAT cmov
cmov	→	AVANCE RECUE ESQUERDA DIREITA AV RC EQ DT
posicionamento	→	FIXAX NAT FIXAY NAT FIXAXY NAT NAT FIXADIRECÇÃO NAT CENTRAR ALEATÓRIO FX NAT FY NAT XY NAT NAT FD NAT
ambiente	→	LÁPISLEVANTADO LÁPISEMBAIXO CORDOLÁPIS NAT CORDOFUNDO NAT APAGAR LL LB CL CF

Notas:

- Com a linguagem *Logo* pretendeu-se desenvolver um ambiente de iniciação à programação para crianças através de um ambiente de programação gráfico, isto é, todos os comandos têm um resultado gráfico dado pelo mover de um «lápiz» (originalmente uma «tartaruga») num plano, desenhando figuras esquemáticas.
- No caso dos comandos de movimento estes têm duas formas, uma com um argumento, e outra sem argumento, neste último caso a movimentação será dada por uma valor previamente definido.
- Como fica claro pelas produções muitos dos comandos têm uma versão abreviada (duas letras).
- Os argumentos são sempre números naturais positivos. No caso das rotações os valores dados podem ser considerados em graus. No caso das cores os valores podem ser considerados combinações «rgb», podendo, eventualmente, ser designadas por nomes apropriados.
- O programa final deve poder ser usado (ou ter versões distintas) como um interpretador, caso em que vai processar os comandos linha à linha, ou como um processador que lê um programa completo de um ficheiro e o processa.
- Deve ser feito um tratamento de erros que permita ao utilizador continuar a usar o interpretador mesmo depois de introduzir um comando errado.
- A representação gráfica da execução do programa [3] deve ser feita tendo por base as rotinas seguintes, `xdriver.h`, `xdriver.c`. Para um exemplo de utilização veja o programa `exemplo.c`. Estes ficheiros estão disponíveis a partir da página da disciplina.

Referências

- [1] Martin Lesser. *Programação em Logo*. Editorial Presença, Lda, Lisboa, 1987.
- [2] Seymour Papert. *Logo : computadores e educação*. Brasiliense, São Paulo, 3a edition, 1988.
- [3] Douglas A. Young. *X Window system: Programming and applications with Xt*. Prentice-Hall, 1989.