

ANALISADOR SINTÁCTICO - O QUE É?

- Autómato finito
- Usa gramática para verificar sintaxe do código fonte
 - Input = Tokens
 - Output = APT (Abstract Parse Tree)

ACÇÕES SEMÂNTICAS

- Acções activadas aquando da escolha de uma produção (regra gramatical)
- Local adequado para executar o “trabalho” do compilador
 - Execução imediata (só para interpretadores)
 - Produção de estruturas de dados (nós da APT)

PRODUÇÃO DE NÓS

- requer “tipagem” dos símbolos da gramática
 - tipo *grande* (classe...) para regras
 - tipo *pequeno* (construtor...) para produções
- assume construção “bottom-up”

TIPOS DE NÓS

```
. . .  
%union {  
    double val;  
    char *name;  
    calc_t_exp exp;  
    calc_t_seq seq;  
}
```

```
%token <val>NUM
```

```
%token <name>ID
```

```
. . .
```

```
%type <exp>exp
```

```
%type <seq>seq
```

ABSTRACT PARSE TREE

- Representa o input, em forma de árvore
- Estrutura próxima da sintaxe concreta, mas mais leve
- Muito ligada às regras e produções da gramática
- Descrição pode ser ambígua
 - Não é usada para fazer parse
 - Usa o parse já efectuado

SINTAXE ABSTRACTA

- Simplificação das “classes” do input
- Várias opções de representação

APT - CONVENÇÕES DE CONSTRUÇÃO

- Depende sempre da gramática
- Escolher uma representação
 - Eficaz
 - Completa
 - Conveniente

APT - LINGUAGENS DE PROGRAMAÇÃO

- Estruturas a representar:
 - Literais
 - Identificadores (variáveis, funções)
 - Expressões
 - Tipos
 - Instruções (statements, declarações)

APT - OUTRO EXEMPLO

120 + 3 * 2 + 1; b = 9 + 2 + 3; c = (b + 1) * b

