1. Compiladores #teórico-prático

semântica
idiomas são ideias expressas por frases, sequências gramaticalmente corretas de palavras, sequencias válidas de letras
léxico

comunicação tem muitas vezes um efeito, seja uma resposta à mensagem ou o despoletar de uma ação

Linguagens de programação partilham características com idiomas, mas
- não podem ter ambiguidade
- ações provocam mudança de estado

parser
análise sintática
—.—
lexer
análise léxica

compiladores | compreensão, interpretação ou tradução

interpretação | execução feita instrução a instrução

tradução | traduzir código fonte em código + próximo do hardware do SO
comum em linguagens de alto nível

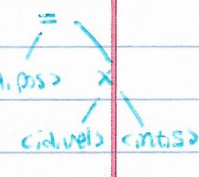
análise léxica | conversão de sequência de caracteres em sequência de elementos lexicais → <tokenName, value>

elimina comentários, espaços em branco...
—.—
nomes reservados da linguagem têm tokens próprios!

pos = vel * S, <id, pos> <=> <id, vel> <*> <int, S> <;>

sintática | validação da conformidade da sequência de elementos lexicais com a estrutura sintática da linguagem

árvore sintática | aproximação à estrutura formal
não inclui alguns elementos lexicais
define sem ambiguidade de ordem



semântica | verificações estáticas, i.é, em tempo de compilação
utiliza, para além da árvore sintática, a tabela de símbolos
verificar se identificador foi declarado array associativo

síntese | após a validação o código é traduzido, interpretado, ou indicado a sua validade

Linguagem | conjunto de símbolos e uma forma válida de descrever sequências válidas dos mesmos com total objetividade

definida em
- **extensão** | enumera todas as possíveis ocorrências
- **compreensão** | formalismos da def. de conjuntos
  - $\{x : p(x)\}$ $\quad \{x \mid p(x)\}$
  - $p(x)$ ↳ predicado sobre a variável
  - x ↳ qualquer elemento do conjunto

conjunto é definido por todos os valores de x para os quais $p(x)$ é verdadeiro

$= \emptyset$ (vazio)
$\{\}, \{\varepsilon\}, A$ e A^* são linguagens válidas

Componentes, termos e designações

símbolo | unidade atómica

alfabeto | conjunto finito não vazio de símbolos

palavra | sequência de símbolos de um dado alfabeto
- ε palavra vazia
- sub-palavra sequência contígua de 0 ou + símbolos de v
- prefixo sequência contígua de 0 ou + símbolos iniciais de v
- sufixo sequência contígua de 0 ou + símbolos terminais de v

fecho | conjunto de todas as palavras definíveis sobre um alfabeto
representa-se por A^* dado o alfabeto A

linguagem | conjunto de palavras consideradas válidas em A
$L \subseteq A^*$

operações sobre palavras

$|v|$ comprimento
número de símbolos

v^R reverso
$v = \{v_1, v_2, \ldots, v_n\}$ $\quad v^R = \{v_n, \ldots, v_2, v_1\}$

$v.v$ concatenação ou produto
símbolos de v seguidos dos de V

v^n potenciação
n réplicas de v

$v^0 = \varepsilon$ $\quad |\varepsilon| = 0$
VIP!

propriedades da concatenação
- $|v.v| = |v| + |v|$
- $v.(v.w) = (v.v).w$ associatividade
- $v.\varepsilon = \varepsilon.v = v$ elemento neutro
- $|v| > 0 \wedge |v| > 0 \Rightarrow v.v \neq v.v$ não comutatividade

operações sobre linguagens

$L1 \cup L2$ **reunião**
$= \{u : u \in L1 \vee u \in L2\}$

$L1 \cap L2$ **interseção**
$= \{u : u \in L1 \wedge u \in L2\}$

$L1 - L2$ **diferença**
$= \{u : u \in L1 \wedge u \notin L2\}$

$\bar{L}$ **complementação**
$= \{u : u \notin L\} = A^* - L$

$L1.L2$ **concatenação**
$= \{uv : u \in L1 \wedge v \in L2\}$

L^n **potenciação**
$L^0 = \{\varepsilon\}$ $\quad L^{n+1} = L^n . L$

L^* **fecho de Kleene**
$= L^0 \cup L^1 \cup L^2 \cup \ldots$
$= \bigcup_{i=0}^{\infty} L^i$
$= \{\varepsilon\} \cup L$

obs para $n > 1$. $L^n \subseteq L$

reunião $\cup$ conjuntos
disjunção $\vee$ predicados

não é requerido que as duas linguagens estejam definidas sobre o mesmo alfabeto

gramática

descreve linguagens por compreensão recorrendo a representações formais e (muitas vezes) recursivas (descreve as seq válidas)

$G = (T, N, S, P)$. $N \cap T = \emptyset$

T alfabeto terminal
(conjunto finito e não vazio de símbolos terminais)
N conjunto finito e não vazio de símbolos não terminais
S símbolo inicial , $S \in N$
P conjunto de regras

regras $\alpha \rightarrow \beta$
α pelo menos um símbolo não terminal
β cadeia de símbolos terminais e não terminais

símbolo terminal símbolo para o qual não existe regra que o possa alterar

hierarquia de Chomsky

tipo - 0 sem restrições
tipo - 1 $|\alpha| \leq |\beta|$ v $S \to \varepsilon$
tipo - 2 $|\alpha| = 1$
tipo - 3 $A \to cB$ v $A \to c$ v $A \to \varepsilon$
A, B não terminais c terminal

obs cada classe do tipo i contém as do tipo $i+1$, $i = 0, 1, 2, \ldots$

autómatos máquinas (algoritmos) capazes de reconhecer uma determinada taxonomia de gramáticas

Analisado na componente teórica