Satyrus III

Pedro Maciel Xavier

25 de agosto de 2020

Sumário

Capítulo 1

Introdução

- SATyrus é uma plataforma
- SATish é a linguagem

1.1 Motivação

1.2 Ficha Técnica

1.3 Implementação

1.4 Uso

Instalação

A instalação pode demandar privilégios de administrador.

```
$ git clone
$ cd Satyrus3
/Satyrus3$ sudo python3 setup.py install
```

```
C:\Users\User> git clone
C:\Users\User> cd Satyrus3
C:\Users\User\Satyrus3> python setup.py install
```

Execução

Escreva seu código em um arquivo de extensão ***.sat***.

```
$ satyrus script.sat
```

```
C:\Users\User> python -m satyrus script.sat
```

Capítulo 2

Conceitos Teóricos

2.1 Compiladores

Um compilador é um programa que transforma o código de um programa em um outro código, numa linguagem potencialmente diferente da linguagem de entrada[3]. O caso de uso mais comum se dá entre linguagens como *C* e *Fortran* que são traduzidas para o *Assembly*, permitindo expressar instruções de máquina dando como entrada expressões de mais alto nível, ou seja, mais próximas da linguagem natural. Uma outra aplicação recorrente para os compiladores é a otimização de programas. Neste caso, a saída pode estar escrita na mesma linguagem que a entrada, representando o mesmo programa, mas com uma sequência de instruções mais eficiente que o original.

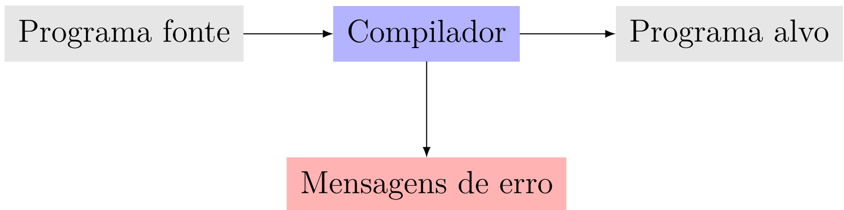

Figura 2.1: A estrutura básica de um compilador.

O processo inteiro de compilação se dá em diversas fases, listadas abaixo:

1. Análise Léxica

2. Análise Sintática

3. Análise Semântica

4. Geração de Código Intermediário

5. Otimização de Código

6. Geração de Código

2.2 Lógica Proposicional

A Lógica Proposicional é um sistema formal capaz de representar elementos do universo de discurso de maneira a evitar ambiguidades. A linguagem da Lógica Proposicional é constituída pela composição dos símbolos de um determinado alfabeto, que apresentamos a seguir.

Alfabeto

O Alfabeto Σ da Lógica Proposicional conta com símbolos para as variáveis, constantes e operadores, além dos símbolos auxiliares (parênteses) usados para alterar a precedência dos operadores. As constantes, que representam os valores Verdadeiro e Falso aparecem em diversas grafias, mas sempre como um par de símbolos. Os usos mais comuns são $\{1, 0\}$, $\{\mathsf{T}, \mathsf{F}\}$ ou até mesmo $\{\top, \bot\}$. Os operadores usuais são $\neg$, $\vee$, $\wedge$, $\rightarrow$ com extensão para o uso de $\leftarrow$ e $\leftrightarrow$. As variáveis são, em geral, denotadas por letras do alfabeto latino[1].

$$\Sigma = \{1, 0, \neg, \vee, \wedge, \rightarrow, \leftarrow, \leftrightarrow, x, y, z, ...\}$$

Tendo o alfabeto, dizemos que Σ^* é o conjunto de todas as palavras que podem ser formadas a partir de Σ. Vejamos uma definição formal. Seja Σ^k o conjunto das palavras de tamanho k formadas pela concatenação de

[1]Em alguns textos, principalmente em Lógica Matemática[**?**][**?**], vemos o uso de letras maiúsculas para representar as variáveis proposicionais, como P, Q e R. Já em publicações que tratam de Lógica Computacional[**?**][**?**] é comum que sejam usadas letras minúsculas nesse caso, como x, y, e z. Neste trabalho vamos nos ater a esta última variante.

símbolos em Σ, temos

$$\begin{aligned}
\Sigma^0 =& \{\Box\}\\
\Sigma^1 =& \Sigma\\
&\vdots\\
\Sigma^{k+1} =& \{r^\frown s : r \in \Sigma^k, s \in \Sigma\}\\
\\
\Sigma^* =& \bigcup_{k\geq 0} \Sigma^k
\end{aligned}$$

Onde escrevemos $\sigma_i^\frown \sigma_j$ para a concatenação entre as cadeias (*strings*) σ_i e σ_j e $\Box$ é uma cadeia vazia (a única cadeia de comprimento zero).

Fórmulas bem formadas

A linguagem em si é descrita pelo subconjunto $\mathscr{L} \subseteq \Sigma^*$, ou seja, está compreendida entre as possíveis combinações dos símbolos de seu alfabeto. Isto não é, contudo, suficiente para que a linguagem tenha o significado almejado. Vamos caracterizar $\mathscr{L}$ como sendo o conjunto das **Fórmulas bem formadas**. Uma Fórmula deste tipo é definida pelas afirmações que seguem. A definição se dá de maneira recursiva.

- Constantes e variáveis são fórmulas atômicas. Toda fórmula atômica é bem formada.
- Se α é uma fórmula bem formada, $\neg\alpha$ e (α) também o são.
- Se α e β são fórmulas bem formadas, $\alpha \vee \beta$, $\alpha \wedge \beta$ e $\alpha \to \beta$ também o são, assim como $\alpha \leftarrow \beta$ e $\alpha \leftrightarrow \beta$.
- Nada mais é uma fórmula bem formada.

2.3 Otimização

$$\min_{x\in X} f(x)$$

Otimização Inteira

Capítulo 3

Tipos

3.1 Números

3.2 Matrizes

3.3 Variáveis

Capítulo 4

Sintaxe do SATish

```
?epsilon: 1.5E-08;

m = 3;
n = 5;

x[m] = {(1) : 1, (m) : -1};
y[n] = {(1) : 0, (n) : +1};

#{---------------------}
* THIS IS SOME MULTI- *
* LINE COMMENT...     *
{---------------------}#

# Integrity Constraints
(int) A:
@{i=[1:m]}        # forall i from 1 to m
${j=[1:n], j > i} # exists j from 1 to n for j greater than i

x[i] → y[j];

# Optmality Constraints
(opt) X[1]:
@{i=[1:m]}         # forall i from 1 to m
@{j=[1:n], i != j} # forall j from 1 to n where i is different from j

```

```
x[i] & y[j];
```

4.1 Comentários

4.2 Diretivas

4.3 Atribuição

4.4 Matrizes

4.5 Definição de Restrições

Capítulo 5

Exemplos

5.1 Coloração de Grafos

Glossário

Referências Bibliográficas

[1] MONTEIRO, B. F. **SATyrus2: Compilando Especificações de Racioncínio Lógico**. Dissertação (Engenharia de Sistemas e Computação) - PESC/COPPE, UFRJ. Rio de Janeiro, 2010.

[2] BENEVIDES, M. **Apostila de Lógica**. Rio de Janeiro, 2015.

[3] AHO, Alfred e ULLMAN, Jeffrey, **Compiladores: Princípios, Técnicas e Ferramentas Livro**, 1986.