

# siC: Uma linguagem baseada em C incluindo pilha, lista e fila como tipos primitivos

Gabriella de Oliveira Esteves, 110118995

<sup>1</sup>Departamento de Ciência da Computação - Universidade de Brasília

## 1. Objetivo

Este trabalho visa projetar e construir uma nova linguagem chamada de siC - Structures in C, baseada na linguagem C. O siC acrescenta três estruturas de dados, lista, pilha e fila, como tipos de dados primitivos e, para manipulá-las, adiciona certas operações próprias para tal.

## 2. Introdução

Um compilador é um programa que recebe como entrada um código fonte e o traduz para um programa equivalente em outra linguagem [1]. Ele pode ser dividido em sete fases, ilustrado na Figura 1.

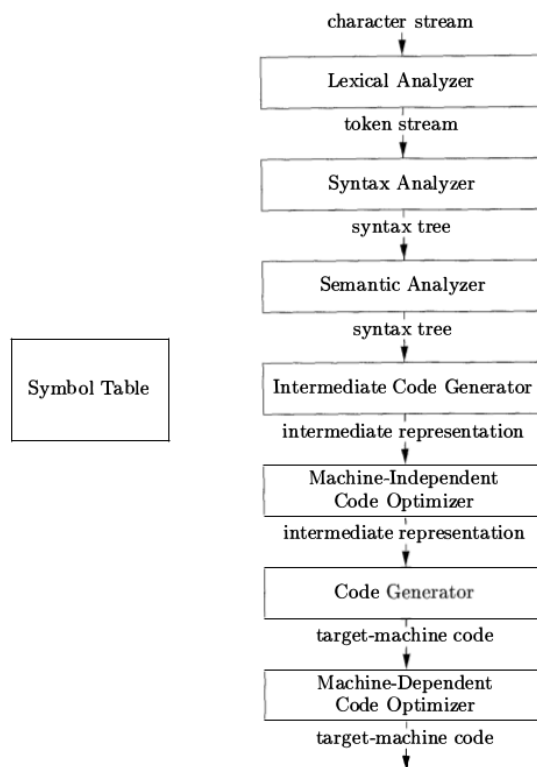


Figura 1. Fases de um compilador

- 1 **Analizador Léxico:** Lê o código fonte e atribui significado à cada sequência de caracteres, agora chamados lexemas. Cada lexema é mapeado para um token, que por sua vez é um par de nome (símbolo abstrato) e atributo (ponteiro para tabela de símbolos);

- 2 **Analizador Sintático:** Constrói uma representação gramatical dos tokens em forma de árvore;
- 3 **Analizador Semântico:** Utiliza a árvore sintática juntamente com a tabela de símbolos para verificar se a consistência semântica é mantida de acordo com a definição da linguagem.
- 4 **Gerador de Código Intermediário:** Converte árvore sintática anotada em código intermediário, com linguagem parecida com assembly e que possui apenas três operadores por linha de código. Nesse sentido, quebra-se estruturas complexas em estruturas mais simples, nesta fase.
- 5, 6 **Otimizador de Código Independente/Dependente de Máquina:** Procura aprimorar o código intermediário com o objetivo de melhorar o código-alvo de alguma forma: o deixando mais rápido, mais curto, consumindo menos energia, etc.
- 7 **Gerador de Código:** Converte o código intermediário no código-alvo, buscando atribuir os registradores às variáveis da maneira ótima.

O foco do projeto será nas fase 1, 2, 3 e 4, porém a princípio serão apresentadas apenas a descrição da linguagem siC, bem como uma breve descrição de sua semântica.

Dois grandes motivos sustentam a escolha do tema deste projeto. Primeiro, uma vez que as estruturas de dados pilha, fila e lista fazem parte dos tipos primitivos de uma linguagem, haverá menos manipulação de ponteiros na mesma, portanto erros envolvendo-os são menos prováveis de ocorrer. Segundo, a linguagem siC é mais alto-nível que C devido à abstração das estruturas de dados básicas, e, de maneira geral, pode ser mais *user-friendly*. Nesse sentido, o usuário (da linguagem) leigo deverá entender como cada estrutura funciona, bem como suas vantagens/desvantagens e usabilidade; porém a implementação de cada uma estará a cargo da própria siC.

### 3. Gramática

A seguir será apresentada a gramática da linguagem siC, que é unicamente baseada em C [2].

```
unary_operator
: ' & '
| ' * '
| ' + '
| ' - '
;
```

### Referências

- [1] A. V. Abo, M. S. Lam, R. Sethi, J. D. Ullman, *Compilers - Principles, Techniques and Tools* 2nd ed. 1986
- [2] ANSI C Yacc grammar, <http://www.quut.com/c/ANSI-C-grammar-y.html>, 18 12 2012.