

Histórico e Conceitos

Eduardo Ferreira dos Santos

Ciência da Computação
Centro Universitário de Brasília – UniCEUB

Julho, 2016

Sumário

- 1 Computadores e máquinas de computar
- 2 Compiladores
- 3 Linguagens de programação

1 Computadores e máquinas de computar

2 Compiladores

3 Linguagens de programação

Computador

- É possível pensar no computador, pelo menos nas primeiras versões, como uma máquina de computar;
- **Computar** também pode ser entendido como **calcular**;
- Os primeiros equipamentos científicos que conhecemos têm origem na mesopotâmia;
- Foi a primeira região a despertar para a ciência.

Ábaco

Soma e subtração simples.



Figura 1.1: Ábaco escolar ¹

¹<https://pt.wikipedia.org/wiki/Ficheiro:Kugleramme.jpg>

Máquinas de calcular [USP, 2016]

- Wilhelm Schickard (1592-1635) construiu a primeira máquina de verdade, que se perdeu na guerra dos trinta anos;
- Blaise Pascal (1623-1662) inventa a primeira calculadora que fazia somas e subtrações;
- Apesar de realizar as operações com sucesso, era cara e difícil de usar;
- Leibniz (1646-1716) aperfeiçoa o projeto de pascal para fazer também multiplicações e divisões, além de simplificar a operação;
- Joseph Marie Jacquard (1752-1834): tear mecânico que utilizava cartões perfurados.

Babbage e Ada

- Charles Babbage (1792-1871) inventa o que viria a ser o pai do computador: o **calculador analítico** [USP, 2016];
- Tinha alguns componentes que viriam a fazer parte da definição moderna de **computador**:
 - **Memória**;
 - Engenho central;
 - Engrenagens e e alavancas para **transferir** dados da memória para o engenho central;
 - Também fazia transferência para dispositivos de entrada e saída de dados;
 - Utilização de cartões perfurados.
- Ada Augusta (1815-1852) pode ser considerada a **primeira programadora**;
- Ada produziu instruções para o engenho analítico;
- Introduziu o conceito de **subrotina**.

Álgebra de Boole

- As máquinas do século XIX utilizavam a base **decimal**;
- George Boole (1815-1864) publicou em 1854 os princípios da **lógica booleana** [USP, 2016];
- As variáveis podem assumir somente valores **binários** (0 e 1);
- A álgebra de boole permitiu, pela primeira vez, **calcular** as configurações do computador;
- Dado um conjunto de entradas, seria possível determinar a saída antes da **computação**.

Máquina de Hollerith



Figura 1.2: Réplica da primeira máquina de tabulação Hollerith localizada no Museu da História da Computação nos EUA ³

Computadores na guerra [USP, 2016]

- Com a segunda guerra mundial aumentam as pesquisas na área;
- Nos EUA, a Marinha, a Universidade de Harvard e a IBM construíram o Mark I em 1944;
- Poderia ser considerada uma implementação do projeto de Babbage;
- Tinha como objetivo melhorar a precisão do cálculo balístico dos mísseis;
- O exército tocou em paralelo o projeto do ENIAC – *Eletronic Numeric Integrator And Calculator*
- O projeto era liderado pelos engenheiros John Presper Eckert (1919-1995) e John Mauchly (1907-1980);
- Ficou pronto apenas em 1946, vários meses depois do fim da guerra.

Mark I



Figura 1.3: Foto do computador Mark I em Harvard ⁴

- 1 Computadores e máquinas de computar
- 2 **Compiladores**
- 3 Linguagens de programação

Conceito de programação

- A ideia de programa de computador começa com a criação de **subrotinas** com Ada Augusta [USP, 2016];
- Seu sonho era conseguir implementar um **laço condicional**: carregar o cartão certo dada uma condição;
- Ao permitir que as máquinas fossem programáveis, ela introduziu um importante conceito da computação.

Formalização [Pinto, 2016]

Computação no dia-a-dia

- Um tipo de computador e um tipo de linguagem de programação;
- Um problema \rightarrow Programador (**você**) \rightarrow Um programa;
- Uma entrada para o programa.

Computação formalizada

- Autômatos finitos;
- Uma linguagem \rightarrow Programador (**você**) \rightarrow Um autômato;
- Uma palavra.

Linguagens

- Teoria de linguagens formais [UNICER, 2001]:
 - Estudo das características, propriedades e aplicações da linguagem formal;
 - Representação da estrutura: **sintaxe**;
 - Determinação do significado: **semântica**.
- É necessário estudar as linguagens formais no domínio da matemática.
 - *uma linguagem é uma forma de comunicação, usada por sujeitos de uma determinada comunidade;*
 - *uma linguagem é o conjunto de SÍMBOLOS e REGRAS para combinar esses símbolos em sentenças sintaticamente corretas;*
 - *uma linguagem é formal quando pode ser representada através de um sistema com sustentação matemática.*

Símbolo

- **Símbolo**: entidade abstrata sem definição formal;
- Ex.: letras, dígitos, etc.
- Ordenação lexicográfica [UNICER, 2001]: igualdade ou precedência;
- Usados como elementos atômicos em definições de sintaxe.

Alfabeto

- **Definição:** sequência finita de símbolos;
- **Exemplos:**
 - $\beta = \{0, 1\}$;
 - $\Gamma = \{a, b, c, d, e, f\}$.
- Uma **palavra** sobre um alfabeto β é uma sequência finita de símbolos de β .
- Ex.: (1, 1, 0, 0, 1) (tupla);
- Representamos apenas como 11001.

Alfabetos e palavras

- $|\rho|$ denota o número de símbolos da palavra $|\rho|$.
- Ex.: $|11001| = 5$
- Uma **linguagem** sobre um alfabeto β é um conjunto de palavras sobre β .
- Ex.: $L = \{1^p \mid p \text{ é primo}\} = \{11, 111, 11111, 1111111, \dots\}$

Computador formal

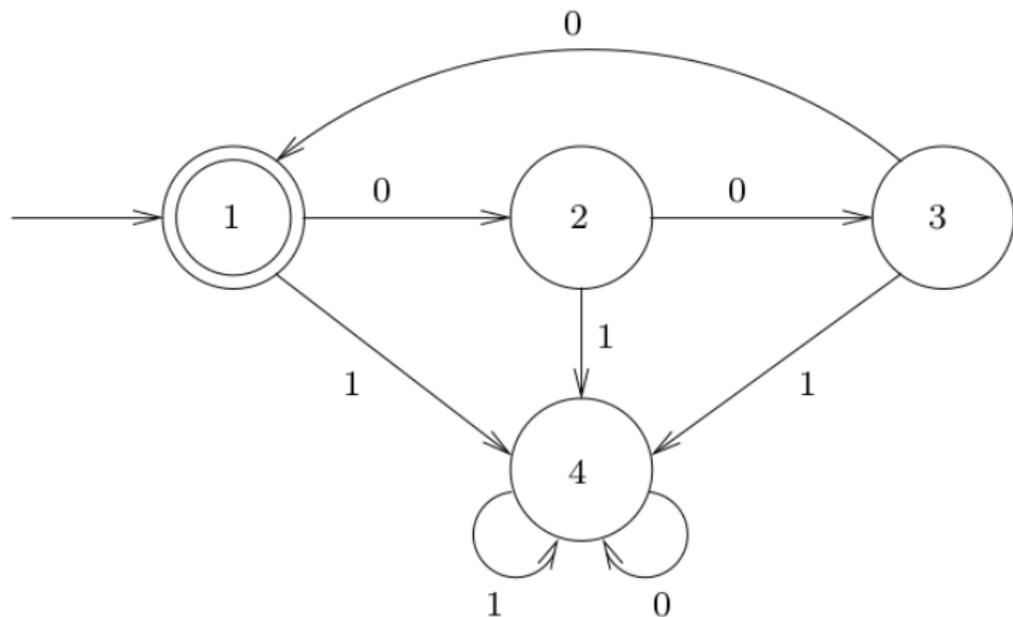


Figura 2.1: Exemplo de computador formal [Pinto, 2016]

Processadores de linguagem

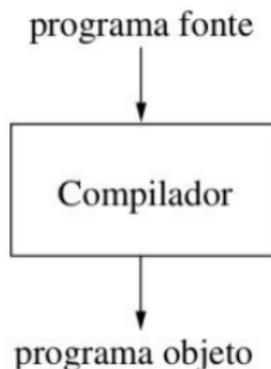


Figura 2.2: Um compilador
[Aho et al., 2007]

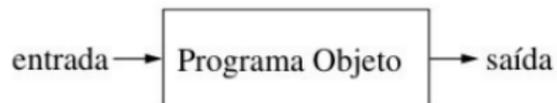


Figura 2.3: Executando o programa objeto [Aho et al., 2007]

Interpretador

- Ao invés de produzir linguagem de máquina, o interpretador **executa diretamente** as operações especificadas no programa fonte sobre as entradas do usuário;
- Normalmente o programa objeto é mais rápido;
- O interpretador oferece melhor diagnóstico de erros.

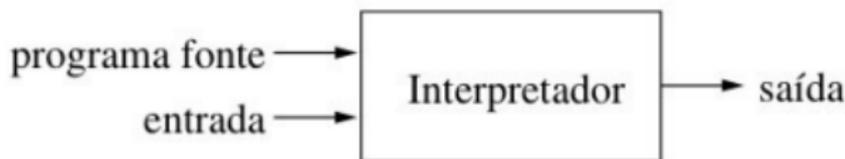


Figura 2.4: Um interpretador [Aho et al., 2007]

Compiladores Java

- O programa Java primeiro gera código intermediário: *bytecode*;
- Os *bytecodes* são interpretados por uma máquina virtual;
- Conceito de compilação universal.

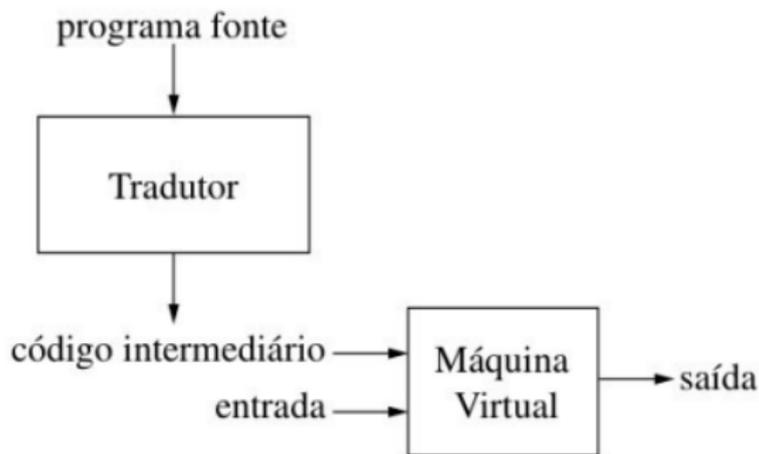


Figura 2.5: Um compilador híbrido [Aho et al., 2007]

Pré-processamento

- Alguns outros programas podem ser necessários para a geração do executável;
- O **pré-processador** é responsável por coletar o programa fonte e, possivelmente, expandir macros em comandos na linguagem fonte.

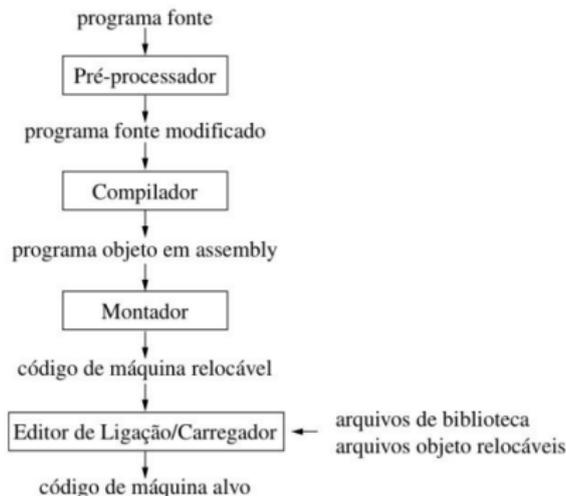


Figura 2.6: Um sistema de processamento de linguagem [Aho et al., 2007]

Modelo de análise e síntese

- A **análise** impõe um modelo gramatical para o código;
- Caso esteja sintaticamente mal formado ou semanticamente incorreto, deve informar qual é o erro;
- Gera também a **tabela de símbolos**, passada para a próxima etapa junto com a apresentação intermediária;
- A parte de **síntese** constrói o programa objeto a partir da tabela de símbolos e da representação intermediária;
- A compilação é organizada em fases, onde cada etapa transforma a representação anterior para a próxima camada.

Fases

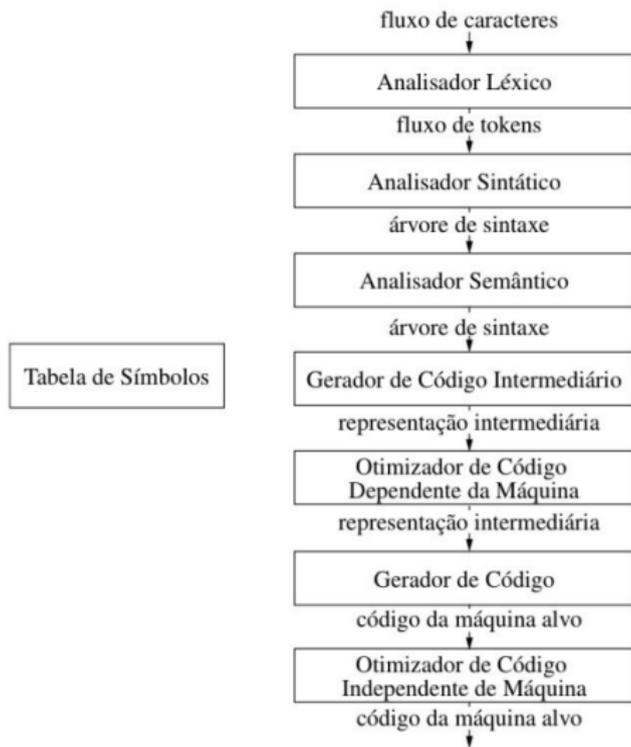


Figura 2.7: Fases do compilador [Aho et al., 2007]

- 1 Computadores e máquinas de computar
- 2 Compiladores
- 3 Linguagens de programação

Computadores e linguagens

- Um computador formal tem o objetivo principal de transformar **linguagem fonte** em **linguagem objeto**;
- A linguagem fonte é uma abstração de alto nível implementada no programa de computador;
- A linguagem objeto é o conjunto de símbolos que serão posteriormente lidos pelo processador;
- O programa objeto pode então ser chamado pelo usuário para processar entradas e produzir saídas [Aho et al., 2007].

Linguagens de programação

- No início eram traduções das instruções de máquina;
- Introdução às linguagens orientados ao cálculo numérico: Fortran, LISP e Cobol;
- Linguagens de **primeira geração**: linguagens de máquina;
- Linguagens de **segunda geração**: linguagens simbólicas ou de montagem, como Assembly;
- Linguagens de **terceira geração**: procedurais de alto nível, como Fortran, LISP, Cobol, etc;
- Linguagens de **quarta geração**: aplicações específicas, como NOMAD para relatórios;
- Linguagens de **quinta geração**: lógica com restrição, tipo Prolog e OPS5.

Classificações

- Linguagens imperativas;
- Linguagens declarativas;
- Linguagens de Von Neumann;
- Linguagens orientadas a objeto;
- Linguagens de *scripting*.

Impactos

Como as mudanças nas linguagens de programação afetam os compiladores?

OBRIGADO!!!
PERGUNTAS???

-  Aho, A., Lam, M., Sethi, R., and Ullman, J. (2007). *Compiladores—Princípios e Ferramentas*. Pearson, 2a. edition.
-  Penn (2016).
Foto do eniac.
-  Pinto, G. (2016).
Notas de aula do Prof. Guilherme Pinto.
-  UNICER (2001).
Apostila de compiladores.
-  USP (2016).
História do computador.