Compiladores
INTRODUÇÃO

Este documento tem alguns direitos reservados:

Atribuição-Uso Não-Comercial-Não a Obras Derivadas 2.5 Portugal
http://creativecommons.org/licenses/by-nc-nd/2.5/pt/

Isto significa que podes usá-lo para fins de estudo.

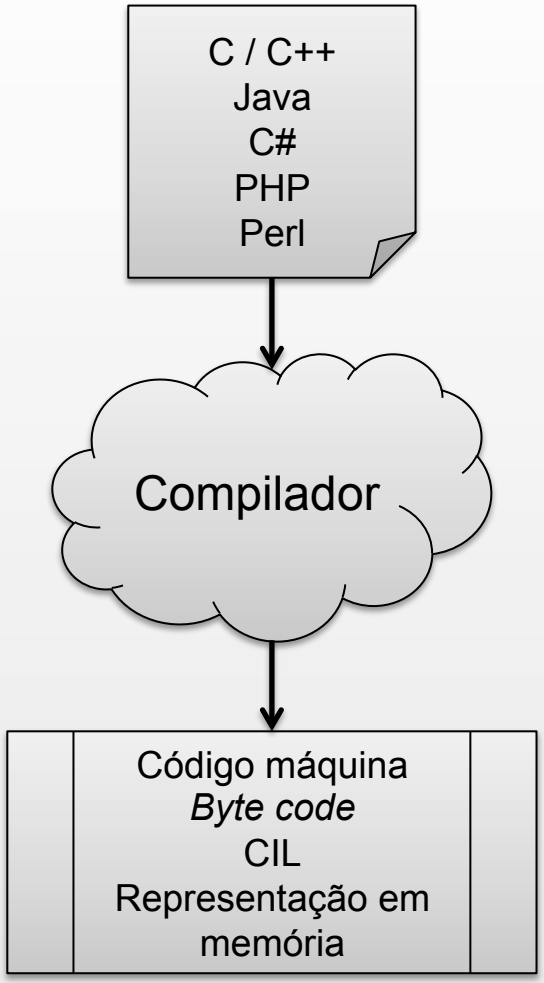
Para outras utilizações, leia a licença completa.
Crédito ao autor deve incluir o nome (“Pedro Freire”) e referência a www.pedrofreire.com.

INTRODUÇÃO

O que são

Um compilador prepara um ficheiro de texto com código-fonte para execução.

O resultado final pode ser outro ficheiro, preparado para execução direta pelo processador (código máquina) ou preparado para execução por uma máquina virtual dedicada *(byte code* ou CIL), ou uma representação em memória para interpretação.



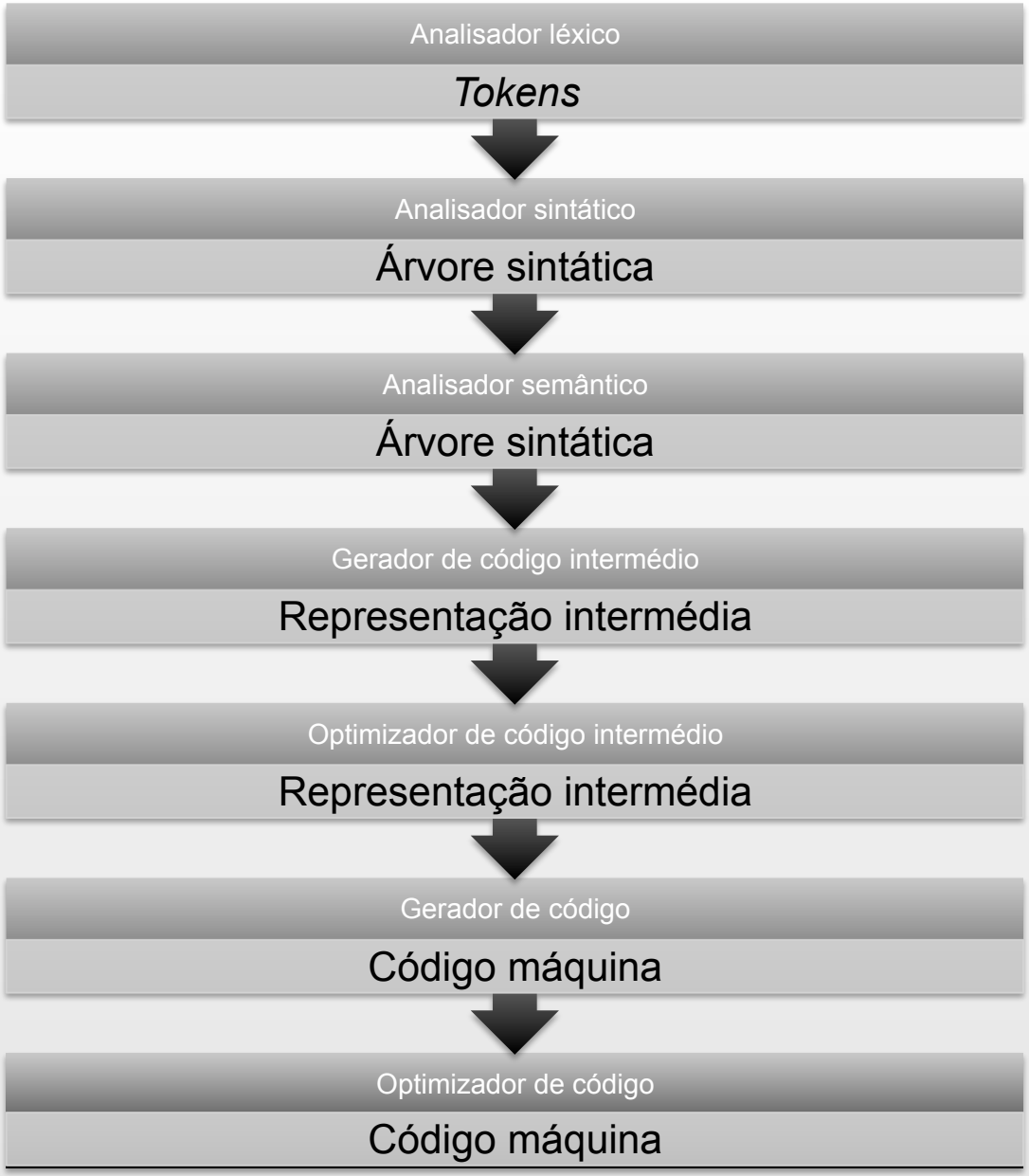
Exemplos de uso

- Compiladores
  - C / C++ / Objective C
  - Pascal
- Intérpretes/Interpretadores
  - JavaScript
  - PHP
  - XML / HTML / CSS
  - Rich Text Format (RTF)
  - Postscript
- Tradutores de linguagens
  - Java
  - C#
- Extractores de informação textual

Java é um exemplo de uma linguagem com um compilador que não gera código executável. Gera *"Java Byte Codes"* que são depois interpretados por outra aplicação.

Etapas

de compilação.

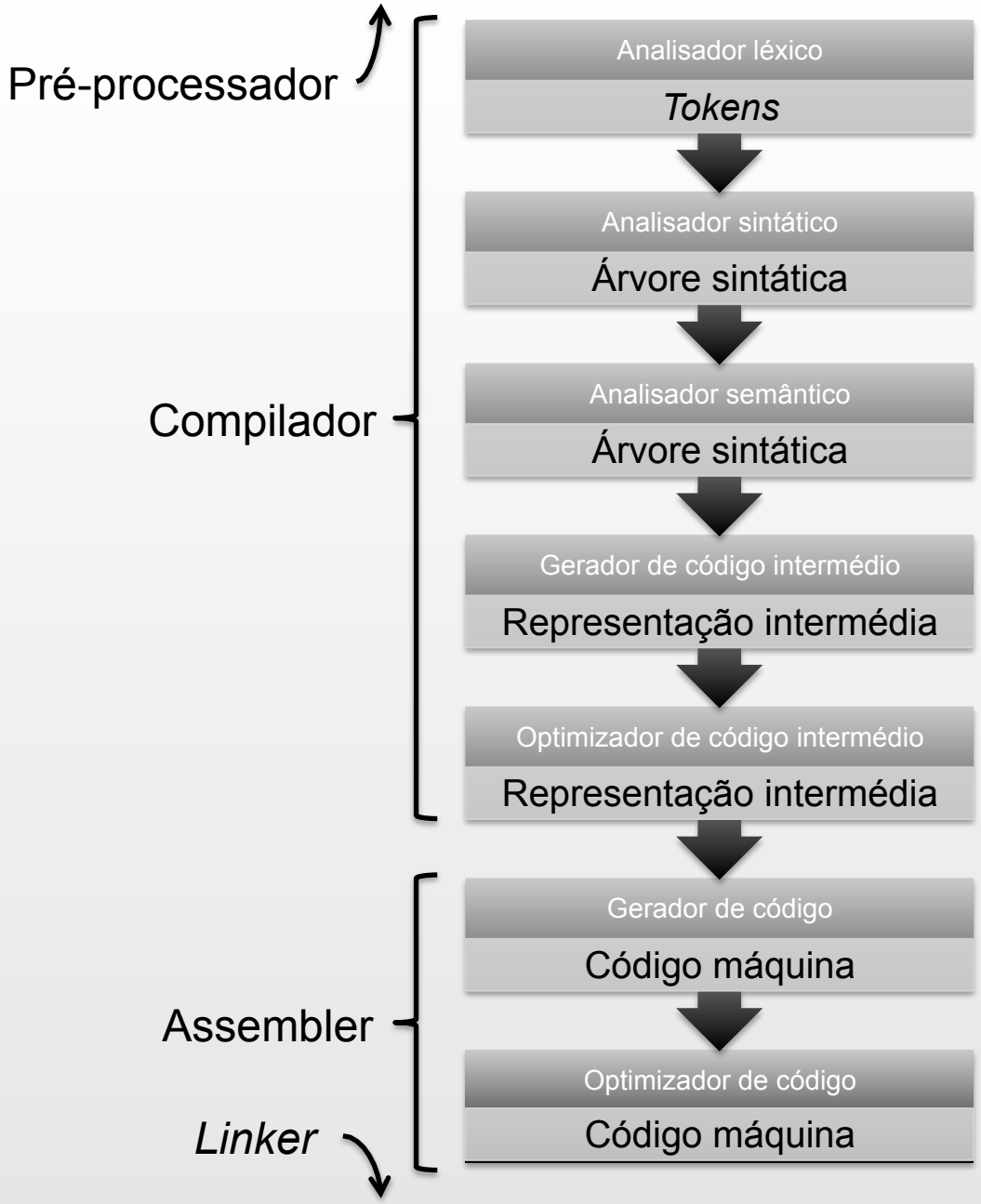


Etapas

de compilação.

O pré-processador transforma um ou mais ficheiros de código-fonte noutro ficheiro de código-fonte. É usado em linguagens como C para implementar `#include`, `#define` e outras diretivas semelhantes.

O *linker* liga um ou mais ficheiros de código máquina (ficheiros objecto, um por cada ficheiro de código-fonte) num último ficheiro executável pronto para execução.

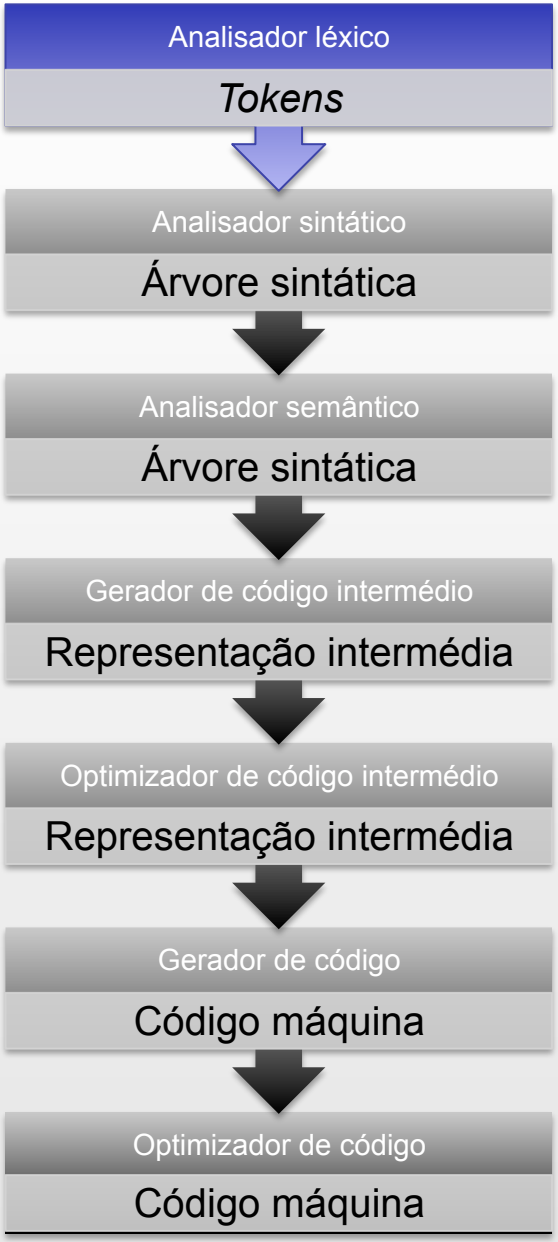


Etapas:
Analisador léxico

Também conhecido como *scanner*.

Lê a sequência de caracteres do ficheiro fonte original e agrupa-os em sequências significativas chamadas lexemas.

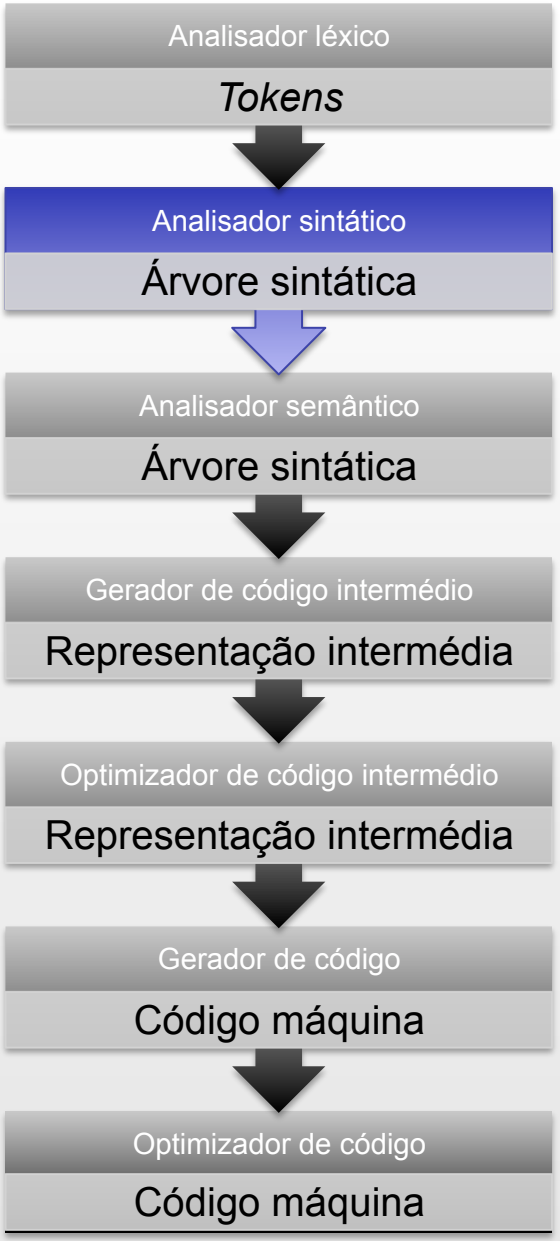
Constrói *tokens* para cada lexema que são um par com o nome do *token* e um atributo que o ajuda a especificar.



Etapas: Analisador sintático

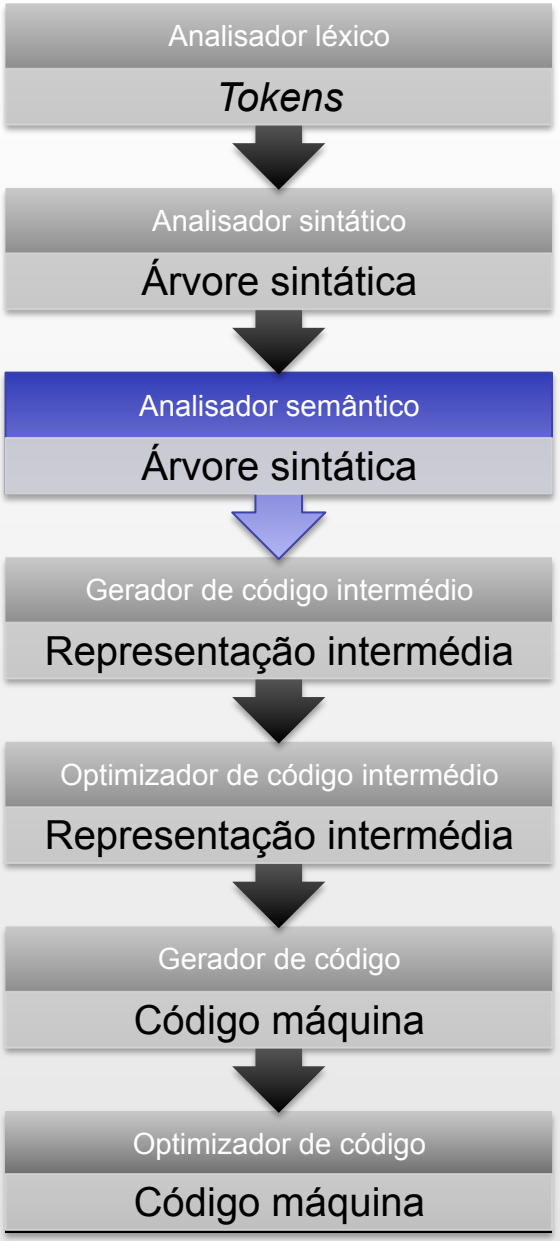
Também conhecido como *parser*.

Lê a sequência de *tokens* produzida pela etapa anterior e produz uma árvore que representa a sua estrutura gramatical.



Etapas:
Analisador semântico

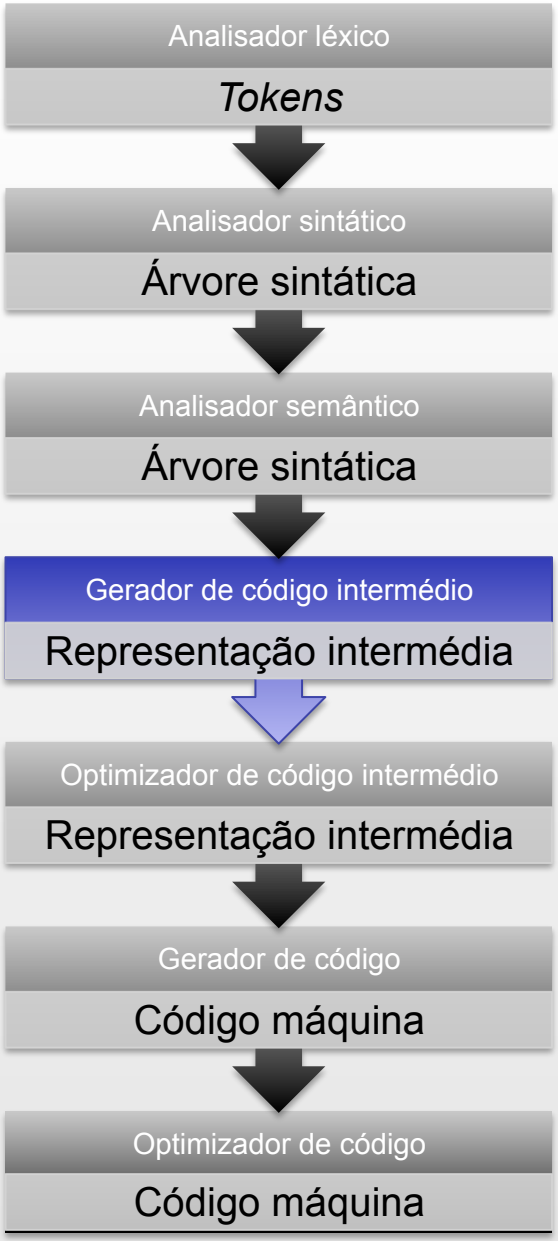
Lê a árvore gerada pela etapa anterior e verifica-a para consistência semântica de acordo com a definição da linguagem (e.g.: verificação de consistência e conversão de tipos de dados).



Etapas:
Gerador de código intermédio

Lê a árvore gerada pela etapa anterior e transforma-a numa sequência de instruções de uma máquina hipotética.

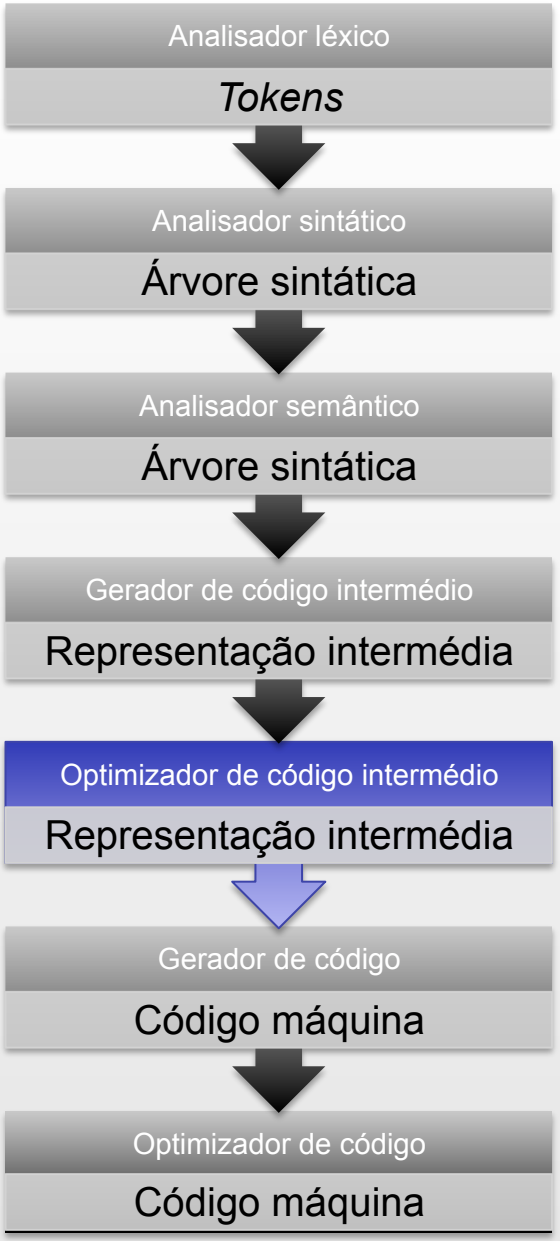
Esta máquina hipotética permite que as etapas seguintes sejam as mesmas para todas as versões do compilador para qualquer tipo de plataforma.



Etapas:
Optimizador de código intermédio

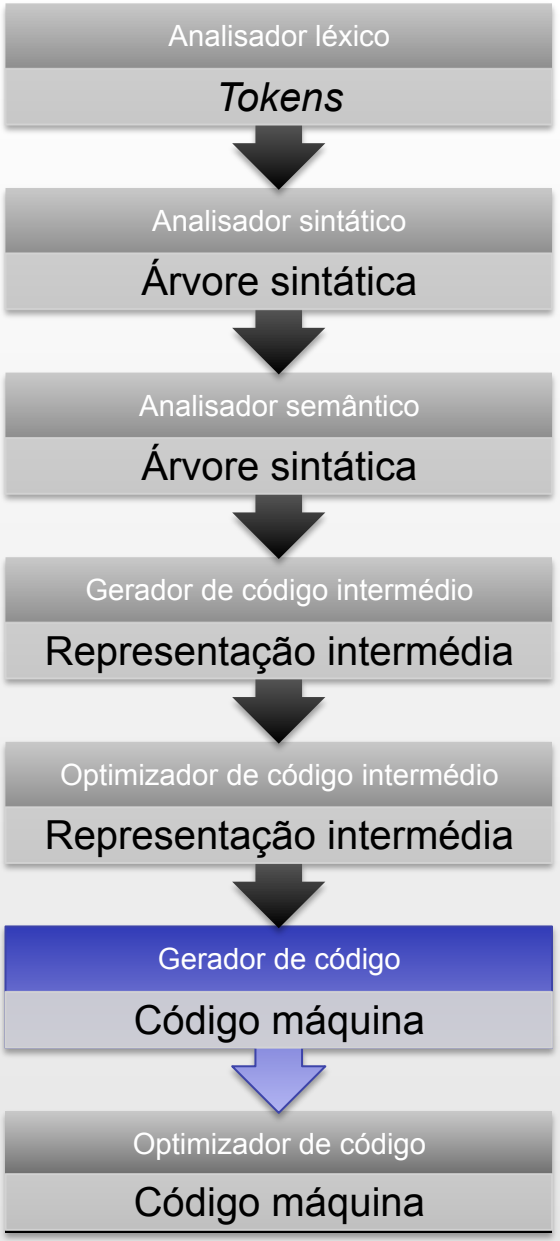
Lê a sequência de instruções gerada pela etapa anterior e optimiza-a de acordo com algum objectivo (código mais rápido ou mais pequeno).

Algumas das optimizações triviais executadas nesta fase são a remoção de variáveis temporárias desnecessárias geradas pela etapa anterior e cálculo antecipado de expressões e outras operações constantes.



Etapas:
Gerador de código

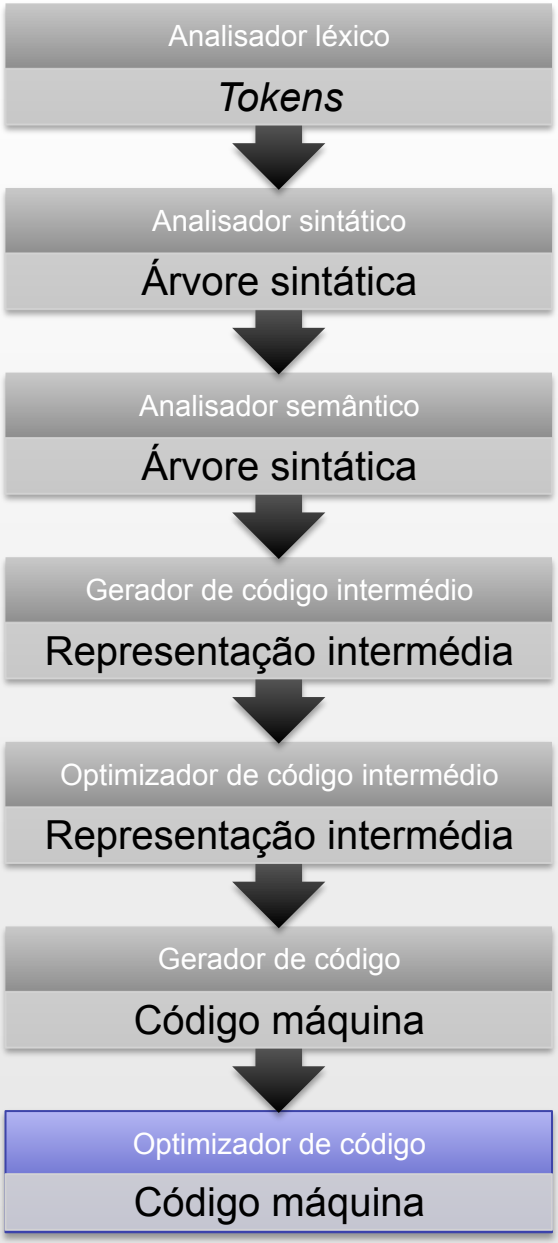
Lê a sequência de instruções gerada pela etapa anterior e transforma-a numa sequência de instruções da plataforma destino desejada.



Etapas:
Optimizador de código

Lê a sequência de instruções gerada pela etapa anterior e optimiza-a de acordo com algum objectivo (código mais rápido ou mais pequeno).

Algumas das optimizações triviais executadas nesta fase são a optimização do uso de registos do processador e a reordenação de instruções para aproveitar oportunidades de paralelização (e.g.: *pipelining)*.



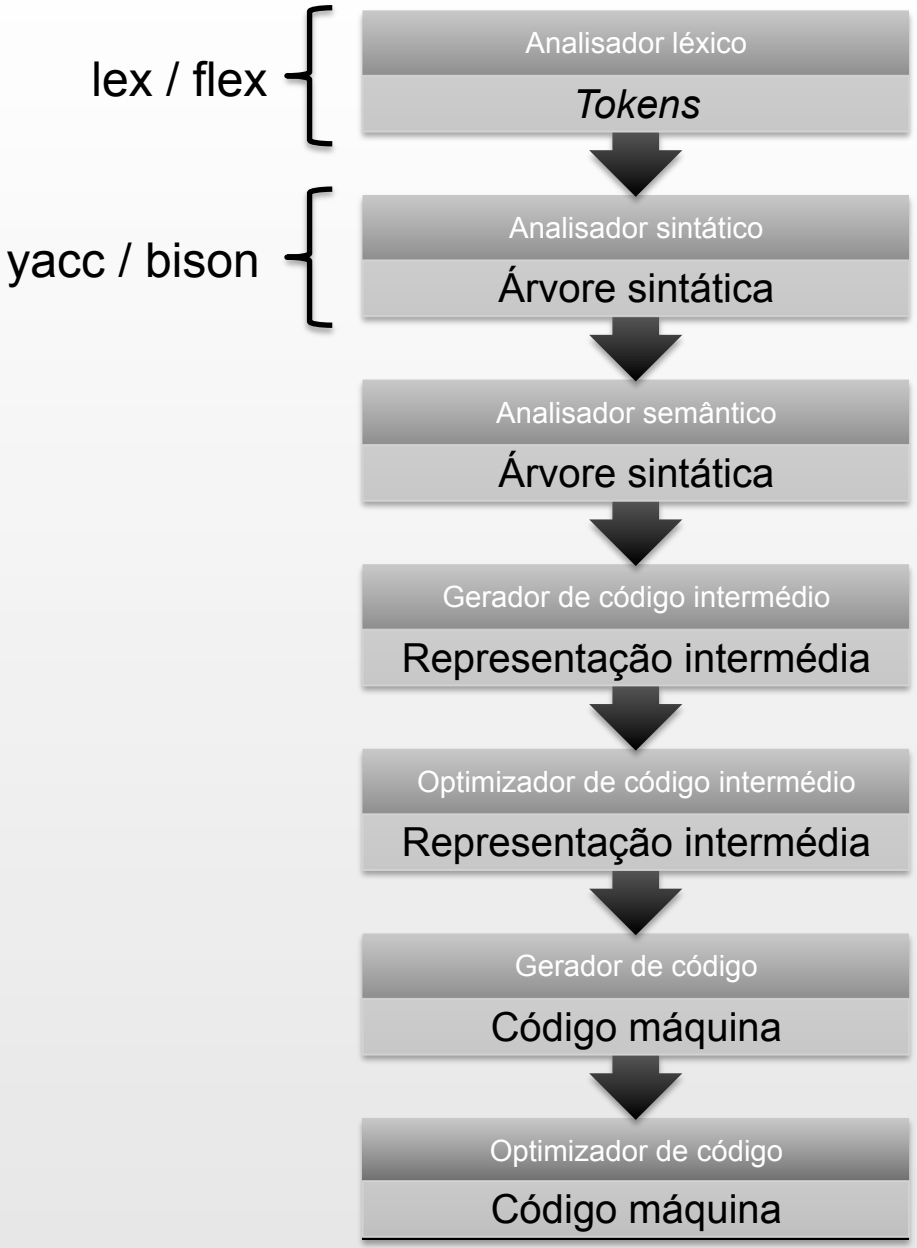
Ferramentas

de geração de compiladores.

O lex (ou a sua versão *open-source*, flex) gera *scanners*.

O yacc (ou a sua versão *open-source*, bison) gera *parsers*.

São ferramentas com décadas de uso, mas hoje em dia temos outras que nos geram em simultâneo um *scanner* e um *parser* (e.g.: ANTLR).



Hiperligações úteis

BIBLIOGRAFIA

- Compilers: Principles, Techniques and Tools – 2nd edition (2ª edição)
  - Alfred Aho, Monica Lam, Ravi Sethi, Jeffrey Ullman
  - Addison Wesley
  - http://dragonbook.stanford.edu
- Compiler Construction Toolkit
  - http://hackingoff.com/compilers/
- GNU Flex
  - http://www.gnu.org/software/flex/
- GNU Bison
  - http://www.gnu.org/software/bison/
- ANTLR
  - http://www.antlr.org