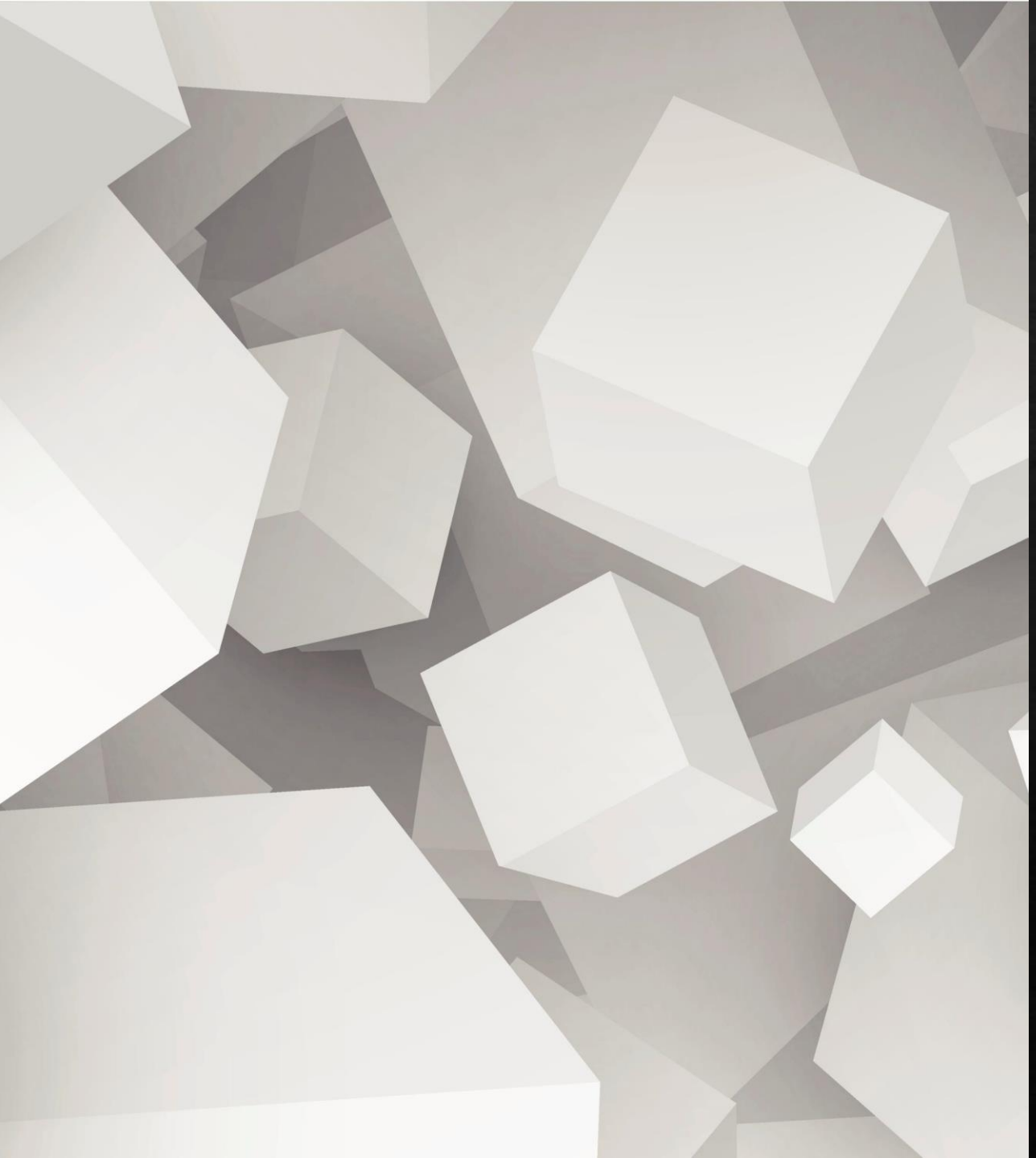
Cmpl

Guilherme Leite



Caraterísticas da Linguagem

- Uma linguagem intermediária entre Python e C
- Pouco complexa e
- Sem tipagem
- Não sensível a indentação
- Turing Completo

Uso de Python e C

Worldwide, Oct 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.49 %	+4.5 %
2		Java	19.57 %	-2.4 %
3		Javascript	8.4 %	+0.1 %
4		C#	7.35 %	-0.4 %
5		PHP	6.34 %	-1.2 %
6		C/C++	5.87 %	-0.4 %

Ranking da Linguagens mais estudadas por progamadores iniciantes

Programming Language	Ratings	Change
Java	16.028%	-0.85%
C	15.154%	+0.19%
Python	10.020%	+3.03%
C++	6.057%	-1.41%
C#	3.842%	+0.30%

Ranking de Linguagens mais usadas na prataforma GitHub em 2019 e a porcentagem de mudança em relação ao ano anterior.

Linguagem Intermediária

Mantendo características do Python, que é familiar para muitos programadores novos e adicionando algumas do C, o Cmpl pode ajudar iniciantes a aprender uma nova linguagem que, apesar de não ser a mais utilizada, tem uma importância muito grande e não é preferida pelos programadores novatos.

Exemplo de Código

test.cmpl :

```
a = 1;
b = 3;
c = scanf() ;
def soma(x, y) {
        z = x + y;
    return z;
 }
d = soma(a,b);
printf(d);
```

EBNF

A EBNF (extended Backus–Naur form) é usado para fazer uma descrição formal de uma linguagem com uma gramática livre de contexto. Uma EBNF consiste símbolos terminais e regras de produção não-terminais que são as restrições relativas como símbolos terminais podem ser combinados em uma sequência válida

A EBNF foi feita com base na do C, com mudanças para se encaixar a nova linguagem

Name	Syntax	Meaning
Alternatives	A | B	Either A or B
Kleene Star	A*	Zero or more instances of A
Kleene Cross	A+	One or more instances of A
Optional	A?	Zero or one instances of A

For example:

```
X ::= 'a' | 'b'+ | ('c' X)
```

matches:

```
a   b   bb   ca   cbbbbbbb
```

Análise Léxica e Sintática

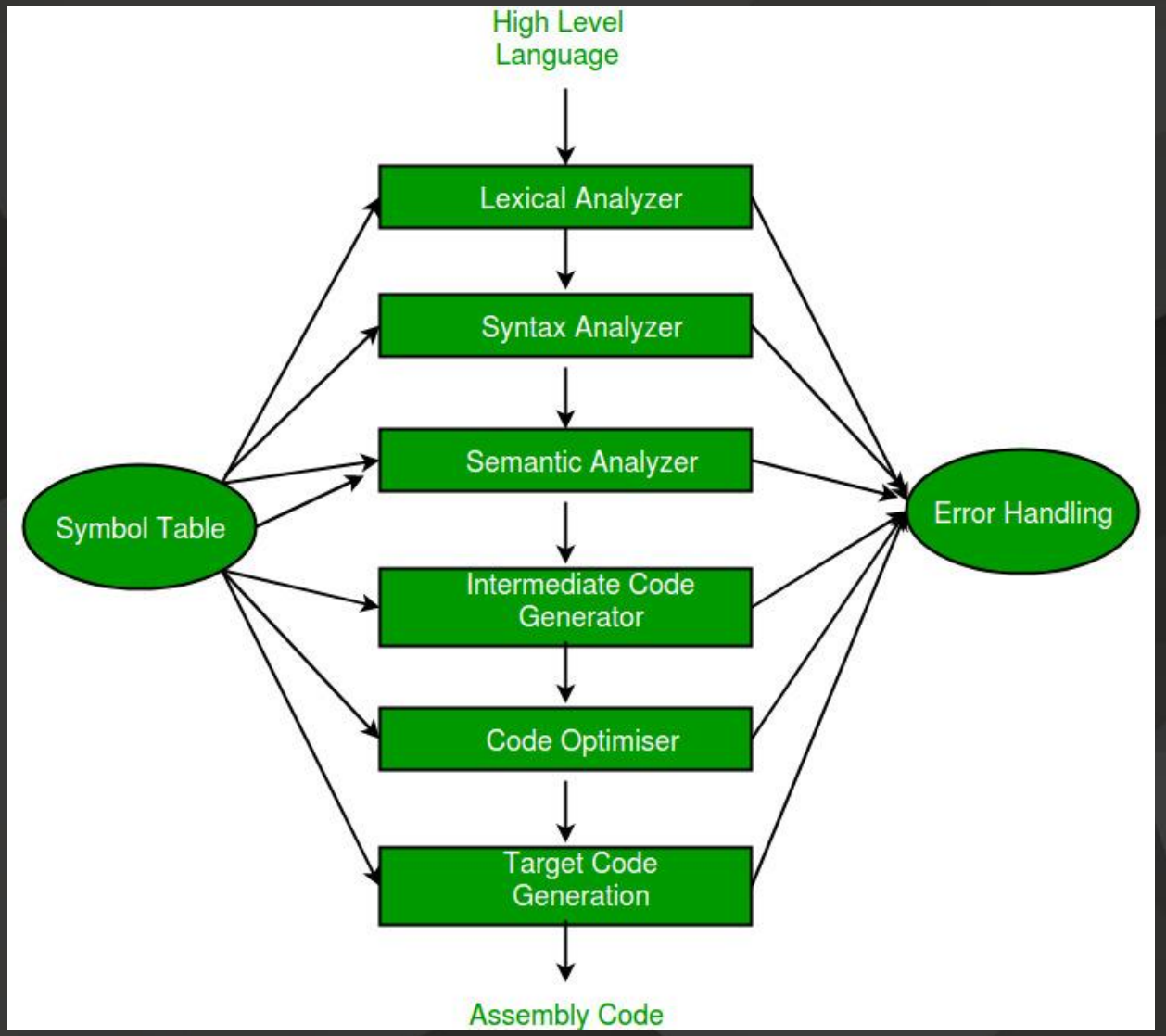
Flex & Bison

Flex e Bison são ferramentas para criar programas que lidam com entradas estruturadas. Eles eram originalmente ferramentas para criar compiladores, mas provaram ser úteis em muitas outras áreas.

O Flex cria *tokens* a partir do arquivo de entrada, enquanto o Bison gera um *parser* para a linguagem a partir da EBNF que lê a sequência de *tokens* e diz se ela corresponde a gramática definida anteriormente.

Compilador

O compilador foi feito usando como base um compilador de PHP implementado em Python, ele analisa o código de entrada .cmpl e cria *tokens* a partir dele, criando uma AST.



Fontes

https://www.gnu.org/software/bison/manual/

https://www.oreilly.com/library/view/flex-bison/9780596805418/ch01.html

https://stackify.com/popular-programming-languages-2018/

https://www.geeksforgeeks.org/top-10-programming-languages-of-the-world-2019-to-begin-with/

http://www.cs.man.ac.uk/~pjj/bnf/ebnf.html

https://karmin.ch/ebnf/index

https://cs.wmich.edu/~gupta/teaching/cs4850/sumII06/The%20syntax%20of%20C%20in%20Backus-Naur%20form.htm

https://www.lysator.liu.se/c/ANSI-C-grammar-y.html