モテる！
筋肉コンパイラ自作入門

～草食男子からの卒業～

自己紹介

戸塚佑太

(@totzyuta)

株式会社 OneBox

コーヒーを淹れていたい

兼 CTO (読み: 雑用)

みんなが一度は考える

モテたい…

だけどどうすれば…


コンパイラ！

コンパイラってつくれるの…？

つくっても意味なさそう…

コンパイラをつくろう

コンパイラをつくると...

1. 計算機が高級言語をどのように解釈していくのか理解することができる
2. 計算機自体やOSの仕組みについても考えれる

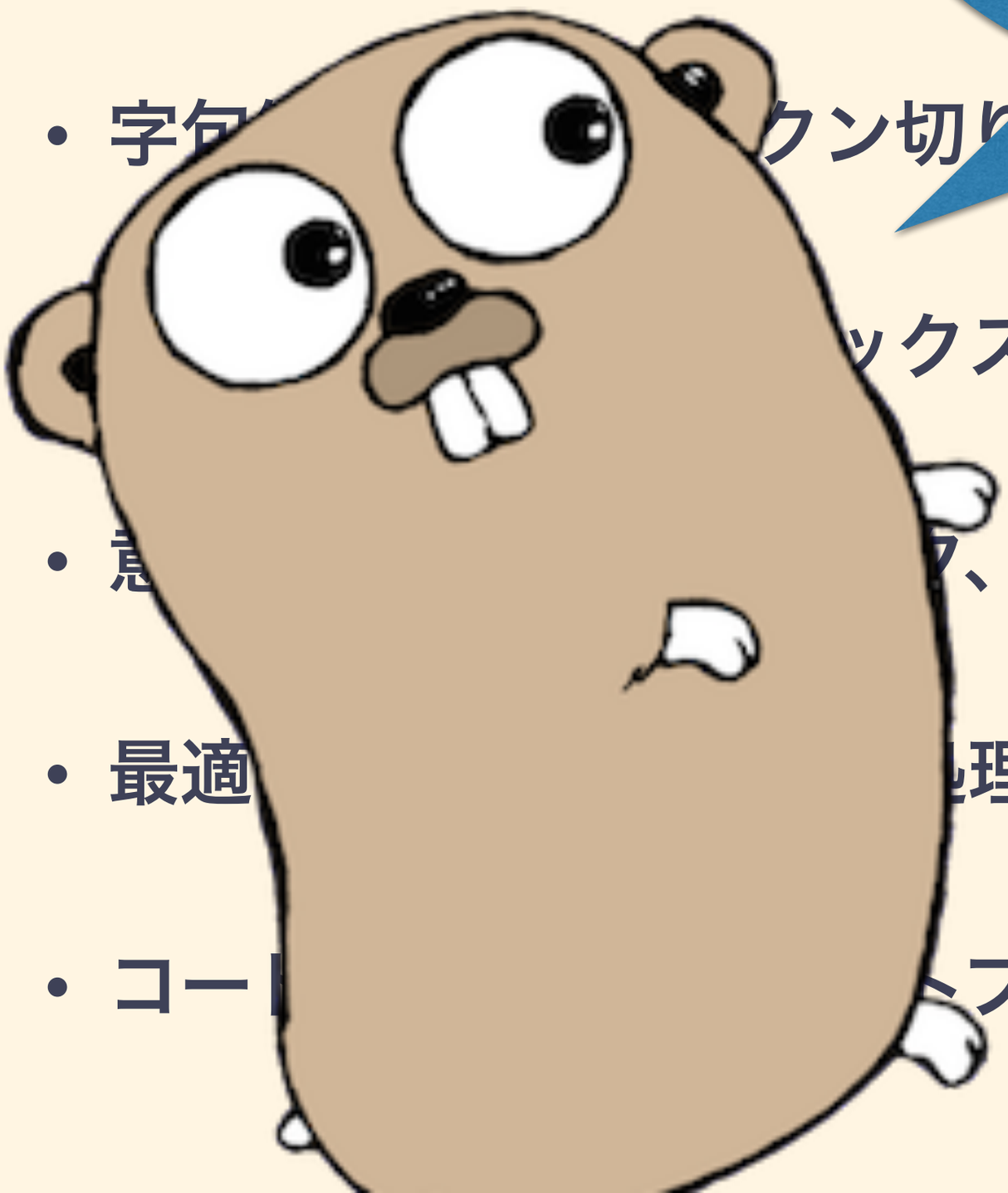
=> モテる！

コンパイラがしてること

コンパイラ

- 字句解析器 => トークン切り出し
- 構文解析器 => シンタックスチェック
- 意味解析 => 型のチェック、未定義変数、関数
- 最適化 => より効率的な処理、メモリ使用
- コード生成 => ターゲットプログラム生成

コンパイラ

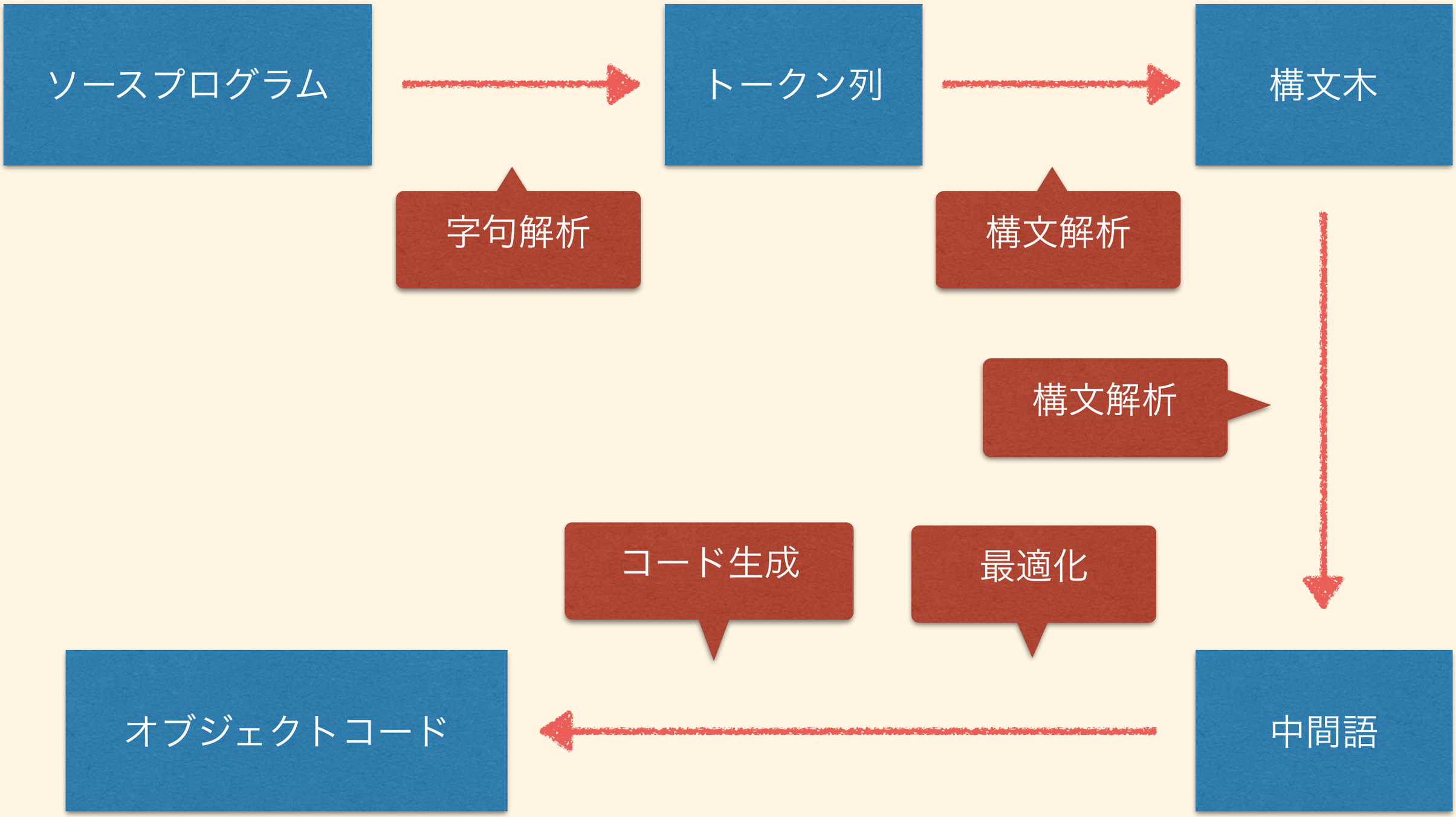


ググってね🙆♡

- 字句
- 意
- 最適
- コー

クン切り
ックスチェック
、未定義変数、関数
理、メモリ使用
トプログラム生成

コンパイルの流れ



コンパイルの流れ

ソースプログラム

ト

ググってね🙆♡

構文解析

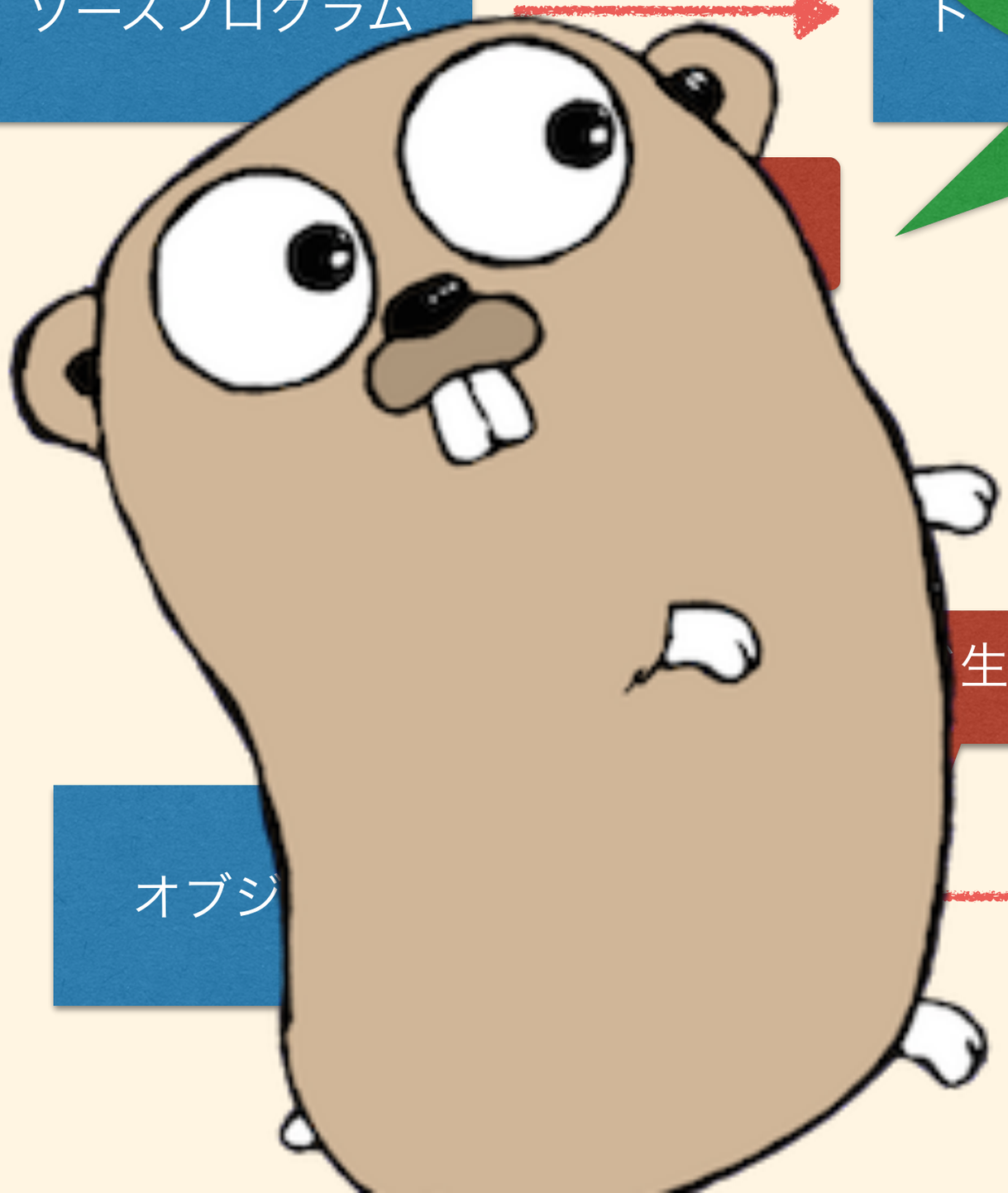
構文解析

生成

最適化

オブジ

中間語



実装

コンパイラを作る手順

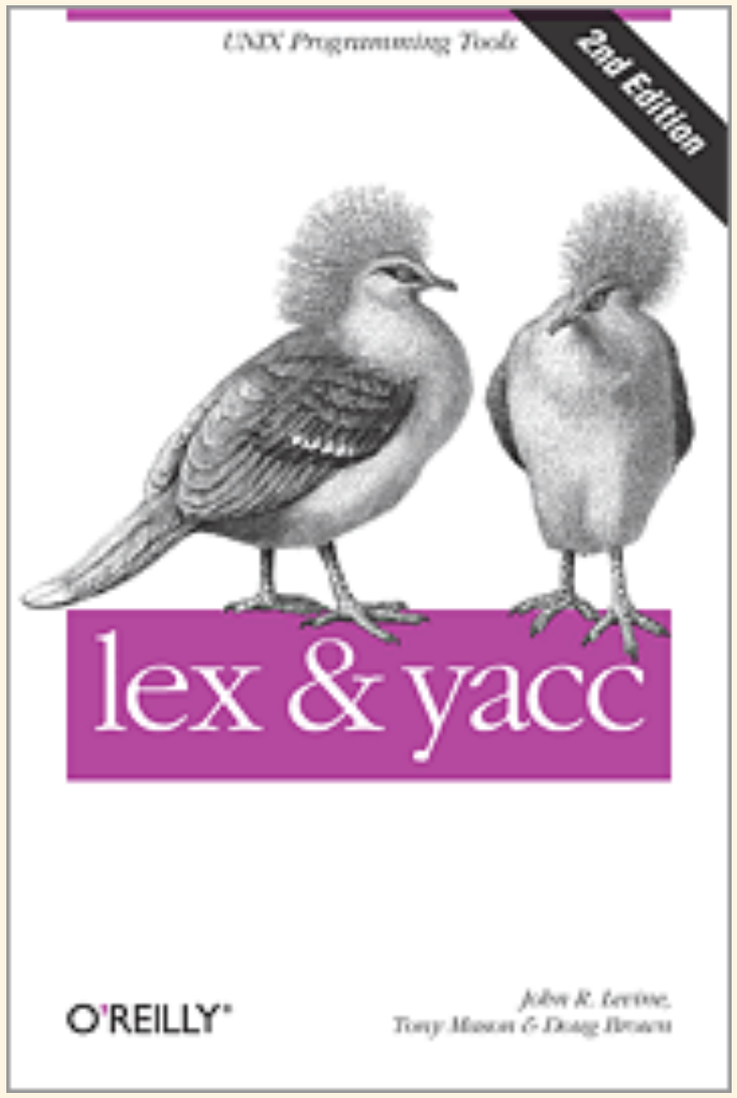
1. （言語を定義する）
2. 字句解析器をつくる
3. 構文解析器をつくる
    (1) 演算子順位構文解析
    (2) 再帰下降型構文解析
4. 最適化（つらい）
5. コード生成処理部分をつくる

便利なツール

字句解析器 & 構文解析器

自動生成ツール

lex & yacc


UNIX Programming Tools
2nd Edition
lex & yacc
O'REILLY
John R. Levine,
Tony Mason & Doug Brown

これを使えば

もっと簡単だ!!

やったぁぁぁぁ

…筋肉で作らないとモテない！

字句解析器をつくる

- 表駆動型オートマン => 終端記号、非終端記号、生成規則、出発記号で構成
- 構造体で状態遷移表を定義して状態を遷移する
- トークンのタイプを定義する
- 偉大な言語は大抵自作

構文解析器をつくる

- 演算子順位構文解析
  - スタックを使って逆ポーランド記法で出力
- 二分木を使って算術式だけの構文木を生成
  - ポインタを使ってノードを表現

構文解析器をつくる

- 再帰下降型構文解析
  - 文法規則ひとつにつき解析関数をひとつ用意
  - 後戻り: 冒頭部分が共通する文法があるときにどちらの生成規則に従えば良いか途中までわからない => 魔術: 文法を変更 or 筋肉でくくりだし
  - 左再帰 => 除去アルゴリズムに従って文法を変更して除去 (言語を書く上での変更なしに変更できる)

いろいろ大変なこともある

一度は筋肉で作ろう！

totzyuta / **espresso**

Watch 2 | Star 2 | Fork 0

[WIP] Tiny C Like Programming Language

102 commits | **3** branches | **1** release | **1** contributor

Branch: **master** ▾ **espresso** / +

To compile fact.esp

totzyuta authored 28 days ago — latest commit `0ff7d57048`

doc	Move generation code to gen.c	28 days ago
objects	Add object codes and result by maps	a month ago
sample-programs	Fix some process	a month ago
.gitignore	merge with gitbucket/master	a month ago
Makefile	Writing a report	a month ago
README.md	Add README	a month ago
define.h	Fix conflicts	a month ago
espresso-bnf.md	Update syntax of espresso	3 months ago
gen.c	To compile fact.esp	28 days ago
lex.c	Writing a report	a month ago
main.c	Writing a report	a month ago
minimum.esp	Developing code generator	a month ago
oparser.c	merge with gitbucket/master	a month ago
parser.c	Move generation code to gen.c	28 days ago
questions.md	Developing code generator	a month ago

Code

Issues 2

Pull requests 0

Pulse

Graphs

HTTPS clone URL

https://github.com

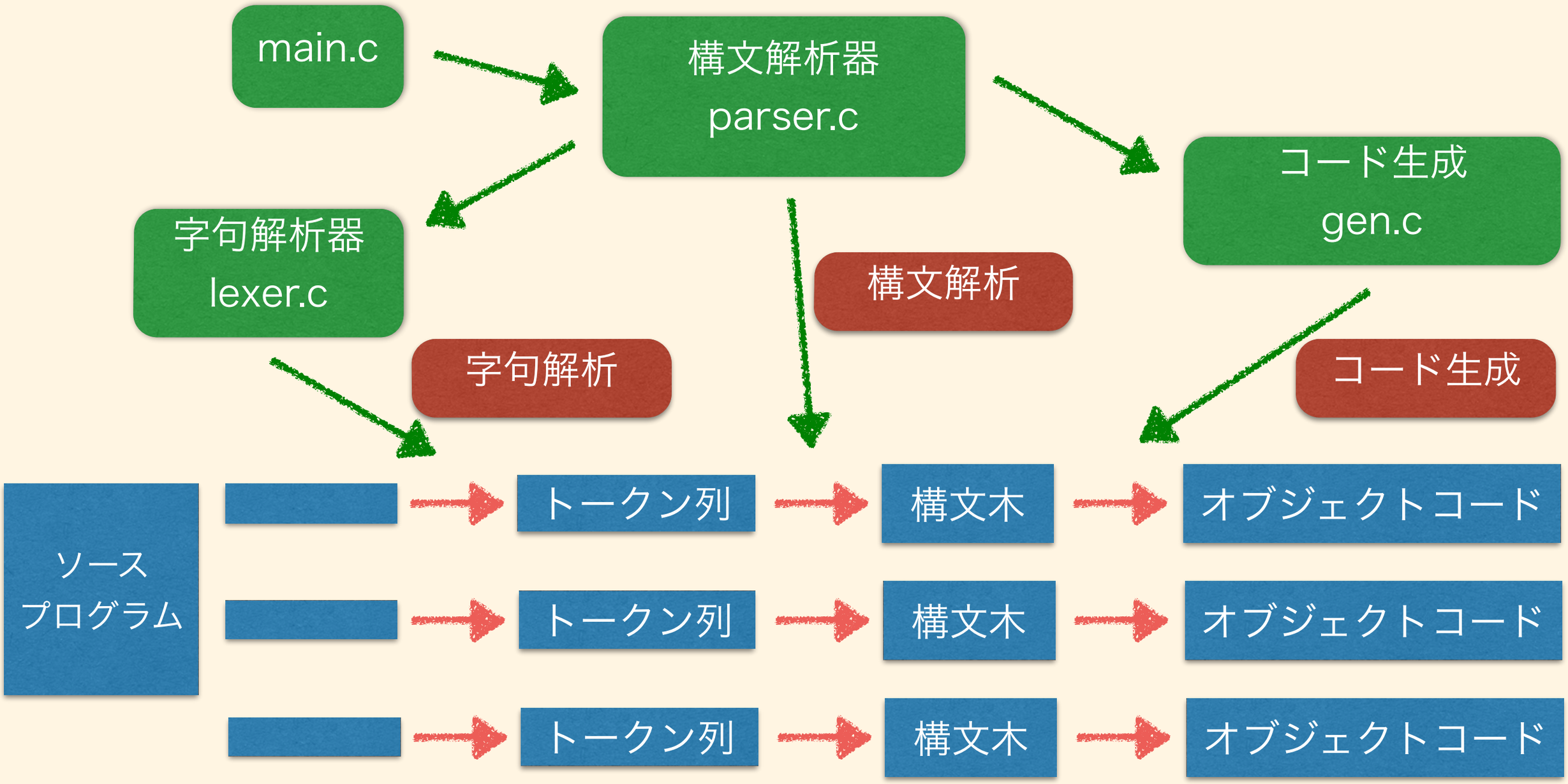
You can clone with HTTPS or Subversion.

Clone in Desktop

Download ZIP

https://github.com/totzyuta/espresso

コンパイルの流れ（espresso）



考えたこと

- 全ての解析器が明確な役割を持つべき
- 細分化された関数
- ファイル分割は大きなフォーカスのモジュール分割
- 雨降って地固まる => 最初は言語仕様固めすぎない方が後で死なない
- スコープの小さいエラー出力 (特に構文解析器)

コンパイラを作ってみて...

- 計算機が高級言語をどのように解釈していくのかちょっと理解することができた
- モテる
- 計算機自体やOSの仕組みについても考えた
- コンピュータサイエンスってこんなん...てなるがその反動でなんかもっと好きになる
- モテる

書籍

- コンパイラー原理・技法・ツール（A.V.エイホ 他）サイエンス社
- 2週間でできる！スクリプト言語の作り方（千葉 滋）
- ふつうのコンパイラをつくろう（青木 峰郎）ソフトバンククリエイティブ
- ７つの言語 ７つの世界（Bruce A. Tate）

OSSなコンパイラ

- Ruby
  - github.com/ruby/ruby
- Golang
  - https://github.com/golang/go
- 8cc
  - github.com/rui314/8cc

楽しいモテライフを！

fine...