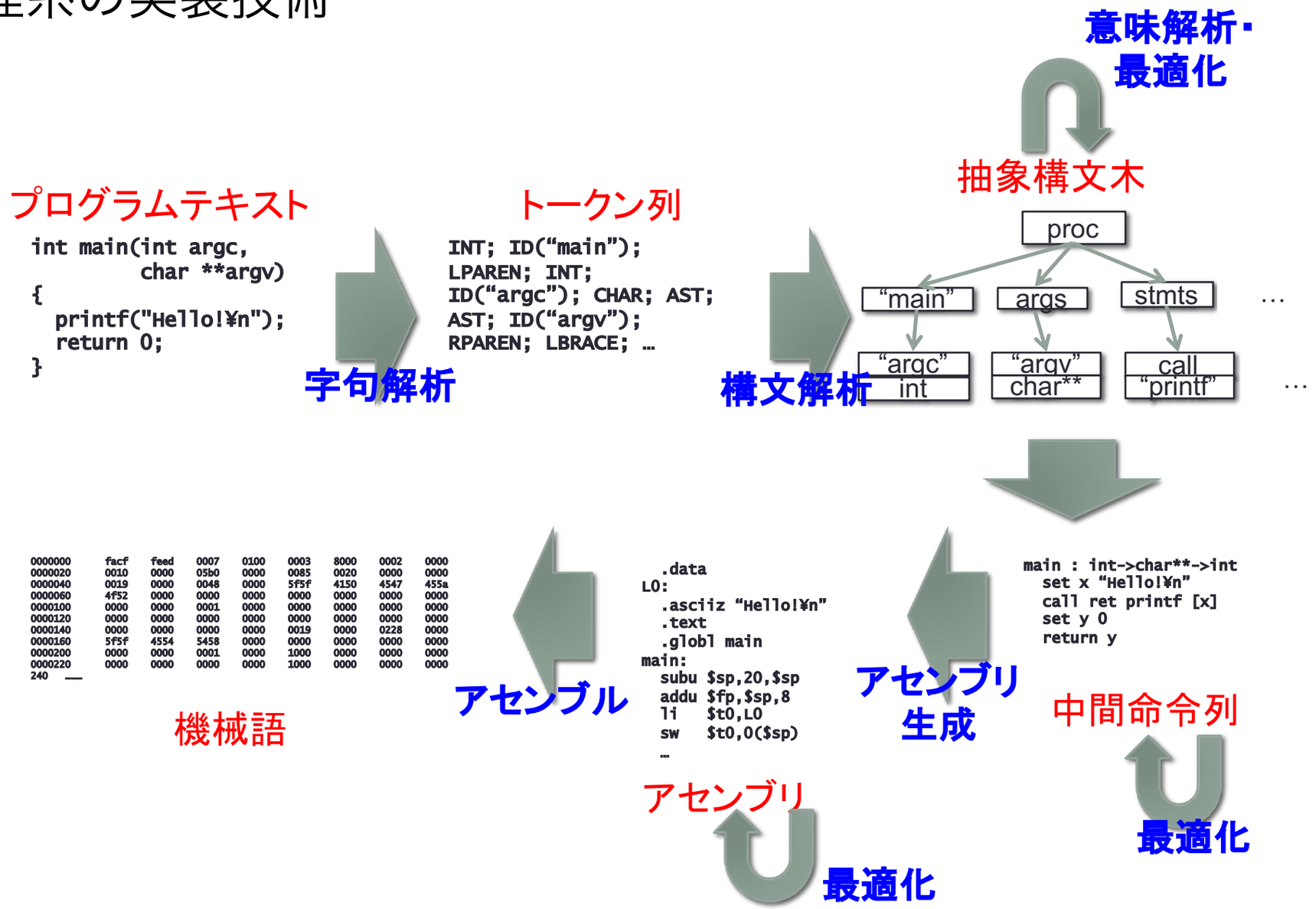
情報ゼミナール 研究室紹介

2022/06/22

馬谷 誠二

研究テーマ

- プログラミング言語に関すること全般
  - 領域特化言語(Domain-Specific Language, DSL)の設計・実装
  - 言語処理系の実装技術



2021年度卒研テーマ

1. ブロックを記述可能なラムダ式のCPython上の実装
2. Python用マクロフレームワークの実現に向けて
3. Racketの構文解析機能syntax-parseの効率化
4. Rossetの生成文法への繰り返し構文の追加の検討
5. Schemeサブセットの操作的意味論のRedexによる実装
6. Redex上のSchemeインタプリタにおける抽象解釈のための継続のヒープ化
7. Redex上のSchemeインタプリタにおけるヒープの有限化による抽象解釈の実現
8. Schemeの為の抽象解釈による制御フロー解析の調査報告
9. RPythonを用いたSchemeサブセット処理系の構築

2021年度卒研テーマ

CPython班

1. ブロックを記述可能なラムダ式のCPython上の実装
2. Python用マクロフレームワークの実現に向けて
3. Racketの構文解析機能syntax-parseの効率化
4. Rosseteの生成文法への繰り返し構文の追加の検討

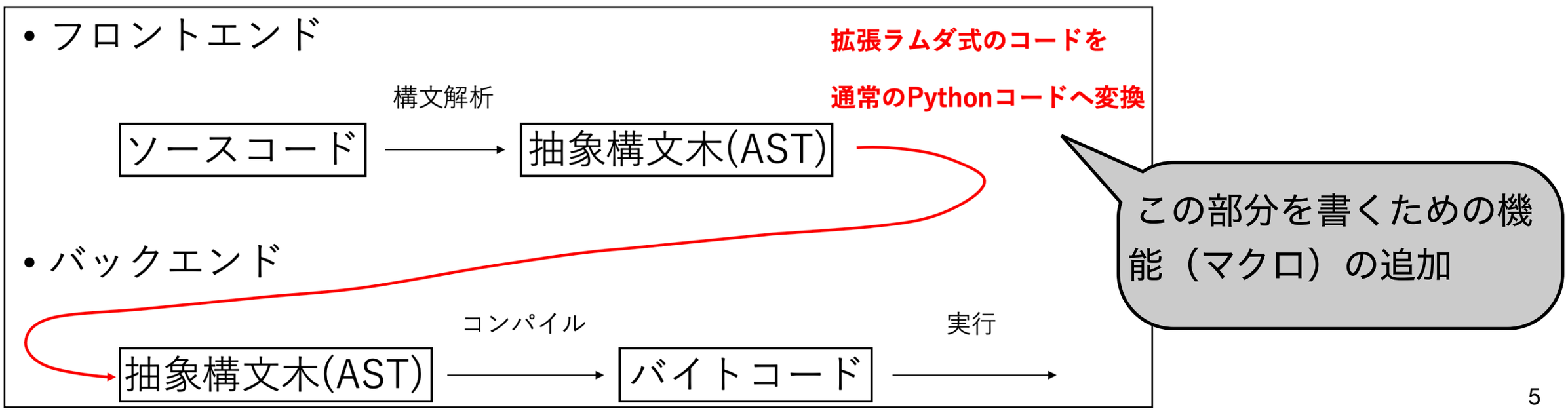
Scheme解析班

5. Schemeサブセットの操作的意味論のRedexによる実装
6. Redex上のSchemeインタプリタにおける抽象解釈のための継続のヒープ化
7. Redex上のSchemeインタプリタにおけるヒープの有限化による抽象解釈の実現
8. Schemeの為の抽象解釈による制御フロー解析の調査報告
9. RPythonを用いたSchemeサブセット処理系の構築

CPython班

- Python言語では**ラムダ式の本体にブロックを書けない**
  - （例） `f = lambda x, y : x + y`
- 条件分岐(if文)や繰り返し処理(for文)のような，より複雑な処理を書くことが可能な**拡張ラムダ式を実装**
  - （例） `f = lambda x, y :`

```
if (x > y):
    x = x + y
```



Scheme解析班

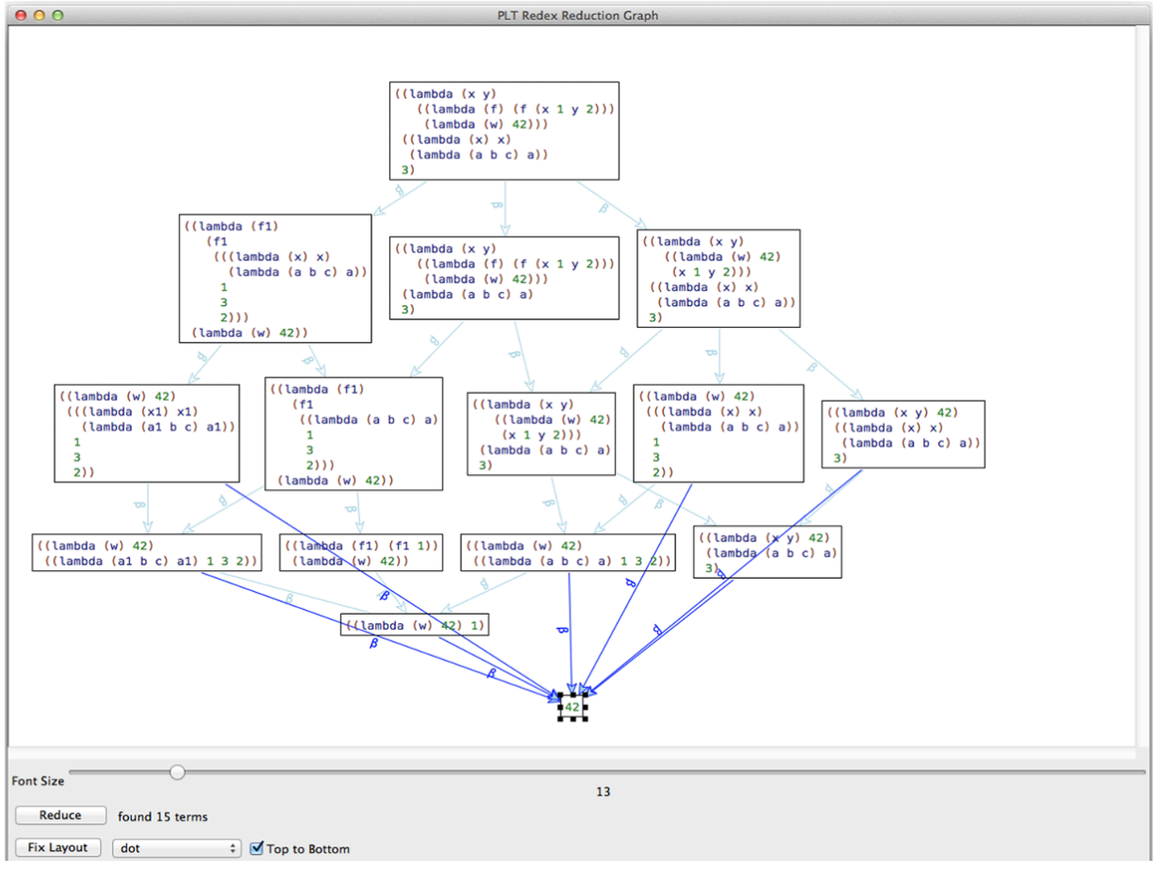
抽象解釈とは

　プログラムを抽象化によって単純化し近似的に実行して解析

抽象解釈の利点

　プログラムを近似的に実行することで不必要な部分を無視し，求めたい性質だけを部分的に求めることができる.

```
(define-language SCM
 (p (store (sf ...) e))
 (sf (x v))
 (e (e e ...) (set! x e) (begin e e ...) (if e e e) x v)
 (v (lambda (x ...) e) n #t #f -unspecified)
 (P (store (sf ...) E))
 (E (v ... E e ...) (set! x E) (begin E e e ...) (if E e e)
    hole)
 (x variable-not-otherwise-mentioned )
 (n number) )
```



勉強方法

- 輪講ゼミ（毎週）
  - 言語処理系の実装技術に関するテキスト
  - もしくは（特に関数型の）プログラミング言語の理解
  - 4年生になってより専門的な論文を読むための準備期間
- 4年生の卒研ゼミへのオブザーバ参加（毎週）
- その他
  - （半年後）横浜キャンパスの研究室立ち上げ
    - Mac 10数台によるネットワーク構築
    - etc.