

# プログラミング言語処理系論 (11)

---

佐藤周行

(情報基盤センター/電気系専攻融合情報学  
コース)

# Effect

---

- プログラムの実行にともない、どのような効果が現れてくるのかを解析したい
    - Memoryのread/write/alloc
    - Binding time
    - Function call
    - Try/catch
    - ...
-

# 実際の例

---

□ 型システムに、注釈をつける形で表現される

□ Def. Language

$$e ::= c \mid x \mid \text{fn}_{\pi} x => e \mid e \ e \mid \dots$$
$$\tau ::= \text{int} \mid \text{bool} \mid \dots \mid \tau \rightarrow \tau$$

---

---

□ 型に、以下のように注釈を加える

□  $\psi ::= \{ \pi \} \mid \psi \cup \psi \mid \phi$

$\underline{\tau} ::= \text{int} \mid \text{bool} \mid \dots \underline{\tau} \rightarrow_{\psi} \underline{\tau}$

□ 注釈つき型を構成するときにどの関数が使われたかを憶えておく

---

---

□ メモリのread/write/allocの解析

□  $\psi$  : effect

□  $\rho$  : region

□  $\tau$  : annotated types

$\psi ::= \{! \pi\} \mid \{\pi :=\} \mid \{\text{new } \pi\} \mid \psi \cup \psi \mid \phi$

$\rho ::= \{\pi\} \mid \rho \cup \rho \mid \phi$

$\underline{\tau} ::= \text{int} \mid \text{bool} \mid \dots \mid \underline{\tau} \rightarrow_{\psi} \underline{\tau} \mid \underline{\tau} \text{ ref } \rho$

---

---

□ read/write/allocを次のように表現する

□  $!x: \tau$  where type of  $x$  is  $\tau$  ref

□  $x := e: \tau$  where type of  $x$  is  $\tau$  ref

□  $\text{new}_{\pi} x := e1$  in  $e2$  where ...

---

□ 以下のルールで  $e: \tau \ \& \ \psi$  を導出していく

$$\hat{\Gamma} \vdash_{\text{SE}} e : \hat{\tau}_c \ \& \ \emptyset \qquad \hat{\Gamma} \vdash_{\text{SE}} x : \hat{\Gamma}(x) \ \& \ \emptyset$$

$$\frac{\hat{\Gamma}[x \mapsto \hat{\tau}_x] \vdash_{\text{SE}} e_0 : \hat{\tau}_0 \ \& \ \varphi_0}{\hat{\Gamma} \vdash_{\text{SE}} \text{fn}_{\pi} x \Rightarrow e_0 : \hat{\tau}_x \xrightarrow{\varphi_0} \hat{\tau}_0 \ \& \ \emptyset}$$

$$\frac{\hat{\Gamma} \vdash_{\text{SE}} e_1 : \hat{\tau}_2 \xrightarrow{\varphi_0} \hat{\tau}_0 \ \& \ \varphi_1 \quad \hat{\Gamma} \vdash_{\text{SE}} e_2 : \hat{\tau}_2 \ \& \ \varphi_2}{\hat{\Gamma} \vdash_{\text{SE}} e_1 e_2 : \hat{\tau}_0 \ \& \ \varphi_1 \cup \varphi_2 \cup \varphi_0}$$

$$\hat{\Gamma} \vdash_{\text{SE}} !x : \hat{\tau} \ \& \ \{!\pi_1, \dots, !\pi_n\} \text{ if } \hat{\Gamma}(x) = \hat{\tau} \ \text{ref} \ \{\pi_1, \dots, \pi_n\}$$

$$\frac{\hat{\Gamma} \vdash_{\text{SE}} e : \hat{\tau} \ \& \ \varphi}{\hat{\Gamma} \vdash_{\text{SE}} x := e : \hat{\tau} \ \& \ \varphi \cup \{\pi_1 :=, \dots, \pi_n :=\}} \text{ if } \hat{\Gamma}(x) = \hat{\tau} \ \text{ref} \ \{\pi_1, \dots, \pi_n\}$$

$$\frac{\hat{\Gamma} \vdash_{\text{SE}} e_1 : \hat{\tau}_1 \ \& \ \varphi_1 \quad \hat{\Gamma}[x \mapsto \hat{\tau}_1 \ \text{ref}(\varrho \cup \{\pi\})] \vdash_{\text{SE}} e_2 : \hat{\tau}_2 \ \& \ \varphi_2}{\hat{\Gamma} \vdash_{\text{SE}} \text{new}_{\pi} x := e_1 \ \text{in} \ e_2 : \hat{\tau}_2 \ \& \ (\varphi_1 \cup \varphi_2 \cup \{\text{new } \pi\})}$$

$$\frac{\hat{\Gamma} \vdash_{\text{SE}} e : \hat{\tau} \ \& \ \varphi}{\hat{\Gamma} \vdash_{\text{SE}} e : \hat{\tau} \ \& \ \varphi'} \text{ if } \varphi \subseteq \varphi'$$

# Effectのご利益

---

- Typeに対する注釈の形で、計算のいろいろな効果を表現
  - Effectに対する導出ができる→Effectを持つ
-



# slicing

---

- Def Program Slice  
a set of programs that may affect the values of some interest (slice criteria)
  - Def Program Slicing  
Making a program slice
-

# Example

---

```
□ int i;  
  int sum = 0;  
  int product = 1;  
  
  for(i = 0; i < N; ++i) {  
    sum = sum + i;  
    product = product * i; }  
  write(sum) write(product);
```

---

Slice criteria

---

```
□ int i;  
  int sum = 0;  
  for(i = 0; i < N; ++i) {  
    sum = sum + i;  
  }  
  write(sum);
```

---

# Slicing

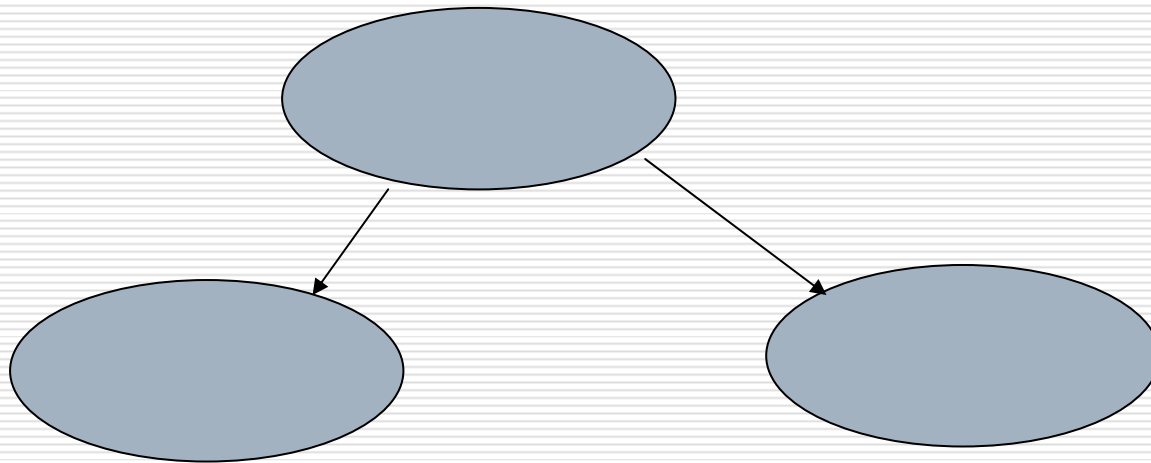
---

- Slice criteriaが与えられたら、関係する変数のDef、Defするとき使用する変数(条件式を含む)...の並びを求める
  - プログラムによっては、スライスを取ることに  
よって、解析対象が非常に小さくなる場合がある  
(ほとんど変わらない場合も多い)
-

# 変数のUse/Def関係によるクラスタリング

---

- $V1 \rightarrow V2$  で、 $V1$ が、 $V2$ のDefの際に使われることを表すと、これらの関係が作るグラフを考えることができる(SSA)



- 
- グラフの連結成分それぞれが塊となる。
  - 連結成分の上位のほうは、たいていのスライスに含まれる可能性が高い
  - 上位には、制御構造に関係する変数が集まっている可能性が高い
  - ...
-