Kontrollflussanalyse 2

0-CFA Analyse

Bernhard Hering

24.Juni.2009

Ludwig Maximilians Universität München

Inhaltsverzeichnis

Kontrollflussanalyse Einführung

- statische Analyse
- vor dem Ausführen eines Programms durchgeführt
- von welchen Block im Programm die Kontrolle an welchen Block übergeben wird
- Graph der Kontrollfluss modelliert
- Ziel Modellierung des Kontrollflusses

Das 0-CFA Verfahren

1. abstrakte Spezifikation, strukturelle Operationalisierung und Korrektheit

---

2. Syntax orientierte Analyse
3. Algorithmus zum Sammeln und Lösen der Bedingungen (Constraints)

Beispiel

$((fn\ x \Rightarrow x)(fn\ y \Rightarrow y))$

- Programmiersprache FUN ähnlich SML
- id_y : Funktion in die id_x einsetzen

Beispiel mit Labels

Labels setzen:
$((fn\ x \Rightarrow x^1)^2(fn\ y \Rightarrow y^3)^4)^5$

Denkbare Lösung für die Menge $(\hat{C}, \hat{\rho})$:

Beispiel mit Labels

Labels setzen:
$((fn\ x \Rightarrow x^1)^2(fn\ y \Rightarrow y^3)^4)^5$

Denkbare Lösung für die Menge $(\hat{C}, \hat{\rho})$:

	$(\hat{C}, \hat{\rho})$
C(1)	$fn\ y \Rightarrow y^3$
C(2)	$fn\ x \Rightarrow x^1$
C(3)	$\emptyset$
C(4)	$fn\ y \Rightarrow y^3$
C(5)	$fn\ y \Rightarrow y^3$
r(x)	$fn\ y \Rightarrow y^3$
r(y)	$\emptyset$

Syntax orientierte Analyse

Spezifikation

$$
\begin{array}{ll}
[con] & (\hat{C},\hat{\rho}) \models_s c^l \ \ always \\
[var] & (\hat{C},\hat{\rho}) \models_s x^l \\
 & \text{iff} \quad \hat{\rho}(x) \subseteq \hat{C}(l) \\
[fn] & (\hat{C},\hat{\rho}) \models_s (fn\ x \Rightarrow e_0)^l \\
 & \text{iff } \{fn\ x \Rightarrow e_0\} \subseteq \hat{C}(l) \quad \wedge \quad (\hat{C},\hat{\rho}) \models_s e_0 \\
[fun] & (\hat{C},\hat{\rho}) \models_s (fun\ f\ x \Rightarrow e_0)^l \\
 & \text{iff } \{fun\ f\ x \Rightarrow e_0\} \subseteq \hat{C}(l) \quad \wedge \quad (\hat{C},\hat{\rho}) \models_s e_0 \\
 & \quad \wedge \quad \{fun\ f\ x \Rightarrow e_0\} \subseteq \hat{\rho}(f) \\
[app] & (\hat{C},\hat{\rho}) \models_s (t_1^{l_1}\ t_2^{l_2})^l \\
 & \text{iff } (\hat{C},\hat{\rho}) \models_s t_1^{l_1} \wedge (\hat{C},\hat{\rho}) \models_s t_2^{l_2} \quad \wedge \\
 & \quad (\forall (fn\ x \Rightarrow t_0^{l_0}) \in \hat{C}(l_1) : \\
 & \quad \hat{C}(l_2) \subseteq \hat{\rho}(x) \wedge \hat{C}(l_0) \subseteq \hat{C}(l)) \\
 & \quad (\forall (fun\ f\ x \Rightarrow t_0^{l_0}) \in \hat{C}(l_1) : \\
 & \quad \hat{C}(l_2) \subseteq \hat{\rho}(x) \wedge \hat{C}(l_0) \subseteq \hat{C}(l))
\end{array}
$$

Spezifikation

$$
\begin{array}{ll}
[if] & (\hat{C},\hat{\rho}) \models_s (\mathit{if}\ t_0^{l_0}\ \mathit{then}\ t_1^{l_1}\ \mathit{else}\ t_2^{l_2})^l \\
& \text{iff } (\hat{C},\hat{\rho}) \models_s t_0^{l_0} \quad \wedge \quad (\hat{C},\hat{\rho}) \models_s t_1^{l_1} \\
& \wedge \quad (\hat{C},\hat{\rho}) \models_s t_2^{l_2} \quad \wedge \\
& \hat{C}(l_1) \subseteq \hat{C}(l) \quad \wedge \quad \hat{C}(l_2) \subseteq \hat{C}(l)) \\
[let] & (\hat{C},\hat{\rho}) \models_s (\mathit{let}\ t_1^{l_1}\ \mathit{in}\ t_2^{l_2})^l \\
& \text{iff} \quad (\hat{C},\hat{\rho}) \models_s t_1^{l_1} \quad \wedge \quad (\hat{C},\hat{\rho}) \models_s t_2^{l_2} \quad \wedge \\
& \hat{C}(l_1) \subseteq \hat{\rho}(x) \quad \wedge \quad \hat{C}(l_2) \subseteq \hat{C}(l)) \\
[op] & (\hat{C},\hat{\rho}) \models_s (t_1^{l_1}\ \mathit{op}\ t_2^{l_2})^l \\
& \text{iff} \quad (\hat{C},\hat{\rho}) \models_s t_1^{l_1} \quad \wedge \quad (\hat{C},\hat{\rho}) \models_s t_2^{l_2}
\end{array}
$$

Anwendung auf Beispiel

- von aussen nach innen
- zuerst anwenden des Konstrukts $[app]$

$$
\begin{aligned}
[app] \quad & (\hat{C}, \hat{\rho}) \models_s (t_1^{l_1}\ t_2^{l_2})^l \\
& \text{iff } (\hat{C}, \hat{\rho}) \models_s t_1^{l_1} \wedge (\hat{C}, \hat{\rho}) \models_s t_2^{l_2} \quad \wedge \\
& \quad (\forall (fn\ x \Rightarrow t_0^{l_0}) \in \hat{C}(l_1) : \\
& \hat{C}(l_2) \subseteq \hat{\rho}(x) \wedge \hat{C}(l_0) \subseteq \hat{C}(l)) \\
& \quad (\forall (fun\ f\ x \Rightarrow t_0^{l_0}) \in \hat{C}(l_1) : \\
& \hat{C}(l_2) \subseteq \hat{\rho}(x) \wedge \hat{C}(l_0) \subseteq \hat{C}(l))
\end{aligned}
$$

- $\hat{C}(l_2) \subseteq \hat{\rho}(x)$: $\hat{C}(l_2)$ wird an x gebunden
- $\hat{C}(l_0) \subseteq \hat{C}(l)$): Modelliert die Parameterübergabe

Anwendung auf Beispiel

Anwendung:

$$[app] \quad (\hat{C}, \hat{\rho}) \models_s ((fn\ x \Rightarrow x^1)^2 (fn\ y \Rightarrow y^3)^4)^5$$
$$\text{iff } (\hat{C}, \hat{\rho}) \models_s (fn\ x \Rightarrow x^1)^2 \wedge (\hat{C}, \hat{\rho}) \models_s (fn\ y \Rightarrow y^3)^4 \quad \wedge$$
$$\hat{C}(4) \subseteq \hat{\rho}(x) \quad \wedge \quad \hat{C}(1) \subseteq \hat{C}(5)$$

Anwendung auf Beispiel 2

- Konstrukt $[fn]$ für Label 2, 4

$$[fn] \quad (\hat{C}, \hat{\rho}) \models_s (fn\ x \Rightarrow e_0)^l$$
$$\text{iff } \{fn\ x \Rightarrow e_0\} \subseteq \hat{C}(l) \quad \wedge \quad (\hat{C}, \hat{\rho}) \models_s e_0$$

Anwendung auf Beispiel 2

- Konstrukt $[fn]$ für Label 2, 4

$$[fn] \quad (\hat{C}, \hat{\rho}) \models_s ((fn\ x \Rightarrow x^1)^2$$
$$\text{iff } fn\ x \Rightarrow x^1 \subseteq \hat{C}(2) \quad \wedge \quad (\hat{C}, \hat{\rho}) \models_s x^1$$

$$[fn] \quad (\hat{C}, \hat{\rho}) \models_s ((fn\ y \Rightarrow y^3)^4$$
$$\text{iff } fn\ y \Rightarrow y^3 \subseteq \hat{C}(4) \quad \wedge \quad (\hat{C}, \hat{\rho}) \models_s y^3$$

Anwendung auf Beispiel 3

- Konstrukt $[var]$ für Label 1, 3

$$[var] \quad (\hat{C}, \hat{\rho}) \models_s x^l$$
$$\text{iff} \quad \hat{\rho}(x) \subseteq \hat{C}(l)$$

Anwendung auf Beispiel 3

- Konstrukt $[var]$ für Label 1, 3

$$[var] \quad (\hat{C}, \hat{\rho}) \models_s x^1 \quad \text{iff} \quad \hat{\rho}(x) \subseteq \hat{C}(1)$$

$$[var] \quad (\hat{C}, \hat{\rho}) \models_s x^3 \quad \text{iff} \quad \hat{\rho}(y) \subseteq \hat{C}(3)$$

Anwendung auf Beispiel 4

Menge der Bedingungen.

$$
\begin{aligned}
&(\hat{C}, \hat{\rho}) = \\
&\{\ (fn\ x \Rightarrow x^1) \subseteq \hat{C}(2) \Rightarrow \hat{C}(4) \subseteq \hat{\rho}(x), \\
&(fn\ x \Rightarrow x^1) \subseteq \hat{C}(2) \Rightarrow \hat{C}(1) \subseteq \hat{C}(5), \\
&fn\ x \Rightarrow x^1 \subseteq \hat{C}(2), \\
&fn\ y \Rightarrow y^3 \subseteq \hat{C}(4), \\
&\hat{\rho}(x) \subseteq \hat{C}(1), \\
&id_y \subseteq \hat{C}(3)\}
\end{aligned}
$$

Korrektheit der Analyse

- $(\hat{C}, \hat{\rho})$ kann unendlich groß werden
- $(\hat{C}, \hat{\rho})$ einschränken auf alle Terme die im Programm konkret vorkommen
- $(\hat{C}, \hat{\rho}) \sqsubseteq (\hat{C_*^T}, \hat{\rho_*^T})$
- wobei $(\hat{C_*^T}, \hat{\rho_*^T})$

$$\textit{wenn} \quad (\hat{C}, \hat{\rho}) \models_s e_* \quad \textit{und} \quad (\hat{C}, \hat{\rho}) \sqsubseteq (\hat{C_*^T}, \hat{\rho_*^T})$$

$$\textit{dann} \quad (\hat{C}, \hat{\rho}) \models e_*$$

$\Rightarrow (\hat{C}, \hat{\rho}) \models_s e_*$ ist auch eine Lösung

Korrektheit der Spezifikation - Beweis

- *wenn* $(\hat{C}, \hat{\rho}) \models_s e_*$ *und* $(\hat{C}, \hat{\rho}) \sqsubseteq (\hat{C_*^T}, \hat{\rho_*^T})$ wird angenommen.
- Coindunktion über $(\hat{C}, \hat{\rho}) \models e_*$
- alle Konstrukte gleich bis auf [app]
- Gleichheit durch Einschränkung gegeben

Algorithmus zum Sammeln der Bedingungen

- Algorithmus $C_*[[e_*]]$
- Eingabe e_*
- Ausgabe der Bedingungen
- Form der Bedingungen:
  $lhs \subseteq rhs$ oder $(\{t\} \subseteq rhs' \Rightarrow lhs) \subseteq rhs$.

Algorithmus zum Sammeln der Bedingungen

Spezifikation

$$
\begin{array}{ll}
[con] & C_*[[c^l]] = \emptyset \\
[var] & C_*[[c^l]] = \{r(x) \subseteq C(l)\} \\
[fn] & C_*[[(fn\ x \Rightarrow e_0)^l]] = \\
 & \{\{fn\ x \Rightarrow e_0\} \subseteq C(l)\} \quad \cup \quad C_*[[e_0]] \\
[fun] & (C_*[[(fun\ x \Rightarrow e_0)^l]] = \{\{fun\ f\ x \Rightarrow e_0\} \subseteq C(l)\} \\
 & \cup \quad C_*[[e_0]] \quad \cup \quad \{\{fun\ f\ x \Rightarrow e_0\} \subseteq r(f)\} \\
[app] & C_*[[(t_2^{l_2}\ t_2^{l_2})^l]] = C_*[[t_1^{l_1}]] \quad \cup \quad C_*[[t_2^{l_2}]] \\
 & \cup \quad \{t\} \subseteq C(l_1) \Rightarrow C(l_2) \subseteq r(x) \\
 & \mid t = (fn\ x \Rightarrow t_0^{l_0}) \in Term_* \\
 & \cup \quad \{t\} \subseteq C(l_1) \Rightarrow C(l_0) \subseteq C(l) \\
 & \mid t = (fn\ x \Rightarrow t_0^{l_0}) \in Term_* \\
 & \cup \quad \{t\} \subseteq C(l_1) \Rightarrow C(l_2) \subseteq r(x) \\
 & \mid t = (fun\ x \Rightarrow t_0^{l_0}) \in Term_* \\
 & \cup \quad \{t\} \subseteq C(l_1) \Rightarrow C(l_0) \subseteq C(l) \\
 & \mid t = (fun\ x \Rightarrow t_0^{l_0}) \in Term_*
\end{array}
$$

Algorithmus zum Sammeln der Bedingungen

Spezifikation

$$
\begin{array}{ll}
[if] & C_*[[if\ t_0^{l_0}\ then\ t_1^{l_1}\ else\ t_2^{l_2})^l]] = \\
 & C_*[[t_0^{l_0}]] \quad \cup \quad C_*[[t_1^{l_1}]] \quad \cup \quad C_*[[t_2^{l_2}]] \quad \cup \\
 & \{C(l_1) \subseteq C(l)\} \quad \cup \quad \{C(l_2) \subseteq C(l)\} \\
[let] & C_*[[(let\ x = t_1^{l_1}\ in\ t_2^{l_2})^l]] = \\
 & C_*[[t_1^{l_1}]] \quad \cup \quad C_*[[t_2^{l_2}]] \quad \cup \\
 & \{\{C(l_1) \subseteq r(x)\} \quad \cup \quad \{C(l_2) \subseteq C(l)\}\} \\
[op] & C_*[[(t_1^{l_1}\ op\ t_2^{l_2})^l]] = C_*[[t_1^{l_1}]] \quad \cup \quad C_*[[t_2^{l_2}]]
\end{array}
$$

Anwendung auf Beispiel

Analog zur Syntax orientierten Spezifikation bis auf [app].
Alle t werden behandelt.
Damit ergibt sich für Menge der Bedingungen
$C_*[[((fn\ x \Rightarrow x^1)^2(fn\ y \Rightarrow y^3)^4)^5]] =$

$\{r(x) \subseteq C(1)$
$fn\ x \Rightarrow x^1 \subseteq C(2)$
$r(y) \subseteq C(3)$
$fn\ y \Rightarrow y^3 \subseteq C(4)$
$(fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(4) \subseteq r(x)$
$(fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(1) \subseteq C(5).$
$(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(4) \subseteq r(y)$
$(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(3) \subseteq C(5)\}$

Korrektheit des Algorithmus

- Berechnet der Algorithmus das selbe?

  wenn $(\hat{C}, \hat{\rho} \sqsubseteq (\hat{C}_*^T, \hat{\rho}_*^T)$ dann $\hat{C}, \hat{\rho} \models_s e_*$
  genau dann wenn $\hat{C}, \hat{\rho} \models_c C_*[[e_*]]$

- Beweis: strukturelle Induktion über e

Algorithmus zum Lösen der Bedingungen

- Eingabe die Menge der Bedingungen $C_*[[e_*]]$
- erstellt Graph
- $(\hat{C}, \hat{\rho})$ (Durch aufzeichnen des Graphs)
- drei Datensätze (Arbeitsliste W, Datenfeld D, Kantenmenge E)

Syntax orientierte Analyse

Spezifikation

Schritt 1: Inizialisieren

W := nil;
for q in Nodes do $D[q] := \emptyset$;
for q in Nodes do $E[q] := nil$;

Schritt 2: Erstellen des Graphs

for cc in $C_*[[e_*]]$ do
case cc of
$\{t\} \subseteq p$: add$(p, \{t\})$;
$p_1 \subseteq p_2$: $E[p_1] := cons(cc, E[p_1])$;
$\{t\} \subseteq p \Rightarrow p_1 \subseteq p_2$: $E[p_1] := cons(cc, E[p_1])$;
$E[p] := cons(cc, E[p])$;

Schritt 3: Iteration

while W $\neq$ nil do
$q := haed(W)$; $W := tail(W)$;
for cc in $E[q]$ do
case cc of
$p_1 \subseteq p_2$: add$(p_2, D[p_1])$;
$\{t\} \subseteq p \Rightarrow p_1 \subseteq p_2$: if $t \in D[p]$ then add $(p_2, D[p_1])$;

Schritt 4: Aufzeichnen der Lösungen

for l in Lab_* do $\hat{C}(l) := D[C(l)]$;
for x in Var_* do $\hat{\rho}(x) := D[r(x)]$;

Unterprogramme: procedure add(q, d) is
iff $\neg(d \subseteq D[q])$
then $D[q] := D[q] \cup d$; $W :=$cons(q, W);

Beispiel

- Eingabe:
  $C_*[[((fn\ x \Rightarrow x^1)^2(fn\ y \Rightarrow y^3)^4)^5]] =$

  $\{r(x) \subseteq C(1)$
  $fn\ x \Rightarrow x^1 \subseteq C(2)$
  $r(y) \subseteq C(3)$
  $fn\ y \Rightarrow y^3 \subseteq C(4)$
  $(fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(4) \subseteq r(x)$
  $(fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(1) \subseteq C(5).$
  $(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(4) \subseteq r(y)$
  $(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(3) \subseteq C(5)\}$

Beispiel - Schritt 1

- Schritt 1

  Initialisieren der Datenstrukturen:

p	$D[p]$	$E[p]$

$W =$

Beispiel - Schritt 2

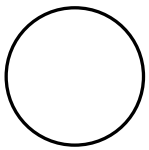
- Schritt 2

  Erstellen des Graphen:
  für jedes $C(l)$ und jedes $r(x)$ ein Konten:

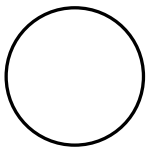
Beispiel - Schritt 2

- Schritt 2

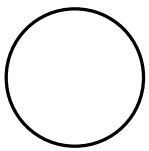
Erstellen des Graphen:
für jedes $C(l)$ und jedes $r(x)$ ein Konten:



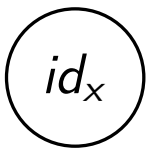
$r(x)$



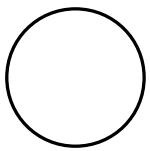
$r(y)$



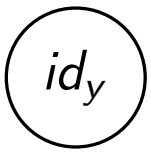
$C(1)$



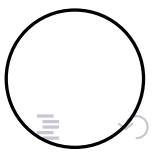
$C(2)$



$C(3)$

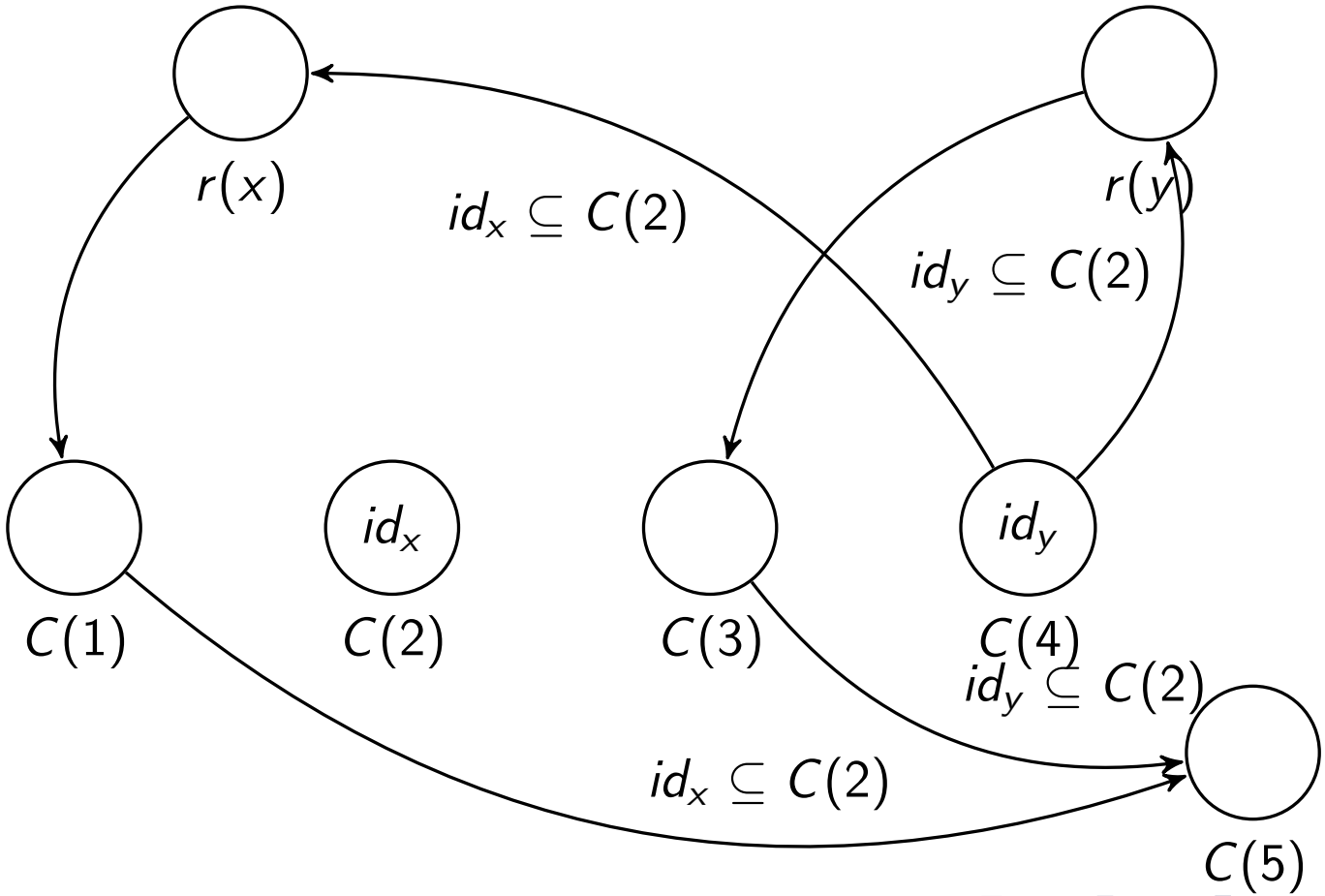


$C(4)$



Beispiel - Schritt 2 b

- Nach Schritt 2
- $W = [C(4), C(2)]$



Beispiel - Schritt 3

- Schritt 3
- Arbeitsliste abarbeiten
- das heißt wird betrachten C(4)

$$[\ (fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(4) \subseteq r(x),$$
$$(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(4) \subseteq r(y)]$$

Beispiel - Schritt 3

W	[C(4),C(2)]	[r(x),C(2)]	
p	$D[p]$	$D[p]$	
C(1)	$\emptyset$	$\emptyset$	
C(2)	id_x	id_x	
C(3)	$\emptyset$	$\emptyset$	
C(4)	id_y	id_y	
C(5)	$\emptyset$	$\emptyset$	
r(x)	$\emptyset$	id_y	
r(y)	$\emptyset$	$\emptyset$	

Beispiel - Schritt 3

- Schritt 3
- wir betrachten r(x)

$[r(x) \subseteq C(1)]$

Beispiel - Schritt 3

W	[C(4),C(2)]	[r(x),C(2)]	[C(1),C(2)]	
p	$D[p]$	$D[p]$	$D[p]$	
C(1)	$\emptyset$	$\emptyset$	id_y	
C(2)	id_x	id_x	id_x	
C(3)	$\emptyset$	$\emptyset$	$\emptyset$	
C(4)	id_y	id_y	id_y	
C(5)	$\emptyset$	$\emptyset$	$\emptyset$	
r(x)	$\emptyset$	id_y	id_y	
r(y)	$\emptyset$	$\emptyset$	$\emptyset$	

Beispiel - Schritt 3

- Schritt 3
- wir betrachten C(1)

$$[(fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(1) \subseteq C(5)]$$

Beispiel - Schritt 3

W	[C(4),C(2)]	[r(x),C(2)]	[C(1),C(2)]	[C(5),C(2)]		
p	$D[p]$	$D[p]$	$D[p]$	$D[p]$		
C(1)	$\emptyset$	$\emptyset$	id_y	id_y		
C(2)	id_x	id_x	id_x	id_x		
C(3)	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$		
C(4)	id_y	id_y	id_y	id_y		
C(5)	$\emptyset$	$\emptyset$	$\emptyset$	id_y		
r(x)	$\emptyset$	id_y	id_y	id_y		
r(y)	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$		

Beispiel - Schritt 3

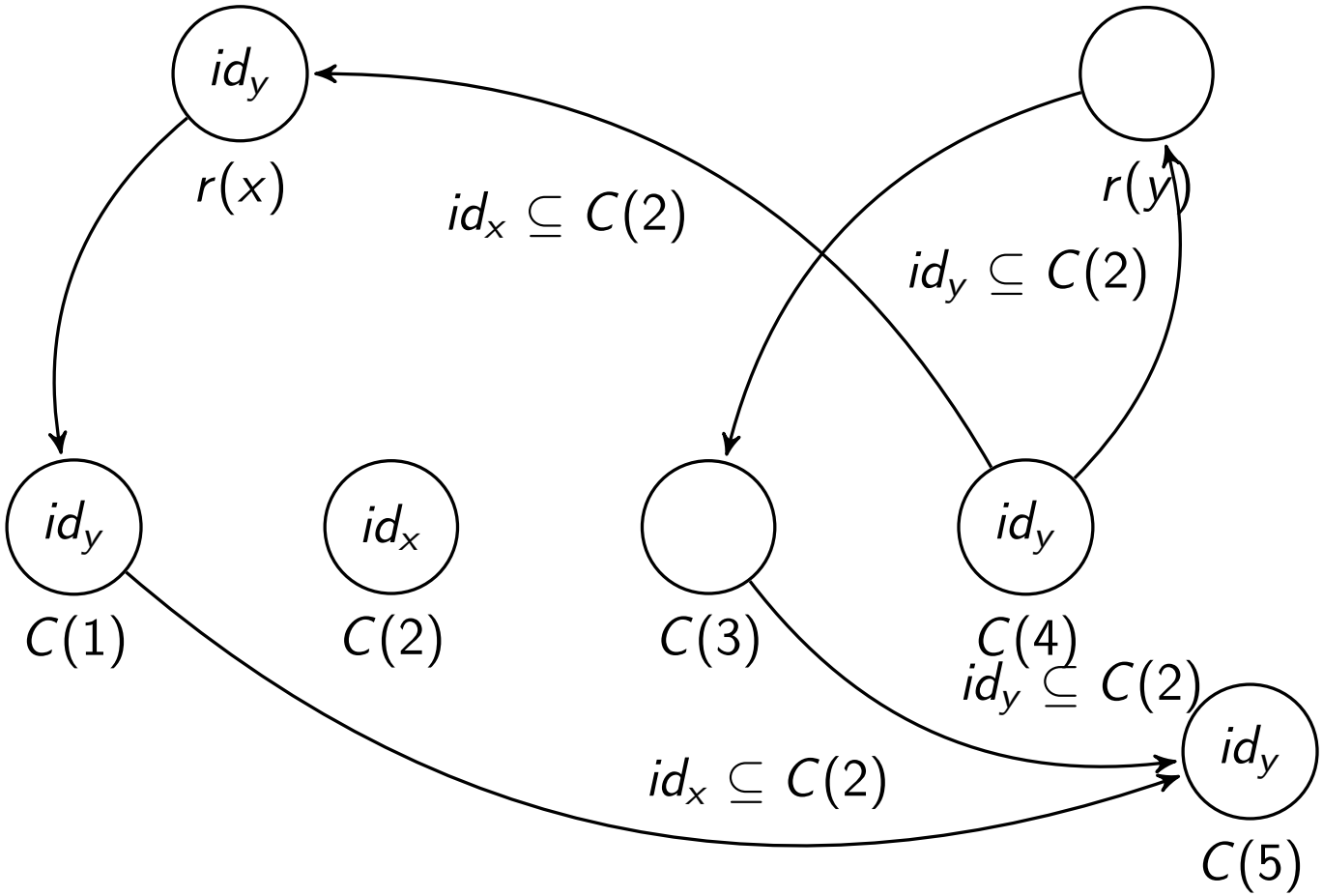
- wir betrachten C(5)
- E(5) ist leer
- wir betrachten C(2)

$$[\ (fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(4) \subseteq r(x),$$
$$(fn\ x \Rightarrow x^1) \subseteq C(2) \Rightarrow C(1) \subseteq C(5)$$
$$(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(4) \subseteq r(y),$$
$$(fn\ y \Rightarrow y^3) \subseteq C(2) \Rightarrow C(3) \subseteq C(5)]$$

Beispiel - Schritt 3

W	[C(4),C(2)]	[r(x),C(2)]	[C(1),C(2)]	[C(5),C(2)]	[C(2)]	[ ]
p	$D[p]$	$D[p]$	$D[p]$	$D[p]$	$D[p]$	$D[p]$
C(1)	$\emptyset$	$\emptyset$	id_y	id_y	id_y	id_y
C(2)	id_x	id_x	id_x	id_x	id_x	id_x
C(3)	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
C(4)	id_y	id_y	id_y	id_y	id_y	id_y
C(5)	$\emptyset$	$\emptyset$	$\emptyset$	id_y	id_y	id_y
r(x)	$\emptyset$	id_y	id_y	id_y	id_y	id_y
r(y)	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Beispiel - Ende



Korrektheit des Algorithmus

- terminiert der Algorithmus?
- berechnet er wirklich die kleinste Lösung?

$$(\hat{C}, \hat{\rho}) = \sqcap\{(\hat{C}', \hat{\rho}') \mid (\hat{C}', \hat{\rho}') \models_c C_*[[e_*]]\}$$

- kleinste Belegung wird erweitert $\Rightarrow$ kleinste Lösung

Zusammenfassung

- Wir haben gezeigt: Der Algorithmus berechnet die kleinste Lösung $(\hat{C}, \hat{\rho})$ aus $C[[e_0]]$
  $(\hat{C}, \hat{\rho}) \models_c C[[e_0]]$
- Die Bedingung die der Algorithmus ausgibt sind auch eine Lösung der Syntax orientierten Analyse
  $(\hat{C}, \hat{\rho}) \models_s e_*$
- Die Syntax orientierte Lösung ist auch eine Lösung der abstrakten Analyse
  $(\hat{C}, \hat{\rho}) \models e_*$

Quellen & Literatur

Noch Fragen?