Theoretische Informatik 1
Große Übung 1

René Maseli
Thomas Haas
TU Braunschweig
Wintersemester 2023/24

Theorem
Sei M eine Menge. $\langle \mathcal{P}(M), \subseteq \rangle$ ist ein vollständiger Verband.

Beweis
$\subseteq$ ist bekanntermaßen reflexiv: Für alle $A \in \mathcal{P}(M)$ gilt $A \subseteq A$, denn $\forall a \in A : a \in A$.
$\subseteq$ ist bekanntermaßen antisymmetrisch: Für alle $A, B \in \mathcal{P}(M)$ mit $A \subseteq B$ und $B \subseteq A$ folgt $A = B$.
$\subseteq$ ist bekanntermaßen transitiv: Für alle $A, B, C \in \mathcal{P}(M)$ mit $A \subseteq B$ und $B \subseteq C$ folgt $A \subseteq C$, denn es gilt $\forall a \in A : a \in B$ und $\forall b \in B : b \in C$ und so folgt $\forall a \in A : a \in C$.

Jedes $X \subseteq \mathcal{P}(M)$ hat einen Join $\bigsqcup X \in \mathcal{P}(M)$: $\bigcup X := \{ m \in M \mid \exists A \in X : m \in A \}$.
Sei $A \in X$.
Es gilt $A \subseteq A \cup \bigcup(X \setminus \{A\}) = \bigcup X$.
Da dies für alle A gilt, ist $\bigcup X$ eine obere Schranke von X.
Sei $S \in \mathcal{P}(M)$ eine obere Schranke von X.
Sei $m \in \bigcup X$.
Per Definition gibt es ein $A \in X$ mit $m \in A$.
Da S eine obere Schranke von X ist, folgt $m \in S$.
Da dies für alle m gilt, folgt $\bigcup X \subseteq S$.
Da dies für alle oberen Schranken S gilt und $\bigcup X$ eine obere Schranke von X ist, ist $\bigcup X = \bigsqcup X$.

Jedes $X \subseteq \mathcal{P}(M)$ hat auch einen Meet $\bigsqcap X \in \mathcal{P}(M)$. $\bigcap X := \{ m \in M \mid \forall A \in X : m \in A \}$.
Sei $A \in X$. Es gilt $\bigcap X = A \cap \bigcap(X \setminus \{A\}) \subseteq A$.
Da dies für alle A gilt, ist $\bigcap X$ eine untere Schranke von X.
Sei $S \in \mathcal{P}(M)$ eine untere Schranke für X.
Sei $m \in S$. Da S eine untere Schranke von X ist, gilt $m \in A$ für alle $A \in X$. Per Definition gilt nun $m \in \bigcap X$.
Da dies für alle m gilt, folgt $S \subseteq \bigcap X$.
Da dies für alle unteren Schranken S gilt und $\bigcap X$ eine untere Schranke von X ist, ist $\bigcap X = \bigsqcap X$.

Zusammen erfüllt $\langle A, \subseteq \rangle$ alle notwendigen Kriterien eines vollständigen Verbandes. □

Theorem

Sei $\langle D, \sqsubseteq \rangle$ eine partielle Ordnung und $K \subseteq D$ eine nichtleere Kette. Falls die aufsteigende Kettenbedingung gilt, hat K ein maximales Element. Falls die absteigende Kettenbedingung gilt, hat K ein minimales Element.

Beweis

Zeige die erste Implikation indirekt, der Beweis für die Zweite ist analog.
Annahme: Es gelte die aufsteigende Kettenbedingung und K habe **kein** maximales Element.
Das heißt jedes Element $x \in K$ ist nicht maximal, also gibt es immer ein Element $y \in K$ mit $y \not\sqsubseteq x$.
Innerhalb der Kette folgt stattdessen $x \sqsubseteq y$.
Daraus lässt sich nun induktiv mindestends eine aufsteigende Kette konstruieren: Da K nicht leer ist, gibt es ein $x_0 \in K$. Da x_i nicht maximal ist, wähle ein beliebiges x_{i+1} mit $x_i \sqsubseteq x_{i+1}$.
Nach ACC kann so eine Kette nicht existieren. Also muss K ein maximales Element besitzen. □

Theorem

Sei $\langle D, \sqsubseteq \rangle$ ein Verband. Der Verband hat genau dann endliche Höhe, wenn ACC und DCC gelten.

Beweis

„⇒": Es sei $\langle D, \sqsubseteq \rangle$ ein Verband mit endlicher Höhe. Zeige ACC, der Beweis für DCC ist analog.
Sei $(x_i)_{i \in \mathbb{N}}$ eine aufsteigende Kette. Ihre Menge $\{x_0, x_1, \ldots\}$ ist eine Kette. Nach Annahme ist diese Menge endlich, also $\{x_0, x_1, \ldots\} = \{a_0, \ldots, a_j\}$ für ein $j \in \mathbb{N}$. Hier seien die a_i aufsteigend sortiert.
Nach Definition der Menge gibt es mindestens einen Index für a_j. Sei $k \in \mathbb{N}$ also der kleinste Index mit $x_k = a_j$. Weil $(x_i)_{i \in \mathbb{N}}$ aufsteigend ist, gilt $x_{k+i} = a_j = x_k$ für alle $i \in \mathbb{N}$, also stabilisiert sich die aufsteigende Kette.
Da dies für jedes $(x_i)_{i \in \mathbb{N}}$ gilt, gilt die aufsteigende Kettenbedingung.

„⇐": Es sei $\langle D, \sqsubseteq \rangle$ ein Verband mit ACC und DCC.
Sei $K \subseteq D$ eine Kette. Falls K leer ist, ist K endlich. Anderenfalls hat K wegen der ACC ein Maximum x_0. Gleichzeitig ist $K \setminus \{x_0\}$ wieder eine Kette. Induktiv konstruieren wir so eine absteigende Kette $(x_i)_{i \in \mathbb{N}}$:

$$x_{i+1} = \begin{cases} x_i & \text{falls } K \setminus \{x_0, \ldots, x_i\} \\ \max(K \setminus \{x_0, \ldots, x_i\}) & \text{sonst} \end{cases}$$

Nach DCC wird diese Kette bei einer Stelle $j \in \mathbb{N}$ stationär: $x_j = x_{j+i}$ für alle $i \in \mathbb{N}$. Das heißt es gilt $K \setminus \{x_0, \ldots, x_j\}$, oder anders dargestellt $K \subseteq \{x_0, \ldots, x_j\}$. Also muss K endlich sein.
Da dies für alle Ketten K gilt, ist die Höhe von $\langle D, \sqsubseteq \rangle$ endlich. □

Definition

Sei a ein (arithmetischer oder boolscher) Ausdruck eines while-Programms. sub(a) ist die Menge der Teilausdrücke von a:

$$\begin{aligned}
\text{sub}(a) &:= \{a\} \text{ falls } a \in \{Var, \mathbb{Z}\} \\
\text{sub}(\text{ nicht } a_1) &:= \{ \text{ nicht } a_1\} \cup \text{sub}(a_1) \\
\text{sub}(a_1 \circ a_2) &:= \{a_1 \circ a_2\} \cup \text{sub}(a_1) \cup \text{sub}(a_2) \text{ für } \circ \in \{+, -, \cdot, <, =, \wedge, \vee\} \\
\text{expressions}(b) &:= \text{sub}(a) \text{ wobei } b \in \left\{[x := a]^l, [a]^l\right\} \\
\text{Exp} &:= \bigcup \{ \text{ expressions}(b) \mid b \in B \} \\
\text{sub}^{-1}(x) &:= \{ e \in \text{Exp} \mid x \in \text{sub}(e) \} \text{ für } x \in \text{Exp}
\end{aligned}$$

Betrachte das folgende while-Programm:

if $[5x + 2 < 7y]^1$ **then**
  $[y := 2y]^2$
end if
while $[(5x + 2) + y < 3x + z]^3$ **do**
  if $[7y < 3x + z]^4$ **then**
    $[z := z - 1]^5$
  else
    $[y := 2y]^6$
  end if
end while
$[x := 3x]^7$

Das Programm induziert u.a. die folgenden Elemente:

$Var := \{x, y, z\}$ $\quad a_1 := 5x + 2 < 7y$ $\quad a_2 := (5x + 2) + y < 3x + z$ $\quad a_3 := 7y < 3x + z$

$B := \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$ $\quad b_1 := [a_1]^1$ $\quad b_2 := [y := 2y]^2$ $\quad b_3 := [a_2]^3$

$b_4 := [a_3]^4$ $\quad b_5 := [z := z - 1]^5$ $\quad b_6 := [y := 2y]^6$ $\quad b_7 := [x := 3x]^7$

$$\text{Exp} = \{1, 2, 3, 5, 7, x, y, z, 3x, 5x, 2y, 7y, z - 1, 5x + 2, 5x + 2 + y, 3x + z, a_1, a_2, a_3\}$$

Available Expressions

Gegeben: Ein Kontrollflussgraph in Programmrichtung

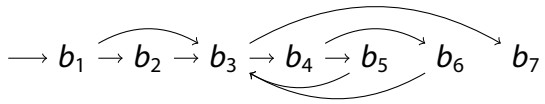
Gesucht: Für jeden Block die Menge der Ausdrücke, die in jedem Weg vor diesem Block schon mal berechnet werden musste, ohne dass zwischendurch eine Variable neu zugewiesen wird.

Betrachte das Datenflusssystem $\langle\langle B, E, F\rangle, \langle \mathcal{P}(\text{Exp}), \supseteq\rangle, \varnothing, TF_1\rangle$ (beachte die umgedrehte Ordnungsrelation, da dies eine **must**-Analyse ist) mit $TF_1 := \{ f_b : \mathcal{P}(\text{Exp}) \to \mathcal{P}(\text{Exp}) \mid b \in B \}$ wobei die Transferfunktionen von folgender Form sind: $f_b(X) := (X \setminus \text{kill}_b) \cup \text{gen}_b$. Dabei sind $\text{kill}_b \in D$ und $\text{gen}_b \in D$ für jeden Block $b \in B$ für Available Expressions so definiert:

$$\text{kill}_b := \begin{cases} \text{sub}^{-1}(x) & \text{falls } b = [x := a]^l \\ \varnothing & \text{sonst} \end{cases}$$

$$\text{gen}_b := \text{expressions}(b) \setminus \text{kill}_b$$

Kontrollflussgraph in Programmrichtung (forward-Analyse) und induziertes Gleichungssystem (der Extremalblock wird hier mit Pfeil aus dem Nichts markiert):



$X_1 = i_1$ $\quad X_2 = f_1(X_1)$ $\quad X_3 = f_1(X_1) \cap f_2(X_2) \cap f_5(X_5) \cap f_6(X_6)$

$X_4 = f_3(X_3)$ $\quad X_5 = f_4(X_4)$ $\quad X_6 = f_4(X_4)$ $\quad X_7 = f_3(X_3)$

Beachte die Verwendung von $\cap$. In Tabellen-Form werden nun sowohl die kill_1- und gen_1-Werte, die Gleichungen des induzierten Gleichungsssystems, als auch die Kleene-Iteration dargestellt.

Anmerkungen:

1. In der Durchführung einer Kleene-Iteration kommt es ausschließlich auf sich ändernde Werte an. Es müssen nur die Blöcke neu berechnet werden, deren Vorgänger-Daten in der vorherigen Iteration verändert wurden.

2. Leere Zellen markieren, wo sich keine Vorgänger-Daten der letzten Iteration verändert haben.

3. Die Pfeile markieren, wo Änderungen aus einer vorigen Iteration zu neuen Berechnungen in der Nächsten führen. (Ebenfalls nicht in Hausaufgaben gefordert.)

b	b_1	b_2	b_3	b_4	b_5	b_6	b_7
kill_b	$\varnothing$	$\text{sub}^{-1}(y)$ $= \{y, 2y, 7y,$ $5x+2+y,$ $a_1, a_2, a_3\}$	$\varnothing$	$\varnothing$	$\text{sub}^{-1}(z)$ $= \{z, z-1,$ $3x+z,$ $a_2, a_3\}$	$\text{sub}^{-1}(y)$ $= \{y, 2y, 7y,$ $5x+2+y,$ $a_1, a_2, a_3\}$	$\text{sub}^{-1}(x)$ $= \{x, 3x, 3x+z,$ $5x, 5x+2,$ $5x+2+y,$ $a_1, a_2, a_3\}$
gen_b	$\text{sub}(a_1)$ $= \{2, 5,$ $7, x,$ $y, 5x,$ $5x+2,$ $7y, a_1\}$	$\text{sub}(2y)$ $\setminus \text{sub}^{-1}(y)$ $= \{2\}$	$\text{sub}(a_2)$ $= \{2, 3, 5,$ $x, y, z, 5x,$ $5x+2, a_2,$ $5x+2+y,$ $3x, 3x+z\}$	$\text{sub}(a_3)$ $= \{3, 7,$ $x, y, z,$ $7y, 3x,$ $3x+z, a_3\}$	$\text{sub}(z-1)$ $\setminus \text{sub}^{-1}(z)$ $= \{1\}$	$\text{sub}(2y)$ $\setminus \text{sub}^{-1}(y)$ $= \{2\}$	$\text{sub}(3x)$ $\setminus \text{sub}^{-1}(x)$ $= \{3\}$
X_b	i_1	$f_1(X_{b_1})$	$f_1(X_{b_1})$ $\cap f_2(X_{b_2})$ $\cap f_5(X_{b_5})$ $\cap f_6(X_{b_6})$	$f_3(X_{b_3})$	$f_4(X_{b_4})$	$f_4(X_{b_4})$	$f_3(X_{b_3})$

$\bot$	Exp	Exp	Exp	Exp	Exp	Exp	Exp
$g_S(\bot)$	$\varnothing$		$1, 2, 3, 5, 7, x,$ $3x, 5x, 5x{+}2$				
$g_S^2(\bot)$		$2, 5, 7,$ $x, y, 5x,$ $5x{+}2, 7y, a_1$	$2, 5, 7, x,$ $5x, 5x{+}2$	$1, 2, 3, 5, 7,$ $x, y, z, 5x, a_2,$ $5x{+}2, 3x{+}z,$ $5x{+}2{+}y, 3x$			$1, 2, 3, 5, 7,$ $x, y, z, 5x, a_2,$ $5x{+}2, 3x{+}z,$ $5x{+}2{+}y, 3x$
$g_S^3(\bot)$			$\ldots$	$2, 3, 5, 7,$ $\ldots$	$1, 2, 3, 5, 7,$ $x, y, z, 5x, a_2$ $5x{+}2, 3x{+}z,$ $5x{+}2{+}y, 3x,$ $7y, a_3$	$1, 2, 3, 5, 7,$ $x, y, z, 5x, a_2$ $5x{+}2, 3x{+}z,$ $5x{+}2{+}y, 3x,$ $7y, a_3$	$2, 3, 5, 7,$ $\ldots$
$g_S^4(\bot)$			$\ldots$		$2, 3, 5, 7,$ $\ldots$	$2, 3, 5, 7,$ $\ldots$	
$g_S^5(\bot)$			$\ldots$				

Die Iteration stabilisiert sich nach 4 Schritten bei $\text{lfp}(g_S) = g_S^4(\bot)$:

$$\text{lfp}(g_S)_{b_1} = \varnothing$$
$$\text{lfp}(g_S)_{b_2} = \{2, 5, 7, x, y, 5x, 5x+2, 7y, a_1\}$$
$$\text{lfp}(g_S)_{b_3} = \{2, 5, 7, x, 5x, 5x+2\}$$
$$\text{lfp}(g_S)_{b_4} = \{2, 3, 5, 7, x, y, 5x, 5x+2, 5x+2+y, 3x, 3x+z, a_2\}$$
$$\text{lfp}(g_S)_{b_5} = \{2, 3, 5, 7, x, y, 5x, 5x+2, 5x+2+y, 3x, 3x+z, 7y, a_2, a_3\}$$
$$\text{lfp}(g_S)_{b_6} = \{2, 3, 5, 7, x, y, 5x, 5x+2, 5x+2+y, 3x, 3x+z, 7y, a_2, a_3\}$$
$$\text{lfp}(g_S)_{b_7} = \{2, 3, 5, 7, x, y, 5x, 5x+2, 5x+2+y, 3x, 3x+z, a_2\}$$