Wiederholung:

- Eine partielle Ordnung (p.O) ist ein Paar $(D, \le)$ mit $\le$ reflexiv, transitiv und anti-symmetrisch.
- $(D, \le)$ ist ein Verband, falls für je zwei Elemente Join $\sqcup$ und Meet $\sqcap$ existieren.
- Ein Verband ist vollständig, falls für alle Teilmengen $X \subseteq D$ Join und Meet existieren.

1.4 Ketten

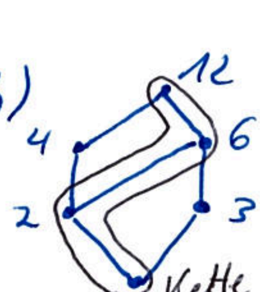
Sei $(D, \le)$ p.O.

- Eine total geordnete Teilmenge $K \subseteq D$ heißt Kette.
  ($\forall k_1, k_2 \in K: k_1 \le k_2$ oder $k_2 \le k_1$)
- Eine Folge $(k_i)_{i \in \mathbb{N}}$ heißt aufsteigende Kette, falls $k_i \le k_{i+1}$ f.a. $i \in \mathbb{N}$.
- Eine Folge $(k_i)_{i \in \mathbb{N}}$ heißt absteigende Kette, falls $k_i \ge k_{i+1}$ f.a. $i \in \mathbb{N}$.
- Eine auf/absteigende Kette $(k_i)_{i \in \mathbb{N}}$ wird stationär, falls $\exists n \in \mathbb{N}: \forall i \ge n: k_i = k_n$.
- $(D, \le)$ hat endliche Höhe, falls jede Kette K in D endlich viele Elemente hat.
- $(D, \le)$ hat beschränkte Höhe, falls es $n \in \mathbb{N}$ gibt, so dass jede Kette höchstens n Elemente hat.

Beispiel:

(a) In $(\mathbb{N}, \le)$ wird jede absteigende Kette stationär.

(b)



Definition (Kettenbedingungen):

Eine p.O. $(D, \leq)$

↳ erfüllt die <u>aufsteigende</u> <u>Kettenbedingung</u> <u>(ACC)</u> (man sagt auch $(D, \leq)$ sei <u>Artinisch</u>), falls

jede aufsteigende Kette $k_0 \leq k_1 \leq \ldots$ stationär wird.

↳ erfüllt die <u>absteigende</u> <u>Kettenbedingung</u> <u>(DCC)</u> (ist <u>Noethersch</u>), falls

jede absteigende Kette $k_0 \geq k_1 \geq \ldots$ stationär wird.

Beachte: (ACC) und (DCC) sind unabhängig von den Verbandsbedingungen.

Lemma:

Eine p.O. hat endliche Höhe gdw. (ACC) und (DCC) erfüllt sind.

Definition (Stetigkeit)

Sei $(D, \leq)$ ein vollständiger Verband. Eine Funktion $f: D \to D$ heißt

(i) <u>$\sqcup$-stetig</u> (<u>aufwärtsstetig</u>), falls für jede Kette K in D gilt

$$f(\sqcup K) = \sqcup f(K) = \sqcup \{ f(k) \mid k \in K \}.$$

(ii) <u>$\sqcap$-stetig</u> (<u>abwärtsstetig</u>), falls für jede Kette K in D gilt

$$f(\sqcap K) = \sqcap f(K).$$

Satz (Monotonie impliziert Stetigkeit):

Sei $(D, \leq)$ ein vollständiger Verband und $f: D \to D$ monoton.

(i) Falls $(D, \leq)$ (ACC) erfüllt, dann ist f $\sqcup$-stetig.

(ii) Falls $(D, \leq)$ (DCC) erfüllt, dann ist f $\sqcap$-stetig.

Beweis:

Sei K eine abzählbare Kette in D, die ohne Einschränkung als Folge dargestellt werden kann:

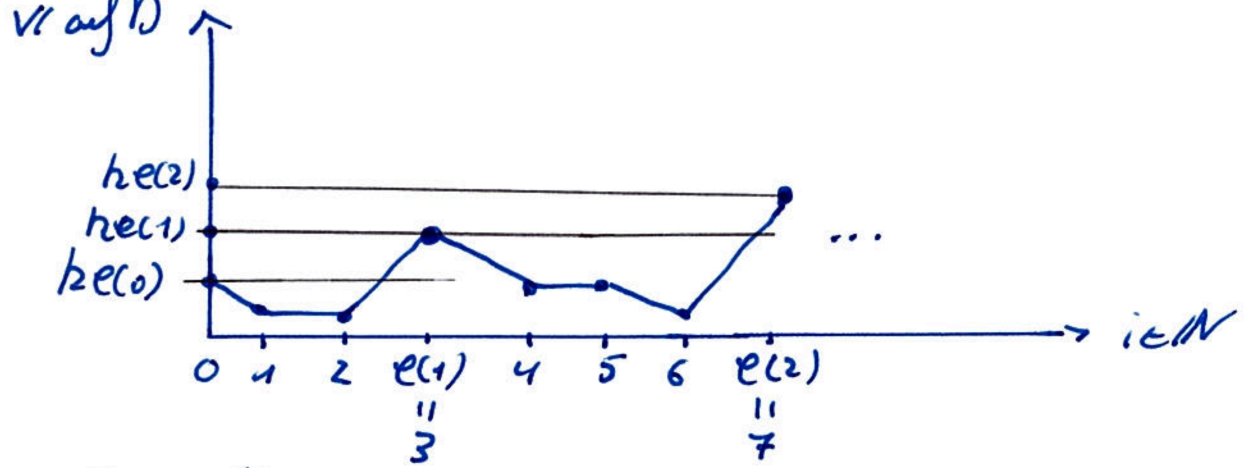
$$K = \{k_0, k_1, \dots\} = (k_i)_{i \in \mathbb{N}}.$$

Wir konstruieren eine aufsteigende Teilfolge $(k_{\ell(i)})_{i \in \mathbb{N}}$ mit

$k_{\ell(0)} := k_0$ $\quad k_{\ell(i+1)} := k_j$ mit $j > \ell(i)$ der kleinste Index, so dass $k_j \geq k_{\ell(i)}$.

Falls es kein solches k_j gibt, wähle $k_{\ell(i+1)} = k_{\ell(i)}$.

Anschaulich:



- Es gilt $\bigsqcup K = \bigsqcup \{k_{\ell(i)} \mid i \in \mathbb{N}\}$

  Für $\sqsupseteq$, beachte $\{k_{\ell(i)} \mid i \in \mathbb{N}\} \subseteq K$.

  Für $\sqsubseteq$, beachte, dass es für jedes Element $k \in K$ einen Index $i \in \mathbb{N}$ gibt mit $k \sqsubseteq k_{\ell(i)}$.

- Mit (ACC) wird $(k_{\ell(i)})_{i \in \mathbb{N}}$ stationär.

  Sei das entsprechende Element $k_{\ell(n)}$.

- Damit folgt

$$f(\bigsqcup K) \overset{(\text{oben})}{=} f(\bigsqcup \{k_{\ell(i)} \mid i \in \mathbb{N}\})$$
$$\overset{(\text{stationär})}{=} f(k_{\ell(n)})$$
$$\overset{(\text{Monotonie})}{=} \bigsqcup \{f(k_{\ell(i)}) \mid i \leq n\}$$

$$\left(f(k_{e(n+1)}) = f(k_{e(n)}) \right) = \bigsqcup \{ f(k_{e(i)}) \mid i \in \mathbb{N} \}$$
$$= \bigsqcup f(K).$$

Der letzte Schritt gilt, da es für jedes $k \in K$ ein $k_{e(i)}$ gibt mit $k \leq k_{e(i)}$.
Per Monotonie folgt dann $f(k) \leq f(k_{e(i)})$. □

<u>Lemma:</u>
Sei $(D, \leq)$ ein vollständiger Verband und $f: D \to D$ monoton.
Die Folge
$$(f^i(\bot))_{i \in \mathbb{N}} \quad \text{mit } f^0(\bot) := \bot \text{ und } f^{i+1}(\bot) := f(f^i(\bot))$$
ist eine aufsteigende Kette.

<u>Beweis:</u>
<u>Zeige:</u> $f^i(\bot) \leq f^{i+1}(\bot)$ f.a. $i \in \mathbb{N}$.
<u>IA:</u> $f^0(\bot) = \bot \leq f(\bot)$, da $\bot = \sqcap D$.
<u>IS:</u> Angenommen $f^i(\bot) \leq f^{i+1}(\bot)$,
dann folgt
$$f^{i+1}(\bot) = f(f^i(\bot))$$
$$\underset{\text{(Induktionsvoraussetzung + Monotonie)}}{\leq} f(f^{i+1}(\bot)) = f^{i+2}(\bot).$$
□

<u>Satz</u> (<u>Knaster</u>, <u>Tarski</u>, <u>Kleene</u>):
Sei $(D, \leq)$ ein vollständiger Verband und $f: D \to D$ monoton.
(i) Ist f $\sqcup$-stetig, dann gilt
$$lfp(f) = \bigsqcup \{ f^i(\bot) \mid i \in \mathbb{N} \}$$
(ii) Ist f $\sqcap$-stetig, dann gilt
$$gfp(f) = \bigsqcap \{ f^i(\top) \mid i \in \mathbb{N} \}.$$

Beweis (i):

Zeige: $\bigsqcup\{f^i(\bot) \mid i \in \mathbb{N}\}$ ist Fixpunkt.

$$
\begin{aligned}
& f(\bigsqcup\{f^i(\bot) \mid i\in\mathbb{N}\}) \\
(f\ \omega\text{-stetig}) \quad &= \bigsqcup\{f^{i+1}(\bot) \mid i \in \mathbb{N}\} \\
(\bot = \sqcap D) \quad &= \bigsqcup\{f^i(\bot) \mid i\in\mathbb{N}\}
\end{aligned}
$$

Zeige: $\bigsqcup\{f^i(\bot) \mid i\in\mathbb{N}\}$ ist kleinste Fixpunkt.

- Betrachte $d \in D$ mit $f(d) = d$
  und zeige $\bigsqcup\{f^i(\bot) \mid i\in\mathbb{N}\}$ ist kleiner.
- Induktion nach $i \in \mathbb{N}$ gibt: $f^i(\bot) \leq d$ f.a. $i\in\mathbb{N}$.

  IA: $f^0(\bot) = \bot \leq d$, da $\bot = \sqcap D$

  IS: Angenommen $f^i(\bot) \leq d$.
  Dann gilt:
  $$f^{i+1}(\bot) = f(f^i(\bot)) \overset{(IH + Mon.)}{\leq} f(d) = d.$$

- Da $f^i(\bot) \leq d$ für alle $i\in\mathbb{N}$,
  folgt $\bigsqcup\{f^i(\bot) \mid i\in\mathbb{N}\} \leq d$. □

Satz:
Sei $(D,\leq)$ ein vollständiger Verband mit (ACC) und (DCC).
Sei $f: D \to D$ monoton.
Dann ist

$$
\begin{aligned}
lfp(f) &= \bigsqcup\{f^i(\bot) \mid i\in\mathbb{N}\} \\
&= f^n(\bot) \quad \text{mit } f^n(\bot) = f^{n+1}(\bot). \\
gfp(f) &= \sqcap\{f^i(\top) \mid i\in\mathbb{N}\} \\
&= f^n(\top) \quad \text{mit } f^n(\top) = f^{n+1}(\top).
\end{aligned}
$$

Beweis: Aus Monotonie folgt Stetigkeit wegen (ACC) und (DCC).
 Dann Knaster, Tarski und Kleene. □

2. Datenflussanalyse:

Ziel: Analysiere das Verhalten von Programmen statisch,
d.h. zur Compile-Zeit.

Ansatz: Fixpunktberechnung auf einer abstrakten Datendomäne.

2.1 While-Programme

Definition (Syntax beschrifteter while-Programme):

Die Syntax von beschrifteten while-Programmen ist durch folgende BNF gegeben:

$a ::= k \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2$

// Arithmetische Ausdrücke

$b ::= t \mid a_1 = a_2 \mid a_1 > a_2 \mid \neg b \mid b_1 \wedge b_2 \mid b_1 \vee b_2$

// Boolesche Ausdrücke

$c ::= [\text{skip}]^l \mid [x := a]^l \mid c_1 ; c_2$

$\mid$ if $[b]^l$ then c_1 else c_2 fi

$\mid$ while $[b]^l$ do c od

// Programme

- Dabei sei $k \in \mathbb{Z}$, $t \in \mathbb{B}$ und $x \in Var$.
- Ferner wird angenommen, dass alle Labels im Programm verschieden sind.
- Beschriftete Befehle werden Blöcke genannt.

Programme lassen sich als Kontrollflussgraphen darstellen

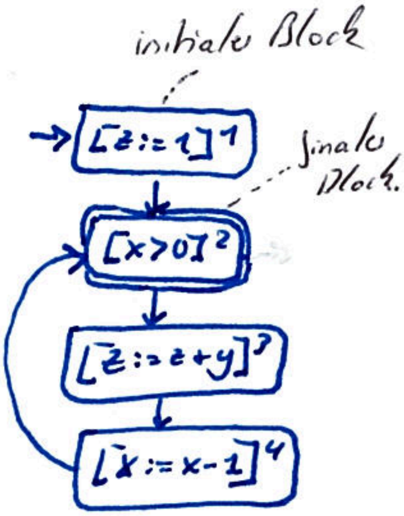
$G = (B, E, F)$,

dabei ist B = Blöcke im Programm
E = Menge an externalen Blöcken (initial oder final)
$F \subseteq B \times B$ = Flussrelation.

- Typischerweise repräsentieren Kontrollflussgraphen die Struktur eines Programms:

$$c = [z:=1]^1$$
$$\underline{\text{while}}\ [x>0]^2\ \underline{\text{do}}$$
$$[z:=z+y]^3;$$
$$[x:=x-1]^4$$

gibt



- Es gibt jedoch Datenflussanalysen, die Programme entgegen der Befehlsfolge (rückwärts) analysieren (Live-Variables zum Beispiel). Daher werden wir bei einer Datenflussanalyse den zugrundeliegenden Kontrollflussgraphen genau festlegen.

- Für Kontrollflussgraphen wird angenommen, dass
  ↳ der initiale Block keine eingehenden Kanten hat.
  ↳ die finalen Blöcke keine ausgehenden Kanten haben.

  Diese Form lässt sich durch Hinzufügen von skip-Befehlen immer herstellen.
  Das obige Beispiel erfüllt die Bedingung für initiale Blöcke, verletzt aber die Bedingung für finale Blöcke.

2.2 Monotone Frameworks

Monotone Frameworks nutzen einen vollständigen Verband als abstrakte Datendomäne und imitieren die Befehle des Programms durch monotone Funktionen.

<u>Definition (Datenflusssystem)</u>:
Ein <u>Datenflusssystem</u> ist ein Tupel $S = (G, (D, \leq), i, f)$
mit
- $G = (B, E, F)$ ein <u>Kontrollflussgraph</u>
- $(D, \leq)$ ein <u>vollständiger Verband</u> mit <u>(ACC)</u>

- $c \in D$ ein Anfangswert für Extremalblöcke
- $f = \{f_b : D \to D \mid b \in B\}$ eine Familie von Funktionen, eine für jeden Block, die alle monoton sind.

Die Datenflussanalyse induziert ein Gleichungssystem

$$X_b = \begin{cases} c & \text{, falls } b \in E \\ \bigsqcup \{ f_{b'}(X_{b'}) \mid (b', b) \in F \} & \text{, sonst.} \end{cases}$$

Ein Vektor $(d_1, \dots, d_{|B|}) \in D^{|B|}$ heißt Lösung von S, falls

$$d_b = \begin{cases} c & \text{, falls } b \in E \\ \bigsqcup \{ f_{b'}(d_{b'}) \mid (b', b) \in F \} & \text{, sonst.} \end{cases}$$

Um den Zusammenhang zwischen den Lösungen des Gleichungssystems von S sowie Fixpunkten herzustellen, definiere die Funktion

$$g_S : D^{|B|} \to D^{|B|}$$
$$(d_1, \dots, d_{|B|}) \mapsto (d_1', \dots, d_{|B|}')$$

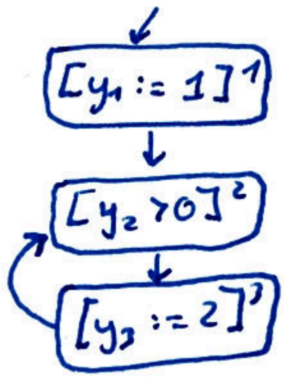
durch

$$d_b' = \begin{cases} c & \text{, falls } b \in E \\ \bigsqcup \{ f_{b'}(d_{b'}) \mid (b', b) \in F \} & \text{, sonst.} \end{cases}$$

Satz
Vektor $\vec{d} = (d_1, \dots, d_{|B|}) \in D^{|B|}$ löst das Gleichungssystem von S gdw. $g_S(\vec{d}) = \vec{d}$, d.h. $\vec{d}$ ist Fixpunkt von g_S.

Beispiel:
Es soll eine Programmanalyse definiert werden, die die Menge an Variablen berechnet, die an einem Programmpunkt geschrieben worden sind.

Betrachte das Programm mit $G =$



$$\begin{aligned} c = \; & [y_1 := 1]^1; \\ & \underline{\text{while}} \; [y_2 > 0]^2 \; \underline{\text{do}} \\ & \quad [y_3 := 2]^3; \\ & \underline{\text{od}} \end{aligned}$$

Das zugehörige Datenflusssystem ist

$$S = (G, (\mathbb{P}(\{y_1, y_2, y_3\}), \subseteq), \emptyset, \{f_1, f_2, f_3\})$$

mit

$$f_1, f_2, f_3 : \mathbb{P}(\{y_1, y_2, y_3\}) \longrightarrow \mathbb{P}(\{y_1, y_2, y_3\})$$

$$\begin{aligned} f_1(X) &:= X \cup \{y_1\} \\ f_2(X) &:= X \\ f_3(X) &:= X \cup \{y_3\} \end{aligned}$$

Das Datenflusssystem induziert das Gleichungssystem

$$\begin{aligned} X_1 &= \emptyset \\ X_2 &= \underbrace{X_1 \cup \{y_1\}}_{= f_1(X_1)} \cup \underbrace{X_3 \cup \{y_3\}}_{= f_3(X_3)} \\ X_3 &= \underbrace{X_2}_{= f_2(X_2)}. \end{aligned}$$

Eine Lösung ist $(\emptyset, \{y_1, y_3\}, \{y_1, y_3\})$.