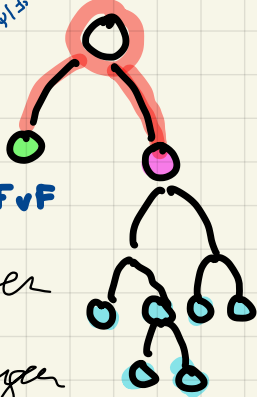
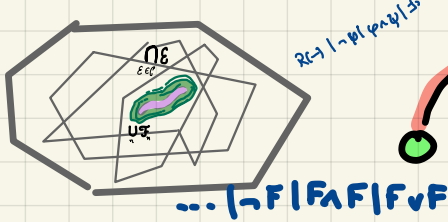


5. Mai 2021
(Nachgeschrieben)



Übersicht

- §1. Top-Down Konstruktion für Mengen
- §2. Bottom-Up Konstruktion für Mengen
- §3. Strukturelle Induktion
- §4. Strukturelle Rekursion
- §5. Präsentation von Mengen durch Schemata

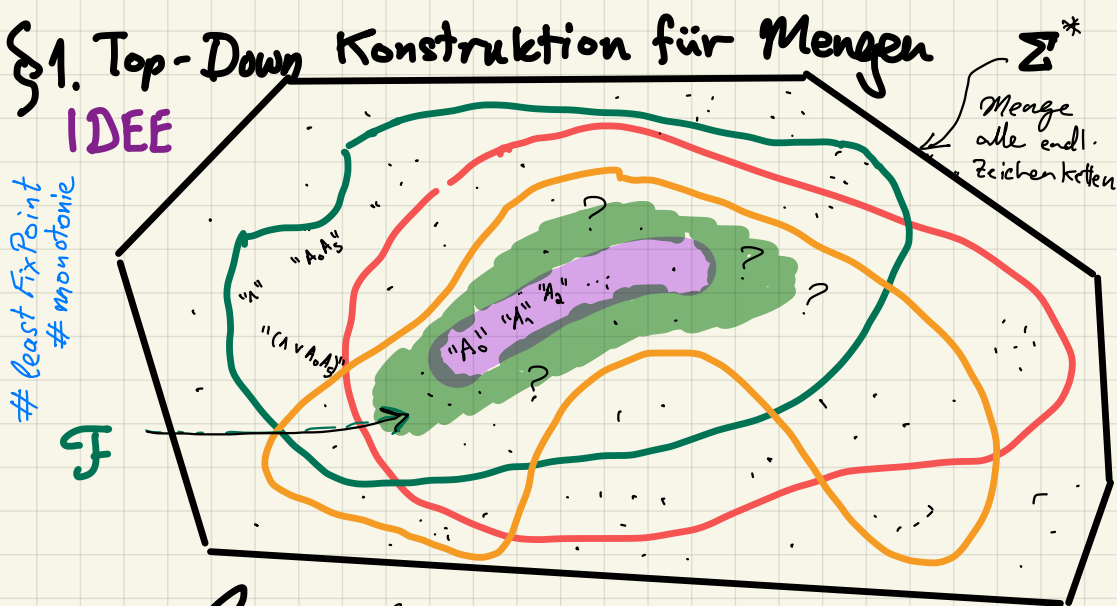
Motivation

In der Logik konstruieren wir mehrfach verschachtelte Ausdrücke, die wir "Formeln" nennen. Um auf diese Metaanalysen "auszuführen" und auf ihnen Funktionen zu bauen, brauchen wir Mittel, um von der Komplexität nicht überfordert zu werden.

Wir werden zwei Aufbauprinzipien einführen und gleichzeitig konkret für den Aufbau der Menge der Formeln, \mathcal{F} , gebrauchen. Wir zeigen, dass beide Sichtweisen zum selben Ergebnis führen, aber nutzen mal die eine, mal die andere aus, um entweder

- einfacher das Induktionssprinzip zu begründen;
- oder konkret Funktionen zu konstruieren. (strukturelle Rekursion)

Zum Schluss wird gezeigt wie Logiker solche Flut und Mengen effizient aufschreiben.



Setze $\mathcal{C} := \{ E \subseteq \Sigma^* \mid E \supseteq \{A_0, A_1, \dots\}$

Z. B. bzgl. des o. s. Bildes:

$\mathcal{C} = \{ \square, \text{red}, \text{green}, \text{orange}, \dots \}$

und $\forall F \in \mathcal{C}; \neg F \in \mathcal{C}$ *
 und $\forall G, H \in \mathcal{C}; (G \cup H) \in \mathcal{C}$
 und $\forall G, H \in \mathcal{C}; (G \cap H) \in \mathcal{C}$ } *

Beob. 1 \mathcal{C} nicht leer, weil u. a. die Menge Σ^* offensichtlich * erfüllt, und somit zu \mathcal{C} gehört □

Setze nun

$$\mathcal{F}_{\text{top-down}} := \bigcap_{E \in \mathcal{C}} E \quad (\text{wird auch } \bigcap \mathcal{C} \text{ geschrieben})$$

Beob. 2 $\emptyset \subset \mathcal{F}_{\text{top-down}} \subseteq \Sigma^*$, wohldefiniert,
 nicht leer weil alle E in \mathcal{C} $\{A_0, A_1, \dots\}$ enthalten
 und somit gilt $\mathcal{F} = \bigcap_{E \in \mathcal{C}} E \supseteq \{A_0, A_1, \dots\}$. □

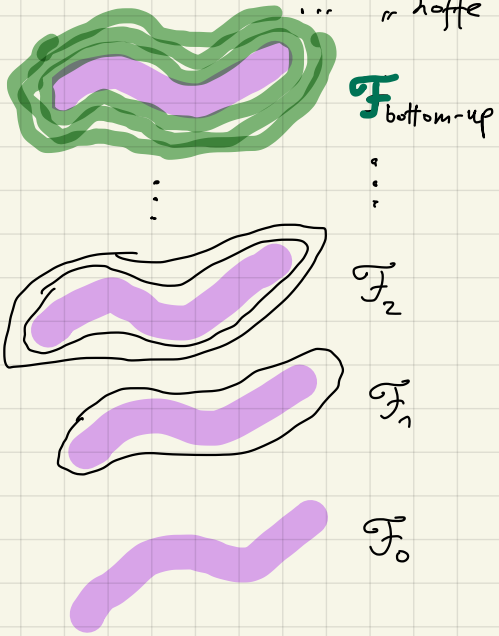
Satz (Übung) $\mathcal{F}_{\text{top-down}}$ selbst liegt in \mathcal{C} . □

Folgerung $\mathcal{F}_{\text{top-down}}$ ist die kleinste Menge die * erfüllt. □

§2. Bottom-Up Konstruktion für Mengen.

IDEE

Farbe mit einer Basismenge an, schließe nach-und-nach unter Operationen ab ... "hoffe", dass Endresultat \mathcal{F} erfüllt.



Konstruktion

Setze $\mathcal{F}_0 := \{A_0, A_1, \dots\}$ und für jedes $n \geq 0$:

$$\mathcal{F}_{n+1} := \mathcal{F}_n \cup$$

$$\left\{ \neg F, (G \cap H), (G \cup H), F, G, H \in \mathcal{F}_n \right\}$$

$$\mathcal{F}_{\text{bottom-up}} := \bigcup_{n \geq 0} \mathcal{F}_n$$

Beob 4 $\{A_0, A_1, \dots\} = \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \dots \subseteq \mathcal{F}_n \subseteq \dots \subseteq \mathcal{F} \quad \square$

Satz 5 (Übung) \mathcal{F} erfüllt \mathcal{F} \square

Theorem $\mathcal{F}_{\text{bottom-up}} = \mathcal{F}_{\text{top-down}}$.

Bew (Skizze) (\supseteq) Da $\mathcal{F}_{\text{bottom-up}}$ erfüllt, und $\mathcal{F}_{\text{top-down}}$ die "kleinste" solche Menge ist.

(\subseteq) Es reicht aus für alle $n \geq 0$ und alle $E \in \mathcal{C}$ zu zeigen, dass $\mathcal{F}_n \subseteq E$. (Übung: benutzt klassische Ind über \mathbb{N} + Tatsache, dass E erfüllt)

$$\mathcal{F}_{\text{bottom-up}} \stackrel{\text{Def 2}}{=} \bigcup_{n \geq 0} \mathcal{F}_n \subseteq \bigcap_{E \in \mathcal{C}} E \stackrel{\text{Def 1}}{=} \mathcal{F}_{\text{top-down}} \quad \square$$

§3. Strukturelle Induktion

HINTERGRUND

Angenommen, wir wollen zeigen, dass alle Formeln, F , eine gegebene (meta) Eigenschaft, $\Phi(\cdot)$, erfüllen.

Ansatz I

Im Hintergrund: es gibt eine binäre Relation

$$F < F' : \Leftrightarrow F \text{ (strikte) Teilformel von } F'$$

Dies ist eine **# Wohlordnung** \rightarrow nachschauen

- Zeige: $\Phi(A_i)$ für alle Atome
 - Zeige: $\Phi(F) \Rightarrow \Phi(\neg F)$
 - Zeige: $\Phi(G), \Phi(H) \Rightarrow \Phi(G \wedge H)$
 - Zeige: $\Phi(G), \Phi(H) \Rightarrow \Phi(G \vee H)$
- \uparrow für alle Formeln F, G, H .

Ansatz II

Setze $\mathcal{E} := \{F \in \mathcal{F} \mid \Phi(F)\}$.

Das Ziel ist äq. zum Ziel:

Zeige, dass $\mathcal{E} = \mathcal{F}$.

- Zeige: $A_i \in \mathcal{E}$ für alle Atome A_i .
- Zeige: $F \in \mathcal{E} \Rightarrow \neg F \in \mathcal{E}$
- Zeige: $G, H \in \mathcal{E} \Rightarrow (G \wedge H) \in \mathcal{E}$
- Zeige: $G, H \in \mathcal{E} \Rightarrow (G \vee H) \in \mathcal{E}$

(+)

Beob. 6

und sind äquivalent, nur mit anderer Betonung.

Satz 7a

Wenn alles gilt, dann $\Phi(F)$
Für alle Formeln $F \in \mathcal{F}$. \square

← indirekter Beweis (hinterrück)

Satz 7b

Wenn alles gilt, dann $\mathcal{E} = \mathcal{F}$,
m.a.W. $\Phi(F)$ für alle $F \in \mathcal{F}$. \square

Bew. aus (+)

folgt, dass $\mathcal{E} \neq \mathcal{F}$ erfüllt. Da \mathcal{F} die kleinste solche Menge ist, gilt $\mathcal{E} = \mathcal{F}$.
und " \subseteq " gilt per Konstruktion. \square

Direkter Beweis.

Bedient: **top-down** Ansicht \rightarrow

(Bew von 7a)

Angenommen, dies sei nicht der Fall.

D.h. es gebe $F \in \mathcal{F}$, so dass $\Phi(F)$ nicht gilt.
Daher ist die Menge

$$\tilde{\mathcal{E}} := \{F \in \mathcal{F} \mid \Phi(F) \text{ gilt nicht}\} \quad (1)$$

nicht leer.

Da $(\mathcal{F}, <)$ eine **Wohlordnung** ist,
existiert ein **minimales** (bzgl. $<$) Element in $\tilde{\mathcal{E}}$.

Sei also

$$\tilde{F} \in \tilde{\mathcal{E}} \text{ minimal in } \tilde{\mathcal{E}}. \quad (2)$$

Wegen der **bottom-up** Konstruktion wissen wir,
dass nur folgende Fälle möglich sind:

Fall 1 \tilde{F} ist atomische Formel, also A_0, A_1, \dots

Dann per I gilt $\Phi(\tilde{F})$.

Fall 2 \tilde{F} setzt sich aus Teilformeln
zusammen. Wegen **Minimalität** von \tilde{F} in $\tilde{\mathcal{E}}$

können diese Teilformeln nicht in $\tilde{\mathcal{E}}$ liegen.

Per Wahl von $\tilde{\mathcal{E}}$ heißt das, dass $\Phi(G)$ für alle
Teilformeln von \tilde{F} gilt.

Aber dann per II gilt $\Phi(\tilde{F})$.

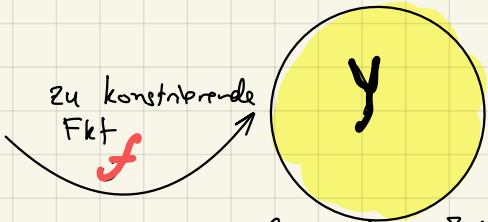
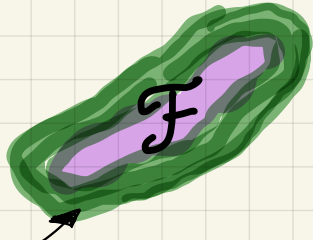
Also gilt in allen Fällen, dass \tilde{F} Eigenschaft Φ
erfüllt. Dies widerspricht (1) + (2).

Daher stimmt die **Annahme** oben nicht \square

§4. Strukturelle Rekursion

rekursives Schema

SETUP



Wegen **Bottom-up** Sichtweise weiß man, für jedes $F \in \mathcal{F}$ exakt eines der Folgenden gilt:

- F ein Atom A_i
- F der Form $\neg G$
- F der Form $(G \wedge H)$
- F der Form $(G \vee H)$

wobei A_i bzw. Teilformeln $G, H \in \mathcal{F}$ eindeutig durch F bestimmt.

IDEE $f(F)$ komplett durch **Formeltyp** (welcher Fall links gilt)

+ Wert von f auf Teilforml bestimmen.

Also ist das **Ziel**, ein (hoffentlich eindeutiges) f zu finden, die

- $f(A_i) = c_i$
- $f(\neg G) = g_{\neg}(f(G))$
- $f(G \wedge H) = g_{\wedge}(f(G), f(H))$
- $f(G \vee H) = g_{\vee}(f(G), f(H))$

erfüllt, wobei $c_i \in Y$ und $g_{\neg}: \mathcal{F} \rightarrow Y$
 $g_{\wedge}, g_{\vee}: \mathcal{F} \times \mathcal{F} \rightarrow Y$

idR „einfach“ zu beschreibende Fkt sind.

Beob 8

Wenn f erfüllt,

dann ist f durch $c_i, g_{\neg}, g_{\wedge}, g_{\vee}$ „erzeugt“ und sehr intuitiv (bspw. programmatisch) zu implementieren. Um bspw.

$$f((A_0 \vee \neg (A_3 \wedge \neg A_4)))$$

zu berechnen, kann man die Komplexität aller Teilteilformeln ausblenden und die Berechnung als

$$g_{\vee}(f(\bullet), f(\bullet))$$

zu verstehen.

„rekursives Schema für f “

Satz 9 Fixiere $Y, c_i \in Y, i \in \mathbb{N}, g_1 : \mathcal{F} \rightarrow Y,$
 $g_1, g_v : \mathcal{F} \times \mathcal{F} \rightarrow Y.$

Dann ex. eine Fkt $f : \mathcal{F} \rightarrow Y$, die das rekursive Schema erfüllt... und sie ist **eindeutig**. \square

Folgerung Wegen Existenz + Eindeutigkeit wird das Präsentieren eines solchen Schemas in der Praxis als die Definition von einer solchen Funktion betrachtet will man

bspw. $l(\cdot) : \mathcal{F} \rightarrow \mathbb{N}$ (Länge) definieren, denn reicht

$$l(A_i) := 0$$

$$l(\neg A) := l(A) + 1$$

$$l((A \wedge H)) := l(A) + l(H) + 1$$

$$l((A \vee H)) := l(A) + l(H) + 1$$

als Definition von l , auch wenn dies nur die erzeugenden Funktionen beschreibt

Bew (Satz 9) (siehe Literatur) Baut auf der **Wohlordnung** der Teilformel-Relation auf.

Die Eindeutigkeit lässt sich per Induktion beweisen. \square

§5. Präsentation von Mengen durch Schemata

Analog zu rekursiven Schemata für durch str. Rek konstruierte Funktionen, können wir auf die Erwähnung des Apparats (top-down/bottom-up) bei dem Aufbau von Mengen verzichten. Es reicht, das Schema zu präsentieren.

Variante 1 Die Menge \mathcal{F} ist die kleinste Menge, so dass

- 1) $A_0, A_1, \dots \in \mathcal{F}$
- 2) $F \in \mathcal{F} \Rightarrow \neg F \in \mathcal{F}$
- 3) $G, H \in \mathcal{F} \Rightarrow (G \wedge H) \in \mathcal{F}$
- 4) $G, H \in \mathcal{F} \Rightarrow (G \vee H) \in \mathcal{F}$.

Variante 2 Die Menge, \mathcal{F} , von Formeln, F , sei durch das Schema

$$\mathcal{F} := A_0, A_1, \dots \mid \neg F \mid (F \wedge F) \mid (F \vee F)$$

gegeben.

ACHTUNG: der doppelte Gebrauch $\uparrow \uparrow$ wirkt problematisch. Dies wird aber als **Grammetik** verstanden, wobei wir hier so etwas wie Typen definieren (vgl. Implementierung in Git Repo \rightarrow /code/aussagenlogik/grammar.lark).

Weitere Beispiele

\mathbb{Q}^+ als Teilmenge von \mathbb{R} : $\tau := 1 \mid \tau^{-1} \mid \tau + \tau \mid \tau \cdot \tau$

NNF als Teilmenge von \mathcal{F} :

$$P := A_0, \neg A_0, A_1, \neg A_1, \dots \mid P \wedge P \mid P \vee P$$