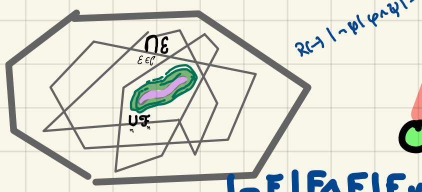
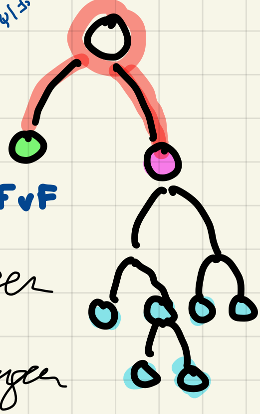
5. Mai 2021
(nachgeschrieben)



$\dots \mid \neg F \mid F \wedge F \mid F \vee F$



Übersicht

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

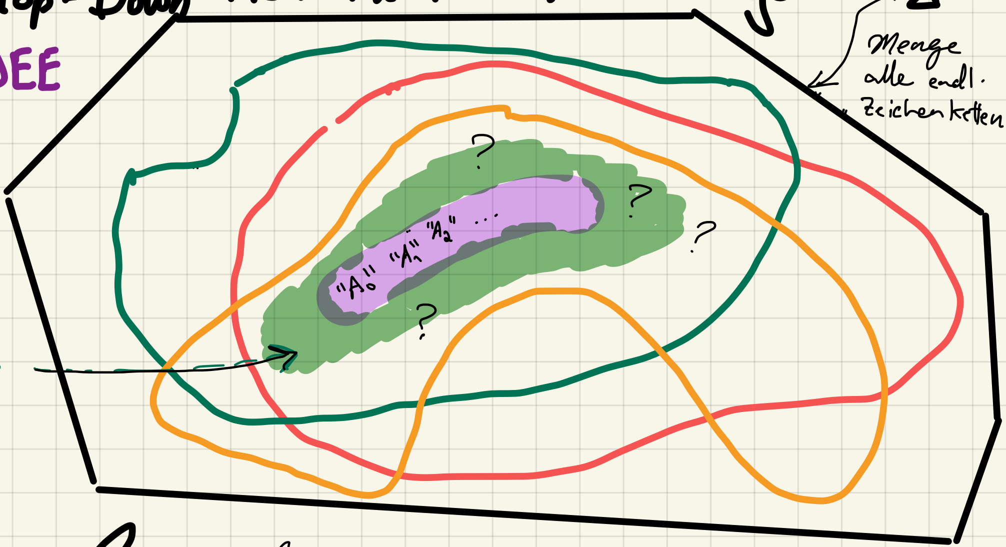
- einfacher das Induktionsprinzip zu begründen;
- oder konkret Funktionen zu konstruieren. (strukturelle Rekursion)

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

§1. Top-Down Konstruktion für Mengen

IDEE

#least FixPoint #monotonie



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

und $\forall F \in \mathcal{E}: \neg F \in \mathcal{E}$

und $\forall G, H \in \mathcal{E}: (G \wedge H) \in \mathcal{E}$

und $\forall G, H \in \mathcal{E}: (G \vee H) \in \mathcal{E} \}$ ⊛

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

$\mathcal{C} = \{$  $, \dots \}$

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

Setze nun

$$\mathcal{F}_{\text{top-down}} := \bigcap_{\mathcal{E} \in \mathcal{C}} \mathcal{E}$$

(wird auch $\bigcap \mathcal{C}$ geschrieben)

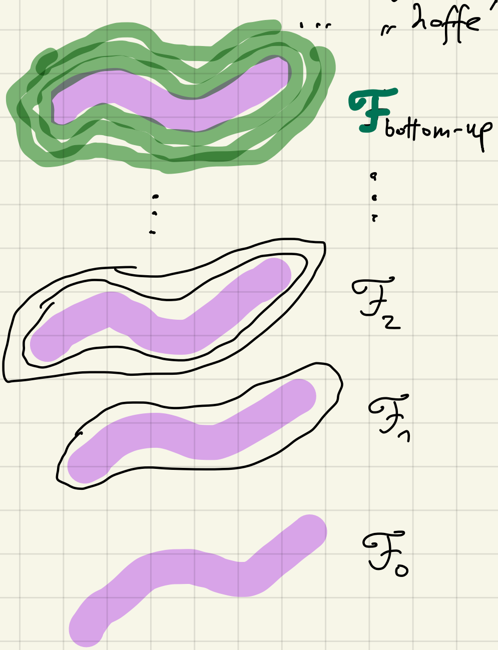
Beob. 2 $\emptyset \subset \mathcal{F}_{\text{top-down}} \subseteq \Sigma^*$, wohldefiniert, nicht leer, weil alle $\mathcal{E}$ in $\mathcal{C}$ $\{A_0, A_1, \dots\}$ enthalten und somit gilt $\mathcal{F} = \bigcap_{\mathcal{E} \in \mathcal{C}} \mathcal{E} \supseteq \{A_0, A_1, \dots\}$. □

Satz 3 (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 Fange mit einer Basismenge an, schließe nach-und-nach unter Operationen ab ... „hoffe", dass Endresultat (*) erfüllt.



Konstruktion

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

$$\mathcal{F}_{n+1} := \mathcal{F}_n \cup \{\neg F, (G \wedge H), (G \vee H) \mid F, G, H \in \mathcal{F}_n\}$$

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

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

Satz 5 (Übung) $\mathbf{F}$ erfüllt (*) □

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

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

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

Dann folgt

$$\mathcal{F}_{bottom-up} \overset{\text{Def}^n}{=} \bigcup_{n \geq 0} \mathcal{F}_n \subseteq \bigcap_{\mathcal{E} \in \mathcal{C}} \mathcal{E} \overset{\text{Def}^n}{=} \mathcal{F}_{top-down}.$$

□

§3. Strukturelle Induktion

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

Ansatz I

Im Hintergrund: es gibt eine binäre Relation
$F \prec F' :\Longleftrightarrow F$ (strikte) Teilfml von F'

Dies ist eine #Wohlordnung → nachschlagen

- 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: $\Phi(G), \Phi(H) \Rightarrow \Phi((G \vee H))$

↳ für alle Formeln F, G, H.

Ansatz II

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

Das Ziel ist äqv. 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 andererer Betonung.

Satz 7a Wenn alles gilt, dann $\Phi(F)$ für alle Formeln $F \in \mathcal{F}$. □

Satz 7b Wenn alles gilt, dann $\mathcal{E} = \mathcal{F}$, m.a.W. $\Phi(F)$ für alle $F \in \mathcal{F}$. □

Direkter Beweis. Bedient „top-down" Ansicht

← indirekter Beweis (bitte wenden)

Bew. aus $(+)$ folgt, dass $\mathcal{E}$ (*) erfüllt. Da $\mathcal{F}$ die kleinste solche Menge ist, gilt $\mathcal{E} \supseteq \mathcal{F}$. Und „$\subseteq$" gilt per Konstruktion. □

(Bew von 7a)

Angenommen, dies sei nicht der Fall.
D.h. es gebe $F \in \mathcal{F}$, so dass $\Phi(F)$ nicht gilt.
Darum ist die Menge

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

nicht leer.
Da $(\mathcal{F}, \prec)$ eine **Wohlordnung** ist,
existiert ein **minimales** (bzgl. $\prec$) 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:

<u>Fall 1</u> $\tilde{F}$ ist atomisch(e Formel), also $A_0, A_1, \ldots$

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

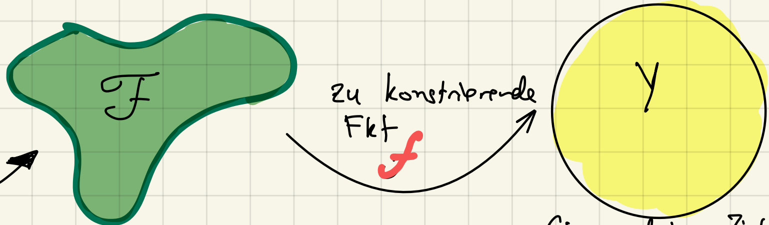
<u>Fall 2</u> $\tilde{F}$ setzt sich aus Teilformeln zusammen. Wegen **Minimalität** von $\tilde{F}$ in $\tilde{\mathcal{E}}$ können diese Teilformeln <u>nicht</u> in $\tilde{\mathcal{E}}$ liegen. Per Wahl von $\tilde{\mathcal{E}}$ heißt das, dass $\Phi(G)$ für alle Teilformeln von $\tilde{F}$ gilt.
Aber dann per II gilt $\Phi(\tilde{F})$.

Also gilt in allen Fällen, dass $\tilde{F}$ Eigenschaft Φ erfüllt. Dies wiederspricht (1) + (2).
Darum stimmt die Annahme oben nicht. $\square$

§ 4. Strukturelle Rekursion

\# rekursives Schema

SETUP



(irgendeine Zielmenge)

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 Teilfml** 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 $c_i \in X$ und $g_\neg : \mathcal{F} \to Y$, $g_\wedge, g_\vee : \mathcal{F} \times \mathcal{F} \to Y$ idR „einfach" zu beschreibende Fkt sind.

„rekursives **Schema** für f"

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. $f((A_0 \vee \neg\neg(A_3 \wedge \neg A_4)))$ zu berechnen, kann man die Komplexität aller Teil teilformeln ausblenden und die Berechnung als

$$g_\vee(f(\ ), f(\ ))$$

zu verstehen.

Satz 9 Fixiere $\mathcal{Y}$, $c_i \in \mathcal{Y}$, $i \in \mathbb{N}$, $g_\neg : \mathcal{F} \to \mathcal{Y}$, $g_\wedge, g_\vee : \mathcal{F} \times \mathcal{F} \to \mathcal{Y}$.

Dann ex. eine Fkt $f : \mathcal{F} \longrightarrow \mathcal{Y}$, 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.** $\ell(\cdot) : \mathcal{F} \longrightarrow \mathbb{N}$ (Länge) definieren, dann reicht

$$\begin{aligned} \ell(A_i) &:= 0 \\ \ell(\neg G) &:= \ell(G) + 1 \\ \ell((G \wedge H)) &:= \ell(G) + \ell(H) + 1 \\ \ell((G \vee H)) &:= \ell(G) + \ell(H) + 1 \end{aligned}$$

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

Analalog zu rekursiven Schemata für durch str. Rek konstruierte Funktionen, können wir auf alle Erwähnung des Apperats (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) $A_0, A_1, \ldots \in \mathcal{F}$
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

$$F := A_0, A_1, \ldots \mid \neg F \mid (F \wedge F) \mid (F \vee F)$$

gegeben.

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

Weitere Beispiele

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

NNF als Teilmenge von $\mathcal{F}$:

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