Aussagenlogik: Hornformeln

- Spezielle Formelklasse
  - Teilklasse der Formeln in Konjunktiver Normalform

    Zur Erinnerung: Wenn F in KNF ist, kann
    F als eine Konjunktion von Klauseln, d.h.,
    als eine Konjunktion von Disjunktionen von Literalen,
    angesehen werden.
  - Charakterisierbar über Implikation als Hauptoperator

    ➔ Grundlage für Logik-Programmierung (z.B. Prolog ➔ SE-3, Datalog ➔ GWV)

    [➔ Hornformeln auf der Basis der Prädikatenlogik]
  - Die Klasse der Hornformeln [➔ Def. 7.1] ist nur eine Teilklasse der Formeln der Aussagenlogik:

    Es gibt sogar Formeln der Aussagenlogik, zu denen es keine äquivalenten Hornformeln gibt [➔ Folie 7.14-7.16].
  - Hornformeln erlauben eine besonders effiziente Berechnung der Prüfung von Erfüllbarkeit und Unerfüllbarkeit [➔ Folie 7.17-7.27].

Grundproblem: Erfüllbarkeit von Formelmengen

Gegeben: Eine Menge von Formeln M.

Frage: Ist M erfüllbar? (Ist es möglich, dass **alle** Formeln in **M** unter derselben Interpretation wahr sind?)

Beobachtung

- Wenn **M** nur Literale enthält, können wir die Frage leicht beantworten (s. Wie auch Erfüllbarkeit von dualen Klauseln.)
  - Beispiel: **M** = $\{\neg A, C, B, \neg D\}$

  ➔ Existieren komplementäre Literale in **M**?
- Wenn **M** Literale und Konjunktionen von Literalen enthält, gilt dasselbe.
  - Beispiel: **M** = $\{\neg A \wedge C, B \wedge \neg D\}$
- Was, wenn **M** Disjunktionen von Literalen enthält?
  - Beispiel: **M** = $\{\neg A \vee C, B \vee D, A \vee \neg B, \neg C \vee \neg D\}$

  ➔ Im schlimmsten Fall muss man verschiedene Belegungen ausprobieren.
  - Beispiel: **M** = $\{\neg A \vee C, B, A \vee \neg B, \neg C \vee \neg D\}$

  ➔ Im günstigen Fall kann man die Belegung direkt bestimmen:
  $\mathcal{A}(B) = \mathbf{1}$, $\mathcal{A}(A) = \mathbf{1}$, $\mathcal{A}(C) = \mathbf{1}$, $\mathcal{A}(D) = \mathbf{0}$.

Hornlogik erlaubt nur, ‚günstige Fälle' auszudrücken !

Hornklauseln & Hornformeln

Definition 7.1 (Hornklausel, Hornformel)

- Eine Klausel K ist genau dann eine *Hornklausel*, falls K **höchstens ein** positives Literal enthält.
- Eine Formel F in konjunktiver Normalform, bei der jede Klausel eine Hornklausel ist, ist eine *Hornformel*.

Beispiele

Hornklauseln		keine Hornklauseln
A	$\neg A$	
$\neg A \vee B$	$\neg A \vee \neg B$	$A \vee B$
$\neg A \vee B \vee \neg C$		$\neg A \vee B \vee C$
$A \vee \neg B \vee \neg C \vee \neg D$		$A \vee \neg B \vee C \vee \neg D$

Typen von Hornklauseln

Eine Hornklausel K enthält **höchstens ein** positives Literal.

➔ Es gibt **drei Typen** von Hornklauseln:

1. K enthält **negative** Literale **und ein positives** Literal: *Regeln*
   $K = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i \vee A_k$,
2. K enthält **nur negative** Literale: *Beschränkungen*
   $K = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i$
3. K enthält **nur ein positives** Literal: *Fakten*
   $K = A_1$

Zum Selbststudium: Grammatik für Hornformel (KNF-Spezialfall)

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{HFK} = \mathcal{As}_{AL} \cup \{\neg, \wedge, \vee,), (\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{HFK} = \{HFK, HKl, L, As\}$ **Startsymbol:** HFK Anmerkung: HFK: Hornformel KNF-Schreibweise HKl: Horn-Klausel L: Literal NL: negatives Literal As: Aussagesymbol (positives Literal)	**Regeln (Produktionen):** P = { HFK → (HFK ∧ HFK) , HFK → HKl , HKl → (NL ∨ HKl) , HKl → (HKl ∨ NL) , HKl → L , NL → ¬As , L → NL , L → As , As → A , As → B , As → C , As → D , … }

Eine „Normalform“ für Hornformeln

Definition 7.2

Eine Hornklausel, in der kein Aussagensymbol mehrfach auftritt, heißt *reduziert*.
Eine Hornformel H heißt genau dann *reduziert*, wenn

1. alle Klauseln in H reduziert sind, und wenn
2. alle Klauseln in H verschieden sind.

Beobachtung

- Wenn in einer Hornklausel K ein Paar komplementärer Literale auftritt, dann ist K eine Tautologie. (Dies kann nur bei Typ 1 der Fall sein).
- Wenn in einer Hornklausel K kein Paar komplementärer Literale auftritt, dann gibt es eine zu K äquivalente reduzierte Hornklausel.

➔ Anwendung der Idempotenz

Satz 7.3 (Existenz einer reduzierten Hornformel)

Für jede nicht allgemeingültige Hornformel existiert eine äquivalente reduzierte Hornformel.
Beweis: Zur Übung.

Exkurs: Syntaxerweiterung: $\top$ und $\bot$

Zur Erinnerung

- Sind F und G *Tautologien*, so gilt: $F \equiv G$.
- Sind F und G *unerfüllbar*, so gilt: $F \equiv G$.

➔ Als Vertreter für die Äquivalenzklasse

- der *Tautologien* führen wir die atomaren Formel $\top$ (*top*) ein.
- der *unerfüllbaren Formeln* führen wir die atomaren Formel $\bot$ (*bottom*) ein.
  1. $\top$ und $\bot$ sind *logische Konstanten* , also keine Aussagesymbole.
  2. Für alle Belegungen $\mathcal{A}$ wird festgesetzt: $\mathcal{A}(\top) = \mathbf{1}$, $\mathcal{A}(\bot) = \mathbf{0}$
  3. Dann gelten folgende Äquivalenzen:

$\neg\top \equiv \bot$	$\neg\bot \equiv \top$
$\top \wedge F \equiv F$ für alle F	$\top \vee F \equiv \top$ für alle F
$\bot \wedge F \equiv \bot$ für alle F	$\bot \vee F \equiv F$ für alle F

  4. Für Implikationen gilt:

$\top \Rightarrow F \equiv F$ für alle F	$F \Rightarrow \bot \equiv \neg F$ für alle F
$F \Rightarrow \top \equiv \top$ für alle F	$\bot \Rightarrow F \equiv \top$ für alle F

Zum Selbststudium: Syntaxerweiterung: $\top$ und $\bot$

Im folgenden arbeiten wir also mit einer erweiterten Logiksprache
Definition 7x

Es sei $\mathcal{As}_{AL}$ eine endliche oder abzählbar unendliche Menge von Aussagesymbolen, so dass $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \top, \bot,), (\} \cap \mathcal{As}_{AL} = \emptyset$.

Die *wohlgeformten Ausdrücke der Aussagenlogik (Formeln) mit* $\top, \bot$ sind induktiv definiert:

1. $\top, \bot$ und alle Aussagensymbole aus $\mathcal{As}_{AL}$ sind (atomare) Formeln.
2. Falls F und G Formeln sind, so sind $(F \wedge G)$, $(F \vee G)$, $(F \Rightarrow G)$ und $(F \Leftrightarrow G)$ (*komplexe*) Formeln.
3. Falls F eine Formel ist, so ist auch $\neg F$ eine (*komplexe*) Formel.
4. Es gibt keine anderen Formeln, als die, die durch endliche Anwendung der Schritte 1–3 erzeugt werden.

- Die Menge aller aussagenlogischen Formeln bezeichnen wir als $\mathcal{L}_{AL}{}^{\top\bot}$.

➔ Für $\mathcal{L}_{AL}{}^{\top\bot}$ müssen bei der strukturellen Induktion im Induktionsanfang auch die Fälle $\top$ und $\bot$ behandelt werden.

Hornklauseln – Implikationsschreibweise

Zu jede Hornklausel gibt es eine äquivalente Formel mit Hauptoperator $\Rightarrow$, so dass in der linken Teilformel kein Junktor außer der Konjunktion auftritt und rechts nur ein Aussagensymbol oder $\bot$.

- Fall 1: K enthält negative Literale und ein positives Literal: *Regeln*

  $K \quad = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i \vee A_k$

  dann kann K in der folgenden Weise umgeformt werden:

  $K \quad \equiv (\neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i) \vee A_k$

  $\equiv \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \vee A_k \quad \equiv (A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow A_k$

- Fall 2: K enthält nur negative Literale: *Beschränkungen*

  $K \quad = \neg A_1 \vee \neg A_2 \vee \ldots \vee \neg A_i$

  $\equiv \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \quad \equiv (A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow \bot$

  [wegen $\neg F \equiv F \Rightarrow \bot$ für alle F]

- Fall 3: K enthält nur ein positives Literal: *Fakten*

  $K \quad = A_1 \quad \equiv \top \Rightarrow A_1$

  [wegen $F \equiv \top \Rightarrow F$ für alle F]

Z. Selbststudium: Grammatik für Hornformel (Implikationsschreibweise)

Vokabular der terminalen Symbole (Symbole, die in der Formel vorkommen): $\Sigma_{HFI} = \mathcal{As}_{AL} \cup \{\wedge, \Rightarrow, \bot, \top,), (\}$ **nicht-terminales Symbol** (Symbole, die für die Erzeugung der Formeln benötigt werden, aber nicht in der Formel vorkommen): $N_{HFI} = \{HFI, HI, L, As\}$ **Startsymbol:** HF Anmerkung: HFI: Hornformel Implikationsschreibweise HI: Horn-Implikation KAt: Konjunktion von Atomen As: Aussagesymbol (positives Literal)	**Regeln (Produktionen):** P = { HFI → HI ∧ HFI , HFI → HI , HI → (KAt ⇒ As) , HI → (KAt ⇒ ⊥) , HI → (⊤ ⇒ As) , KAt → As ∧ KAt , KAt → As , As → A , As → B , As → C , As → D , … }

Exkurs: Hornklauseln – Logik-Programmierung

Die drei Typen von Hornklauseln entsprechen drei Grundtypen von Ausdrücken der Logik-Programmierung (➔ SE-3, Regelsprachen mit definiten Klauseln: Datalog ➔ GWV):

Fakten (z.B. einer Wissensbasis)

- Hans ist der Sohn von Peter: ISTSOHNVON(HANS, PETER).
- Peter ist der Sohn von August: ISTSOHNVON(PETER, AUGUST).
- Fall 3: nur ein positives Literal. A_k . entspricht $T \Rightarrow A_k$

Regeln (z.B. einer Wissensbasis)

- Wenn Hans Sohn von Peter und Peter Sohn von Ben ist, dann ist Hans Enkel von Ben: ISTENKELVON(HANS, BEN) :– ISTSOHNVON(HANS, PETER), ISTSOHNVON(PETER, BEN).
- Fall 1: negative Literale (rechts!) und ein positives Literal (links!)
  A_k :– A_1 , A_2 , … , A_i . entspricht $(A_1 \wedge A_2 \wedge \ldots \wedge A_i) \Rightarrow A_k$

Anfragen / Ziele (Modelliert über Beschränkungen)

- Ist Hans Enkel von Ben und Ben Sohn von Peter? :
  ? :– ISTENKELVON(HANS, BEN), ISTSOHNVON(BEN, PETER).
- Fall 2: nur negative Literale. ? :– A_1 , … , A_i . entspricht $(A_1 \wedge \ldots \wedge A_i) \Rightarrow \perp$

[".", ":–", "," und "?" sind Symbole der Logik-Programmierung]

Zum Selbststudium (insbesondere im Zusammenhang mit SE 3)

Logik-Programme, die auf Aussagenlogik basieren

- sind sehr ausdrucksschwach, da sie keine Variablen verwenden können
- bei der Behandlung der Prädikatenlogik wird die Beziehung zu PROLOG noch enger.

Dass Anfragen als Klauseln mit negativen Literalen zu verstehen sind,

- ergibt sich aus Satz 5.9 (→ $F \models G$ GDW. $F \wedge \neg G$ unerfüllbar ist.)
- Eigentlich will man ja wissen, ob aus den Fakten und Regeln die Anfrage folgt (wobei die Teile der Anfrage konjunktiv verknüpft werden).
- Dazu fügt man die Negation der Anfrage mit den Fakten und Regeln zusammen und prüft, ob sich daraus ein Widerspruch ergibt. Ist dies der Fall, dann folgt die Anfrage aus dem Rest.
- Werden mehrere Anfragen gleichzeitig beantwortet, dann ergibt sich aus einem gefundenen Widerspruch, dass (mindestens) eine der Anfragen aus den Fakten und Regeln folgt.

Zum Selbststudium: Logik-Programm und Hornformeln

- B, C und G sind Fakten.
- Wenn C, dann auch A und wenn G dann auch D.
- Ergibt sich daraus A, B und D? oder ergibt sich E?
  - Modelliert als Test: Wird die Beschränkung $(\neg A \vee \neg B \vee \neg D)$ bzw. $\neg E$ verletzt?

Fakten	B .	Ziele	? :– A , B , D .
	C .		? :– E .
	G .		
Regeln	A :– C .		
	D :– G .		

$$B \wedge C \wedge G \wedge (\neg C \vee A) \wedge (\neg G \vee D) \wedge (\neg A \vee \neg B \vee \neg D) \wedge \neg E$$

umgeformt als Konjunktion von Implikationen

$$\equiv \quad (T \Rightarrow B) \wedge (T \Rightarrow C) \wedge (T \Rightarrow G) \wedge (C \Rightarrow A) \wedge (G \Rightarrow D) \wedge$$
$$((A \wedge B \wedge D) \Rightarrow \bot) \wedge (E \Rightarrow \bot)$$

Gesucht

Ein einfaches Verfahren, dass für **jedes** Logik-Programm die (Un)erfüllbarkeit prüft.

Was ist durch Hornformeln ausdrückbar?

- Es gibt Formeln der Aussagenlogik, zu denen es keine äquivalente Hornformel gibt.
- Der entsprechende Wahrheitswertverlauf ist nicht in Hornform ausdrückbar.
- Hornformel-Logik ist gegenüber der Aussagenlogik eingeschränkt (in ihrer Ausdrucksstärke).

Beispiel: Es gibt zu $A \vee B$ keine äquivalente Hornformel.

Es gibt keine zu A ∨ B äquivalente Hornformel

- Wenn es eine zu A ∨ B äquivalente Hornformel H* gäbe, so würde es hierzu eine äquivalente reduzierte Hornformel H geben. [Satz 7.3: Existenz reduzierter Hornformel]
- Diese äquivalente reduzierte Hornformel H müsste – falls sie ausschließlich aus den Aussagesymbolen A und B aufgebaut ist – aus den *(Horn-)Klauseln*

  A B ¬A ¬B A ∨ ¬B ¬A ∨ B ¬A ∨ ¬B

  gebildet sein, d.h. eine Konjunktion aus einigen dieser Klauseln sein.

	A	B	A ∨ B	¬A	¬B	A ∨ ¬B	¬A ∨ B	¬A ∨ ¬B
$\mathcal{A}_1$	1	1	1	0	0	1	1	0
$\mathcal{A}_2$	1	0	1	0	1	1	0	1
$\mathcal{A}_3$	0	1	1	1	0	0	1	1
$\mathcal{A}_4$	0	0	0	1	1	1	1	1

- Aus der Tafel lassen sich zwei Begründungen dafür ablesen, dass es keine zu A ∨ B äquivalente Hornformel H gebildet aus den Aussagesymbolen A und B gibt.
  - Jede der Hornklauseln wird unter mindestens einer der Belegung $\mathcal{A}_1$, $\mathcal{A}_2$ oder $\mathcal{A}_3$ zu **0** ausgewertet. Da sich die Bewertung **0** in einer Konjunktion durchsetzt, kann keine der Klauseln zu H gehören, die ja – wie A ∨ B – zu **1** ausgewertet werden soll.
  - Nur A und B werden von $\mathcal{A}_4$ zu **0** ausgewertet. Wäre aber A oder B Konjunkt von H, dann würde für H auch bei $\mathcal{A}_2$ oder $\mathcal{A}_3$ eine **0** entstehen.

Zum Selbststudium: Es gibt keine zu A ∨ B äquivalente Hornformel

➔ Können andere Aussagesymbole dazu beitragen, dass eine zu A ∨ B äquivalente Hornformel gebildet wird?

Gesetzt den Fall, es kommt ein weiteres Aussagesymbol C ins Spiel.

- Es können 10 neue Hornklauseln über A, B und C gebildet werden (zusätzlich zu C und ¬C).

	A	B	C	A ∨ B	A ∨ ¬C	B ∨ ¬C	¬A ∨ C	¬A ∨ ¬C	¬B∨ C	¬B∨ ¬C	A ∨ ¬B∨ ¬C	¬A ∨ B∨ ¬C	¬A ∨ ¬B∨ C	¬A ∨ ¬B∨ ¬C
$\mathcal{A}_1$	1	1	0	1	1	1	0	1	0	1	1	1	0	1
$\mathcal{A}_2$	1	0	0	1	1	1	0	1	1	1	1	1	1	1
$\mathcal{A}_3$	0	1	0	1	1	1	1	1	0	1	1	1	1	1
$\mathcal{A}_4$	0	0	0	0	1	1	1	1	1	1	1	1	1	1
$\mathcal{A}'_1$	1	1	1	1	1	1	1	0	1	0	1	1	1	0
$\mathcal{A}'_2$	1	0	1	1	1	0	1	0	1	1	1	0	1	1
$\mathcal{A}'_3$	0	1	1	1	0	1	1	1	1	0	0	1	1	1
$\mathcal{A}'_4$	0	0	1	0	0	0	1	1	1	1	1	1	1	1

- Weitere Überlegungen wie zuvor.

Erfüllbarkeitstest für Hornformeln: Markierungsalgorithmus

Eingabe: Eine Hornformel F in Implikationsschreibweise

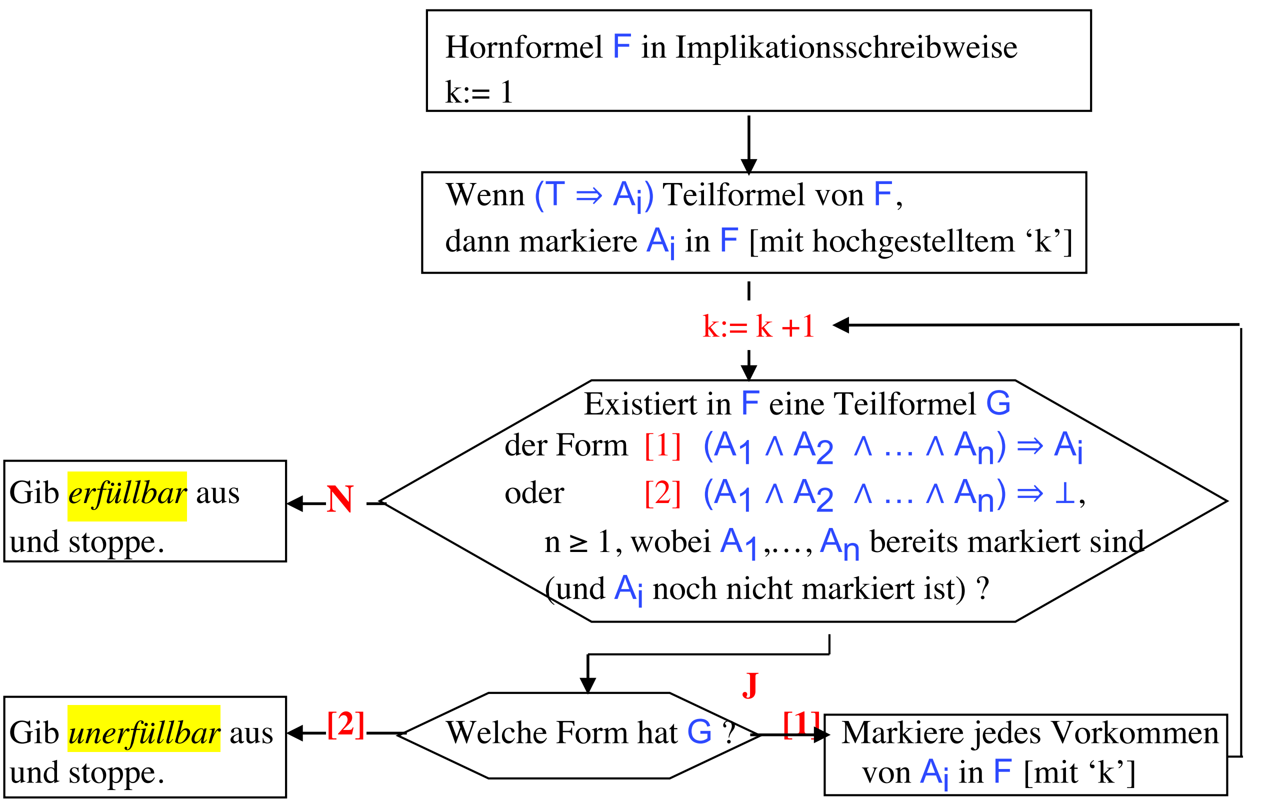
- **Grundidee**: Markiere Aussagesymbole, die in jeder Belegung, die F erfüllt, auf **1** abgebildet werden (müssen).
  1. Versehe jedes Vorkommen eine Aussagensymbols A_i in F mit einer Markierung, falls es in F eine Teilformel der Form $(T \Rightarrow A_i)$ gibt.
  2. WHILE es gibt in F eine Teilformel G der Form
     [1] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ oder der Form
     [2] $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \bot$,
     $n \geq 1$, wobei $A_1, \ldots, A_n$ bereits markiert sind (und A_i noch nicht markiert ist)

     DO IF G hat die Form [1]
     THEN markiere jedes Vorkommen von A_i in F
     ELSE gib *unerfüllbar* aus und stoppe.
  3. Gib *erfüllbar* aus und stoppe.

     Die erfüllende Belegung wird durch die Markierung gegeben:

$$\mathcal{A}(A_i) = \begin{cases} \mathbf{1}, \text{ falls } A_i \text{ eine Markierung besitzt} \\ \mathbf{0}, \text{ sonst} \end{cases}$$

Markierungsalgorithmus: Struktur des Ablaufs



Erfüllbarkeitstest für Hornformeln: Beispiel-1

$(\neg A \vee \neg B \vee E) \wedge C \wedge A \wedge (\neg C \vee B) \wedge (\neg A \vee D) \wedge (\neg C \vee \neg D \vee A) \wedge \neg E \wedge \neg G$

Als Konjunktion von Implikationen:

$((A \wedge B) \Rightarrow E) \wedge (T \Rightarrow C) \wedge (T \Rightarrow A) \wedge (C \Rightarrow B) \wedge$
$(A \Rightarrow D) \wedge ((C \wedge D) \Rightarrow A) \wedge (E \Rightarrow \perp) \wedge (G \Rightarrow \perp)$

Markierung 1. Runde: $((^1A \wedge B) \Rightarrow E) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1A) \wedge (^1C \Rightarrow B) \wedge$
$(^1A \Rightarrow D) \wedge ((^1C \wedge D) \Rightarrow {}^1A) \wedge (E \Rightarrow \perp) \wedge (G \Rightarrow \perp)$

Markierung 2. Runde: $((^1A \wedge {}^2B) \Rightarrow E) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1A) \wedge (^1C \Rightarrow {}^2B) \wedge$
$(^1A \Rightarrow {}^2D) \wedge ((^1C \wedge {}^2D) \Rightarrow {}^1A) \wedge (E \Rightarrow \perp) \wedge (G \Rightarrow \perp)$

Markierung 3. Runde: $((^1A \wedge {}^2B) \Rightarrow {}^3E) \wedge (T \Rightarrow {}^1C) \wedge (T \Rightarrow {}^1A) \wedge (^1C \Rightarrow {}^2B) \wedge$
$(^1A \Rightarrow {}^2D) \wedge ((^1C \wedge {}^2D) \Rightarrow {}^1A) \wedge (^3E \Rightarrow \perp) \wedge (G \Rightarrow \perp)$

➔ unerfüllbar

Erfüllbarkeitstest für Hornformeln: Beispiel-2

$F := (A \vee \neg B) \;\wedge\; (\neg C \vee \neg A \vee D) \;\wedge\; (\neg A \vee \neg B) \;\wedge\; D \;\wedge\; \neg E$

als Konjunktion von Implikationen:

$(B \Rightarrow A) \;\wedge\; ((C \wedge A) \Rightarrow D) \;\wedge\; ((A \wedge B) \Rightarrow \perp) \;\wedge\; (T \Rightarrow D) \;\wedge\; (E \Rightarrow \perp)$

Markierung:

$(B \Rightarrow A) \wedge ((C \wedge A) \Rightarrow {}^1D) \wedge ((A \wedge B) \Rightarrow \perp) \wedge (T \Rightarrow {}^1D) \wedge (E \Rightarrow \perp)$

Einteilung der Klauseln nach Form [1] und Form [2]:

[1] $B \Rightarrow A$ [2] $(A \wedge B) \Rightarrow \perp$

$(C \wedge A) \Rightarrow {}^1D$ $E \Rightarrow \perp$

➔ erfüllbar

Erfüllende Belegung:

	A	B	C	D	E	F
$\mathcal{A}$	0	0	0	1	0	1

Zum Selbststudium

Welches Ergebnis liefert der Markierungsalgorithmus

- Wenn es keine Klauseln der Form $(T \Rightarrow A_i)$ gibt?
- Wenn es keine Klauseln der Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$ gibt?

Begründen Sie, warum dieses Ergebnis korrekt ist.

- Greifen Sie dazu auch die Erläuterungen zu 7.4 und 7.6 zu.

Was passiert,

- Wenn es keine Klauseln der Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A$ gibt?

Korrektheit & Terminierung des Markierungsalgorithmus

Satz 7.4: Der Markierungsalgorithmus für Hornformeln ist korrekt.
Er stoppt nach spätestens n Markierungsschritten.

(n = Anzahl der Aussagensymbole in F)

Anmerkungen

1. Der Markierungsalgorithmus setzt voraus, dass die Eingabe eine Hornformel ist.
2. Korrektheit: Der Markierungsalgorithmus gibt bei erfüllbaren Hornformeln *erfüllbar* und bei unerfüllbaren Hornformeln *unerfüllbar* aus.
3. Der Erfüllbarkeitstest durch Markierung benötigt maximal n Markierungsschritte, während eine Berechnung der Wahrheitswerttafel gegebenenfalls 2^n Belegungen berücksichtigen muss.

➔ Markierungsalgorithmus korrespondiert zur Fragebeantwortung in der Logik-Programmierung (➔ SE-3).

Terminierung des Markierungsalgorithmus

Voraussetzungen: F ist eine Formel, in der genau n Aussagensymbole vorkommen

Beweis

Im ersten Schritt (1.) werden einige Aussagensymbole markiert, dann ist der Schritt abgeschlossen.
In jedem weiteren Markierungsschritt (2. / WHILE-Schleife) wird mindestens ein weiteres Aussagensymbol markiert.
Ist keine weitere Markierung möglich, so wird die WHILE-Schleife verlassen.
➔ Maximal n Markierungsschritte.
Dann stoppt das Verfahren mit der Ausgabe „*erfüllbar*“ oder „*unerfüllbar*“.

➔ Markierungsalgorithmus hat *linearen Aufwand*.

Korrektheit der Markierung: Beweis (1)

Lemma 7.5: Wenn $\mathcal{A}$ ein Modell für die Hornformel F ist,
so gilt für die durch den Algorithmus markierten Symbole A_i: $\mathcal{A}(A_i) = \mathbf{1}$.

Voraussetzungen: Def. 3.1, 3.3, 4.18, 7.1

Beweis

Es sei $\mathcal{A}$ ein Modell für die Hornformel F.
Dann macht $\mathcal{A}$ auch alle Konjunkte von F wahr, also die Hornklauseln, aus denen F aufgebaut ist.

Behauptung

$\mathcal{A}$ macht die im Markierungsschritt k markierten Aussagensymbole wahr.
Vollständige Induktion über die Nr. des Markierungsschrittes, in dem A_i markiert wurde.

Induktionsanfang

In Markierungsschritt Nr. 1 (das ist Schritt 1 des Algorithmus) werden nur Aussagensymbole A_i markiert, für die eine Klausel $(T \Rightarrow A_i)$ in F existiert.
$\mathcal{A}$ macht $(T \Rightarrow A_i)$ wahr und $(T \Rightarrow A_i)$ ist zudem mit A_i äquivalent ist. Also macht $\mathcal{A}$ A_i wahr.

Korrektheit der Markierung: Beweis (2)

Induktionsannahme
Für alle $j < k$ gelte: $\mathcal{A}$ macht die im Markierungsschritt Nr. j markierten Aussagensymbole wahr.

Induktionsschritt
Der k-te Markierungsschritt erfolgt innerhalb der WHILE-Schleife. Dabei werden nur solche Aussagensymbole A_i markiert, für die eine Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ existiert, und $A_1, A_2, \ldots, A_n$ wurden schon (in einem früheren Schritt) markiert.
Da $\mathcal{A}$ $A_1, A_2, \ldots, A_n$ (Induktionsannahme) und die Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow A_i$ wahr macht (s.o.), macht $\mathcal{A}$ auch A_i wahr.

Resümee
In jedem Markierungsschritt gilt, dass die markierten Aussagensymbole durch $\mathcal{A}$ wahr gemacht werden.

Korrektheit der Klassifikation „*unerfüllbar*“: Beweis

Lemma 7.6a: Wenn „*unerfüllbar*“ ausgegeben wird, so ist die Eingangsformel F unerfüllbar.

Beweis
Die Klassifikation „*unerfüllbar*“ wird ausgegeben (Schritt 2 „ELSE“), wenn die Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$ in F enthalten ist, wobei $n \geq 1$ und $A_1, \ldots, A_n$ bereits markiert sind.

Beweis durch Widerspruch
Annahme: In F ist die Klausel $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$ enthalten, der Markierungsalgorithmus markiert $A_1, \ldots, A_n$ und trotzdem gibt es ein Modell $\mathcal{A}$ von F.

Da $\mathcal{A}(F) = \mathbf{1}$, ist für $1 \leq i \leq n$ auch: $\mathcal{A}(A_i) = \mathbf{1}$, da die A_i markiert werden. (s. Lemma 7.5)
Dann ist auch: $\mathcal{A}(A_1 \wedge A_2 \wedge \ldots \wedge A_n) = \mathbf{1}$ (Def 3.1)
Und damit $\mathcal{A}((A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp) = \mathbf{0}$ also auch $\mathcal{A}(F) = \mathbf{0}$. (Def 3.1)

Also Da $\mathcal{A}(F) = \mathbf{1}$ und $\mathcal{A}(F) = \mathbf{0}$ nicht beides gelten kann, ist die Annahme widersprüchlich und es kann keine erfüllende Belegung für F geben, wenn der Markierungsalgorithmus „*unerfüllbar*“ ausgibt.

Korrektheit der Klassifikation „*erfüllbar*“: Beweis

Lemma 7.6b: Wenn in Schritt 3 „*erfüllbar*“ ausgegeben wird, so liefert die Markierung ein Modell $\mathcal{A}$ für F, gemäß $\mathcal{A}(A_i) = \begin{cases} \mathbf{1}, \text{ falls } A_i \text{ eine Markierung besitzt} \\ \mathbf{0}, \text{ sonst} \end{cases}$.

Voraussetzung: Def. 3.1, Lemma 7.5

Beweis

Die Klassifikation „*erfüllbar*“ wird ausgegeben (Schritt 3), wenn der Schritt 2 erfolgreich durchlaufen wurde. Wir müssen zeigen, dass in diesem Fall $\mathcal{A}$ alle Klauseln wahr macht.

Sei K eine beliebige Klausel aus F.

- Falls K eine atomare Formel ist, so wird K in Schritt 1 markiert. Es gilt: $\mathcal{A}(K) = \mathbf{1}$.
- Falls K die Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow B$ hat, so können 2 Fälle auftreten:
- Alle A_i und B sind markiert (wg. Schritt 2), dann gilt: $\mathcal{A}(K) = \mathbf{1}$.
- Mindestens ein A_i ($1 \leq i \leq n$) ist nicht markiert, also: $\mathcal{A}(A_i) = \mathbf{0}$.
  Dann gilt aber auch: $\mathcal{A}(K) = \mathbf{1}$.
- Falls K die Form $(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow \perp$ hat, so ist mindestens ein A_i ($1 \leq i \leq n$) nicht markiert, denn anderenfalls wäre *unerfüllbar* ausgegeben worden.
  Also: $\mathcal{A}(A_i) = \mathbf{0}$ und deswegen $\mathcal{A}(K) = \mathbf{1}$.

Zum Selbststudium: Hornformeln – Logik-Programmierung:

„Logik-Programm“

Fakten		Regeln		Ziele	
	B .		A :– C .		? :– A , B , D .
	C .		D :– G .		? :– E .
	G .				

als Konjunktion von Implikationen: $(T \Rightarrow B) \wedge (T \Rightarrow C) \wedge (T \Rightarrow G) \wedge$
$(C \Rightarrow A) \wedge (G \Rightarrow D) \wedge ((A \wedge B \wedge D) \Rightarrow \perp) \wedge (E \Rightarrow \perp)$

Erfüllbarkeitstest für Hornformeln – Logik-Programmierung

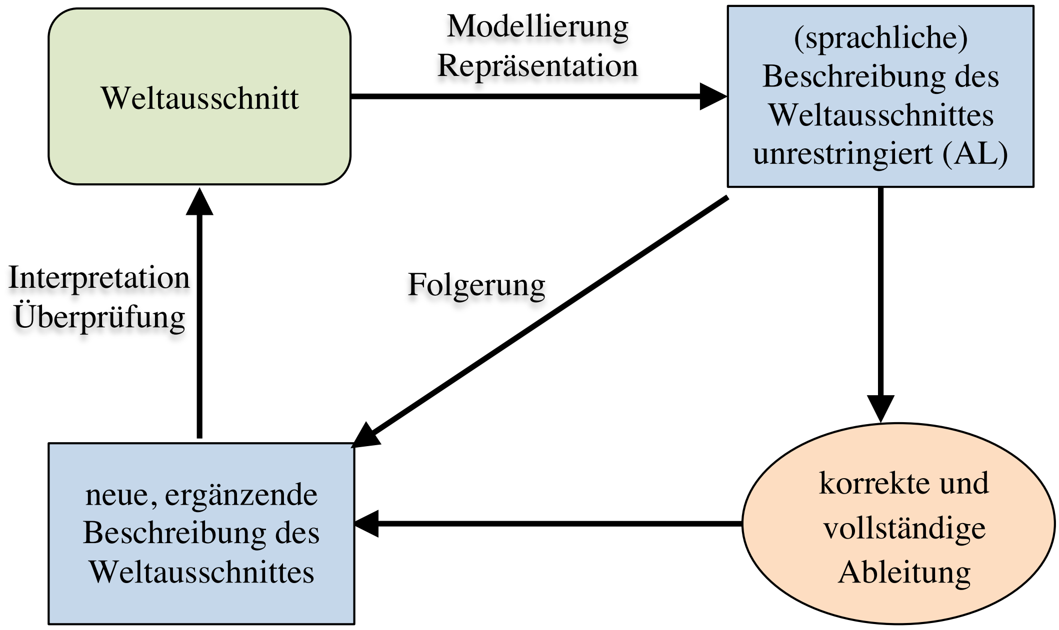
Markierung 1. Runde: $(T \Rightarrow {}^{1}B) \wedge (T \Rightarrow {}^{1}C) \wedge (T \Rightarrow {}^{1}G) \wedge$
$({}^{1}C \Rightarrow A) \wedge ({}^{1}G \Rightarrow D) \wedge ((A \wedge {}^{1}B \wedge D) \Rightarrow \perp) \wedge (E \Rightarrow \perp)$

Markierung 2. Runde: $(T \Rightarrow {}^{1}B) \wedge (T \Rightarrow {}^{1}C) \wedge (T \Rightarrow {}^{1}G) \wedge$
$({}^{1}C \Rightarrow {}^{2}A) \wedge ({}^{1}G \Rightarrow {}^{2}D) \wedge (({}^{2}A \wedge {}^{1}B \wedge {}^{2}D) \Rightarrow \perp) \wedge (E \Rightarrow \perp)$

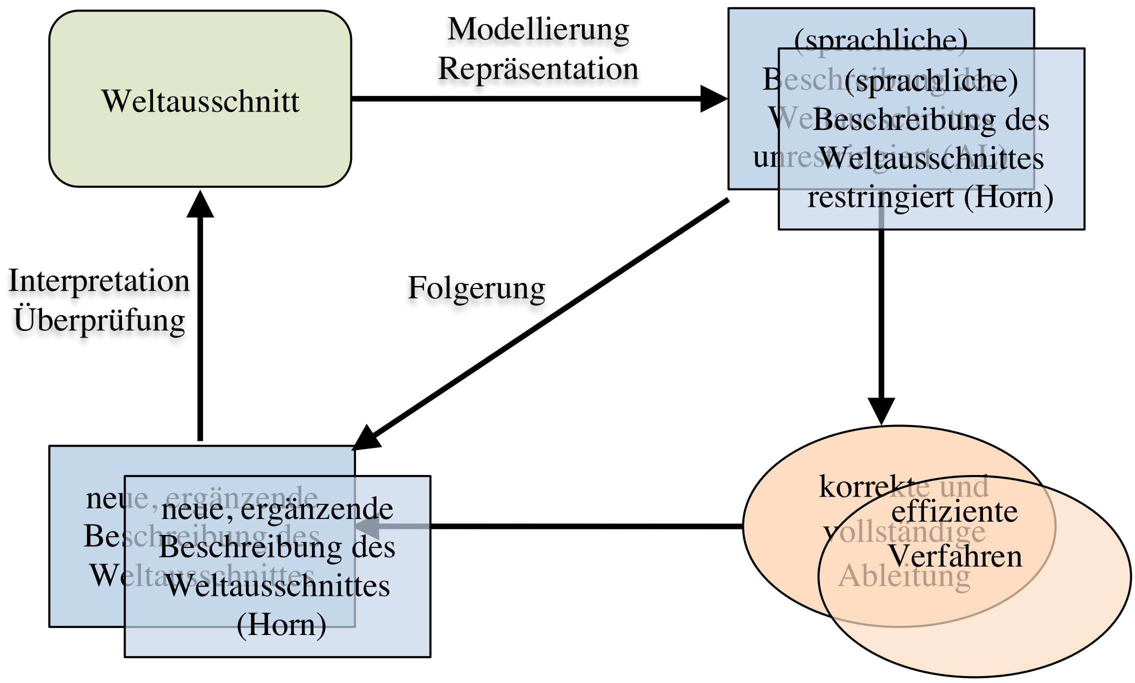
➔ unerfüllbar

➔ $(A \wedge B \wedge D)$ folgt aus den Fakten und Regeln.

Modellierung, Folgerung und Ableitung in Logik



Modellierung, Folgerung und Ableitung in Hornlogik



Wichtige Konzepte dieser Vorlesung

- Hornklausel, Hornformel, KNF-Darstellung, Implikationsschreibweise
- Logische Konstanten ($\top$: top, $\perp$: bottom)
- Beschränkte Ausdrucksfähigkeit von Hornformeln (es gibt Wahrheitswertverläufe, zu denen es keine Hornformeln gibt)
- Markierungsalgorithmus: Grundidee, Funktionsweise, Termination, Korrektheit