Computational Logic
Algorithmische Logik

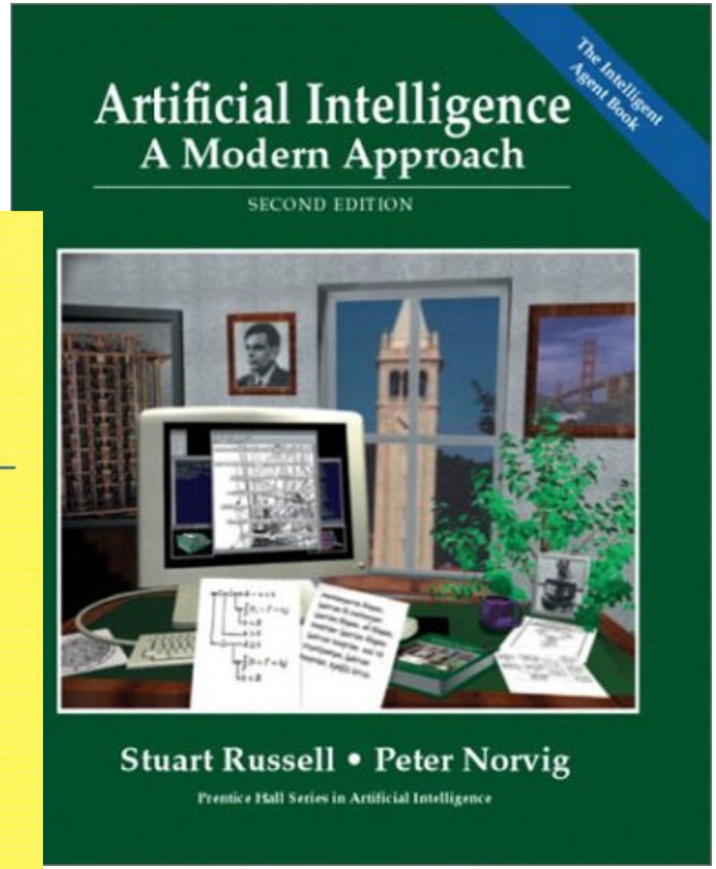
Introduction

Ralf Moeller
Hamburg Univ. of Technology

Literature

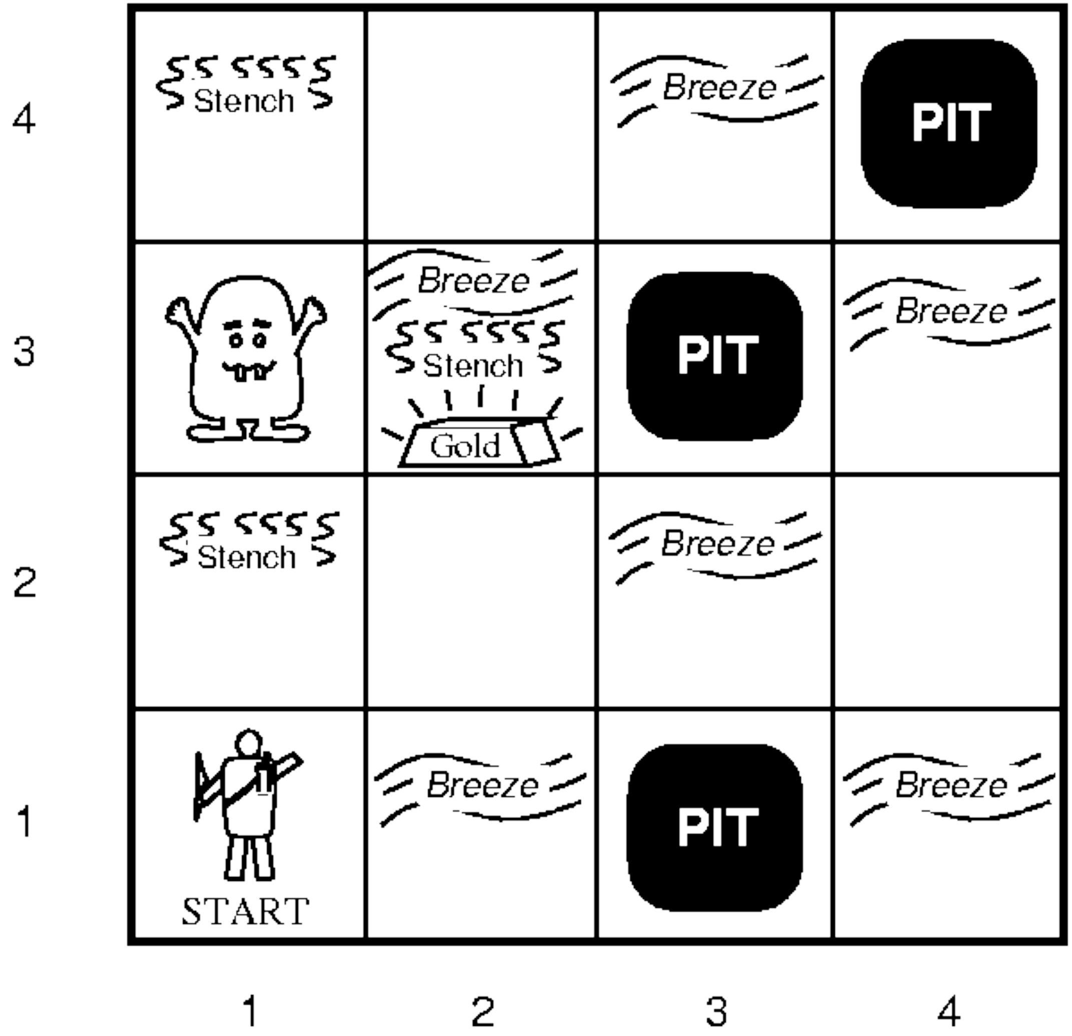
COMPANION WEBSITE
CWS
informatik
Martin Kreuzer
Stefan Kühling
Logik für Informatiker
PEARSON
Studium

HochschulTaschenbuch
HTb
Uwe Schöning
Logik für Informatiker
5. Auflage
Spektrum



Artificial Intelligence
A Modern Approach
SECOND EDITION
The Intelligent Agent Book
Stuart Russell • Peter Norvig
Prentice Hall Series in Artificial Intelligence

Die Wumpus-Welt



START: Anfang

Wumpus: Monster (Tod)

PIT: Fallgrube (Tod)

Stench: Gestank (neben Wumpus)

Breeze: Luftzug (neben Fallgrube)

Glitter: Gold

Stoß:
Agent läuft gegen Wand

Agent hat 1 Pfeil, der horizontal oder vertikal den Wumpus tötet, Schrei überall zu hören

Die Wumpus–Welt (2)

Agent kann seinen Standort nicht wahrnehmen

Wahrnehmungen:

[Stench, Breeze, Glitter, None, None]

Gestank, Luftzug, Glitzer, kein Stoß, kein Schrei

Anfangszustand:

Position [1,1], nach rechts gewandt,
irgendwo der Wumpus, das Gold, drei Fallgruben

Ziel:

Hole Gold und verlasse danach die Höhle

Die Wumpus–Welt (3)

Aktionen:

vorwärts gehen

nach rechts drehen

nach links drehen

Gold greifen

Pfeil Schießen

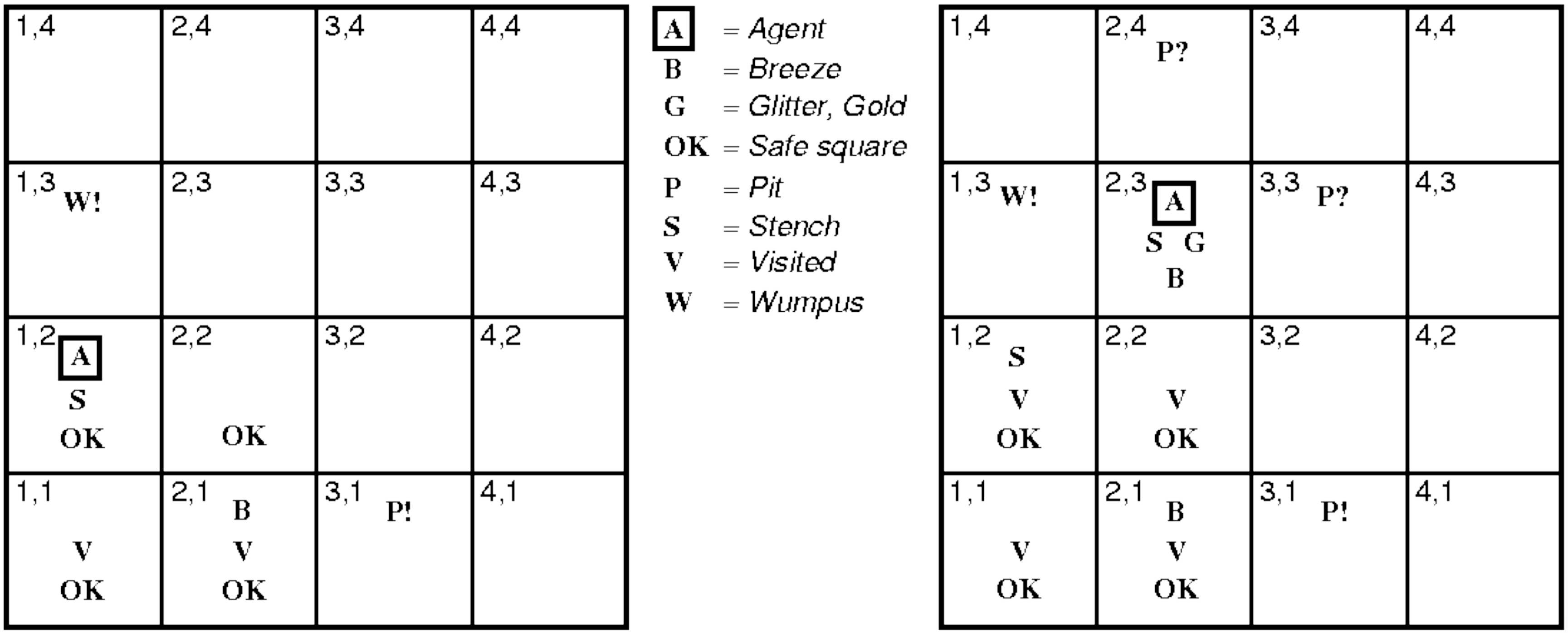
Höhle verlassen (falls Standort [1,1])

Exploration einer Beispielwelt

[1,2] und [2,1] sind sicher:

(a) (b)

Exploration einer Beispiel–Welt



(a) (b)

Logik in der Informatik

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $0 \ldots 1$
Fuzzy logic	degree of truth	degree of belief $0 \ldots 1$

Wumpus: Formalisierung

Grundbausteine:

1) ***atomare Aussagen***

- D : *„Der Wumpus ist tot“*
- $W_{1,3}$: *„Der Wumpus ist auf [$1,3$]“*

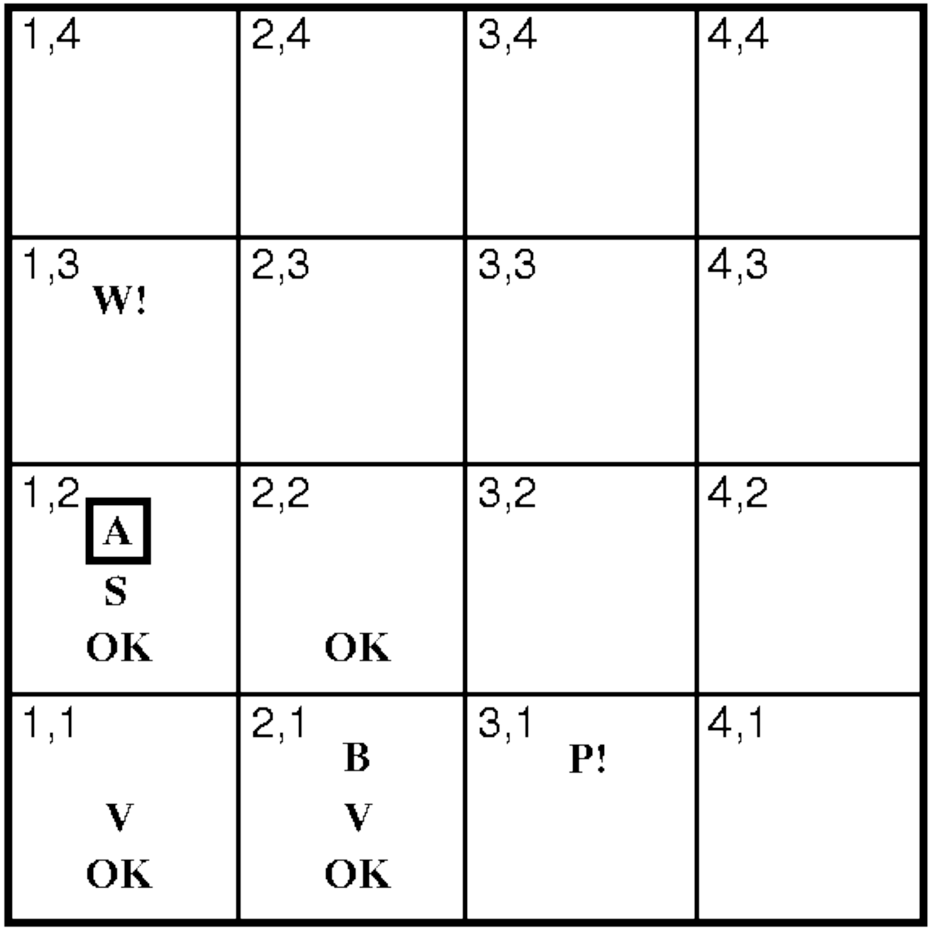
*die **wahr** oder **falsch** sein können*

2) ***logische Konnektoren***

(und, oder, nicht etc.)

*zur Konstruktion von **Formeln***

Formalisierung des Beispiels



$\boxed{\mathbf{A}}$ = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

aktuelle Situation:

$\neg S_{1,1}$	$\neg B_{1,1}$
$\neg S_{2,1}$	$B_{2,1}$
$S_{1,2}$	$\neg B_{1,2}$

Formalisierung des Beispiels (2)

Beschreibung der Domäne (Spielregeln):

R_1: $\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$
R_2: $\neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$
R_3: $\neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$
R_4: $S_{1,2} \Rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$
...

Gilt $WB \models W_{1,3}$?

Die Wissensbasis WB als Klauselmenge:

Der aktuelle Zustand: $\{\neg S_{1,1}\}, \{\neg S_{2,1}\}, \{\neg S_{1,2}\}, \ldots$

Die Domäne:

R_1: $\{S_{1,1}, \neg W_{1,1}\}, \{S_{1,1}, \neg W_{1,2}\}, \{S_{1,1}, \neg W_{2,1}\}$
R_2: $\ldots, \{S_{2,1}, \neg W_{2,2}\}, \ldots$
R_3: $\ldots$
R_4: $\{\neg S_{1,2}, W_{1,3}, W_{1,2}, W_{2,2}, W_{1,1}\}$

Logisches Schließen

Wie können wir ableiten,
welche Aktionen der Agent ausführen soll?

Negative Selektion:
Schließe alle beweisbar gefährlichen Aktionen aus

$$A_{1,1} \wedge East_A \wedge W_{2,1} \Rightarrow \neg Forward$$

Positive Selektion:
Schlage nur Aktionen vor, die beweisbar sicher sind.

$$A_{1,1} \wedge East_A \wedge \neg W_{2,1} \Rightarrow Forward$$

Zur Beschreibung der Instruktion
„Gehe nicht vorwärts, wenn der Wumpus
auf dem nächsten Feld sitzt."
benötigt man 64 Klauseln (16 Felder x 4 mögliche Richtungen).

Wumpus: Aussagenlogik

Die Wumpus-Welt kann in Aussagenlogik modelliert werden.

Für ***jedes einzelne*** *Feld müssen entsprechende Regeln aufgestellt werden:*

R_1: $\neg S_{1,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

R_2: $\neg S_{2,1} \Rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$

R_3: $\neg S_{1,2} \Rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$

Alle atomaren Aussagen müssen mit einem ***Zeitindex*** *versehen werden.*
Bei einem Horizont von 100 Zeitschritten werden 6400 Regeln zur Aktionsbeschreibung benötigt.

Wumpus: Prädikatenlogik

*Prädikatenlogik 1. Stufe (FOL) ermöglicht die **Strukturierung** atomarer Aussagen.*

*Es gibt **Objekte**, **Funktionen** und **Relationen**.*

***Beispiel**:*

„Auf allen Feldern, die dem Aufenthaltsort des Wumpus unmittelbar benachbart sind, nimmt man einen unangenehmen Geruch wahr."

$$\forall f_1, f_2 [[Auf(f_1, wumpus) \wedge Benachbart(f_1, f_2)] \Rightarrow Geruch(f_2)]$$

(Aussagenlogik: 16 Formeln der Form

$$W_{i,j} \Rightarrow [S_{i-1,j} \vee S_{i+1,j} \vee S_{i,j-1} \vee S_{i,j+1}])$$

Prädikatenlogik erlaubt kompaktere Darstellung.

Wumpus: Formalisierung (2)

Suppose a wumpus-world agent is using an FOL KB
and perceives a smell and a breeze (but no glitter) at $t = 5$:

$\text{TELL}(KB, Percept([Smell, Breeze, None], 5))$
$\text{ASK}(KB, \exists a \;\; Action(a, 5))$

I.e., does the KB entail any particular actions at $t = 5$?

Answer: Yes, $\{a/Shoot\}$ $\leftarrow$ substitution (binding list)

Given a sentence S and a substitution σ,
$S\sigma$ denotes the result of plugging σ into S; e.g.,
$S = Smarter(x, y)$
$\sigma = \{x/Hillary, y/Bill\}$
$S\sigma = Smarter(Hillary, Bill)$

$\text{ASK}(\text{KB}, \text{S})$ returns some/all σ such that $KB \models S\sigma$

Wumpus: Formalisierung (3)

"Perception"

$\forall b, g, t \;\; Percept([Smell, b, g], t) \;\Rightarrow\; Smelt(t)$

$\forall s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

Reflex: $\forall t \;\; AtGold(t) \;\Rightarrow\; Action(Grab, t)$

Reflex with internal state: do we have the gold already?

$\forall t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$

$Holding(Gold, t)$ cannot be observed

$\Rightarrow$ keeping track of change is essential

Folgerungen

Properties of locations:

$\forall l, t \;\; At(Agent, l, t) \land Smelt(t) \;\Rightarrow\; Smelly(l)$

$\forall l, t \;\; At(Agent, l, t) \land Breeze(t) \;\Rightarrow\; Breezy(l)$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall y \;\; Breezy(y) \;\Rightarrow\; \exists x \;\; Pit(x) \land Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall x, y \;\; Pit(x) \land Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

Definition for the *Breezy* predicate:

$$\forall y \;\; Breezy(y) \;\Leftrightarrow\; [\exists x \;\; Pit(x) \land Adjacent(x, y)]$$

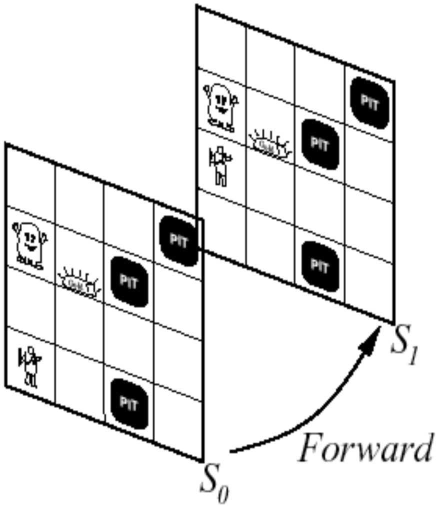
Umgang mit Veränderungen

Facts hold in situations, rather than eternally
E.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$

Situation calculus is one way to represent change in FOL:

Adds a situation argument to each non-eternal predicate
E.g., Now in $Holding(Gold, Now)$ denotes a situation

Situations are connected by the $Result$ function
$Result(a, s)$ is the situation that results from doing a is s



Beschreibung von Aktionen

"Effect" axiom—describe changes due to action

$\forall s \;\; AtGold(s) \;\Rightarrow\; Holding(Gold, Result(Grab, s))$

"Frame" axiom—describe non-changes due to action

$\forall s \;\; HaveArrow(s) \;\Rightarrow\; HaveArrow(Result(Grab, s))$

Frame problem: find an elegant way to handle non-change

(a) representation—avoid frame axioms

(b) inference—avoid repeated "copy-overs" to keep track of state

Qualification problem: true descriptions of real actions require endless caveats—what if gold is slippery or nailed down or ...

Ramification problem: real actions have many secondary consequences—what about the dust on the gold, wear and tear on gloves, ...

Beschreibung von Aktionen (2)

Successor-state axioms solve the representational frame problem

Each axiom is "about" a predicate (not an action per se):

P true afterwards $\Leftrightarrow$ [an action made P true

$\vee$ P true already and no action made P false]

For holding the gold:

$$\forall a, s \;\; Holding(Gold, Result(a, s)) \;\Leftrightarrow$$
$$[(a = Grab \wedge AtGold(s))$$
$$\vee\, (Holding(Gold, s) \wedge a \neq Release)]$$

Pläne machen ...

Initial condition in KB:

$At(Agent, [1, 1], S_0)$

$At(Gold, [1, 2], S_0)$

Query: $\text{ASK}(KB, \exists s \;\; Holding(Gold, s))$

i.e., in what situation will I be holding the gold?

Answer: $\{s/Result(Grab, Result(Forward, S_0))\}$

i.e., go forward and then grab the gold

This assumes that the agent is interested in plans starting at S_0 and that S_0 is the only situation described in the KB

Überblick

- Aussagenlogik
- Prädikatenlogik
- Beschreibungslogik / Modallogik
- Zeitlogiken

Syntax der Aussagenlogik

Eine *atomare Formel* hat die Form A_i (wobei $i = 1, 2, 3, \ldots$).

Formeln werden durch folgende induktive Definition festgelegt:

1. Alle atomaren Formeln sind Formeln
2. Für alle Formeln F und G sind $(F \wedge G)$ und $(F \vee G)$ Formeln.
3. Für jede Formel F ist $\neg F$ eine Formel.

Abkürzungen:

A, B, C oder P, Q, R oder $\ldots$	statt	$A_1, A_2, A_3 \ldots$
$(F_1 \rightarrow F_2)$	statt	$(\neg F_1 \vee F_2)$
$(F_1 \leftrightarrow F_2)$	statt	$((F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2))$
$(\bigvee_{i=1}^{n} F_i)$	statt	$(\ldots((F_1 \vee F_2) \vee F_3) \vee \ldots \vee F_n)$
$(\bigwedge_{i=1}^{n} F_i)$	statt	$(\ldots((F_1 \wedge F_2) \wedge F_3) \wedge \ldots \wedge F_n)$

Begriff der induktiven Definition

- Zunächst einfachste Einheiten (Atome) festlegen
  - Beispiel: atomare Formeln der Aussagenlogik
- Dann erklären wie aus einfachen Einheiten komplexere Einheiten konstruiert werden können
  - Beispiel: Bildung allgemeiner Formeln wie $F \wedge G$, $F \vee G$, $\neg F$

Noch ein paar Abkürzungen ...

- Sei A eine beliebige atomare Formel (Variable)
- $\top$ stehe für $A \vee \neg A$
- $\perp$ stehe für $A \wedge \neg A$

Semantik der Aussagenlogik

Die Elemente der Menge $\{0,1\}$ heißen *Wahrheitswerte*.
Eine *Belegung* ist eine Funktion $\mathcal{A}\colon D \to \{0,1\}$, wobei D eine Teilmenge der atomaren Formeln ist. Wir erweitern $\mathcal{A}$ zu einer Funktion $\overline{\mathcal{A}}\colon E \to \{0,1\}$, wobei $E \supseteq D$ die Menge aller Formeln ist, die nur aus den atomaren Formeln in D aufgebaut sind.

$$\overline{\mathcal{A}}((F \wedge G)) = \begin{cases} 1 & \text{falls } \overline{\mathcal{A}}(F) = 1 \text{ und } \overline{\mathcal{A}}(G) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\overline{\mathcal{A}}((F \vee G)) = \begin{cases} 1 & \text{falls } \overline{\mathcal{A}}(F) = 1 \text{ oder } \overline{\mathcal{A}}(G) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\overline{\mathcal{A}}((\neg F)) = \begin{cases} 1 & \text{falls } \overline{\mathcal{A}}(F) = 0 \\ 0 & \text{sonst} \end{cases}$$

Wir schreiben $\mathcal{A}$ statt $\overline{\mathcal{A}}$.

Modelle

Sei F eine Formel und $\mathcal{A}$ eine Belegung.

Falls $\mathcal{A}$ für alle in F vorkommenden atomaren Formeln definiert ist, so heißt $\mathcal{A}$ zu F *passend*.

Sei $\mathcal{A}$ passend zu F:

Falls $\mathcal{A}(F) = 1$	so schreiben wir	$\mathcal{A} \models F$
	und sagen	F *gilt unter* $\mathcal{A}$
	oder	$\mathcal{A}$ *ist ein Modell für* F
Falls $\mathcal{A}(F) = 0$	so schreiben wir	$\mathcal{A} \not\models F$
	etc.	

Gültigkeit und Erfüllbarkeit

Eine Formel F heißt *erfüllbar*, falls F mindestens ein Modell besitzt, andernfalls heißt F *unerfüllbar*.

Eine (endliche oder unendliche!) Menge von Formeln M heißt *erfüllbar*, falls es eine Belegung gibt, die für jede Formel in M ein Modell ist.

Eine Formel F heißt *gültig* (oder *allgemeingültig* oder *Tautologie*) falls jede zu F passende Belegung ein Modell für F ist. Wir schreiben $\models F$, falls F eine Tautologie ist, und $\not\models F$ sonst.

Aufgabe

	Gültig	Erfüllbar	Unerfüllbar
A			
$A \vee B$			
$A \vee \neg A$			
$A \wedge \neg A$			
$A \rightarrow \neg A$			
$\neg A \rightarrow A$			
$A \rightarrow B$			
$A \rightarrow (B \rightarrow A)$			
$A \rightarrow (A \rightarrow B)$			
$A \leftrightarrow \neg A$			
$A \leftrightarrow (B \leftrightarrow A)$			

Aufgabe

Gelten die folgenden Aussagen?

				J/N	Gegenb.
Wenn	F gültig	dann	F erfüllbar		
Wenn	F erfüllbar	dann	$\neg F$ unerfüllbar		
Wenn	F gültig	dann	$\neg F$ unerfüllbar		
Wenn	F unerfüllbar	dann	$\neg F$ gültig		

Aufgabe

Gelten die folgenden Aussagen?

							J/N	Gegenb.
Wenn	$(F \rightarrow G)$ gültig	und	F gültig	dann	G gültig			
Wenn	$(F \rightarrow G)$ erfüllbar	und	F erfüllbar	dann	G erfüllbar			
Wenn	$(F \rightarrow G)$ gültig	und	F erfüllbar	dann	G erfüllbar			

Folgerung und Äquivalenz

Eine Formel G heißt eine *Folgerung* der Formeln $F_1, \ldots, F_k$ falls für jede Belegung, die sowohl zu $F_1, \ldots, F_k$ als auch zu G passend ist, gilt:

> Wenn $\mathcal{A}$ Modell von $\{F_1, \ldots, F_k\}$ ist, dann ist $\mathcal{A}$ auch Modell von G.

Wir schreiben $F_1, \ldots, F_k \models G$, falls G eine Folgerung von $F_1, \ldots, F_k$ ist.

Zwei Formeln F und G heißen (*semantisch*) *äquivalent*, falls für alle Belegungen $\mathcal{A}$, die sowohl für F als auch für G passend sind, gilt $\mathcal{A}(F) = \mathcal{A}(G)$. Hierfür schreiben wir $F \equiv G$.