Prädikatenlogik 1. Ordnung SYNTAX

Signatur Σ (Func, Pred)

= Abzählbares ∞ Set

Func + arity
- 0: Konstanten
- n: Terme
- ↳ liefern Werte aus Domain zurück

Pred + arity
- Für Atom
- ↳ liefern T/F zurück

Begriffe

Klausel
= Literal ∨ Literal
- ⊤ Leere {Klausel}, Leere ∧
- ⊥ Leere Klausel, Leere ∨, Leere {Literals}

Atom
= Wahrheitswert (PLO Prädikat (PL1))

Literal
= Atom, ¬Atom

ground
= ohne Var.

Σ-Terme (Σ, Var)

- B1: ∀ Elem ∈ Var = Σ-Term
- B2: ∀ cs ∈ Func ∈ Σ = Σ-Term
- S1: ∀ FS ∈ Func ∈ Σ mit Σ-Termen als Param = Σ-Term

FOFS (Σ, Var)

PL1

- B1: ∀ atoms = Σ-Formulas
- B2: ⊤ & ⊥ = Σ-Formulas
- S1: φ_1 & φ_2 = Σ-Formulas
  → φ_1 o φ_2 o = {→, ↔, ∧, ∨, ⊕, ¬} = Σ-Formulas
- S2: x ∈ Var & φ = Σ-Formulas
  → ∀x φ & ∃x φ = Σ-Formulas

PLO

BV = set of boolean vars
- B1: ∀ p ∈ BV = formula
- B2: ⊤ & ⊥ = formula
- S1: φ = formula → ¬φ = formula
- S2: φ_1 & φ_2 = formulas
  → φ_1 o φ_2 o = {¬, ∧, ∨, ↔, →}

Formel

Geschlossen
Keine freie Vars

Cleansed
∀x, x ∈ Var: x = frei ≠
x = gebunden
∀ Quantor bindet 1 Var

Variablen

frei
→ Nicht an Quantor gebunden, ~~Konstante~~
⇒ ~~Fkt~~ - Param

gebunden
gebunden durch Quantor (∀, ∃)

Σ-Struktur $\mathcal{I}$

= Interpretation $\langle \mathcal{U}, I, \alpha \rangle$

Nicht leere Menge von Sym.
= Domain

= Interp.fkt bildet auf D.
ab
Erfüllt:
- $c \in CS : I(c) \in \mathcal{U}$
- $f \in FS : I(f) : \mathcal{U} \to \mathcal{U}^n$
- $p \in PS : I(p) \subseteq \mathcal{U}^n$

Variablenbelegung

Evaluierung

Term (Σ, Var)

t in $\mathcal{I}\langle \mathcal{U}, I, \alpha \rangle$
B1: $x \in Var : \mathcal{I}(x) = \alpha(x)$
B2: $c \in CS : \mathcal{I}(c) = I(c)$
S1: $f \ln \in Func$ & $t_1, \dots, t_n$ = Σ-Term
$\mathcal{I}(f(t_1, \dots, t_n)) = (I f)(\mathcal{I}(t_1), \dots, \mathcal{I}(t_n))$

FOFS(Σ-Formel)

Closed for. in $\mathcal{I}\langle \mathcal{U}, I, \alpha \rangle$
B1: $p \ln \in Pred$ & $t_1, \dots, t_n$:
$\mathcal{I}(p(t_1, \dots, t_n)) = 1 \iff (\mathcal{I}(p(t_1)), \dots, \mathcal{I}(p(t_n)))$
S1: Für $\circ = \{\neg, \wedge, \vee, \to, \leftrightarrow\}$
gilt dasselbe wie in PLO
S2: $\mathcal{I}(\forall x \varphi) = \min_{c \in \mathcal{U}} = K$
S3: $\mathcal{I}(\exists x \varphi) = \max_{c \in \mathcal{U}} = K$
$K = \{J(y) \mid J = \langle \mathcal{U}, I, \alpha' \cup \{x \leftarrow c\} \rangle\}$

Prädikatenlogik 1. Ordnung SEMANTIK

Gültigkeit, ek. Mod(φ)

$\mathcal{I}(\varphi)$	
1	$\mathcal{I}$ = model of φ → satisfiable OR $\neg\varphi$ not valid
0	$\mathcal{I}$ falsible → φ falsible
valid unsat.	$\forall\, \mathcal{I}(\varphi) = 1$ or $\neg\varphi$ unsat $\models \varphi \equiv \top$ $\forall\, \mathcal{I}(\varphi) = 0$ $\models \varphi \equiv \bot$

Equivalent:
$\varphi \equiv \psi$ IFF (⟺) $Mod(\varphi) \equiv Mod(\psi)$
⟹ $\forall\, \mathcal{I} \in \Sigma : \mathcal{I} \models \varphi \iff \mathcal{I} \models \psi$

All four statements in each line amount the same

entailment(s)	validity	satisfiability	equivalence
$\varphi \models \psi$	$\varphi \to \psi$ valid	$\varphi \wedge \neg\psi$ unsat	$(\varphi \to \psi) \equiv \top$
$\top \models \psi$	ψ valid	$\neg\psi$ unsat	$\psi \equiv \top$
$\top \not\models \neg\psi$	$\neg\psi$ not valid	ψ sat	$\neg\psi \not\equiv \top$
$\varphi \models \psi$ and $\psi \models \varphi$	$\varphi \leftrightarrow \psi$ valid	$\varphi \leftrightarrow \neg\psi$ unsat	$\varphi \equiv \psi$

Entailment

Let W be a set of closed formulas
→ then W entails φ ($W \models \varphi$),
IFF $Mod(W) \subseteq Mod(\varphi)$
⟺

ERL
Let I be an interpretation
& $I \models \psi_1 \leftrightarrow \psi_2$
Then $I \models \phi[\psi_1] \leftrightarrow \phi[\psi_2]$

ERT
Let $\psi_1 \equiv \psi_2$.
Then $\phi[\psi_1] \equiv \phi[\psi_2]$

Reduction to satisfiability

Reduce validity, entailment, equivalence to satisfiability

1. Validity
   - $\neg\varphi$ is unsatisfiable iff φ is valid
2. Entailment
   - φ entails ψ ($\varphi \models \psi$) iff $\varphi \to \psi$ is valid (apply Deduction Thm)
   - Hence, $\varphi \models \psi$ iff $\varphi \wedge \neg\psi$ (i.e., $\neg(\varphi \to \psi)$) is unsatisfiable
3. Equivalence
   - φ is equivalent to ψ ($\varphi \equiv \psi$) iff $\varphi \leftrightarrow \psi$ is valid
   - Hence, $\varphi \equiv \psi$ iff $\varphi \models \psi$ and $\psi \models \varphi$ hold
   - Consequently, $\varphi \equiv \psi$ iff $\varphi \wedge \neg\psi$ and $\psi \wedge \neg\varphi$ are unsatisfiable

Sound and complete procedure for satisfiability is sufficient!

Entailment (or logical implication)

- So far, $\models$ relates a Σ-structure and a formula.
- We want to allow a set of formulas on the left side.
- Important: a (finite) set of formulas coincides with the conjunction of its elements, i.e., $\{\varphi_1, \dots, \varphi_n\}$ is $\bigwedge_{i=1}^{n} \varphi_i$.
- Important: an empty conjunction is 1 in all Σ-structures i.e., it is equivalent to $\top$.

Definition

Let W be a set of closed formulas. Then W entails φ,
$W \models \varphi$, if and only if $Mod(W) \subseteq Mod(\varphi)$

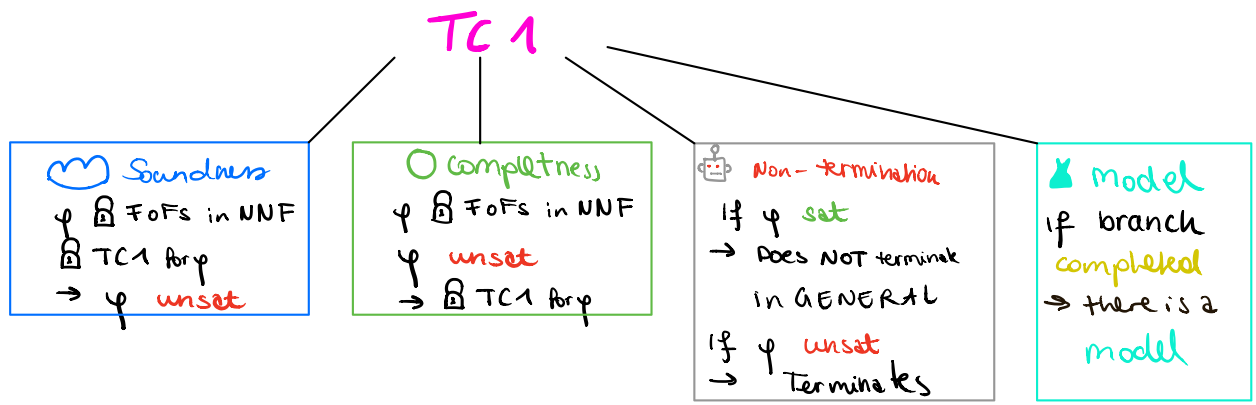
Entailment is a very important concept when we consider KBSs!

Basics of the NNF translation for PL1

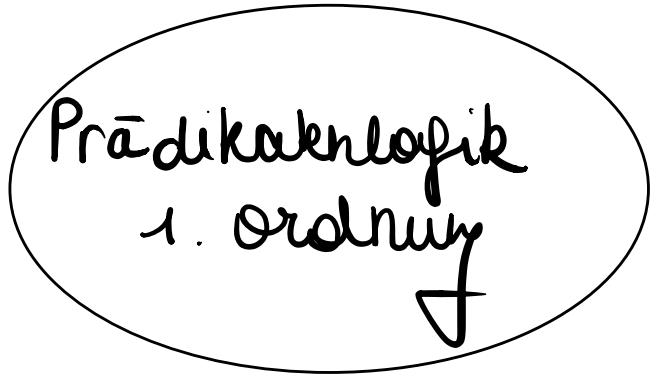
Perform the following 4 steps:

1. Replace all $\leftrightarrow$ by $\rightarrow$ using $(\varphi \leftrightarrow \psi) \equiv ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$
2. Replace all $\oplus$ using $(\varphi \oplus \psi) \equiv ((\varphi \vee \psi) \wedge (\neg\varphi \vee \neg\psi))$ (why?)
3. Replace all $\rightarrow$ using $(\varphi \rightarrow \psi) \equiv (\neg\varphi \vee \psi)$
4. Replace the left side of the following equivalences by the right side (order does not matter!)

$$\begin{array}{rcl} \neg\forall x\,\varphi & \equiv & \exists x\,\neg\varphi \\ \neg(\varphi \vee \psi) & \equiv & \neg\varphi \wedge \neg\psi \\ \varphi \vee \top & \equiv & \top \\ \varphi \wedge \bot & \equiv & \bot \\ \varphi \wedge \top & \equiv & \varphi \\ \varphi \vee \bot & \equiv & \varphi \\ \neg\neg\varphi & \equiv & \varphi \end{array} \qquad \begin{array}{rcl} \neg\exists x\,\varphi & \equiv & \forall x\,\neg\varphi \\ \neg(\varphi \wedge \psi) & \equiv & \neg\varphi \vee \neg\psi \\ \top \vee \varphi & \equiv & \top \\ \bot \wedge \varphi & \equiv & \bot \\ \top \wedge \varphi & \equiv & \varphi \\ \bot \vee \varphi & \equiv & \varphi \end{array}$$



- Quantifier rules are conceptually simple, but sufficient for our purpose later
- For first-order theorem proving, advanced quantifier rules are widely used which use Skolem functions in general
- Additionally, usually free variable tableaux are used which use unification in order to determine the term t
- The use of sophisticated quantifier rules and unification result in better/faster implementation because some problems wrt permutability of inferences are avoided



$$\frac{\forall x\,\varphi}{\varphi\{x \leftarrow t\}}$$

If a model satisfies a universally quantified formula, it also satisfies the formula (without the quantifier), where the former quantified (and now free) variable is substituted with some (ground) term. The prescription is to use terms which occur in the tableau.

$$\frac{\exists x\,\varphi}{\varphi\{x \leftarrow a\}}$$

If a model satisfies an existentially quantified formula $\exists x\,\varphi$, then it also satisfies the formula $\varphi\{x \leftarrow a\}$, where the former quantified (and now free) variable is substituted with a fresh new Skolem constant.

$$\frac{\varphi \wedge \psi}{\begin{array}{c}\varphi \\ \psi\end{array}}$$

This rule can be applied if φ and ψ are not both is on the current branch

$$\frac{\varphi \vee \psi}{\varphi \mid \psi}$$

This rule can be applied if neither φ nor ψ on the current branch

$$\frac{\exists x\,\varphi}{\varphi\{x \leftarrow a\}}$$

This rule can be applied if $\varphi\{x \leftarrow b\}$ (for a Skolem constant b) is not on the current branch

- Applicability conditions prevent redundant rule applications
- For the $\forall$-rule, no restriction can be given in general!

Adjectives and nouns

- Formalize *"Programming is fun."*
- Possibility 1: $programming(fun)$
- Possibility 2: $fun(programming)$
- Which one should be preferred in many cases? It's the second possibility!

Translate adjectives to predicates and nouns to argument terms!

Pronouns

- How do we deal with pronouns, i.e., (s)he, it, nothing, etc.?
- Formalize *"Sue adores Pat who adores her."*
  - Restating it yields *"Sue adores Pat and Pat adores Sue."*
  - The formula is: $adore(sue, pat) \wedge adore(pat, sue)$
- Formalize *"Nothing is striped."*
  - Restating it yields *"There does not exist anything which is striped."*
  - The formula is: $\neg\exists x\, striped(x)$

Pronouns and connectives

- Formalize *"Something is spotted and it is hungry."*
- Can we translate the two substatements independently?
- No, *it* refers to *something*, i.e., the variable introduced for *something* must be "global"
- Therefore, the structure is $\exists x(x \text{ is spotted and } x \text{ is hungry})$
- The formula is: $\exists x(spotted(x) \wedge hungry(x))$

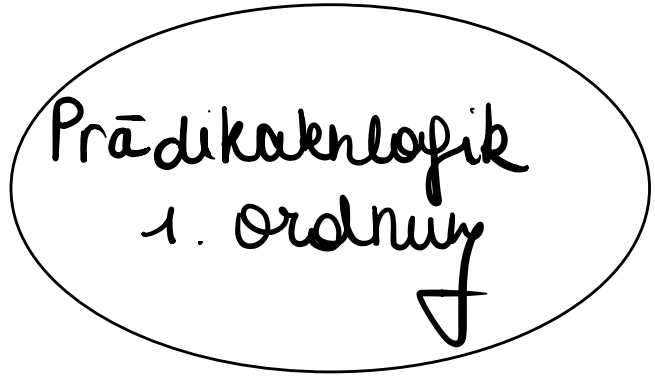
If pronouns are linked across a connective, deal with the pronouns before the connective.

- Example: Give a PL1 formula for the following

  "Every child is younger than its mother."
- We use the following predicates:
  - x *is a child*: $child(x)$
  - y *is the mother of* x: $mother(y, x)$
- We choose the formula:

$$\forall x \forall y((child(x) \wedge mother(y, x)) \rightarrow younger(x, y))$$

- Is this a "good" formal spec of what we want to express?
- No, our formalization does not take into account that every kid has exactly one genetic mother



- The formalization becomes more concise with a FS
- Take the term $m(x)$ to denote the *mother of* x
- Then the above statement can be formalized as:

$$\forall x(child(x) \rightarrow younger(x, m(x)))$$

- Clarify always the "type" of a symbol you use
- m has to be (and is) a FS, not a PS
- Recall the syntax of PL1! Only terms are allowed at argument positions of terms and predicates

Use equality

- Formalize the statement:

  "Andy and Paul have the same maternal grandmother."
- Use the equality predicate $=/2$ (written infix as usual)
- A possible formula expressing this statement is

$$\forall x \forall y \forall u \forall v(mother(x, y) \wedge mother(y, andy) \wedge mother(u, v) \wedge mother(v, paul) \rightarrow x = u)$$

- Looks complicated! Can we formalize it simpler?
- Using equality and functions often makes life easier:

$$m(m(andy)) = m(m(paul))$$

is much simpler than the formula above

Interlude: The equality relation

- The equality relation $=$ is a binary predicate of the form $=(s, t)$ (usually written infix $s = t$)
- The relation satisfies the following properties:
  - reflexivity: $\forall x\ (x = x)$
  - symmetry: $\forall x \forall y\ (x = y \rightarrow y = x)$
  - transitivity: $\forall x \forall y \forall z\ ((x = y\ \wedge\ y = z) \rightarrow x = z)$
- Two possibilities to deal with equality in KBS:
  1. Equip the inference procedure with rules for equality
  2. Compile properties into the KB (i.e., into the formula)

Qualifiers and exceptions

- Formalize the statement:

  "All instructors except John like PCs"

- First establish the overall structure, ...

  All (instructors except John) $likes(x, pc)$

- ... then formulate the sort information correctly

  $\forall x\,((inst(x) \wedge x \neq john) \rightarrow likes(x, pc))$

- Clearly, $x \neq john$ stands for $\neg(x = john)$

- Formalize *"All rational people abhor violence."*
- Reformulate with qualifier: $\forall$ rational x (x abhor violence)
- *Rational* is a qualifier (type, sort) for x
- Then the formula is $\forall x(rational(x) \rightarrow abhor(x, violence))$

Qualification is translated into an implication $\rightarrow$, if the corresponding quantifier is $\forall$.

$\exists$-quantifiers and qualifiers again

- Formalize *"Sue likes somebody who likes logic."*
- Reformulate: $\exists$ (person who likes logic) x (Sue likes x)
- Formula is $\exists x\,(person(x) \wedge likes(x, logic) \wedge likes(sue, x))$
- Pronoun *who* links the substatements conjunctively

Get the structure of the sentence correct before dealing with quantifiers.

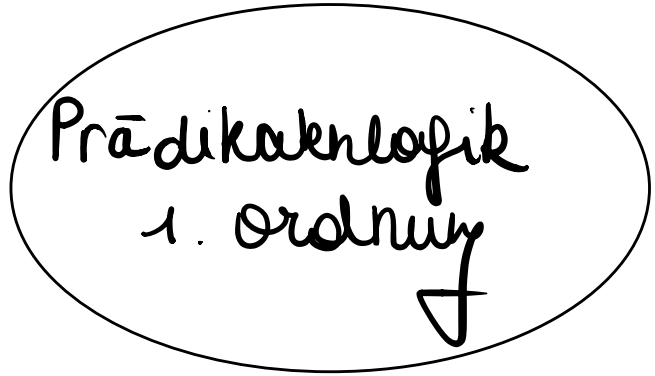
Qualification is translated into a conjunction $\wedge$, if the corresponding quantifier is $\exists$.

Pitfalls of quantifier alternations

- Formalization of *"Everybody has a (genetic) mother."* is

  $\forall x \exists y\, mother(y, x)$

- You first "fix" the x and then asks for x's mother
- The mother therefore depends on the chosen x!
- With $\exists y \forall x\, mother(y, x)$ ...
  - you first "fix" the mother y and then
  - state that everybody has the fixed y as the mother
- The result is: *"Somebody is mother of everybody."*



Implications and quantifiers

- $\forall$ formalize words like *all*, *every*, *any*, *everything*, etc.
- $\exists$ formalize words like *some*, *someone*, *something*, etc.
- Formalize

  "If someone is tall, then the door frame will be knocked."

- Formula is $(\exists x\, tall(x)) \rightarrow knocked(doorframe)$
- This is equivalent to $\forall x\,(tall(x) \rightarrow knocked(doorframe))$
- Formalize *"If someone is tall, then he will bump his head."*
- Here, we have to deal with the quantifier first because the pronoun *he* is linked to *someone*
- Hence, the formula is $\forall x\,(tall(x) \rightarrow bumphead(x))$

Classical Logic vs Commensense Logic

Classical Logic	Commensense Logic
monoton $\forall S, S' \& A$: $(S \models A) \wedge (S \subseteq S')$ $\rightarrow S' \models A$	~~monoton~~
Brauche komplette Infos	weniger strikte Reasoning patterns ("typically") → muss conclusions später vllt ändern, wenn mehr info dazu kommt
Implizites wissen → Explizit wissen	

flexible

Definitions

~~freie~~ / vars

- Theorie: Set von 🔒 Formeln
- complete Theory
  T complete IFF $\forall$ ground atom P: $P \in T$ / $P \notin T$
- Deduktiver Schluss:
  $Cn(T) = \{\varphi \mid T \models \varphi \ \& \ \varphi$ 🔒$\}$
  - > Inflationaryness: $T \subseteq Cn(T)$
  - > Idempotenz: $Cn(T) = Cn(Cn(T))$
  - > Monotonicity: $T \subseteq T' \rightarrow Cn(T) \subseteq Cn(T')$
  - $\forall$ valid formulas: $Cn(\emptyset)$ $(\neq \emptyset !)$
  - $Cn(T)$ = set of $\forall$ formulas IFF T inkonsistent
  - $\varphi \in Cn(T) \rightarrow Cn(T \cup \{\varphi\}) = Cn(T)$

- Consistency theory T
  $CWA(T)$ = ~~consistent~~ IFF
  ground atoms $P_1, \dots, P_n$
  s.t. $T \models P_1 \vee \dots \vee P_n$
  BUT: $T \not\models P_i$ $i \in \{1, \dots, n\}$
- Definite Horn Klauseln (= null / 1 Pl)
  T = set 🔒 DHK → $CWA(T)$ = consistent
- Domain closure axiom (DCA)
  Kann verwenden, um set von Konstanten zu restriken
  $DCA: \forall x (x = c_1 \vee \dots \vee x = c_n)$

Non-monotonic Reasoning

Formalismen

2 Methoden, um mit nicht kompletter/unklarer Information umzugehen

- Quantitativ: Probability Theory
- Qualitativ: Non-monotonic reasoning

→ Computational approach: ↑SP

CWA

= Methode zur effizienten "representation" von negativen Fakten
→ Nur Positive Information wird gespeichert.
↳ jeder Fakt, d. nicht abgeleitet werden kann → falsch)

CWA = complete
⇒ completes T bzgl adding ⊖facts

$CWA(T) = \{\varphi \mid T \cup T_{asm} \models \varphi, \varphi$ 🔒$\}$
$= Cn(T \cup T_{asm})$

$T_{asm} = \{\neg P \mid P$ ground atom, $T \not\models P\}$

→ CWA is logical closure of $\forall$ assumptions

- CWA relative to a predicate symbol (PS) $q \Longrightarrow$ application of the CWA only to predicates containing the PS q.

  Formally:

  $$T^q_{asm} = \{\neg P \mid P \text{ ground atom with PS } q,\ T \not\models P\};$$
  $$\mathrm{CWA}^q(T) = \{\varphi \mid T \cup T^q_{asm} \models \varphi,\ \varphi \text{ closed}\} = Cn(T \cup T^q_{asm}).$$

- CWA relative to a set $\{q_1, \dots, q_n\}$ of predicate symbols
  $\Longrightarrow \mathrm{CWA}^{\{q_1,\dots,q_n\}}(T)$ similarly defined!

Default logic circumscription modal nm logics

Regeln (Cau & Default)

= Inferenz regel

$\delta = \frac{\varphi : \psi_1, \ldots, \psi_n}{\chi}$

φ ... Prerequisite

$\psi_1, \ldots, \psi_n$ ... justifications

χ ... Konsequenzen

Normal

$\frac{\varphi : \psi}{\psi}$

→ immer E

wenn φ bekannt & $\forall \psi \in \{\psi_1, \ldots, \psi_i\}$ konsistent angenommen angenommen werden können, dann kann χ gefolgert werden

Basic

= Methode, um mit (inf.) Ausnahmen umzugehen

- Repräsentieren Fakten, typischerweise halten (Abgeleitet durch Default Regeln)
- D.R. erweitern klassische Inferenzregeln durch Konsistenz Konditionen
- Wissensbasis = D.T.

in Kl. L. ♡ Inf.

Extension

= $T = (W, \Delta)$

- Properties:
  - > E = Theory (FOFS)
  - > $E \subseteq W$
  - > $Cn(E) = E$
    - → wollen mehr wissen ableiten
  - > Geschlossen unter Anw. Defaults

$\delta = \frac{\varphi : \psi_1, \ldots, \psi_n}{\chi} \in \Delta$ anwendbar auf E

→ $\chi \in E$, wo: δ anwendbar zu E IFF $\varphi \in E$ & $\neg\psi_1 \notin E, \ldots, \neg\psi_n \notin E$

Default Theorie

$T = (W, \Delta)$

- Set FOFs: Premissen → sicher, aber kann unvollständig sein
- = Defaults: Plausible Schlüsse

wende Δ auf W an → neues Wissen

Default Logik

Kl. L.

K. K. ✱

DT

DT → Grounding

Γ_T ~~monoton~~

$T = (W, \Delta)$, S = FOFS

Γ_T = collection von $\forall$ sets F FOFS, d. folg. Properties erfüllen

1. $Cn(F) = F$
2. $W \subseteq F$
3. F abgeschlossen im relativen Kontext S

$\Rightarrow \Gamma_T(S) = \bigcap_{F \in H(S)} F$ Durchschnitt $\forall$ mengen, die 1-3 erfüllen

Fixed Point { E von T IFF $\Gamma_T(E) = E$

$\forall \delta = \frac{\varphi : \psi_1, \ldots, \psi_n}{\chi} \in \Delta$:

IF $\varphi \in F$ & $\neg\psi_i \notin S$ THEN $\chi \in F$

Kontext E = E von T IFF reproduces itself

Computing extensions

Determining the operator Γ_T for $T = (W, \Delta)$:

1. Classical reduct Δ_E:
   $\Delta_E := \{\varphi/\chi \mid (\varphi : \psi_1, \ldots, \psi_n/\chi) \in \Delta \text{ and } \{\neg\psi_1, \ldots, \neg\psi_n\} \cap E = \emptyset\}$
   - φ/χ is the *residue* of $(\varphi : \psi_1, \ldots, \psi_n/\chi)$.
   - φ/χ represents a *classical inference rule*:
     - if φ is derivable, then χ is asserted.
2. $Cn^{\Delta_E}(W) := Cn(W \cup \bigcup_{i \geq 0} E_i)$, with
   $E_0 := \{\chi \mid \varphi/\chi \in \Delta_E \text{ and } W \models \varphi\}$;
   $E_i := \{\chi \mid \varphi/\chi \in \Delta_E \text{ and } W \cup E_{i-1} \models \varphi\}$.
   ☞ $Cn^{\Delta_E}(W)$ is the deductive closure of W under classical logic together with the inference rules in Δ_E.
3. Then, the following can be shown: $\Gamma_T(E) = Cn^{\Delta_E}(W)$.

⟹ In other words, $Cn^{\Delta_E}(W)$ constitutes a *proof-theoretical characterisation* of $\Gamma_T(E)$.

Definition

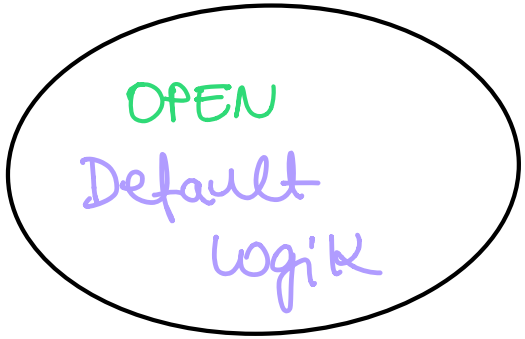
= Offenheit hängt nur von d. Formeln d. Defaults ab

→ z.B.: Closure finden

- Alle offenen Defaults werden mit Termen gebaut as Termen in T

Bei ∃ Quantor: ⇒ Skolemisierung

Imp. → Expl.



Answer Set Programming

deklarativ

Negationen

- **not** = neg. as failure
  - $a :- b_1, \dots, b_k, not\ b_{k+1}, \dots, not\ b_n$
  - If $b_1, \dots, b_k$ derivable & $b_{k+1}, \dots, b_n$ NOT derivable → a
  - Default-R.: $\frac{b_1 \wedge \dots \wedge b_k : \neg b_{k+1}, \dots, \neg b_n}{a}$
- $\neg$ = klassische Neg

Definition

Basis Idee: Horn Klauseln als Regeln

$a \vee \neg b_1 \vee \dots \vee \neg b_n$

$\Rightarrow a :- b_1, \dots, b_n$

wenn • herleitbar → folgere a

ASP

= Deklarative Methode zum & von Suchproblemen

Suchprob. in Ab. von D.T. darstellen

→ Wie?

Prob. wird in Ab. von T. von formalem Sys. + Semantik dargestellt, s.t.:

⇒ AS / Model = Lsg gefunden durch Modelle d. T.

Requirement: { Codieren & Rek. in polyn. Zeit

Answer Set

= min. Modell von klassischem Resultat

Sei r = Rule & $\delta(r)$ = Default Regel

P (Logic programm)

⇒ Dann:

Semantik von P durch AS gegeben → korrespondieren mit Extension d. default theory $(\emptyset, \delta(P))$

$\delta(P) = \{\delta(r) \mid r \in P\}$

Kann nicht inkonsistent sein

AS liefern Lsg zu codierten Problemen = alle Literale, die von P abgeleitet werden können

Default Logic

Für Lsg von Suchprob.

→ Repräsentieren Problem P in Abhängigkeit von Theorie T s.t. Lsg gegeben durch Extension von T

ASP vs Traditionelle wissensrep.

ASP

Prob. wird in Ab. von T. von formalem Sys. + Semantik dargestellt

⇒ Models = AS

MODELS

Traditionelle wissensrep.

Info in Wissensbasis → Ab fragen durch wg. Konsequenz Relation

PROOFS (Formal) Kalkül (Prolog)

Answer Set Programming

SYNTAX & SEMANTIK

Regeln

= geordnetes $\mathcal{T}$:

$a_1 \vee \dots \vee a_m :- b_1, \dots, b_k, not\ b_{k+1}, \dots, not\ b_n$

Head — Body

wenn $b_1, \dots, b_k$, $b_{k+1}, \dots, b_n$ ableitbar & Nicht

$\Rightarrow a_1, \dots, a_m$

- Fakt
- Basic
- Nondisj

& No $\neg$ = Normal

Horn

Ground (No vars)

Safe

safe var (not)
= Var nicht schwach negiert im Body
→ Entweder $\neg$ o. gar nicht negiert

Programm

= End. Set R

↳ Hier: Disj. Log. P

Definitionen

- Logik Pr.: Set ϕ (= nicht-leer)
  - Cs — Var — Ps — $\neq$S
- Term
- Atom $(p(t_1, \dots, t_n))^{n \geq 0}$ von ϕ

Literal (a, $\neg a$)

Klassisches Modell

Semantik einer Regel

$a_1 \vee \dots \vee a_m :- b_1, \dots, b_k, not\ b_{k+1}, \dots, not\ b_n$

IFF

$b_1, \dots, b_k \subseteq M$ & $\{b_{k+1}, \dots, b_n\} \cap M = \emptyset$

$\Rightarrow \{a_1, \dots, a_m\} \cap M \neq \emptyset$

Sprich: "wenn immer ⊕ in M & ⊖ nicht in M dann muss mind. 1 d. Kopf(.) in M sein"

wenn Rumpf T → Kopf T

Answer Set

= wird definiert durch reduKt

→ Idaphiera: Γ_T

$P^M = \{a_1 \vee \dots \vee a_m :- b_1, \dots, b_k \mid a_1 \vee \dots \vee a_m :- b_1, \dots, b_k, not\ b_{k+1}, \dots, not\ b_n \in P,\ \{b_{k+1}, \dots, b_n\} \cap M = \emptyset\}$

Erfüllt Konsistenz b.

$a_1 \vee \dots \vee a_m :- b_1, \dots, b_k, not\ b_{k+1}, \dots, not\ b_n \in P$ wahr

$\{b_{k+1}, \dots, b_n\} \cap M = \emptyset$

True under M

Basic

Interpretation M = TS von ground P. IFF min. model von P^M

$\Rightarrow$ No $N \subseteq M$ also in P^M

Konsistenz

$M = \{l_1, \dots, l_n\}$ ground

M konsistent IFF

NO atom s.t. $\{p, \neg p\} \subseteq M$

Interpretation = kons. M

Wahrheitswerte von atom unter Interpretation (M)

a	a true under M	a false under M	undefined
a	$a \in M$	$\neg a \in M$	X
$\neg a$	$\neg a \in M$	$a \in M$	X
X	X	X	✓
not l	$l \notin M$	$l \in M$	X

Grounding

Verwendet d. HB (Herbrand base) um zu grounden

→ A S von Prog. P mit var ist folgendermaßen definiert:

→ als AS von grnd(P)

(M = AS von grnd(P) ⇒ (M ⊆ HB(P) = AS)

= Menge aller ground atome, d. aus PS & CS aus P gebildet werden können

Guess vs Check

Guess ?

Disjunktive Regeln "erraten"

Lsg Kandidaten

Check ✓

"Constraints" checken

Zulässigkeit ("admissibility")

Answer Set Programmierg

Model-basierte Diagnose

- Findet Gründe für einen "Failure" von einem techn. System
- verwendet ein Model (= korr. Verhalten Sys) um Gründe heraus zu finden

⊕	⊖
Deckt große Menge an Fehlern, Missverh., etc. ab	Probleme können zu "hard" sein

Diagnose Prob.

In DLV

⟨ H T O ⟩

H ... Hypothesen (= Raum d. mögl. defekten Komponenten) — Ground atoms

T ... Theorie (= Model) — Logic program

O ... Beobachtungen (= konkreter Fehlverhalten) — Ground Literals

Consistency-based

= set C ⊆ H s.t.

T ∪ O ∪ {¬h | h ∈ H \ S}

is consistent

⟨ H T O ⟩

ground atoms mit Pred ab(.) (= abnormal)

logic prog

ground literals

Abductive

= Erklärungen für beobachte Phänomene gemäß bekannten gesetzen in app. Domäne finden

= A ⊆ H s.t.

T ∪ A ⊨ O

⟨ H T O ⟩

ground atoms mit Pred nicht nur ab(.) (= abnormal)

logic prog

ground literals

O muss von sowohl T als auch H ableitbar sein

⊨ brave konsq. Relation

Disjunction 1/2

- Disjunction in logic programs differs from classical disjunction.
- In particular, the one-to-one correspondence between default logic extensions and answer sets holds only for *normal programs*.
  - Consider the program $P = \{a \vee b : -\}$ and its corresponding default theory $\delta(P) = (\emptyset, \{\top : / a \vee b\})$.
  - ➥ P has two answer sets, $\{a\}$ and $\{b\}$, while $\delta(P)$ has *one extension*, namely $Cn(\{a \vee b\})$.
- Disjunction is *minimal*

  $P = \{a \vee b \vee c : -\}$

  $\Longrightarrow$ has answer sets $\{a\}$, $\{b\}$, and $\{c\}$
- actually *subset minimal*

  $P = \{a \vee b : - ,$
  $a \vee c : - \}$

  $\Longrightarrow$ has answer sets $\{a\}$ and $\{b, c\}$

Probabilistic Reasoning

Definition

= verwendet Extension von propositional logic

- wende Grade von "belief" auf Propositions (= Zusicherungen, d. kpl. Binge wahr sind) an
- Baris elem: Random variables
  ⇒ haben Domain

- **Boolesch**: isDaddyP = true
- **Diskret**: Werte aus abzählbarer Domain { sonnig, wolkig, ... }
- **Kontinuierlich**: Über ab zählbare Domain z.B.: 4.02

Basic

= quantitativ

- **Probabilistic R.** = kl. T. + Prob. T.
- **Dempster-Shafer theory** = generalisierte Prob. ignorance + uncertainty
- **Fuzzy logic**: Non-discrete set-elem relation

Joint probability distribution (JPD)

Mapping, wo $\forall$ atomares Event einer Zahl $P(\omega)$ zugeordnet, s.t.

1. $0 \leq P(\omega) \leq 1$
2. $\sum_{\omega \in \Omega} P(\omega) = 1$

$\omega = \{ V_1 = v_1, \dots, V_n = v_n \}$

Marginalisierung

= Wahrsch., dass eine gewisse # von Var, einen gewissen Wert annimmt

$$P(V_i = v_i) = \sum_{V_i = v_i} P(V_1, \dots, V_n)$$

Bedingte Wahrscheinlichk.

$P(V_i = v_i \mid V_j = v_j)$

= Wahrscheinlichkeit, dass $V_i = v_i$ unter d. Bedingung $V_j = v_j$ gilt

$$P(V_i = v_i \mid V_j = v_j) = \frac{P(V_i, V_j)}{P(V_j)}$$

- **Produktregel**

$$P(V_1, \dots, V_n) = \prod_{i=1}^{n} P(V_i \mid V_{i+1}, \dots, V_n)$$

→ Unabhängig von Reihenfolge

- **Satz von Bayes**

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

- **Conditional Independence**

V cond. independent von B mit gegeb. Evidenz E, wenn $P(V \mid B, E) = P(V \mid E)$

zusätzliches Wissen wird über B erzeugt kein zusätzl. Wissen zu V

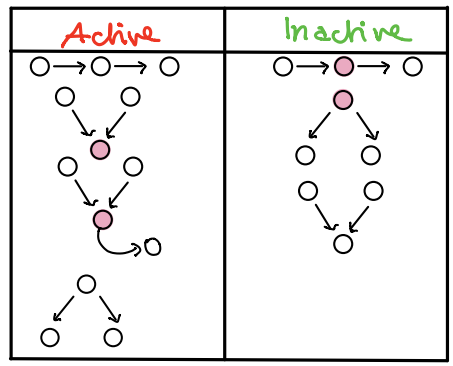
Atomare Ergebnis

Zuweisung aller Zufallsvar = Spezifikation d. Welt-zustand

- **MUTEX**: schließen sich gegenseitig aus
- **Exhaustive**: Es mus immer genau ein Ereignis stattfinden

D-Separation

Nur wenn es keine aktive Pfade cond. independent



Bayes Network

DAG zur Darstellung einer JDP inkl. Unabhängigkeit zw. d. Var

- gerichtete Kanten
  $V_1 \to V_2$: V_1 Elternteil von V_2
- Knoten = Zufallsvar
  - > Bedingte Wahrscheinlichkeit
  - > $\forall$ Knoten ist cond. indep. von seinen Nicht-Abkömmlingen mit Evidenz d. Eltern

Schließen

- **Kausales**: Ursache → Effekt
- **Diagnostisch**: Effekte → Ursache
- **Zwischenkausal**: zw. Ursache von persönl. Effekten