Einführung in Wissensbasierte Systeme
Zusammenfassung

March 27, 2014

1 Prädikatenlogik

Ground Atome	Ground Atome sind Atome ohne Variablen.
Literal	Ein Literal ist ein Atom oder ein negiertes Atom.
Klausel	Eine Klausel ist eine disjunktion von Literalen.
Eigenschaften der logischen Implikation $\models$	- Monotonie: $W \models \psi \rightarrow W \cup \{\varphi\} \models \psi$ - Deduktionstheorem: $W \cup \{\varphi\} \models \psi \Longleftrightarrow W \models \varphi \rightarrow \psi$ - Kontrapositionstheorem: $W \cup \{\varphi\} \models \neg\psi \Longleftrightarrow W \cup \{\psi\} \models \neg\varphi$ - Kontradiktionstheorem: $\not\models W \cup \{\varphi\} \Longleftrightarrow W \models \neg\varphi$

2 Nonmonotonic Reasoning

Monotonie	$(S \models A) \wedge (S \subseteq S') \; \rightarrow \; S' \models A$
Deduktiver Schluss Cn(T)	$Cn(T) = \{ \; \varphi \mid T \models \varphi \; \wedge \; \varphi \, is \, closed\}$
Eigenschaften von Cn(T)	- $T \subseteq Cn(T)$ "inflationaryness" - $Cn(T) = Cn(Cn(T))$ "idempotenz" - $T \subseteq T' \rightarrow Cn(T) \subseteq Cn(T')$ "monotonie" - $Cn(\emptyset)$ ist die Menge aller gültigen Formeln $\rightarrow Cn(\emptyset) \neq \emptyset$ - $Cn(T)$ ist die Menge aller Formeln $\Longleftrightarrow$ T ist konsistent. - $\varphi \in Cn(T) \; \rightarrow \; Cn(T \cup \{\varphi\}) = Cn(T)$
$CWA(T)$	$T_{asm} = \{\neg P \mid P$ ist ein ground atom, $T \not\models P\}$ $CWA(T) = \{\varphi \mid T \cup T_{asm} \models \varphi, \varphi$ ist geschlossen $\} = Cn(T \cup T_{asm})$

Eigenschaften der CWA

- CWA ist nicht monoton
- Intuitiv ist die CWA(T) der logische Schluss aller Annahmen.
- Berechnung erfolgt immer relativ zu einem fixen Vokabular (z. B. alle Konstanten in der KB)
- Wenn T eine konsistente Theorie ist, dann ist CWA(T) inkonsistent falls es ground atoms $A_1, ..., A_n$ gibt, sodass $T \models A_1 \vee ... \vee A_n$, aber $T \not\models A_i$ für alle $i = 1, ..., n$
- Zusätzliche Information kann zu einer inkonsistenten CWA führen.
- DCA kann verwendet werden um die Menge an möglichen Konstanten einzuschränken.
- Wenn die Theorie T nur aus Horn Klausen besteht, ist die CWA(T) konsistent.

Vollständige Theorie

Eine Theorie ist vollständig genau dann wenn, für jedes ground atom P, $P \in T$ oder $\neg P \in T$ gilt. Die CWA(T) ist vollständig.

$CWA^q(T)$

$T^q_{asm} = \{\neg P \mid P$ ist ein ground atom mit PS q, $T \not\models P\}$
$CWA^q(T) = \{\varphi \mid T \cup T^q_{asm} \models \varphi, \varphi$ ist geschlossen $\} = Cn(T \cup T^q_{asm})$

Default Theory $T = (W, \Delta)$

W: Menge von geschlossenen Formeln von first-order logic.
Δ: Menge von Default-Regeln.

Geschlossene Default Theorie

Eine Default Theorie ist geschlossen, wenn alle Formeln in Δ geschlossen sind (W ist per Definition geschlossen).

Eingenschaften von Extensions

Extensions sind nicht monoton (aus $W \subseteq W'$ folgt nicht $E \subseteq E'$).

Anwendung von Defaults

$\delta \in \Delta$ ist anwendbar auf E genau dann wenn $\varphi \in E$ und $\neg\varphi_1 \notin E, ..., \neg\varphi_n \notin E$ gilt.

Anwendung von Defaults relativ zum Kontext

Gegeben F (deduktiv geschlossene Menge von Formeln) und K (beliebige Menge von Formeln = Kontext). Ein Default ist Anwendbar auf F relativ zu K wenn $\varphi \in F$ und $\neg\psi_1, ..., \neg\psi_n \notin K$ gilt.

Fixpunktoperator $\Gamma_T(S)$

$\Gamma_T(S)$ ist die kleinste Menge F aus geschlossenen Formeln für die gilt:
- F ist deduktiv geschlossen, d. h. $Cn(F) = F$,
- $W \subseteq F$ und
- F ist abgeschlossen bez. der Anwendung der Defaults relativ zum Kontext S, d. h. für alle $\delta \in \Delta$ gilt, dass wenn $\varphi \in F$ und $\neg\varphi_1 \notin S, ..., \neg\varphi_n \notin S$ gilt, $\chi \in F$ auch gilt. "Wenn ich einen Default anwenden kann, dann ist das dadurch entstandene Wissen bereits bekannt."

Extension

E ist eine Extension von $T = (W, \Delta)$ genau dann wenn $\Gamma_T(E) = E$ gilt (d. h. wenn E ein Fixpunkt von Γ_T ist).

Berechnung von Extensions I	Es muss der Fixpunktoperator Γ_T bestimmt werden: 1. Klassisches Redukt Δ_E: $\Delta_E := \{\varphi/\gamma \ \mid (\varphi : \psi_1, ..., \psi_n/\gamma) \in \Delta, \ \{\neg\psi_1, ..., \neg\psi_n\} \cap E = \emptyset\}$ 2. $Cn^{\Delta_E}(W) := Cn(W \cup \bigcup_{i\geq 0} E_i)$, wobei $E_0 := \{\gamma \mid \varphi/\gamma \ \in \Delta_E$ und $W \models \varphi\}$ und $E_i := \{\gamma \mid \varphi/\gamma \ \in \Delta_E$ und $W \cup E_{i-1} \models \varphi\}$ gilt. 3. $\Gamma_T(E) = Cn^{\Delta_E}(W)$ $\Longrightarrow Cn^{\Delta_E}(W)$ ist der deduktive Schluss von W unter klassischer Logik zusammen mit den Inferenz Regeln in Δ_E.
Berechnung von Extensions II	$T = (W, \Delta)$, dann gilt: E ist eine Extenstion von $T \iff Cn^{\Delta_E}(W) = E$ Alle Mengen der Form $Cn(W \cup \mathcal{C})$, $\mathcal{C} \subseteq \{\gamma \mid (\varphi : \psi_1, ..., \psi_n/\gamma) \in \Delta\}$, sind mögliche Kanditaten für Extensions.
Semi-rekursive Beschreibung von Extensions	Sei E eine Menge geschlossener Formeln und $T = (W, \Delta)$ eine geschlossene Default Theorie. E ist eine Extension von T genau dann wenn $E = \bigcup_{i\geq 0} E_i$, wobei $E_0 = W$ und $E_{i+1} = Cn(E_i) \cup \{\gamma \mid (\varphi : \psi_1, ..., \psi_n/\chi) \in \Delta, E_i \models \varphi$ und $\neg\psi_1, ..., \neg\psi_n \notin E\}$ gilt.
Normale Defaults	Normale Defaults haben die Form $\varphi : \psi/\psi$. Normale Defaults haben immer eine Extension.
Skolemisierte Formel	http://de.wikipedia.org/wiki/Skolemform
Prenex Form	Eine Formel ist in Prenex Normal Form wenn alle Quantoren am Anfang stehen.
Schließen von Default Theorien	Eine skolemisierte (= geschlossene?) Default Theorie entsteht durch ersetzen jeder Annahme und jedes Defaults von W und Δ durch ihre skolemisierten Formen. Skolemisierung eines Defaults $\delta = (A : B_1, ..., B_n/C)$: 1. $\forall C$ ist der universal closure $(\forall_{x_1}, ..., \forall_{x_2} C)$, wobei $x_1, ..., x_n$ freie Variablen in C sind. 2. Skolemisiere C, keine der Skolem-Therme darf in $A, B_1, ..., B_n$ vorkommen. Das Ergebnis ist C_1 3. Eliminiere alle Quantoren, das Ergebnis ist C_2. 4. Aus $\delta = (A : B_1, ..., B_n/C)$ wird jetzt $\delta = (A : B_1, ..., B_n/C_2)$
$TERMS(T)$	Sei $T = (W, \Delta)$ eine Default Theorie. Die Menge $TERMS(T)$ beinhaltet alle ground Therme, die aus den Funksionssymbolen in T konstuiert werden können.
Closure von T	$\overline{T}$ bezeichnet den closure of T. Wenn T geschlossen ist, dann ist $\overline{T} = T$. Andernfalls ersetze alle offene Defaults einer skolemisierten Form von T duch ihre Instanzen aus $TERMS(T)$, das Ergebnis ist $\overline{T}$.

Extensions von offenen Default Theorien	Sei $T = (W, \Delta)$ eine offene Default Theorie. E ist eine Extension von T genau dann wenn $E = F \cap \mathcal{L}_T$ gilt, wobei F eine Extension von $\overline{T}$ und $\mathcal{L}_T$ die Menge aller Formeln ist, die durch die in T vorkommenden Prädikat- und Funktionssymbole gebildet werden können, ist.

3 Answer Set Programming

Klausel vs. Regel	Aus $a \vee \neg b_1 \vee ... \vee \neg b_n$ wird $a : -b_1, ..., b_n$
Defaults	Aus $b_1 \wedge ... \wedge b_n : \neg c_1, ..., \neg c_m / a$ wird `a :- b1, ..., bn, not c1, ..., not cm.` Wenn $b_1, ..., b_n$ ableitbar und $c_1, ..., c_m$ nicht ableitbar sind, dann schließe auf a.
Semantik eines Programmes	Die Semantik eines Programmes ist durch seine Answer Sets gegeben.
Negation as failure	Wird über *not* definiert.
Programm	Ein Programm ist eine definierte Menge aus Regeln.
Eigenschaften von Programmen	Eine Regel $a_1 \vee ... \vee a_m : -b_1, ..., b_k, not\ b_{k+1}, ..., not\ b_n$ ist - ein Fakt wenn $n = 0$ und $m \geq 1$, - basic wenn $n = k$ und $m \geq 1$, - nicht-disjunktiv wenn $m = 1$ - normal wenn sie nicht-disjunktiv ist und keine starke Negation $\neg$ enthält, - horn wenn sie normal und basic ist und - ground wenn alle ihre Literale ground sind. Ein programm ist *, wenn all seine Regeln * sind.
Konsistente Menge aus ground Literalen	Eine Menge M von ground Literalen ist konsistent genau dann wenn es kein atom p gibt sodass $\{p, \neg p\} \subseteq M$ gilt
Interpretation	Eine konsistente Menge aus ground Literalen wird Interpretation genannt. Ein Literal l ist $true$ unter einer Interpretation M wenn $l \in M$ gilt und $false$ andernfalls. Ein Default Literal *not* l ist $true$ unter einer Interpretation M wenn $l \notin M$ gilt und $false$ andernfalls.
Semantik von ground Programmen	M ist ein klassisches Modell einer Regel wenn es ein klassisches Modell der entsprechenden logischen Formel ist. M ist ein klassisches Modell eines Programmes P wenn es ein klassisches Modell jeder Regel ist aus P ist.
Answer Set I	Ein Answer Set wird auch stabiles Modell (stable model) genannt.

Reduct P^M	Sei P ein Programm und M eine Interpretation, dann ist das Redukt P^M wie folgt definiert: $P^M = \{a_1 \vee ... \vee a_m : -b_1, ..., b_k \mid$ $a_1 \vee ... \vee a_m : -b_1, ..., b_k, not\ b_{k+1}, ..., not\ b_n \in P, \{b_{k+1}, ..., b_n\} \cap M = \emptyset\}$
Answer Set II	M ist ein Anser Set genau dann wenn es eine minimale Menge von Literalen ist und ein Modell von P^M ist.
Beziehung zwischen Anwser Sets und Extensions	Lediglich die Answer Sets von normalen Programmen können 1:1 in die Extensions der zugehörigen default Theorie übergeführt werden.
Killing Klausel	Die Klausel `p :- not p.` eliminiert alle Answer Sets. Die Klausel `p :- q, not r, not p.` eliminiert alle Anwser Sets, die `q` enthalten und `r` nicht enthalten. `:- q, not r.` ist die Kurzschreibweise für `p :- not p, q, not r.`
Diagnose Problem $\mathcal{P}$	Ein Diagnose Problem $\mathcal{P}$ ist ein Tripel $\langle H, T, O \rangle$ H ist eine Menge aus Hypothesen, in DLV eine Menge aus ground Atomen. T ist die Theorie, in DLV ein Logik Programm. O ist eine Menge aus Beobachtungen, in DLV eine Menge aus ground Literalen.
Konsistenzbasierende Diagnose	Bei einer konsistenzbasierender Diagnose ist H (Hypothesen) eine Menge von ground Atomen mit Predikatnamen $ab(.)$. Eine konsistenzbasierende Diagnose ist eine Menge $S \subseteq H$, mit der $T \cup O \cup S \cup \{\neg h \mid h \in H \backslash S\}$ konsistent (d. h. es existiert zumindest ein Answer Set) ist.
Abduktion	Abduktion bezeichnet das Finden von Erklärungen für beobachtete Phänomene entsprechend den bekannten Gesetzen der Domäne.
Abduktive Diagnose	Bei abduktiver Diagnose ist H eine beliebige Menge von ground Atomen (nicht nur $ab(.)$). Eine abduktive Diagnose ist eine Menge $S \subseteq H$ sodass $T \cup S \models O$ gilt.

4 Probabilistic Reasoning

Zufallsvariablen	Es gibt boolean, diskrete und fortlaufende Zufallsvariablen.
Atomares Event	Allen Zufallsvariablen V werden Werte v zugewiesen (bem.: Möglicherweise vergleichbar mit einer Interpretation der klassischen Logik). Atomare Events schließen sich gegenseitig aus (es kann immer nur eines "aktiv" sein). Es muss immer ein atomares Event "aktiv" sein.
Joint Probability Distribution	Die JPD weißt einem atomaren Event einen Wahrscheinlichkeitswert zu, geschrieben als $P(V_1, ..., V_n) = x$, mit $0 \leq x \leq 1$. Die Summe aller JDP der atomaren Events muss 1 sein.

Marginalisation	Die extraktion der Verteilung aus einer Untermenge einer gegebenen Menge von Variablen wird marginalisation genannt, z. B. die Berechnung von $P(V_{i_1}, ..., V_{i_k})$, $1 \leq i_j \leq n$, wenn $P(V_1, ..., V_n)$ gegeben ist.
Berechnung der Marginalisation	$P(V_{i_1} = v_{i_1}, ..., V_{i_k} = v_{i_k}) = \sum_{V_{i_1}=v_{i_1},...,V_{i_k}=v_{i_k}} P(V_1, ..., V_n)$ Bem.: "Die marginalisierte Wahrscheinlichkeit $P(V_i = v_1)$ ist die Summe der JPD, bei denen $V_{i_1} = v_{i_1}, ..., V_{i_k} = v_{i_k}$ gilt."
Bedingte Wahrscheinlichkeit	$P(V_i \mid V_j) = \frac{P(V_i, V_j)}{P(V_j)}$
Produktregel	$P(V_1, ..., V_n) = \prod_{i=1}^{n} P(V_i \mid V_{i+1}, ..., V_n)$ Wichtig für die Praxis, da bedingte Wahrscheinlichkeiten i. d. R. leichter als die JPD hergeleitet werden können.
Bayes Regel	Aus $P(V_i, V_j) = P(V_i \mid V_j)P(V_j)$ und $P(V_i, V_j) = P(V_j \mid V_i)P(V_i)$ kann $P(V_i \mid V_j) = \frac{P(V_j \mid V_i)P(V_i)}{P(V_j)}$ hergeleitet werden. Die Regel kann wie folgt interpretiert werden: $P(Cause \mid Effect) = \frac{P(Effect \mid Cause)P(Cause)}{P(Effect)}$ Die Werte der rechten Seite der Gleichung ist in medizinischen Anwendungsgebieten of recht gut bekannt.
Nützliches	$P(a \mid b) + P(\neg a \mid b) = 1$ $P(a \mid b) + P(a \mid \neg b) = 1$ (ev. nicht richtig!)
Bedingte Unabhängigkeit	Sei $\mathbf{V}_i$ und $\mathbf{E}$ eine Menge von Variablen und V eine Variable. Dann ist V bedingt unabhängig von $\mathbf{V}_i$ bei gegebenen Wissen $\mathbf{E}$ gebau dann wenn $P(V \mid \mathbf{V}_i, \mathbf{E}) = P(V \mid \mathbf{E})$ gilt. Zusätzliches Wissen $\mathbf{V}_i$ hat nicht zusätzliches Wissen über V zur folge.
Gegenseitige bedingte Unabhängigkeit	$V_1, ..., V_n$ sind gegenseitig (mutually) bedingt unabhängig wenn jede Variable von jeder bedingt unabhängig ist: $P(V_i \mid V_1, ..., V_{i-1}, V_{i+1}, ..., V_n, \mathbf{E}) = P(V_i \mid \mathbf{E})$
Produktregel für gegenseitige bedingte Unabhängigkeit	$P(V_1, ..., V_n \mid \mathbf{E}) = \prod_{i=1}^{n} P(V_i \mid \mathbf{E})$ Im Spezialfall $\mathbf{E} = \emptyset$ gilt daher $P(V_1, ..., V_n) = P(V_1)P(V_2)...P(V_n)$
Bayessche Netzwerke	Ein Bayessches Netzwerk ist gerichteter Graph, welcher jede JPD, unter Mitberücksichtigung der Unabhängigkeit zwischen den Variablen, darstellt. Jedes Bayessche Netzwerk ist ein gerichteter, azyklischer Graph, wobei jeder Knoten eine Zufallsvariable repräsentiert und eine bedingte Wahrscheinlichkeitsverteilung $P(V \mid Parents(V))$ zugewiesen bekommt. $Parents(V)$ ist die Menge von Elternknoten von V. Wenn es eine Verbindung $V_1 \longrightarrow V_2$ gibt, dann ist V_1 ein Elternknoten von V_2.

Eigenschaften Bayesscher Netzwerke	Für alle Knoten $\mathbf{W}$, welche weder Elternknoten oder Nachfolgeknoten von V sind, gilt $P(V \mid \mathbf{W}, Parents(V)) = P(V \mid Parents(V))$. $P(V_1, ..., V_n) = \prod_{i=1}^{n} P(V_i \mid Parents(V_i))$
Kausaler Schluss	Von der Ursache zur Auswirkung.
Diagnostischer Schluss	Von der Auswirkung zur Ursache.
Interkausaler Schluss	Zwischen Ursachen einer bekannten Auswirkung.
D-Separation	Zwei Knoten X und Y werden durch $\mathbf{E}$ d-separiert genau dann wenn X und Y bedingt unabhängig bez. gegebener Evidenz $\mathbf{E}$ sind. Zwei Knoten X und Y werden durch $\mathbf{E}$ d-separiert genau dann wenn jeder ungerichtete Pfad zwischen X und Y durch $\mathbf{E}$ blockiert ist. Ein Pfad P ist blockiert relativ zu $\mathbf{E}$ wenn es einen Knoten Z auf dem Pfad P gibt der eine folgenden Bedingungen erfüllt: - $Z \in \mathbf{E}$ und eine Kante ist ein-, die andere ist ausgehend, - $Z \in \mathbf{E}$ und beide Kanten auf sind ausgehend oder - weder Z noch ein Nachfolger von Z sind $\in \mathbf{E}$ und beide Kanten sind eingehend.