1 Aussagenlogik AL: Verknüpfung von Aussagen

Syntax

- atomare Formeln A,B,C sind AL-Formeln
- F und G AL-Formeln $\Rightarrow$ (F$\wedge$G),(F$\vee$G) und $\neg$ F AL-Formeln
- müssen in endlich vielen Schritten gebildet werden können
- echtes Teilwort heißt Teilformel

Semantik (Wahrheitswerte: $\{1,0\}$)

- Belegungen der atomaren Formeln,z.B. $\mathcal{A}(A_1) = 1$
- $(G \vee H)$: Disjunktion (ODER)
- $(G \wedge H)$: Konjunktion (UND)
- $\neg G$: Negation (NICHT)

Bemerkungen

- Syntaxbaum (Wurzelbaum): Zur Bestimmung des Wahrheitswerts
- F = $((A_1 \wedge A_3) \vee \neg A_4)$
- $D_F = \{A_1, A_3, A_4\}$
- $\mathcal{A}$: $\mathcal{A}(A_1) = 1$, $\mathcal{A}(A_3) = 0$, $\mathcal{A}(A_4) = 1$
- $\mathcal{A}'$: $\mathcal{A}'(A_1) = 1$, $\mathcal{A}'(A_3) = 0$, $\mathcal{A}'(A_4) = 0$
- $\mathcal{A}'$ ist erfüllende Belegung für F
- D sind atomare Formeln von F ($|D|$ = n $\Rightarrow 2^n$ mögl. Belegungen)
- $\mathcal{A}$(F) ist von der Reihenfolge der Auswertung unabhängig
- passende Belegung $\mathcal{A}$: D $\rightarrow \{0,1\}$, falls $D_F \leq$ D (Jede Formel hat Wert)
- erfüllende Belegung (Modell für F): $\mathcal{A}$(F) = 1
- Formel ist erfüllbar, wenn es min. ein Modell gibt
- Formel heißt gültig: jede passende Belegung ist erfüllende Belegung (Tautologie) ($\neg$F ist unerfüllbar)
- Es gibt Formeln die nicht erfüllbar sind (heißen Widerspruch)
- Es gibt Tautologien: $\mathcal{A}$(F) = 1 oder $\mathcal{A}$(F) = 0

- Semantische Äquivalenz F $\equiv$ G

  falls gilt $\mathcal{A}$: D $\rightarrow \{0,1\}$ mit $\mathcal{A}$(F) = $\mathcal{A}$(G)

  z.B. Nachzuprüfen durch Wahrheitswert(WW)-Tafel

	A	B	$(A \vee B)$	$((A \vee B) \vee B)$
$\mathcal{A}_1$	0	0	0	0
$\mathcal{A}_2$	0	1	1	1
$\mathcal{A}_3$	1	0	1	1
$\mathcal{A}_4$	1	1	1	1

  Aufwand: 2^n bei n atomaren Formeln

- Implikation: F impliziert G (F $\rightarrow$ G) $\equiv$ ($\neg$ F $\vee$ G)

  Implikation nur dann nicht erfüllt, wenn F erfüllt, aber G nicht

- Folgerung (aus F folgt G) F $\mid =$ G

  $\mathcal{A}$(F) = 1 $\Rightarrow$ $\mathcal{A}$(G) = 1

  F $\equiv$ G $\Leftrightarrow$ F $\mid =$ G und G $\mid =$ F

- Deduktionstheorem

  F $\mid =$ G g.d.w. (F $\rightarrow$ G) ist Tautologie

- Einheits-Resolution

  $(((A \vee B) \wedge \neg B) \rightarrow A)$ ist Tautologie

  Resolution benutzt man um die Erfüllbarkeit rein syntaktisch zu entscheiden

- Erfüllbarkeit bei gegebener Belegung prüfen: linearer Aufwand (Syntaxbaum)

- Möglichkeiten um Erfüllbarkeit zu prüfen: WW-Tafel, Horn-Algorithmus, 2-Sat-Graph, Resolution

2 Äquivalenzen und Normalformen

- Äquivalenzen
  - $F \equiv F$
  - $F \equiv G \Rightarrow G \equiv F$
  - $F \equiv G, G \equiv H \Rightarrow F \equiv H$
- Ersetzbarkeitstheorem

  $F \equiv G$. H ist Formel in der F als Teilformel vorkommt $\Rightarrow H \equiv H'$ mit H′ entstanden aus H, wobei F durch G ersetzt wurde

  (Beweis per struktureller Induktion)
- Semantische Äquivalenzen

$(F \wedge F) \equiv F \equiv (F \vee F)$	*Idempotenz*
$(F \wedge G) \equiv (G \wedge F)$	*Kommutativitaet*
$(F \vee G) \equiv (G \vee F)$	*Kommutativitaet*
$((F \wedge G) \wedge H) \equiv (F \wedge (G \wedge H))$	*Assoziativgesetze*
$((F \vee G) \vee H) \equiv (F \vee (G \vee H))$	*Assoziativgesetze*
$(F \wedge (F \vee G)) \equiv F$	*Absorptions−*
$(F \vee (F \wedge G)) \equiv F$	*gesetze*
$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$	*Distributivgesetze*
$\neg\neg F \equiv F$	*Doppelnegation*
$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$	*De Morgan*
$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$	
$(F \rightarrow G) \equiv (\neg G \rightarrow \neg F)$	
$(\neg F \vee G) \equiv (\neg\neg G \vee \neg F)$	

- Normalformen

  Literal: atomare Formel oder die Negation einer atomaren Formel

  Konjunktive Normalform (KNF): F = $\bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m_i} L_{i,j})$

  Disjunktive Normalform (DNF): F = $\bigvee_{i=1}^{n} (\bigwedge_{j=1}^{m_i} L_{i,j})$
- Beispiel zur Umformung KNF/DNF

  $(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$

- Algorithmus zur Normalisierung (o.E. KNF)
  1. Erstze in F jedes Vorkommen einer Teilformel
     $\neg\neg G$ durch G,
     $\neg(G \wedge H)$ durch $(\neg G \vee \neg H)$,
     $(G \vee H)$ durch $(\neg G \wedge \neg H)$
     bis keine derartige Teilformel mehr vorkommt
  2. Ersetze jedes Vorkommen einer Teilformel
     $(F \vee (G \wedge H))$ durch $((F \vee G) \wedge (F \vee H))$
     $((F \wedge G) \vee H)$ durch $((F \vee H) \wedge (G \vee H))$
     bis keine derartige Teilformel mehr vorkommt
     $\Rightarrow$ Resultat KNF, die semantisch äquivalent zur Ausgangsformel F
- Normalisierung mittels WW-Tabelle
  1. Betrachte Belegungen bei denen Formel erfüllt
     $\Rightarrow$ verknüpfe mit $\wedge$ und diese Teilformeln mit $\vee \Rightarrow$ DNF
  2. Betrachte Nullstellen $\mathcal{A}$(F) und konstruiere für jede Nullstelle eine Klausel
     $\mathcal{A}$(F) = 0 von $(\neg A \vee \neg B \vee C)$ (110) und verknüpfe diese mit $\wedge \Rightarrow$ KNF
- Satz

  Für jede Formel F $\in$ AL gibt es (min.) eine semantisch äquivalente Formel in KNF und auch (min.) eine semantisch äquivalente in DNF

  Beweis per struktureller Induktion(mit Rekursionsschema)

  Mittels Normalisierung ergibt sich $|D_F|$ = n

  KNF: n_0 Literale, DNF: n_1 Literale

  mit $n_0 + n_1 = 2^n$

  KNF und DNF haben zusammen $2^n * n$ Literale
- Anzahl semantisch unterschiedl. Formeln (n atomare Formeln)

  Menge aller 1-Stelle in WW-Tabelle (m = 2^n Belegungen)

  $1 + m + \frac{m(m-1)}{2} + ... = 2^m \Rightarrow 2^{2^n}$ versch. Formeln

  (Anzahl unterschiedl. WW-Tabellen)
- Bemerkungen
  - KNF / DNF sind nicht eindeutig bestimmt
  - Jede Formel induziert Boolsche Funktion f: $\{0,1\}^n \rightarrow \{0,1\}$, $F \rightarrow \mathcal{A}$(F)
    semantisch äquivalente Formeln haben gleiche boolsche Funktion
    zu jeder boolsche Funktion ex. Formel mit $\mathcal{A}$(F) = f
    Um jede boolsche Fkt mittels $\wedge, \vee, \neg$ realisieren zu können brauchen wir (im worst-case-Fall) mindestens $\frac{2^n}{log_2(n+5)} - 1$ Zeichen , $\frac{1}{5} * \frac{2^n}{log_2(n+5)}$ Gatter

3 Hornformeln

- F ist Hornformel $\Leftrightarrow$ F in KNF und jede Klausel enthält höchstens ein pos. Literal
- Horn-Algorithmus

  Input: Hornformel F = $\bigwedge_{i=1}^{n} F_i$ (m Klauseln) und $F_i = \bigvee_{j=1}^{n_i}$

  1. Besteht eine Klausel aus genau einem pos. Literal so markiere dieses pos. Literal
  2. Streiche jede Klausel in der ein markiertes Literal positiv vorkommt (insbes. jede Formel aus (1))
  3. Streiche in jeder Klausel jedes negative Literal das markiert ist

  Output: bleit nach Streichen eine Klausel ohne Literale (leere Klausel) übrig, so gebe aus:

  "F ist nicht erfüllbar" andernfalls "F ist erfüllbar" mit $\mathcal{A} = \left\{ \begin{array}{ll} 1 & \text{falls A markiert wurde} \\ 0 & \text{sonst} \end{array} \right\}$

  (diese erfüllende Belegung ist auch minimales Modell)
- Aufwand des HA

  n atomare Formeln mit F = $C_1 \wedge ... \wedge C_m$

  N Anzahl der in F insgesamt genannten Literale, o.E. N $\leq$ n * m

  Pro Scan O(N) für 1. und O(N) für 2.,3 und höchstens n Scans

  $\Rightarrow$ Gesamtformel $\in O(n * N)$ (polynomial mit der Eingabelänge)
- Zum Beweis des HA (Induktion)
  - bricht nach höchstens n Markierungsschritten ab
  - Restformel bleibt immer Hornformel
  - Verringerung der Klauselzahl

4 k-Sat-Formeln

Eine Formel F in KNF bei der jede Klausel k Literale enthält heißt k-SAT-Formel
SAT-Problem $\Leftrightarrow$ Erfüllbarkeitsproblem (satisfiability)

4.1 2-Sat-Formeln

- Jede 2-Sat-Klausel erzeugt eine semantisch äquivalente Implikation
- 2-SAT-Graphen, Implikationsgraph
  1. Für jede atomare Formel A_i von F reserviere eine Ecke für A_i und $\neg A_i$
  2. Für jede Klausel $(L_1 \vee L_2)$ führe ein Pfeil $(\neg L_1 \rightarrow L_2)$ und $(\neg L_2 \rightarrow L_1)$
- Satz (Unerfüllbarkeit)

  Die 2-SAT-Formel F ist genau dann nicht erfüllbar, wenn es im 2-SAT-Graphen einen (Implikationen-)Kreis gibt, der eine atomare Formel A_i von F und $\neg A_i$ berührt

  Beweis: Übung ?
- Konstruktion einer erfüllenden Belegung falls ein solcher Kreis nicht existiert:

  Wähle Literal L, das noch nicht belegt ist und für das Kein Weg von L nach $\neg L$ im 2-Sat-Graphen existiert mit 1 und alle L aus erreichbaren Literale (auch mit 1)

4.2 Max SAT

Menge M = $\{C_1, ..., C_m\}$ von m Klauseln mit F = $C_1 \wedge ... \wedge C_m$ nicht erfüllbar.
Problem: Wie groß ist die Anzahl der gleichzeitig erfüllbaren Klauseln maximal?
$\overline{m}$: maximale Anzahl der gleichzeitig erfüllbaren Klauseln
F erfüllbar $\Leftrightarrow \overline{m}$ = m
$\overline{m} \geq \frac{m}{2}$ gilt immer (mit Belegung und Komplementärbelegung)

- zufällige Belegung

  Wahscheinlichkeit für best. Belegung p($\mathcal{A}$) = $\frac{1}{2^k}$ (für eine Klausel)

  Wahrscheinlichkeit, dass Klausel C_i erfüllt: $\frac{1}{2^k}(2^k - 1)$ und nicht erfüllt: $\frac{1}{2^k}$ (da genau 1 nicht-Modell existiert)

  Erwartungswert der Anzahl Z der erfüllten Klauseln von f (für k-SAT-Formeln)

  $$\text{E(Z)} = \sum_{i=1}^{m} p(C_i) = \sum_{i=1}^{m} \frac{2^k - 1}{2^k} = m\frac{2^k - 1}{2^k} = m(1 - \frac{1}{2^k})$$

- Es muss mindestens eine Belegung geben, die mindestens $m(1 - \frac{1}{2^k})$ Klauseln gleichzeitig erfüllt (Erdös)

- Algorithmus von D.S. Johnson

  Input: atomare Formeln $A_1, ..., A_n$, Klauselmenge M = $\{C_1, ..., C_m\}$, C_i sei k_i-Klausel (enthält k_i Literale)

  1. Setze Gewicht w(C_i) $\leftarrow \frac{1}{2^{k_i}}$ für i = 1,...,m

     r $\rightarrow$ w(M) = $\sum_{i=1}^{m} w(C_i)$ w($\varnothing$) = 0
  2. UNERFÜLLT = M
  3. for i = 1,...,n do
     sei ERFÜLLBA$R^{(1)}$ die Teilmenge aller Klauseln aus UNERFÜLLT, die A_i enthält
     sei ERFÜLLBA$R^{(0)}$ die Teilmenge aller Klauseln aus UNERFÜLLT, die $\neg A_i$ enthält
  4. wähle den Wahrheitswert t (für A_i), so dass das Gewicht von ERFÜLLBA$R^{(t)}$ maximal ist
  5. markiere atomare Formel A_i mit Wahrheitswert t
  6. lösche ERFÜLLBA$R^{(t)}$ aus UNERFÜLLT
  7. verdopple das Gewicht jeder Klausel in ERFÜLLBA$R^{(1-t)}$
  8. end for

  Output: die Belegung $\mathcal{A}$ erfüllt bei k-SAT-Formeln min. $m * (1 - \frac{1}{2^k})$ Klauseln gleichzeitig (aber nicht notwendigerweise Maximalzahl)

- Bemerkungen
  - Gewichtsreduktion min. so groß wie Gewichtszunahme von w(UNERFÜLLT)
  - bei Abbruch enthält UNERFÜLLT die durch M nicht erfüllten Klauseln
  - das Gewicht jeder nicht erfüllten Klausel C_i wurde genau k_i mal verdoppelt
    w(UNERFÜLLT) $\leq \frac{m}{2^k}$
  - Anzahl der bei M erfüllten Klauseln z: z $\geq m * (1 - \frac{1}{2^k})$
  - Aufwand des Johnson-Algorithmus
    pro Iteration O(m^2), insg: O(nm^2)
  - i.A. gilt nicht $z_{Johnson} = \overline{m}$
  - Entscheidungsproblem der Erfüllbarkeit von 3-Sat-Formeln ist NP-vollständig
    falls Johnson immer $\overline{m} = z_{Johnson}$ wäre er Verfahren für die Entscheidung der Erfüllbarkeit von 3-Sat-Formeln
- NP-Vollständig

  vollständige Schwierigkeit aller Sprachen aus Komplexitätsklasse NP
- NP-hart

  Problem ist mindestens so schwer wie jedes beliebige Problem aus NP

4.3 Randomisierte Erfüllbarkeitstests für k-SAT-Formeln

- Algorithmus von Uwe Schöning

  Input: Klauselmenge M = $\{C_1, ..., C_m\}$ (KNF) und zugeh. atomare Formeln $A_1, ..., A_n$ und $|C_i|$ = k, i = 1,...,m

  Ist F = $C_1 \wedge ... \wedge C_m$ erfüllbar?

  1. Wähle eine Belegung $\mathcal{A}$ zufällig und gleichverteilt aus der Menge aller Belegungen für $\mathcal{A}$(F)
     ($\mathcal{A}$: Startbelegung)
  2. FOR 3n Schritte DO
  3. IF $\mathcal{A}$(F) = 1
  4. Stop: "F ist erfüllbar und $\mathcal{A}$ ist Modell für F"
  5. END IF
  6. Wähle Klausel c $\in$ M die bei $\mathcal{A}$ nicht erfüllt ist
  7. Wähle zufällig und gleichverteilt ein Literal L aus dieser Klausel C, C = C_i = $(L_{i_1} \vee ... \vee L_{i_n})$
  8. flippe Belegung von L
  9. END FOR
  10. Abbruch ohne erfüllende Belegung "Kein Erfolg bei diesem Durchgang"
      Wiederhole diese Durchgänge t mal t $\in \mathbb{N}_{>0}$ falls die bisherigen Durchgänge erfolglos waren

  Output: bei Erfolg wie in (4)

  bei t-facher Wiederholung ohne Erfolg: "F ist nicht erfüllbar"

- Bemerkungen zu Schönig-Algorithmus

  1. erfüllbare Formel F in 3-KNF $\Rightarrow$ Erfolgswahrscheinlichkeit pro Durchlauf p = $(\frac{3}{4})^n$
  2. erfüllbare Formel F mit falschen Output mit Fehlerwahrscheinlichkeit $\leq e^{-tp}$

- Klauseldichte: $\frac{m}{n} = \frac{Anzahl\ Klauseln}{Anzahl\ atomare\ Formeln}$

- Erfüllbarkeitsrate und Aufwand

- Max-2-Sat ist NP-hart

  Idee: zu jeder Klausel C_i von F konstruiere eine 2-SAT-Klauselmenge

  mit F = $C_1 \wedge ... \wedge C_m$ eine beliebige 3-SAT-Formel, mit $C_i = (A \vee B \vee C)$ i = 1,...,m.

  $M_1 : A, B, C, D$ (D neue atomare Formel)

  $M_2 : (\neg A \vee \neg B), (\neg B \vee \neg C), (\neg C \vee \neg A)$

  $M_3 : (A \vee \neg D), (B \vee \neg D), (C \vee \neg D)$

  $M_i := M_1 \cup M_2 \cup M_3$ hat genau 10 2-SAT-Klauseln und m zusätzliche neue atomare Formeln $D_1, ..., D_m$.

  Lösung von MAX-2-Sat ist min. so schwer wie die Entscheidung ob 3-Sat-Formel erfüllbar oder nicht.

  Max-2-Sat: Optimierungsproblem

  3-Sat: Entscheidungsproblem

  Die folgenden Aussagen sind äquivalent

  1. Max-2-Sat-Instaz M hat Lösung $\geq$ 7m
  2. 3-Sat-Instanz F hat Lösung "JA"

5 Resolution

Syntaktisches Verfahren zur Entscheidung der (Un-) Erfüllbarkeit einer Formel (KNF)

- Resolvieren

  Seien $C_1 = \{L_{1,1}, ..., L_{1,n_1}\}$ und $C_2 = \{L_{2,1}, ..., L_{2,n_2}\}$ zwei Klauseln mit $L_{1,i} = \neg L_{2,j}$

  R = $(C_1 - L_{1,i}) \vee (C_2 - L_{2,j})$ Der Resolvent von C_1 und C_2 (nach $L_{1,i}$)

  () leere Klausel (auch ein kleines Quadrat)

- Resolutionslemma

  F in KNF, R Resolvent zweier Klauseln von F: F $\equiv F \cup R$

  Beweis:

  - passende Belegung für F passt auch für $F \cup R$
  - durchgehen der Fälle mit $\mathcal{A}$(F) = 1
    a) $\mathcal{A}$(L) = 1
    $\Rightarrow \mathcal{A}(C_2) = 1$ und $\mathcal{A}(\neg$ L) = 0
    $\mathcal{A}(C_2$ - L) = 1 $\Rightarrow \mathcal{A}$(R) = 1
    b) $\mathcal{A}$(L) = 0
    $\Rightarrow \mathcal{A}(C_1) = 1$ und $\mathcal{A}$(L) = 0
    $\mathcal{A}(C_1$ - L) = 1 $\Rightarrow \mathcal{A}$(R) = 1

- Deduktion, Deduktionskreis

  Herleitung der leeren Klausel aus einer Klauselmenge F ist endliche Folge $(C_1, ..., C_q)$ von Klauseln mit C_q = () und C_i (i = 1,...,q) ist resolviert aus Klauseln $F \cup \{C_1, ..., C_{i-1}\}$

  Deduktion der leeren Klausel $\Rightarrow$ F unerfüllbar

- Deduktionstheorem

  F $\mid =$ G $\Leftrightarrow$ F $\rightarrow$ G ist Tautologie $\Leftrightarrow F \wedge \neg G$ ist unerfüllbar

- Resolutionsschritte

  Res(F) = $F \cup \{R \mid$ R ist Resolvent zweier Klauseln von F$\}$

  $Res^0(F)$ = F

  $Res^{n+1}(F)$ = Res($Res^n(F)$), n = 0,1,...

- Resolventenhülle

  $Res^*(F) = \bigcup_{n=0}^{\infty} Res^n(F)$

  F = $Res^0(F) \subseteq Res^1(F) \subseteq ... \subseteq Res^*(F)$

  F $\equiv Res^*(F)$

  aus n atomaren Formeln können höchstens 4^n Klauseln entstehen

- Resolutionssatz der Aussagenlogik

  Klauselmenge F unerfüllbar $\Leftrightarrow () \in Res^*(F)$

  Beweis:

  - Korrektheit
    sei $() \in Res^*(F) \Rightarrow$ i > 0 mit $() \in Res^i(F) \Rightarrow Res^{i-1}$ unerfüllbar $\Rightarrow$ F unerfüllbar
  - Vollständigkeit
    F unerfüllbar: per Induktion (nach Anzahl der versch. atomaren Formeln) erreicht man Stelle an der leere Klausel resolviert wird

  F $| =$ G $\Leftrightarrow () \in Res^*(F \wedge \neg G)$

- Resolutionsalgorithmus

  Input: Formel F in KNF

  1. transformiere ggf. F in Mengenschreibweise
  2. n $\leftarrow$ 0, Res[0] $\leftarrow$ F
  3. Wiederhole
     n $\rightarrow$ n+1, Res[n] $\leftarrow$ Res(Res(n-1))
     bis ($() \in$ Res[n] oder Res[n] = Res[n-1]
  4. ist $() \in$ Res[n] than "F ist unerfüllbar" else "F ist erfüllbar"

- Bemerkungen

  1. F enthält 2n Literale: $Res^*(F) \in \Theta 2^{2^n}$
  2. Aufwand zum Erfüllbarkeitstest: wieder bei $\frac{m}{n}$ (Klauseldichte) 4,3 hoch (Phasensprung)

6 Unendliche Formelmengen / Endlichkeitssatz

Vor: höchstens abzählbar unendlich vielen atomaren Formeln

- Eine Menge M von Formeln ist erfüllbar genau dann wenn jede der endlichen Teilmengen von M erfüllbar ist

  Beweis:

  - $\mathcal{A}$ Modell für M $\Rightarrow$ dann erfüllt $\mathcal{A}$ auch jeden endl. Teilmenge
  - für jede endliche Teilmenge existiert Modell
    Sei M_n Menge aller Formeln aus M, die höchstsn $A_1, A_2, ..., A_n$ atomare Formeln enthalten
    $\Rightarrow$ für jedes M_n gibt es endl. Menge von Formeln $\{F_1, ..., F_k\}$, sodass jede Formel aus M_n semantisch äquivalent ist zu einer dieser Formeln
    $\Rightarrow$ Jedes Modell für M_n ist auch Modell für $\{F_1, ..., F_k\}$ und umgekehrt
    Kernproblem: aus Modellen für M_i ein Modell für M (Konstruktion iterativ)
    Ausdünnen der Indexmange I (bleibt unendlich)
    iterativ konstruierte Belegung ist Modell für F