Logik für Informatiker
Logic for Computer Scientists

Till Mossakowski

Wintersemester 2014/15

Klausur

- Über die Zulassung zur Klausur entscheiden die Übungsleiter, alle Teilnehmer erhalten nach dem Semester eine E-Mail mit dem Entscheid an die E-Mail-Adresse, über die sich zur Übung eingeschrieben wurde.
- Die Klausur ist am Montag, dem 16. Februar 2015, von 13:30 bis 15:30 Uhr.
  Die Plätze sind spätestens 15 min vor Beginn einzunehmen.
  Es sind keine Hilfsmittel zugelassen, Papier und Schreibutensilien sind mitzubringen.
  Die Räume sind folgende.
  - G26-H1:
    Benotete Leistung (Prüfung) für INF, IngIF, WIF.
  - G29-307:
    Benotete Leistung (Prüfung) für CV.
    Unbenotete Leistung (Schein) für CV, INF, IngIF, WIF.

PL1-Strukturen

Korrektheit von $\mathcal{F}$ für PL1

Theorem Wenn $\mathcal{T} \vdash S$ gilt, dann ist S eine PL1-Folgerung der Menge $\mathcal{T}$.

Beweis: Durch vollständige Induktion über die Anzahl der Beweisschritte.

Wir werden zeigen, dass jeder Satz, der in einem Schritt im Rahmen eines Beweises auftritt, eine PL1-Folgerung aus denjenigen Annahmen ist, die zu diesem Schritt in Kraft sind.

Eine Annahme ist in einem Schritt in Kraft, falls sie eine Annahme des aktuellen Unterbeweises ist oder eine vorherige Annahme eines Beweises einer höheren Ebene.

Für Regeln, für die Unterbeweise mit neuen Konstanten eingeführt werden müssen, brauchen wir die Erfüllungsinvarianz.

Vollständigkeit der Formaxiome

Die grundlegenden Formaxiome

1. $\neg\exists x\,(Cube(x) \wedge Tet(x))$
2. $\neg\exists x\,(Tet(x) \wedge Dodec(x))$
3. $\neg\exists x\,(Dodec(x) \wedge Cube(x))$
4. $\forall x\,(Tet(x) \vee Dodec(x) \vee Cube(x))$

SameShape – Einführungs- und Beseitigungsaxiome

1. $\forall x \forall y ((Cube(x) \wedge Cube(y)) \rightarrow SameShape(x, y))$
2. $\forall x \forall y ((Dodec(x) \wedge Dodec(y)) \rightarrow SameShape(x, y))$
3. $\forall x \forall y ((Tet(x) \wedge Tet(y)) \rightarrow SameShape(x, y))$

4. $\forall x \forall y ((SameShape(x, y) \wedge Cube(x)) \rightarrow Cube(y))$
5. $\forall x \forall y ((SameShape(x, y) \wedge Dodec(x)) \rightarrow Dodec(y))$
6. $\forall x \forall y ((SameShape(x, y) \wedge Tet(x)) \rightarrow Tet(y))$

Vollständigkeit der Formaxiome

Zwei Strukturen heißen isomorph, wenn es eine Bijektion zwischen ihren Gegenstandsbereichen gibt, die verträglich mit ihren Erweiterungen von Prädikatsymbolen und Interpretationen der Individuenkonstanten sind.

Wir betrachten die Sprache *Cube*, *Tet*, *Dodec*, *SameShape*.

Lemma Für jede Struktur, die die Formaxiome erfüllt, gibt es eine isomorphe Tarski's-World-Struktur.

Theorem Es sei S ein Satz.
Wenn S eine Tarski's-World-Folgerung von $\mathcal{T}$ ist, dann ist S auch eine PL1-Folgerung von $\mathcal{T}$, erweitert um die Formaxiome.

PL1-Resolution

PL1-Resolution

- Verallgemeinerung der aussagenlogischen Resolution für die Logik erster Stufe (PL1).
- Ein Beweissystem, das eine effiziente Implementierung erlaubt.
- Viele automatische PL1-Beweiser basieren auf der Resolution, z. B. SPASS, Prover9, Vampire.
- Auch interaktive Beweiser für Logiken höherer Stufe basieren auf der Resolution, z. B. Isabelle, HOL, HOL-light.

Erfüllbarkeit und logische Folgerung

Logische Folgerungen können auf die (Un-)Erfüllbarkeit zurückgeführt werden:

Die logische Folgerung $\mathcal{T} \models S$ gilt

genau dann, wenn

$\mathcal{T} \cup \{\neg S\}$ unerfüllbar ist.

Man beachte: Der Gegenstand der Resolution ist die Erfüllbarkeit.

Skolemisierung

Der Satz

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

ist logisch äquivalent zum Satz in der Logik zweiter Stufe

$$\exists f\, \forall x\, Neighbor(x, f(x))$$

In der Logik erster Stufe haben wir die Skolemsche Normalform

$$\forall x\, Neighbor(x, f(x))$$

Theorem über die Skolemsche Normalform

Theorem

Der Satz $S \equiv \forall x \, \exists y \, P(x, y)$ ist genau dann erfüllbar, wenn seine Skolemisierung $\forall x \, P(x, f(x))$ erfüllbar ist.

Jede Struktur, die die Skolemisierung erfüllt, erfüllt auch den Satz S.

Und jede Struktur, die den Satz S erfüllt, kann in eine Struktur überführt werden, welche die Skolemisierung erfüllt. Dazu interpretieren wir das Funktionssymbol f durch eine Funktion, welche jedem Objekt b im Gegenstandsbereich ein Objekt c zuweist, so dass sie die Wff $P(x, y)$ erfüllen.

Unifikation von Termen

$$\{P(f(a)), \forall x \neg P(f(g(x)))\}$$

ist erfüllbar, aber

$$\{P(f(g(a))), \forall x \neg P(f(x))\}$$

nicht. Das kann man durch Unifikation sehen.

Die Terme t_1 und t_2 sind unifizierbar, wenn es möglich ist, einige oder alle Variablen in t_1 und t_2 so durch Terme zu ersetzen, dass die aus der Substitution resultierenden Terme syntaktisch identisch sind.

Beispiel

$$f(g(z), x), \quad f(y, x), \quad f(y, h(a))$$

sind unifizierbar durch die Substitution
mit $h(a)$ für x und $g(z)$ für y.

Wiederholung Pränexe Form: Regeln für Konjunktion und Disjunktion

$$\forall x\, Q \wedge P \rightsquigarrow \forall x\, (Q \wedge P) \qquad \exists x\, Q \wedge P \rightsquigarrow \exists x\, (Q \wedge P)$$

$$P \wedge \forall x\, Q \rightsquigarrow \forall x\, (P \wedge Q) \qquad P \wedge \exists x\, Q \rightsquigarrow \exists x\, (P \wedge Q)$$

$$\forall x\, Q \vee P \rightsquigarrow \forall x\, (Q \vee P) \qquad \exists x\, Q \vee P \rightsquigarrow \exists x\, (Q \vee P)$$

$$P \vee \forall x\, Q \rightsquigarrow \forall x\, (P \vee Q) \qquad P \vee \exists x\, Q \rightsquigarrow \exists x\, (P \vee Q)$$

Man beachte, dass x keine freie Variable in P sein darf.

Falls x eine freie Variable in P ist, können wir die Bedingung erreichen, indem wir folgende Regel anwenden.

$\forall x\, Q \rightsquigarrow \forall y\, Q[y/x]$

Dabei bedeutet $Q[y/x]$, dass in Q alle freien Vorkommen von x durch y ersetzt werden.

Wiederholung Pränexe Form: Regeln für Negation, Implikation, Äquivalenz

$\neg\forall x\, P \rightsquigarrow \exists x\, \neg P$ $\qquad$ $\neg\exists x\, P \rightsquigarrow \forall x\, \neg P$

$\forall x\, Q \to P \rightsquigarrow \exists x\, (Q \to P)$ $\qquad$ $\exists x\, Q \to P \rightsquigarrow \forall x\, (Q \to P)$

$P \to \forall x\, Q \rightsquigarrow \forall x\, (P \to Q)$ $\qquad$ $P \to \exists x\, Q \rightsquigarrow \exists x\, (P \to Q)$

$P \leftrightarrow Q \rightsquigarrow (P \to Q) \wedge (Q \to P)$

Man beachte, dass in der zweiten und dritten Zeile x keine freie Variable in P sein darf.

Alpha-Umbenennung (Ersetzen gebundener Variablen)

Der Algorithmus zur Herstellung der Pränexen Form erwartet, dass alle Variablen in der Formel veschieden sind.
Das kann durch α-Umbenennung erreicht werden.

$$\forall x\, P(x) \rightsquigarrow \forall y\, P(y)$$

$$\exists x\, P(x) \rightsquigarrow \exists y\, P(y)$$

Resolution für PL1

Wie zeigt man, dass eine Menge $\mathcal{T}$ von Sätzen unerfüllbar ist?

1. Bringe jeden Satz S aus $\mathcal{T}$ in pränexe Form, wie etwa
$$\forall x_1\, \exists y_1\, \forall x_2\, \exists y_2\, \ldots\, P(x_1, y_1, x_2, y_2, \ldots)$$
2. Jeder der resultierenden Sätze wird skolemisiert, wie etwa
$$\forall x_1\, \forall x_2\, \ldots\, P(x_1, f_1(x_1), x_2, f_2(x_1, x_2), \ldots)$$
wobei verschiedene Skolemfunktionssymbole für verschiedene Sätze verwendet werden.
3. Bringe jede quantorenfreie Matrix P in KNF, wie etwa
$$P_1 \wedge P_2 \wedge \cdots \wedge P_n$$
wobei jedes P_i eine Disjunktion von Literalen ist.
4. Distributiere die universellen Quantoren in jedem Satz über die Konjunktionen und tilge die Konjunktionszeichen:
$$\forall x_1\, \forall x_2\, \ldots\, P_i$$

Resolution für PL1 (Fortsetzung)

5. Tausche die gebundenen Variablen in jedem der resultierenden Sätze so aus, dass keine Variable in zwei von diesen Sätzen vorkommt.
6. Wandele jeden der resultierenden Sätze in eine Menge von Literalen um, durch Tilgen der universellen Quantoren und Disjunktionszeichen. Auf diese Weise gelangen wir zu einer Menge von Resolutionsklauseln.
7. Wende die Resolution und Unifikation auf diese Menge von Klauseln an:

$$\frac{\{C_1, \ldots, C_m\},\ \{\neg D_1, \ldots, D_n\}}{\{C_2\theta, \ldots C_m\theta, D_2\theta, \ldots, D_n\theta\}}$$

falls $C_1\theta = D_1\theta$ (θ ist ein Unifikator von C_1 und D_1).

Beispiel I

Ist das folgende Argument gültig?

$$\begin{array}{|l} \forall x\,(\mathrm{Cube}(x) \vee \mathrm{Tet}(x)) \\ \exists x\,\neg\mathrm{Cube}(x) \\ \hline \exists x\,\mathrm{Tet}(x) \end{array}$$

Umformuliert: Ist die folgende Satzmenge unerfüllbar?

$$\begin{array}{l} \forall x\,(Cube(x) \vee Tet(x)) \\ \exists x\,\neg Cube(x) \\ \neg\exists x\; Tet(x) \end{array}$$

Schritt 1: Pränexe Normalform

$$\forall x\,(Cube(x) \vee Tet(x))$$
$$\exists x\,\neg Cube(x)$$
$$\forall x\,\neg Tet(x)$$

Schritt 2: Skolemisierung

$$\forall x\,(Cube(x) \vee Tet(x))$$
$$\neg Cube(c)$$
$$\forall x\,\neg Tet(x)$$

Da vor dem Existenzquantor kein Allquantor steht, benötigen wir ein 0-stelliges Funktionssymbol, d. h. eine Individuenkonstante c.

Schritt 3: In konjunktiver Normalform sind die Sätze bereits.

Schritt 4: Weglassen der Konjunktionen: ebenfalls nichts zu tun.

Schritt 5: Umbenennung gebundener Variablen

$$\forall x\,(Cube(x) \vee Tet(x))$$
$$\neg Cube(c)$$
$$\forall y\,\neg Tet(y)$$

Schritt 6: Tilgen der Allquantoren und Konjunktionen, und Schritt 7: Anwenden der Resolution

1. $\{Cube(x),\ Tet(x)\}$
2. $\{\neg Cube(c)\}$
3. $\{\neg Tet(y)\}$
4. $\{Tet(c)\}$ 1,2 mit c für x
5. $\square$ 3,4 mit c für y

Beispiel II

Ist das folgende Argument gültig?

$$\begin{array}{|l} \forall x\,(\mathrm{Cube}(x) \rightarrow \exists y\,\mathrm{BackOf}(y,x)) \\ \forall x\,\forall y\,(\mathrm{BackOf}(x,y) \rightarrow \mathrm{Large}(y)) \\ \hline \forall x\,(\mathrm{Cube}(x) \rightarrow \mathrm{Large}(x)) \end{array}$$

Umformuliert: Ist die folgende Satzmenge unerfüllbar?

$$\begin{array}{l} \forall x\,(\mathit{Cube}(x) \rightarrow \exists y\,\mathit{BackOf}(y,x)) \\ \forall x\,\forall y\,(\mathit{BackOf}(x,y) \rightarrow \mathit{Large}(y)) \\ \neg\forall x\,(\mathit{Cube}(x) \rightarrow \mathit{Large}(x)) \end{array}$$

Schritt 1: Pränexe Normalform

$$\forall x\, \exists y\, (Cube(x) \rightarrow BackOf(y, x))$$
$$\forall x\, \forall y\, (BackOf(x, y) \rightarrow Large(y))$$
$$\exists x\, \neg(Cube(x) \rightarrow Large(x))$$

Schritt 2: Skolemisierung

$$\forall x\,(Cube(x) \rightarrow BackOf(f(x), x))$$
$$\forall x\,\forall y\,(BackOf(x, y) \rightarrow Large(y))$$
$$\neg(Cube(c) \rightarrow Large(c))$$

Da vor dem ersten Existenzquantor ein Allquantor steht, benötigen wir hier ein 1-stelliges Funktionssymbol f, für den anderen Existenzquantor eine Individuenkonstante c.

Schritt 3: Konjunktive Normalform

$$\forall x\,(\neg Cube(x) \vee BackOf(f(x), x))$$
$$\forall x\,\forall y\,(\neg BackOf(x, y) \vee Large(y))$$
$$Cube(c) \wedge \neg Large(c)$$

Schritt 4: Weglassen der Konjunktionen

$$\forall x\,(\neg Cube(x) \vee BackOf(f(x), x))$$
$$\forall x\,\forall y\,(\neg BackOf(x, y) \vee Large(y))$$
$$Cube(c)$$
$$\neg Large(c)$$

Schritt 5: Umbenennung gebundener Variablen

$$\forall z\,(\neg Cube(z) \vee BackOf(f(z), z))$$
$$\forall x\,\forall y\,(\neg BackOf(x, y) \vee Large(y))$$
$$Cube(c)$$
$$\neg Large(c)$$

Schritt 6: Tilgen der Allquantoren und Konjunktionen, und Schritt 7: Anwenden der Resolution

1. $\{\neg Cube(z),\ BackOf(f(z), z)\}$
2. $\{\neg BackOf(x, y),\ Large(y)\}$
3. $\{Cube(c)\}$
4. $\{\neg Large(c)\}$

5. $\{BackOf(f(c), c)\}$ 1,3 mit c für z
6. $\{Large(c)\}$ 2,5 mit c für y, $f(c)$ für x
7. $\square$ 4,6

Beispiel III

Ist das folgende Argument gültig?

$$\begin{array}{|l} \forall x\,(P(x,b) \vee Q(x)) \\ \forall y\,(\neg P(f(y),b) \vee Q(y)) \\ \hline \forall y\,(Q(y) \vee Q(f(y)) \end{array}$$

Umformuliert: Ist die folgende Satzmenge unerfüllbar?

$$\begin{array}{l} \forall x\,(P(x,b) \vee Q(x)) \\ \forall y\,(\neg P(f(y),b) \vee Q(y)) \\ \neg\forall y\,(Q(y) \vee Q(f(y)) \end{array}$$

Schritt 1: Pränexe Normalform

$$\forall x\,(P(x,b) \vee Q(x))$$
$$\forall y\,(\neg P(f(y),b) \vee Q(y))$$
$$\exists y\,\neg(Q(y) \vee Q(f(y))$$

Schritt 2: Skolemisierung

$$\forall x\,(P(x,b) \vee Q(x))$$
$$\forall y\,(\neg P(f(y),b) \vee Q(y))$$
$$\neg(Q(c) \vee Q(f(c))$$

Da vor dem Existenzquantor kein Allquantor steht, benötigen wir ein 0-stelliges Funktionssymbol, d. h. eine Individuenkonstante c.

Schritt 3: Konjunktive Normalform

$$\forall x\,(P(x,b) \vee Q(x))$$
$$\forall y\,(\neg P(f(y),b) \vee Q(y))$$
$$\neg Q(c) \wedge \neg Q(f(c))$$

Schritt 4: Weglassen der Konjunktionen

$$\forall x\,(P(x,b) \vee Q(x))$$
$$\forall y\,(\neg P(f(y),b) \vee Q(y))$$
$$\neg Q(c)$$
$$\neg Q(f(c))$$

Schritt 5: Umbenennung gebundener Variablen: nichts zu tun.

Schritt 6: Tilgen der Allquantoren und Disjunktionen, und Schritt 7: Anwenden der Resolution

1. $\{P(x,b),\ Q(x)\}$
2. $\{\neg P(f(y),b),\ Q(y)\}$
3. $\{\neg Q(c)\}$
4. $\{\neg Q(f(c))\}$

5. $\{Q(y), Q(f(y))\}$ 1,2 mit $f(y)$ für x
6. $\{Q(f(c))\}$ 3,5 mit c für y
7. $\square$ 4,6

Beipiel IV

Ist das folgende Argument gültig?

Von

> *"Alle bewundern jemanden, der sie bewundert, wenn sie nicht Meier bewundern."*

folgt

> *"Es gibt Personen, die sich gegenseitig bewundern, von welchen zumindest einer Meier bewundert."*

Formalisierung

$$\begin{array}{|l} \forall x\,[\neg A(x,m) \rightarrow \exists y(A(x,y) \wedge A(y,x))] \\ \vdash \\ \exists x\,\exists y\,[A(x,m) \wedge A(x,y) \wedge A(y,x)] \end{array}$$

Umformuliert: Ist die folgende Satzmenge unerfüllbar?

$$\forall x\,[\neg A(x,m) \rightarrow \exists y\,(A(x,y) \wedge A(y,x))]$$
$$\neg\exists x\,\exists y\,[A(x,m) \wedge A(x,y) \wedge A(y,x)]$$

Schritt 1: Pränexe Normalform

$$\forall x\, \exists y\, [\neg A(x,m) \rightarrow (A(x,y) \wedge A(y,x))]$$
$$\forall x\, \forall y\, \neg[A(x,m) \wedge A(x,y) \wedge A(y,x)]$$

Schritt 2: Skolemisierung

$$\forall x\, [\neg A(x,m) \rightarrow (A(x,f(x)) \wedge A(f(x),x))]$$
$$\forall x\, \forall y\, \neg[A(x,m) \wedge A(x,y) \wedge A(y,x)]$$

Schritt 3: Konjunktive Normalform

$$\forall x\, [(A(x,m) \vee A(x,f(x))) \wedge (A(x,m) \vee A(f(x),x))]$$
$$\forall x\, \forall y\, [\neg A(x,m) \vee \neg A(x,y) \vee \neg A(y,x)]$$

Schritt 4: Weglassen der Konjunktionen

$$\forall x\,(A(x,m) \vee A(x,f(x)))$$
$$\forall x\,(A(x,m) \vee A(f(x),x))$$
$$\forall x\,\forall y\,[\neg A(x,m) \vee \neg A(x,y) \vee \neg A(y,x)]$$

Schritt 5: Umbenennung gebundener Variablen

$$\forall x\,(A(x,m) \vee A(x,f(x)))$$
$$\forall y\,(A(y,m) \vee A(f(y),y))$$
$$\forall z\,\forall w\,[\neg A(z,m) \vee \neg A(z,w) \vee \neg A(w,z)]$$

Schritt 6: Tilgen der Allquantoren und Disjunktionen, und Schritt 7: Anwenden der Resolution

1. $\{A(x, m), A(x, f(x))\}$
2. $\{A(y, m), A(f(y), y)\}$
3. $\{\neg A(z, m), \neg A(z, w), \neg A(w, z)\}$
4. ... [Der Rest ist Übungsaufgabe]

Die Ableitungsprozedur FO Con in Fitch

Die Ableitungsprozedur FO Con in Fitch
basiert auf automatischer Deduktion, ähnlich der Resolution.

Allerdings gilt:

Das Problem "Ist ein gegebenes PL1-Argument gültig?"
ist nicht entscheidbar (Church).

Folglich liefert die Ableitungsprozedur FO Con in Fitch bei manchen Eingaben kein Ergebnis.

Ausblick

Jenseits der Aussagenlogik

- Mehrsortige Logik

  Von Variablen, Konstanten, Prädikate und Funktionen gibt es veschiedene Typen

  Zum Beispiel: $\forall n\colon Nat\ \forall l\colon List\ head(cons(n,l)) = n$

- Logiken mit partiellen Funktionen

  $D(f(x))$ ("$f(x)$ ist definiert")

- Logiken höherer Stufe

  $\forall f\colon s \to t \ldots,\ \ \forall p\colon Pred(t) \ldots$

$$\begin{aligned}
&\forall u\, \forall v\, (Path(u,v) \leftrightarrow \\
&\quad \forall R\, \{[\forall x\, \forall y\, \forall z\, (R(x,y) \wedge R(y,z) \to R(x,z)) \\
&\qquad\qquad \wedge \forall x\, \forall y\, (DirectWay(x,y) \to R(x,y))] \\
&\qquad\qquad\qquad\qquad\qquad\qquad \to R(u,v)\})
\end{aligned}$$

Modallogik

- Modallogik

  $\Box P$ ("Es ist notwendig, dass P")

  $\Diamond P$ ("Es ist möglich, dass P")

  Weitere Lesarten von $\Box P$:

  Es sollte sein, dass P
  Es ist bekannt, dass P
  Es ist beweisbar, dass P
  Immer P (temporale Logik)

Temporale Logik

- Temporale Logik

  $\Box P$ ("immer in der Zukunft ist P wahr"),

  $\Diamond P$ ("irgendwann einmal in der Zukunft ist P wahr"), and

  $\bigcirc P$ ("im nächsten Schritt ist P wahr")

  Beispiel: $\Box bank_account > 0$ (sehr unrealistisch)

Weitere Modelle der Modal- und temporalen Logik

- Temporale Logik der Aktionen (TLA)

  $\Box[state' = f(state)]_{state}$

  "Entweder ändert sich in der Zukunft der Zustand 'state' nie oder der nächste Zustand ist $f(state)$".

- Dynamische Logik

  $[p]P$

  "Nach jeder Abarbeitung des Programms p ist P wahr"

  $<p>P$

  "nach einer Abarbeitung des Programms p ist P wahr"

Exotische Modallogiken

- Agentenlogik, z. B. ATL:

  Die Agenten A and B haben die Möglichkeit zu einem Telefongespräch, falls sie kooperieren.

- Logik für Sicherheiten, z. B. ABLP:

  A steuert P

  "Agent A hat die Erlaubnis, die Aktion P auszuführen"

Logiken für Wissensrepräsentation / Semantic Web

- Beschreibungslogik, z. B. $\mathcal{ALC}$:

  $$Elephant \doteq Mammal \sqcap \exists bodypart.Trunk \sqcap \forall color.Grey$$

  steht für

  $$\forall x\,[Elephant(x) \leftrightarrow (Mammal(x) \wedge \exists y\,(bodypart(x,y) \wedge Trunk(y)) \wedge \forall z\,(color(x,z) \rightarrow Grey(z)))]$$

Mehrwertige Logiken

- Dreiwertige Logik

  Die Wahrheitswerte sind WAHR, FALSCH und UNDEFINIERT.

- Object Constraint Logic (OCL)

  für UML – 'unified modeling language' (vereinheitlichte Modellierungssprache)

```
-- Managers get a higher salary than employees
inv Branch2:
  self.employee->forall(e | e <> self.manager
    implies self.manager.salary > e.salary)
```

Mehrwertige Logiken (Fortsetzung)

- Fuzzy-Logik

  Die Wahrheitswerte liegen im Intervall $[0, 1]$ und korrespondieren zu verschiedenen Graden der Wahrheit.

  Zum Beipiel: Peter ist ziemlich groß, ist groß, ist sehr groß.

Noch exoterischere Logiken

- Parakonsistente Logiken

  für Datenbanken, lokale Inkonsistenz ist ok, aber sollte nicht zu globaler Inkonsistenz führen

- Nicht-monotone Logiken

  "neue Fakten machen vorherige Argumente ungültig", zum Beispiel
  $Bird(x) \vdash CanFly(x)$
  $\{Bird(x), Penguin(x)\} \vdash \neg CanFly(x)$

- Lineare Logik (Resourcen-gebundene Logik)

  $A \otimes A \vdash B$

  "wir können B beweisen, falls wir A zweimal nutzen dürfen"

Wozu braucht man so viele Logiken?

- Sie decken verschiedene Aspekte der komplexen Welt oder/und von Softwaresystemen ab.
- Eine "große" Logik, die alles abdeckt, wäre zu schwerfällig.
- Gute Nachricht: Die meisten der Logiken basieren auf der Aussagenlogik oder Logik erster Stufe (PL1).
- Die meisten der Logiken benutzen dieselben zentralen Begriffe (obwohl immer auf die spezielle Logik angepasst).
  - Erfüllung eines Satzes in einem Modell,
  - Logische Folgerung,
  - Gültigkeit, Erfüllbarkeit,
  - Beweissystem und dessen Korrektheit und Vollständigkeit.