Algorithmische Semantik und Modellgenerierung

Johannes Dellert

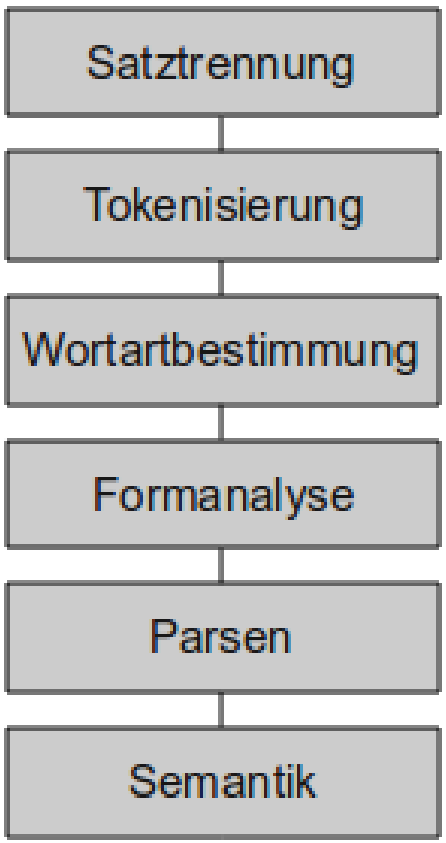
Universität Tübingen

06.07.2012

Übersicht

1. Algorithmische Semantik
   - Die Standard-Toolchain der Computerlinguistik
   - FOL als Formalismus zur Bedeutungsrepräsentation
   - Übersetzung syntaktischer Strukturen in FOL
   - Semantische Fragestellungen
   - Anwendungen der Algorithmischen Semantik
2. Modellgenerierung
3. Linguistisch motivierte Heuristiken

Die Standard-Toolchain der Computerlinguistik



Beispiel

"Der Tisch steht am Fenster." $\Rightarrow$
(der,Tisch,steht,an,dem,Fenster) $\Rightarrow$
(der_D,Tisch_N,steht_V,an_P,
dem_D,Fenster_N) $\Rightarrow$
(der_D[nom,sing,masc],
Tisch_N[nom,sing,masc],
stehen_V[3rd,sing,pres,indicative],...) $\Rightarrow$
(S (DP ((D (der))(NP (N Tisch))))
(VP (V steht) (PP ((P an)
(DP ((D (dem))(NP (N Fenster)))))))) $\Rightarrow$
$\exists x \exists y (tisch(x) \wedge \forall z (tisch(z) \rightarrow x = z)$
$\wedge\ fenster(y) \wedge \forall z (fenster(z) \rightarrow y = z)$
$\wedge\ stehen(x) \wedge an(x, z))$

FOL zur Beschreibung von Satzbedeutungen

Motivation aus der Geschichte der Logik

- bis ins 19. Jahrhundert (Boole) wurde Logik in natürlicher Sprache betrieben, ohne dass man sich formaler Systeme bewusst war
- beinahe alle klassischen Erkenntnisse wie z.B. die Syllogismen sind trotzdem ohne weiteres in FOL formalisierbar

Motivation aus heutiger Sicht

- linguistisch sinnvolle Modellierung braucht sowohl Entitäten (z.B. Personen, Ereignisse und mögliche Situationen) als auch Relationen (z.B. Eigenschaften, temporale Zusammenhänge, Kausalstruktur)
- Modelltheorie und Inferenz sind sehr gut erforscht
- interessiert man sich besonders für Phänomene wie Modalitäten und zeitliche Strukturen, so nutzt man gerne auch Modallogiken und Temporallogiken; allerdings kann man viele dieser Logiken auch recht gut in FOL “emulieren”

Typische Probleme prädikatenlogischer Repräsentation

Problem 1: Natürlichsprachliche Quantoren

- NL weist eine Vielzahl von **quantorenähnlichen Konstrukten** auf: der, ein, zwei, drei, ..., kein, manche, wenige, viele, alle, jeder ...
- die meisten davon lassen sich in FOL ausdrücken
- manche aber auch nicht: "die meisten" erfordert Logik höherer Stufe, weil wir über die Kardinalität von Mengen reden müssen, z.B. $[\![\text{die meisten arbeiten}]\!] = |\{x|\mathit{arbeiten}(x)\}| \geq |\{x|\neg\mathit{arbeiten}(x)\}|$

Problem 2: Funktionen auf Prädikaten?

- **Adverbien** modifizieren die Bedeutung von Adjektiven und Verben: sehr, ziemlich, kaum, ..., wahrscheinlich, offenbar, ..., hinauf, ...
- natürlich wäre z.B. $[\![\text{Klaus ist sehr mutig}]\!] = \text{sehr}(\text{mutig}(\text{klaus}))$
  $\Rightarrow$ Funktion von Relationen in Relationen, kein Standard-FOL!

Warum trotzdem "nur" Prädikatenlogik?

Theoretische Semantik

- wir erstreben eine elegante Modellierung von Erkenntnissen darüber, wie die Semantik natürlicher Sprache funktioniert
- wir kümmern uns nicht um die Stärke der Logiken, die wir benutzen

Algorithmische Semantik

- wir wollen mit den semantischen Repräsentationen Folgerungsbeziehungen ausrechnen (z.B. um sicherzugehen, dass eine theoretische Modellierung die richtigen Vorhersagen trifft)
- für Logiken höherer Stufe gibt es aber fast keine und nur sehr experimentelle Implementierungen von Inferenzalgorithmen
- FOL ist eigentlich zu schwach, aber gerade noch handhabbar $\Rightarrow$ notwendiger Kompromiss aus Ausdruckskraft und Handhabbarkeit

Übersetzung in FOL: Grundprinzipien

Kompositionale Semantik

Kompositionalität: Die Bedeutung eines Ausdrucks ergibt sich in funktionaler Abhängigkeit aus den Bedeutungen der Teilausdrücke.

- selbstverständlich für Informatik, und insbesondere Logik mit ihren induktiven Definitionen und Induktionsbeweisen über Formelstruktur
- in der Linguistik allerdings heftig umstritten, aus gutem Grund: wie ergibt sich $[\![\text{ins Gras beißen}]\!]$ aus $[\![\text{ins}]\!]$, $[\![\text{Gras}]\!]$ und $[\![\text{beißen}]\!]$?
- für die Berechnung semantischer Repräsentationen müssen wir einen gewissen Grad an Kompositionalität voraussetzen

Lexikalisierte Semantik

Lexikalismus: "Alle Information steckt in den Wörtern."

- Lexikoneinträge enthalten alle Information darüber, wie sich ein Wort syntaktisch mit anderen Wörtern kombinieren lässt, und wie sich deren Bedeutung dadurch ändert

Lambda-Kalkül als Glue Language

Lexikalische Semantik von Wörtern

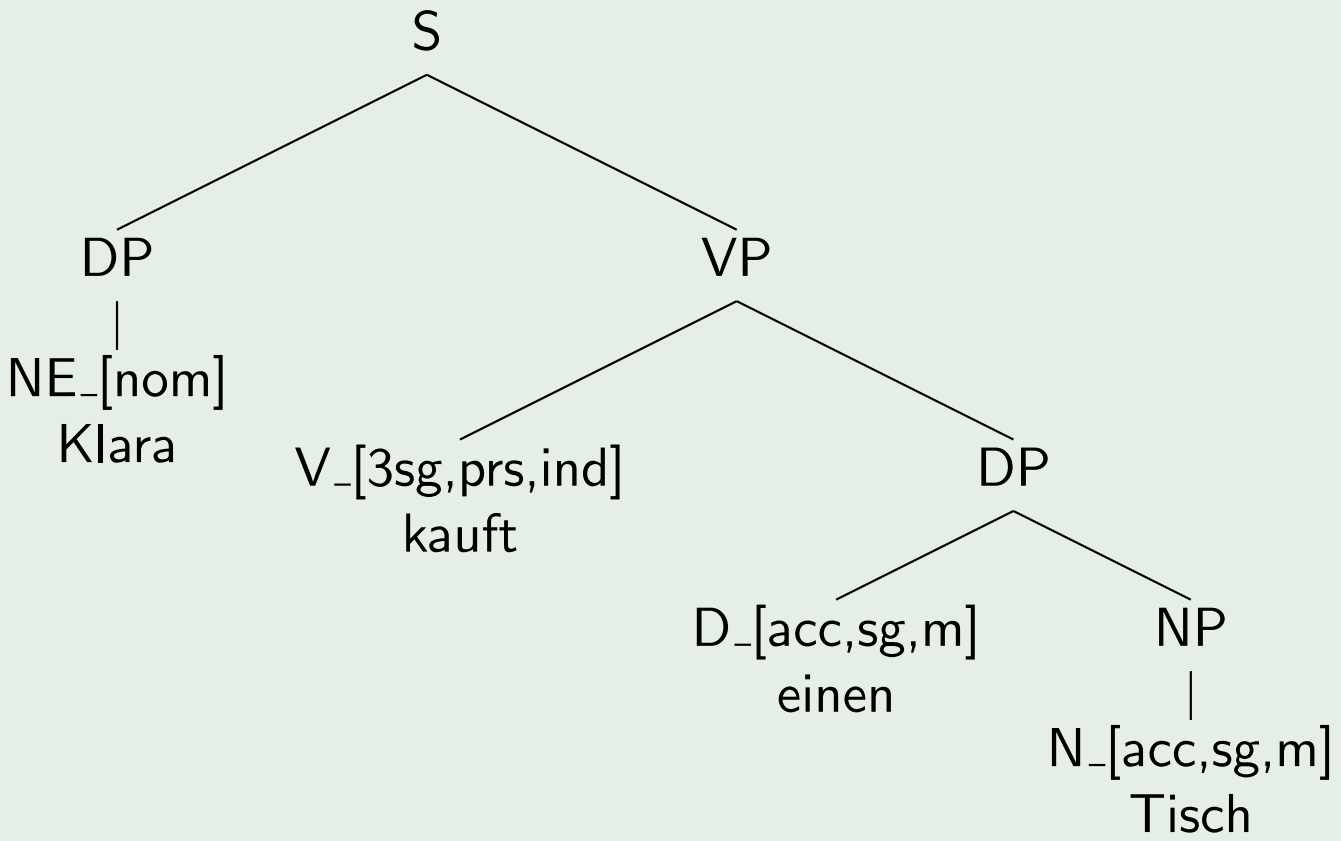
- Konzepte sind Funktionen von Individuen in Wahrheitswerte, z.B. $[\![\text{Tisch}]\!] = \mathit{tisch} : D \to \{\mathit{true}, \mathit{false}\}$, in λ-Notation $\lambda x.\mathit{tisch}(x)$
- Eigennamen sind Konstanten: $[\![\text{Klara}]\!] :=$ klara, mit Denotat aus D
- Verben sind Relationen oder Funktionen von Tupeln aus Individuen in Wahrheitswerte: $[\![\text{liebt}]\!] = \mathit{lieben} : D \times D \to \{\mathit{true}, \mathit{false}\}$, in λ-Notation $\lambda y.\lambda x.\mathit{lieben}(x, y)$

Kompositionale Semantik von Konstituenten

- wir errechnen die Semantik von Satzteilen kompositional aus der Semantik ihrer Teile mittels **Funktionalapplikation** (β-Konversion)
- Beispiel: $[\![\text{Klara liebt Stephan}]\!] = [\![\text{liebt Stephan}]\!]([\![\text{Klara}]\!])$
  $= [\![\text{liebt}]\!]([\![\text{Stephan}]\!])([\![\text{Klara}]\!])$
  $= [\lambda y.\lambda x.\mathit{lieben}(x, y)](\mathit{stephan})(\mathit{klara})$
  $= [\lambda x.\mathit{lieben}(x, \mathit{stephan})](\mathit{klara}) = \mathit{lieben}(\mathit{klara}, \mathit{stephan})$

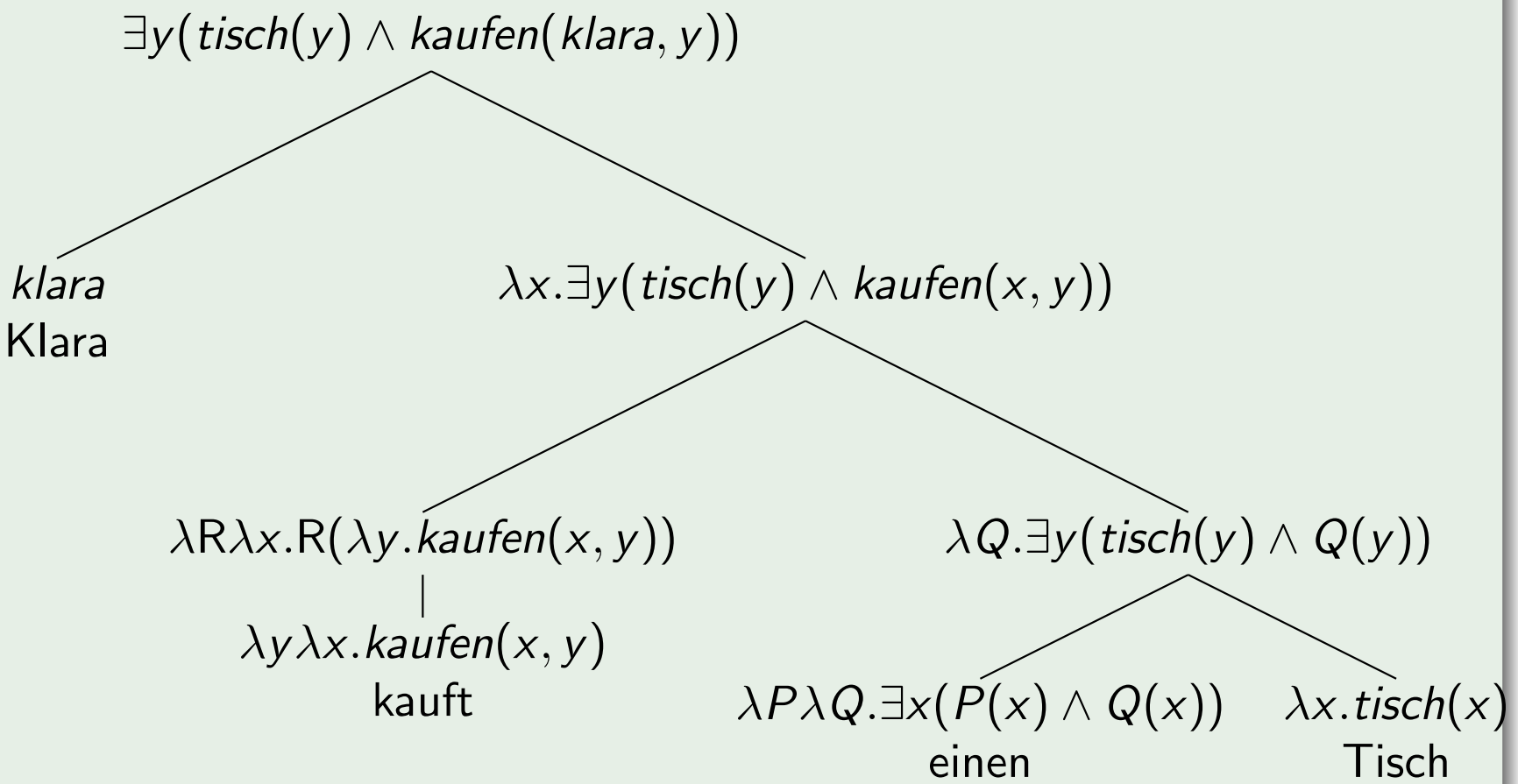
Beispiel: Übersetzung eines Satzes in FOL

Syntaktische Analyse



Beispiel: Übersetzung eines Satzes in FOL

Errechnung der semantischen Repräsentation



Die drei Grundfragen der Algorithmischen Semantik

Query Task (= Model Checking)

Gegeben das Modell M einer Situation und die semantische Repräsentation ϕ eines Satzes oder Diskurses, gilt ϕ in der Situation M oder nicht?

Consistency Checking Task (= Erfüllbarkeit)

Enthält die semantische Repräsentation ϕ eines Satzes oder Diskurses einen Widerspruch, oder ist sie in sich konsistent, ist also eine Situation M denkbar, in der ϕ gilt?

Informativity Checking Task (= Nicht-Allgemeingültigkeit)

Gegeben einen Diskurskontext Γ, enthält die semantische Repräsentation ϕ des nächsten Satzes neue Information, gilt also nicht $\Gamma \models \phi$?

Anwendungen der Algorithmischen Semantik

Question Answering

Beantwortung natürlichsprachlicher Fragen zu einem gegebenen Modell, zum Beispiel für eine neue Generation von Expertensystemen

Knowledge Base Extraction

kompakte Repräsentationen von Sachverhalten aus Texten (z.B. der Wikipedia) extrahieren und als Axiome oder Modelle vorhalten

Recognizing Textual Entailment

Entscheidung, ob eine Aussage implizit in einem Text enthalten ist

Consistency Checking

für technische Dokumentation oder Maschinenübersetzungen,
in ferner Zukunft vielleicht sogar für Gesetzestexte

Literatur: Algorithmische Semantik

In Prolog:

Patrick Blackburn & Johan Bos: **Representation and Inference for Natural Language** - A First Course in Computational Semantics. CSLI Publications, 2005.

- beschränkt auf Prädikatenlogik & Unterspezifikation
- Schwerpunkt auf Inferenz, Ergebnis ist kleines Dialogsystem

In Haskell:

Jan van Eijck & Christina Unger: **Computational Semantics with Functional Programming**. Cambridge University Press, 2010.

- Schwerpunkt auf Übersetzung, auch in Modallogiken
- nur Model Checking, sonst kein Automatisches Beweisen

Übersicht

1 Algorithmische Semantik

2 Modellgenerierung
- Was ist Modellgenerierung?
- Möglichkeiten der Implementierung
- Rolle in der Linguistik

3 Linguistisch motivierte Heuristiken

Modellgenerierung: Grundlagen

Definition

Sei ϕ eine FOL-Formel. Dann ist **Modellgenerierung** das Problem, eine Interpretation I zu konstruieren mit $\models_I \phi$.

Wichtige

- ein solches Modell ist ein konstruktiver Erfüllbarkeitsbeweis
- FOL-Erfüllbarkeit ist aber nicht einmal semi-entscheidbar
  $\Rightarrow$ es gibt keine vollständigen Verfahren!
- Vollständigkeit für endliche Erfüllbarkeit ist aber möglich (Enumeration immer größerer Herbrand-Strukturen + Model Checking, dann liefert Satz von Herbrand Vollständigkeit)

Modellgenerierung: The Big Picture

Zusammenhang mit Theorembeweisen

- **Theorembeweiser** können nur zeigen, dass eine Formel in allen Strukturen oder in keiner Struktur erfüllt ist
- **Modellgeneratoren** finden Beispiele oder Gegenbeispiele und zeigen damit Erfüllbarkeit oder Nicht-Allgemeingültigkeit

	Konsistenz (= Erfüllbarkeit)	Informativität (= Ungültigkeit)
positiver Beweis (**Modellgenerierung**)	$\exists I : \models_I \varphi$ $\exists I : \models_I \varphi$	$\exists I : \models_I \neg\varphi$ $\neg\forall I : \models_I \varphi$
negativer Beweis (**Theorembeweisen**)	$\neg\exists I : \models_I \varphi$ $\forall I : \models_I \neg\varphi$	$\forall I : \models_I \varphi$ $\forall I : \models_I \varphi$

Ein erster anschaulicher Ansatz

Modellgenerierung mit semantischen Tableaus

$$\exists x \exists y(rabbit(x) \wedge carrot(y) \wedge eat(x, y))$$
$$\exists y(rabbit(c_1) \wedge carrot(y) \wedge eat(c_1, y))$$
$$rabbit(c_1) \wedge carrot(c_2) \wedge eat(c_1, c_2)$$
$$\boxed{rabbit(c_1)}$$
$$carrot(c_2) \wedge eat(c_1, c_2)$$
$$\boxed{carrot(c_2)}$$
$$\boxed{eat(c_1, c_2)}$$

Zwei erfolgreiche Ansätze zur Implementierung

Ansatz 1: Reduktion auf SAT (MACE-style)

- suche inkrementell endliche Modelle der Größe m
- Quantoren lassen sich nun ersetzen durch Konjunktionen oder Disjunktionen jeweils über das ganze endliche Universum
- eine AL-Variable für jede Prädikatsymbol-Argument-Kombination
- alle verbleibenden komplexen Strukturen (wie Ungleichungen zwischen Variablen) herunterbrechen auf einfache Literale
- zusätzliche Klauseln erzwingen funktionale Eigenschaften
- SAT-Solver liefert Menge wahrer Relationstupel (= Modell)

Ansatz 2: Direkte Lösung als Constraint-Problem (SEM-style)

- fülle Zellen der Funktionstabellen durch Constraintpropagation
- priorisiere Zellen mit starken Constraints $\Rightarrow$ Suchraumreduktion
- Vermeidung isomorpher Modelle durch Least Number Heuristic

Literatur: Implementierung von Modellgeneratoren

Quellen zu Ansatz 1 (MACE-style)

- W. McCune: **Automatic Proofs and Counterexamples for Some Ortholattice Identities**. *Information Processing Letters 65*. (1998)
- K Claessen & N. Sörensson: **New Techniques that Improve MACE-style Model Finding**. In *Model Computation - Principles, Algorithms, Applications*. (2003) (beschreibt das **Paradox**-System)

Quellen zu Ansatz 2 (SEM-style)

- H. Zhang & J. Zhang: **Generating Models by SEM**. CADE 1996.
- **MACE** selbst ist seit Version 4 SEM-style: http://www.mcs.anl.gov/research/projects/AR/mace4/

Weiterer guter Ansatz: Hyperresolution

- siehe z.B. Fermüller et al.: **Resolution Decision Procedures**, das ist Kapitel 25 im *Handbook of Automated Reasoning (2001)*

Linguistische Anwendungen für Theorembeweiser

Beweis der Allgemeingültigkeit (direkt):

$\forall x(witwe(x) \rightarrow frau(x)) \wedge witwe(y) \rightarrow frau(y)$ ✓

Beweis der Inkonsistenz (über Negation):

$\forall x(witwe(x) \rightarrow frau(x)) \wedge witwe(y) \wedge \neg frau(y) \Rightarrow$
$\exists x(witwe(x) \wedge \neg frau(x)) \vee \neg witwe(y) \vee frau(y)$ ✓

Beweis von Redundanz bzw. Entailment (über Implikation):

Text: $\forall x(witwe(x) \rightarrow frau(x)) \wedge witwe(y)$ Hypothese: $frau(y) \Rightarrow$
$\forall x(witwe(x) \rightarrow frau(x)) \wedge witwe(y) \rightarrow frau(y)$ ✓

Linguistische Anwendungen für Modellgeneratoren

Beweis der Konsistenz bzw. Erfüllbarkeit (direkt):

$\forall x(\mathit{witwe}(x) \rightarrow \mathit{frau}(x)) \wedge \mathit{frau}(y) \wedge \neg \mathit{witwe}(y)$:
erhalten z.B. Modell voller Frauen, von denen keine eine Witwe ist ✓

Finden von Gegenbeispielen (über Negation):

$\forall x(\mathit{frau}(x) \rightarrow \mathit{witwe}(x))$ negieren zu $\exists x(\mathit{frau}(x) \wedge \neg \mathit{witwe}(x))$,
jedes Modell davon ist ein **Gegenbeispiel** für die ursprüngliche Formel ✓

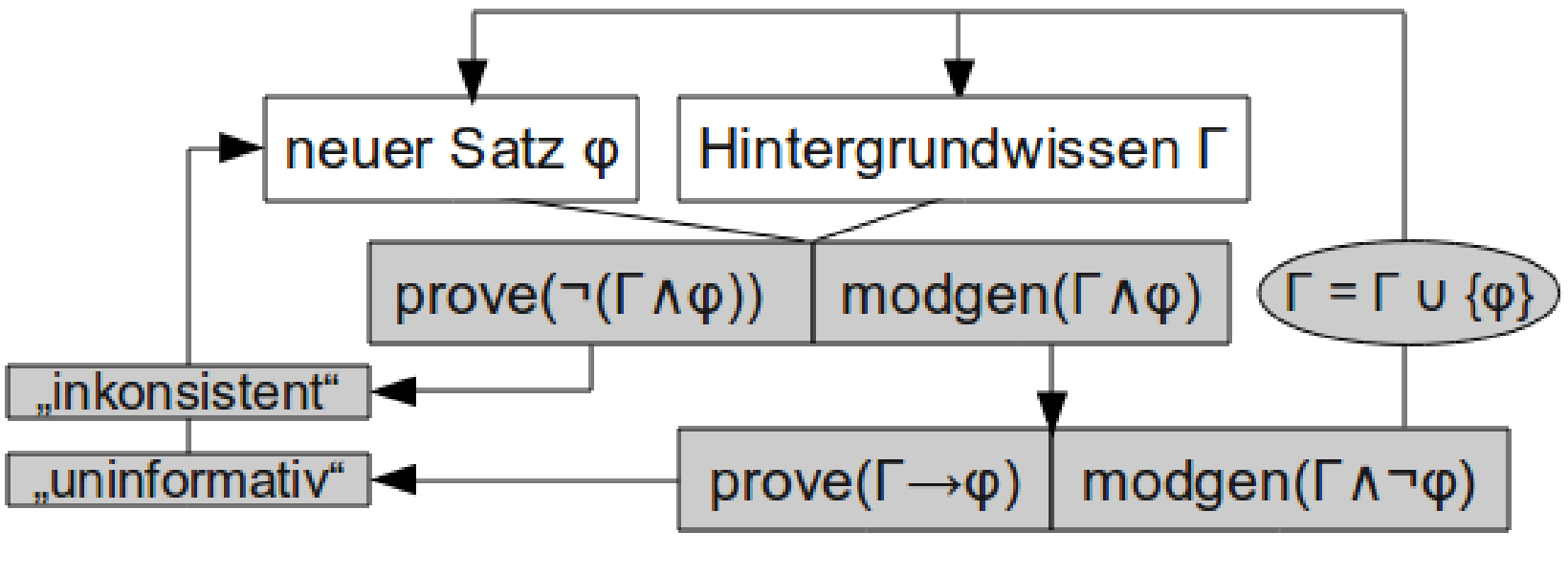
Beweis der Informativität (über Implikation):

Text: $\forall x(\mathit{witwe}(x) \rightarrow \mathit{frau}(x))$ Neues Wissen: $\exists y\ \mathit{witwe}(y)$
$\forall x(\mathit{witwe}(x) \rightarrow \mathit{frau}(x)) \wedge \forall y\ \neg \mathit{witwe}(y)$ hat Modell (siehe oben), damit muss das neue Wissen nicht der Fall sein und ist informativ ✓

Beispiel einer Architektur

Parallele Architektur für ein Dialogsystem (Blackburn & Bos 2006)

- paralleler Aufruf eines Theorembeweisers und eines Modellgenerators
- Konsistenz und Informativität lassen sich fast immer entscheiden



Modelle in der Linguistik

FOL-Strukturen als kognitive Modelle von Diskursen

- erfüllende Modelle zeigen, was der Fall sein sollte, damit ein Diskurs wahr wird; sie sind damit Bilder von beschriebenen Situationen
- sie können gesehen werden als die Annahmen eines Hörers; Modellgenerierung simuliert daher **Akkomodation** (Hörer trifft zusätzliche Annahmen, um einen Diskurs sinnvoll zu machen):
  Meine Schwester lernt gerade Esperanto. $\Rightarrow$
  Ich habe eine Schwester, die gerade Esperanto lernt.
- in Modellen sind auch **Anaphern** aufgelöst:
  Ein Haus steht dort. Es ist groß und schön. $\Rightarrow$
  Ein Haus steht dort. Das Haus ist groß und schön.

Literatur: Modellgenerierung in der Linguistik

Eine ausführliche Dissertation

Karsten Konrad: **Model Generation for Natural Language Interpretation and Analysis**. Springer (2004)

- experimentelles System KIMBA für Modellgenerierung in HOL
- zwei interessante Anwendungen in der Diskursinterpretation
- umfassendes Verzeichnis weiterer relevanter Literatur

Ein kurzes Papier mit Konzepten

Johan Bos: **Exploring Model Building for Natural Language Understanding**. Inference in Computational Semantics (ICoS) 2003.

- keine Betrachtung der technischen Aspekte
- gute linguistische Motivation, interessante Ideen für Anwendungen

Übersicht

1 Algorithmische Semantik

2 Modellgenerierung

3 Linguistisch motivierte Heuristiken
- Das Problem der Minimalität
- Eigene Arbeiten zu Heuristiken

Minimale und nicht-minimale Modelle

Minimale Modelle

- ein Modell $(\mathfrak{A}, \beta)$ einer Formel ϕ heißt **minimal**, wenn für jedes Modell $(\mathfrak{A}', \beta')$ von ϕ gilt $|D(\mathfrak{A})| \leq |D(\mathfrak{A}')|$
- übliche Modellgeneratoren gehen inkrementell über die Domänengrößen vor und produzieren daher immer minimale Modelle
- sehr praktisch in der mathematischen Anwendung: minimale Gegenbeispiele sind in der Regel die nützlichsten
- für die Linguistik ungünstig (siehe nächste Folie)

Nichtminimale Modelle

- die meisten Formeln besitzen Modelle beliebiger Größe
- müssen also die erwünschten Modelle anderweitig eingrenzen
- üblich ist das Erwzingen größerer Modelle durch zusätzliche Axiome
- nötig in der Linguistik (siehe nächste Folie)

Das Problem der Minimalität

Beispiel: Ein unbrauchbares minimales Modell

- Formel: $\exists x \exists y(mann(x) \wedge apfel(y) \wedge essen(x, y))$
- Modell: $(\{c_0\}, \{mann \mapsto \{c_0\}, apfel \mapsto \{c_0\}, essen \mapsto \{(c_0, c_0)\}\})$

Lösungansatz 1: Weltwissen

- erzwingen größere Modelle durch zusätzliche Axiome:
  $\forall x(apfel(x) \rightarrow obst(x))$, $\forall x(obst(x) \rightarrow pflanze(x))$,
  $\forall x(mann(x) \rightarrow mensch(x))$, $\forall x(mensch(x) \rightarrow \neg pflanze(x))$
- Problem 1: Wo kriegen wir diese Axiome her? (es gibt Ansätze)
- Problem 2: Die Theorien werden sehr groß.

Lösungsansatz 2: Heuristiken

- Heuristiken, die auf linguistisch sinnvollere Modelle hinarbeiten
- Modellgeneratoren, die solche Heuristiken unterstützen

Eigene Arbeiten

OletinMB: Ein erster Versuch

- **OletinMB** ist ein Modellgenerator, entstanden im Rahmen einer Hausarbeit, um das Potential nicht-minimaler Modellgenerierung für Zwecke der algorithmischen Semantik zu evaluieren (*oletin* ist ein finnischer Neologismus für *Annahmen treffendes Werkzeug*)
- basiert auf einer Implementierung semantischer Tableaus in Java, weil sich darauf leichter Heuristiken entwickeln lassen
- beherrscht die Input- und Outputformate von MACE, kann also MACE in bestehenden Architekturen ersetzen und so getestet werden
- zahlreiche Optimierungen sorgen für eine für ein Tableausystem respektable und sogar halbwegs brauchbare Performance

Eigene Arbeiten

PRIDAS: Prioritized Identity Assumptions

- die wichtigste Schnittstelle in OletinMB für die Kontrolle des Suchprozesses; die Priorität von Variableninstanziierungen in δ-Regeln kann frei definiert werden als Funktion interner und externer Parameter
- Beispiele für **interne Parameter**: Topologie des Tableaus, propagierte Constraints, Muster in Variablenvorkommen
- Beispiele für **externe Parameter**: beliebige Gewichtungsfunktionen (z.B. konzeptuelle Ähnlichkeit zwischen Prädikatensymbolen)
- die einfachsten denkbaren PRIDAS-Funktionen:
  - immer die Einführung einer neuen Konstante als letztes probieren: entspricht minimaler Modellgenerierung, Verhalten wie bei MACE
  - immer zuerst eine neue Konstante einführen: keinerlei Identitätsannahmen und größtmögliche Modelle, terminiert nicht bei verschachtelten Quantoren

Eigene Arbeiten

Ergebnisse mit OletinMB

- Weltwissen ist nicht mehr unabdingbar für plausible Modelle, in der Architektur von Blackburn & Bos sind die Ergebnisse auch ohne Weltwissensaxiome mit denen von MACE + Weltwissen vergleichbar
- für aus kurzen Zeitungstexten generierte Formeln (für RTE) funktioniert die Konsistenzprüfung in der Regel sehr gut, kann von der Geschwindigkeit her mit MACE mithalten (nicht mit Paradox)
- bei Informativität nicht konkurrenzfähig, in sehr dünn besetzten Suchräumen ist der Tableau-Ansatz viel zu ineffizient

Zukünftige Vorhaben

- PRIDAS-Heuristik in Paradox einbauen (oder selbst nachbauen)
- Suche nach formal fassbaren Eigenschaften linguistischer Modelle
- Nutzung in neuer Architektur, die Lernerantworten auf semantische Korrektheit prüft (Analyse nicht-standardkonformer Sprache)