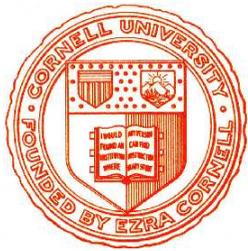


Inferenzmethoden

Einheit 18

Offene Themen



1. Effizientere allgemeine Beweissuche
 2. Höhere Formen des Schließens
 3. Anwendungen
-
4. Ausblick

VERDICHTUNG DER ALLGEMEINEN BEWEISSUCHE

- **Spezialanalyse ersetzt explizite Beweisschritte**
 - Verdichtete Form liefert dieselbe Information wie ausführlicher Beweis
- **Viele Beweise brauchen Klauselinstanzen**
 - Explizite Kopien der Klausel vergrößern Matrix und Suchraum
 - Alternative: Dynamische Aufspaltung oder Faktorisierung von Klauseln
- **Manche Beweise durchlaufen Zyklen**
 - Eine Kette von Argumenten wird mehrfach benutzt
 - Ähnlich zu Schleifen in Programmen muß ein “Ausstieg” vorhanden sein
 - Vermeide Kopien durch Bestimmung der “Zyklusmultiplizität”
 - Nichtterminierende (tautologische) Zyklen sind für Beweise unbrauchbar
- **Viele Anwendungen haben große Faktenmengen**
 - Ausgangsmatrix zu groß für effiziente Suche mit allgemeinen Verfahren
 - Schneller Zugriff erfordert Datenbanktechniken und Indizierung

- **Leistungssteigerung durch Verdichtung hat Grenzen**
 - Es gibt kaum syntaktische Kriterien zur Auswahl der besten Strategie
 - Aussagenlogisches Beweisen ist bereits \mathcal{NP} -vollständig
 - Kann man Nichtdeterminismus ausnutzen?
- **Verteile alternative Wege auf mehrere Prozessoren**
 - Verarbeite Alternativen (Konnektionsmengen, Substitutionen, ...) simultan
 - Lasse alternative Beweiser parallel laufen
 - Die erste korrekte Lösung reicht als Beweis
- **Aufteilung in unabhängige Teilprobleme schwierig**
 - Unabhängige Probleme können isoliert gelöst werden
 - Abhängigkeiten zwischen Problemen benötigen Kommunikation (z.B. bei simultaner Unifikation mehrerer Konnektionen)
- **Praktische Effizienzsteigerungen selten erreicht**
 - Nur bei massiver Parallelität (mehrere tausend Prozessoren)

HÖHERE FORMEN DES SCHLIESSENS

- **Analogieschließen**

- Übertragung einer Beweisidee auf neue, ähnliche Probleme
- Wichtig für Entdeckung neuer Beweise
- Konzept der “Ähnlichkeit” bisher schwer zu formulieren

- **Abstraktion**

- Anhebung der Ebene des Schließens von Literalen auf “Konzepte”
- Elimination überflüssiger formaler Details bei Beweissuche
- Wichtig für Verständnis und Planung von Beweisen
- Allgemeine Mechanismen bisher kaum verstanden

- **Beweisplanung**

- Schematisierte Form der Abstraktion
- Makro-Operatoren formulieren abstraktere Beweisschritte
- KI-Planer sucht Kette von Operatoren, die zum Ziel führen können
- Assoziierte Beweistaktiken prüfen, ob Beweis im Detail funktioniert
- Teilerfolge im Zusammenhang mit unverdichteten Beweisern

HÖHERE FORMEN: VERWENDUNG DER META-EBENE

- **Taktiken: Meta-Programmierung von Strategien**
 - Formuliere Anweisungen zur Manipulation der Objektsprache in Meta-Sprache
 - **Sicher**: Meta-Programme verwenden im Endeffekt nur Basisinferenzen
 - **Flexibel**: Benutzer können eigene deduktive Strategien ergänzen
- **Meta-Inferenz: Beweise Aussagen über Strategien**
 - Formuliere Aussagen über Zusammenhang zwischen Logik und Strategie
 - Welche Schlüsse sind in der Logik überhaupt erlaubt?
 - Für welche (syntaktischen) Formelklassen sind die Strategien erfolgreich
 - Formulierung in separater Meta-Logik (Logical Framework)
- **Reflektion: Meta-Inferenz in der Objektlogik**
 - Möglich, wenn Meta- und Objektsprache syntaktisch ähnlich
 - Erlaubt Beweise über Strategien innerhalb der Objektlogik
 - Nur eingeschränkt in Logik erster Stufe möglich
 - Mechanismen bisher zu wenig verstanden

- **Menschliches Schließen basiert auf Bedeutung**
 - Syntaktische Struktur wird nur als Beschreibungsform verstanden
 - Veranschaulichung (Betrachtung einfacher Modelle) liefert Beweisidee
 - Konkreter Beweis prüft, ob Beweisidee im Detail tragfähig ist oder verfeinert, modifiziert, verworfen werden muß
- **Unterstütze Beweissuche durch semantische Analyse**
 - Untersuche konkrete Modelle (Beispiele) einer Formel
 - Eliminiere Teilziele, die von diesen Beispielen widerlegt werden
 - Beende zugehörigen Pfad der Beweissuche als erfolglos
 - Setze zurück und beginne alternativen Suchpfad
- **Erfolgreich für Geometrische Beweisverfahren**
 - Basismethode von Gelernter (1959) formuliert für Horn Formeln
 - Erweiterbar auf volle Prädikatenlogik
 - Automatisierung schwer, da Erzeugung von Gegenbeispielen nötig

ANWENDUNGEN

- **Logische Programmiersprachen** (Prolog)
 - Hornklausellogik + Kontrollmechanismen (cut!)
 - Turingmächtig, effiziente “Compiler” vorhanden
- **Formale Mathematik**
 - Automatische Beweise für mathematische Teiltheorien
 - Interaktive Beweise für jegliche Form von Mathematik
- **Programmverifikation**
 - Höchste Stufe des Software-TÜV
 - Prüfe Korrektheit, Zuverlässigkeit, Sicherheit, ...
- **Programmsynthese**
 - Erzeugung korrekter Algorithmen aus formalen Spezifikationen
 - Programmverifikation während der Programmentwicklung
- **Wissensrepräsentation & Expertensysteme**

mehr als ein akademisches Spielzeug

- **Automatisierte Logik und Programmierung** (WS16/17)
 - Konstruktive Typentheorie (Logik höherer Stufe + ...)
 - Beweiseditoren, Taktiken, Entscheidungsprozeduren
 - Automatisierte Softwareentwicklung
- **Vorlesungen von Dr. E. Richter und Prof. T. Schaub**
 - Fuzzy Logik, Nicht-monotones Schließen, Kategorientheorie
 - Logikprogrammierung, KI, Wissensrepräsentation, ...
- **Mitarbeit in der Forschung** (Bachelor-/Masterarbeiten)
 - Problemsammlungen, Benchmarks von Beweisern \mapsto J. Otten
 - Implementierung effizienter Konnektionsbeweiser \mapsto J. Otten
 - Arithmetisches Beweisen \mapsto J. Otten, C. Kreitz
 - Anwendungen: Verifikation, Sicherheit, ... \mapsto C. Kreitz, E. Richter, N. Brede
 - ⋮