

# Logik für Informatiker: Sommersemester 2016

## • Aussagenlogik

- **Syntax** der Aussagenlogik: Formeln
- **Strukturelle Induktion** für Aussagenlogik
- **Semantik** der Aussagenlogik
  - \* Valuation (Wertebelegung, Modell)
  - \* Wahrheitstafel für die logischen Operatoren
  - \* Auswertung von Formeln in einer Interpretation
  - \* Modell einer Formel(menge)
  - \* Definitionen:
    - allgemeingültige Formeln (Tautologien);
    - erfüllbare/unerfüllbare Formeln
    - Folgerung und Äquivalenz
  - \* Unerfüllbarkeit und Allgemeingültigkeit:  
 $F$  ist allgemeingültig gdw.  $\neg F$  ist unerfüllbar.
  - \* Allgemeingültigkeit/Folgerung/Unerfüllbarkeit  
 $F \models G$  gdw.  $\models F \rightarrow G$  gdw.  $F \wedge \neg G$  unerfüllbar.  
 $F \equiv G$  gdw.  $\models F \leftrightarrow G$ .  
 $N \models G$  gdw.  $N \cup \{\neg G\}$  unerfüllbar.
- **Kalküle** zur Überprüfung der Erfüllbarkeit (Teil 1):
  - \* 1. Wahrheitstafelmethode
  - \* 2. Äquivalenzumformung (Substitutionstheorem)
- **Normalformen:**
  - \* Definitionen: Literal, Klausel
  - \* Konjunktive Normalform (KNF); Disjunktive Normalform (DNF)
    - Herstellung aus der Wahrheitstafel
    - Herstellung durch Äquivalenzumformung
  - KNF: Mengenschreibweise; Subsumption
- **Erfüllbarkeitsproblem** (SAT-Problem)
  - \* Definition
  - \* **Theorem** (ohne Beweis): SAT ist ein NP-vollständiges Problem
  - \* Teilklassen des Erfüllbarkeitsproblems:
    - Erfüllbarkeit für Formeln in 3-KNF: NP-vollständig
    - Erfüllbarkeit für Formeln in 2-KNF: PTIME entscheidbar
    - Erfüllbarkeit für Formeln in DNF: PTIME entscheidbar
    - **Horn-Formeln**  
Theorem: Erfüllbarkeit von Horn-Formeln ist in quadratischer Zeit entscheidbar.  
Erfüllbarkeitstest für Horn-Formeln: Markierungsalgorithmus.
  - \* Allgemeingültigkeit, Erfüllbarkeit, Unerfüllbarkeit: Entscheidbar
- **Kalküle** zur Überprüfung der Erfüllbarkeit (Teil 2):
  - \* 3. Der aussagenlogische Resolutionkalkül (Mengennotation):  
Resolutionsregel; Korrektheit, Vollständigkeit
  - \* 4. Semantische Tableaux; Klauseltableaux (Korrektheit, Vollständigkeit)  
Einschränkungen des Suchraums (Regularität; Schwache Konnektionsbedingung (Connection calculus); Starke Konnektionsbedingung (Modellelimination); Strikte Tableaux)  
Ohne 1-Resolution (unvollständig!)

- **Prädikatenlogik**

- **Syntax** der Prädikatenlogik

- \* Terme, Atome, Formeln; Literale; Klauseln
  - **Strukturelle Induktion** – Terme
- \* Gebundene und freie Variablen
- \* Substitutionen
  - Substitution eines Termes für eine Variable;
  - Substitution allgemein
  - Anwendung einer Substitution
  - **Unifikation** (allgemeinste Unifikatoren; Algorithmus nach Martelli/Montanari zur Berechnung allgemeinsten Unifikatoren)

- **Semantik** der Prädikatenlogik

- \*  $\Sigma$ -Struktur ( $\Sigma$ -Interpretation,  $\Sigma$ -Modell)
- \* (Variablen-) Belegung (Valuation)
- \* Wert eines Terms in einer  $\Sigma$ -Struktur bzgl. einer Valuation
- \* Wahrheitswert einer Formel in einer  $\Sigma$ -Struktur bzgl. einer Valuation
- \* Gültigkeit und Erfüllbarkeit
- \* Folgerung und Äquivalenz
- \* Eigenschaften von Quantoren; Umbenennung von Variablen

- **Unentscheidbarkeit:** Es gibt  $\Sigma$ , so dass Gültigkeit( $F$ ) unentscheidbar (ohne Beweis)

- **Normalformen** (NNF, Bereinigung, Pränexnormalform); Skolemisierung; KNF

- **Herbrand-Interpretationen**

- **Kalküle** zur Überprüfung der Erfüllbarkeit:

- \* Prädikatenlogische Resolution:  
Resolutionsregel; Faktorisierung  
(Korrektheit, Vollständigkeit)
- \* Prädikatenlogische Tableaux; Tableaux mit freien Variablen  
Prädikatenlogische Klauseltableaux  
(Korrektheit, Vollständigkeit)

- **Prolog:** Grundlegendes Verständnis des Prinzip der Programmierung in Prolog (Prolog-Programme als Datenbasis, deren Einträge Fakten und Regeln sind; Rekursion).

- Insbesondere: Grundlegendes Verständnis vom Umgang mit Listen in Prolog und der Rekursion (über Listen) in Prolog.

**Bemerkungen:**

- Die Resultate der Beweise sollten bekannt sein (z.B. die Korrektheit und Vollständigkeit von Kalkülen, etc.).
- Die Details der Beweise werden in der Klausur nicht abgefragt.
- Bei konstruktiven Beweisen (z.B. Kodierung in 3-SAT, Umformung zu Pränexnormalform) wird das Vorgehen in dem Beweis später als “Algorithmus” verwendet. In solchen Fällen, sollten Sie im Stande sein, die Konstruktionen (bzw. Algorithmen) anzuwenden.