Variationen: andere Logiken — andere Spiele

am Beispiel zweier wichtiger (Familien von) Logiken in der Informatik

- **MSO, monadische Logik zweiter Stufe**
  Erweiterung von FO: Quantoren über Teilmengen
  $\rightarrow$ formale Sprachen, concurrency
- **ML, Modallogik**
  Fragment von FO: beschränkte Quantoren über Elemente
  $\rightarrow$ temporale Spezifikation, Wissensrepräsentation

hier: zugehörige Spiele und Beispiele für ihren Nutzen

MSO: monadische zweite Stufe

hier über Σ-Wortstrukturen, zu $S = \{<\} \cup \{P_a \colon a \in \Sigma\}$

Elementvariable: $x_1, x_2, \ldots$

Mengenvariable: $X_1, X_2, \ldots$ für Teilmengen der Trägermenge

zu Syntax und Semantik von MSO(S)

atomare Formeln: $x_i = x_j$, $x_i < x_j$, $P_a x_i$, $X_i x_j$
AL Junktoren $\wedge, \vee, \neg$ wie üblich
Quantifizierung über Elemente: $\forall x_i \varphi$, $\exists x_i \varphi$ wie in FO
Quantifizierung über Teilmengen: $\forall X_i \varphi$, $\exists X_i \varphi$

Beispiele: Ordnungen/Wörter ungerader Länge
allgemeiner: reguläre Sprachen
MSO-Kodierung von DFA/NFA

MSO-Spiel

Konfigurationen $(\mathcal{W}, \mathbf{Q}, \mathbf{m}; \mathcal{W}', \mathbf{Q}', \mathbf{m}')$
mit markierten Elementen $\mathbf{m}/\mathbf{m}'$ und Teilmengen $\mathbf{Q}/\mathbf{Q}'$

zwei Zugvarianten { weiteres Elementepaar markieren; weiteres Teilmengenpaar markieren

Ehrenfeucht-Fraïssé Satz für MSO:

II hat Gewinnstrategie in $G^q_{\mathsf{MSO}}(\mathcal{W}, \mathbf{Q}, \mathbf{m}; \mathcal{W}', \mathbf{Q}', \mathbf{m}')$
gdw. $\mathcal{W}, \mathbf{Q}, \mathbf{m} \equiv^{\mathsf{MSO}}_q \mathcal{W}', \mathbf{Q}', \mathbf{m}'$

auch im MSO-Spiel sind Gewinnstrategien
verträglich mit Konkatenation

daraus: **Satz von Büchi**

MSO-definierbare Eigenschaften von Σ-Wortstrukturen
entsprechen genau den regulären Σ-Sprachen

ML: Modallogik

hier über Σ-Transitionssystemen,
zu $S = \{E_a \colon a \in \Sigma\} \cup \{P_i \colon 1 \leqslant i \leqslant n\}$

Formeln von ML(S) sprechen über einzelne Zustände in
Σ-Transitionssystemen mit atomaren
Zustandseigenschaften $p_i \leftrightsquigarrow P_i$

Syntax von ML(S)

atomare Formeln: $\bot$, $\top$, p_i (wie AL_n)
AL Junktoren $\wedge, \vee, \neg$ wie üblich
modale Quantifizierung: $\Box_a \varphi$, $\Diamond_a \varphi$ für jedes $a \in \Sigma$

Semantik von ML(S) als Fragment von FO(S):

$$\Box_a \varphi(x) \equiv \forall y\Big(E_a xy \to \varphi(y)\Big) : \quad \forall y\Big((x \xrightarrow{a} y) \to \varphi(y)\Big)$$

$$\Diamond_a \varphi(x) \equiv \exists y\Big(E_a xy \wedge \varphi(y)\Big) : \quad \exists y\Big((x \xrightarrow{a} y) \wedge \varphi(y)\Big)$$

ML-Spiel: Bisimulation

Konfigurationen:
$(\mathcal{Q}, q; \mathcal{Q}', q')$ mit je *einem* markierten Zustand

Zugabtausch in einer Runde:
I bewegt Spielstein längs einer E_a-Kante in $\mathcal{Q}$ oder in $\mathcal{Q}'$
II antwortet in der anderen Struktur

von $(\mathcal{Q}, q; \mathcal{Q}', q')$ zu Nachfolgekonfiguration $(\mathcal{Q}, r; \mathcal{Q}', r')$
mit $(q, r) \in E_a^{\mathcal{Q}}$ und $(q', r') \in E_a^{\mathcal{Q}'}$

Gewinnbedingung:
I/II verlieren wenn sie nicht ziehen können
II verliert wenn sich aktuelle Positionen atomar unterscheiden

Ehrenfeucht-Fraïssé Satz für ML:

II hat Gewinnstrategie in $G^n_{\mathsf{ML}}(\mathcal{Q}, q; \mathcal{Q}', q')$
gdw. $\mathcal{Q}, q \equiv^{\mathsf{ML}}_n \mathcal{Q}', q'$

Modallogik und Bisimulation

Bisimulationsspiel:
unbeschränktes (unendliches) ML-Spiel $G^\infty_{\mathsf{ML}}(\mathcal{Q}, q; \mathcal{Q}', q')$

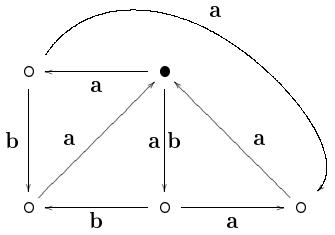
beschreibt vollständige Prozess-Äquivalenz / mehr als $\equiv^{\mathsf{ML}}$

zentrale Resultate:

- ML-Formeln unterscheiden nicht zwischen bisimulationsäquivalenten Situationen
- über endlich verzweigten Systemen fällt ML-Äquivalenz mit Bisimulationsäquivalenz zuammen (Hennessy-Milner)
- ML erfasst genau diejenigen FO-Eigenschaften, die Bisimulationsäquivalenz respektieren (van Benthem)

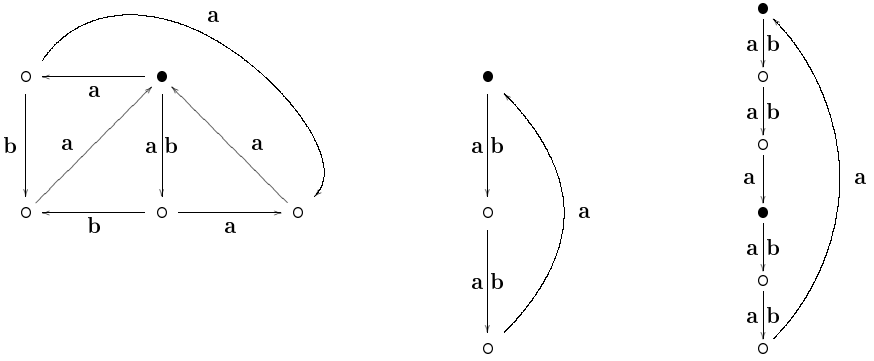
Modallogik und Bisimulation: Beispiel

in Transitionsssystem zu $\Sigma = \{a, b\}$:



- $\models \Diamond_a \Box_b \Diamond_a \Box_b \Diamond_a \Box_b \bot$

Bisimulationsäquivalenz: Beispiel



- $\models \Diamond_a \Box_b \Diamond_a \Box_b \Diamond_a \Box_b \bot$

ML-Formeln unterscheiden nicht zwischen bisimulationsäquivalenten Situationen

Wiederholung

– was Sie unbedingt wissen/können müssen

Formalismen
Syntax (AL, FO, Formeln, Terme, freie Variablen, etc.)

Normalformen (DNF, KNF, pränexe Normalform)
syntaktische Manipulationen: Substitution, Skolemisierung

Beweiskalküle (Resolutionsmethode, Sequenzenregeln)

Inhaltliches Verstehen
Semantik von Formeln, Modellbeziehung

Formeln lesen können, Terme/Formeln in Strukturen auswerten

Formalisierungen in AL und FO angeben

semantische Beziehungen: Äquivalenzen, Folgerungsbeziehung,
Erfüllbarkeitsäquivalenz
semantische Kriterien: Erfüllbarkeit, Allgemeingültigkeit,
Korrektheit, ...

Wiederholung

zentrale Begriffe/Konzepte inhaltlich beherrschen
im Kontext sinnvoll anwenden

zentrale Sätze und Resultate: kennen
interpretieren
anwenden

zentrale Sätze
Kompaktheit (Endlichkeitssätze),
Herbrand-Modelle,
Reduktionen von FO auf AL,
Korrektheits- und Vollständigkeitsaussagen zu Kalkülen
Entscheidbarkeit und Unentscheidbarkeit

Wiederholung: Beispiele

AL-Formeln auswerten (systematisch: Wahrheitstafel)
AL-Formeln auf Folgerung bzw. Äquivalenz untersuchen
natürlichsprachliche Bedingungen in AL formalisieren
Unerfüllbarkeit mittels Resolution nachweisen
Allgemeingültigkeit formal im Sequenzenkalkül nachweisen

Folgerungsbeziehungen reduzieren auf
Unerfüllbarkeit/Allgemeingültigkeit
Kompaktheitssatz anwenden
Kalküle rechtfertigen (z.B. Korrektheit von Regeln)

Wiederholung: Beispiele

Umgang mit Strukturen

auch spezielle Strukturen und Klassen wie z.B.
Graphen, Transitionssysteme, relationale DB-Strukturen,
Wortmodelle, linear-temporale Abfolgen, $\mathcal{N}$

Auswerten von Termen und Formeln in Strukturen

PNF, Skolemisieren, Substitutionen ausführen

Herbrandmodelle beschreiben/untersuchen

Unerfüllbarkeit durch Reduktion auf AL nachweisen

GI-Resolution und Sequenzenkalkül in Beispielen

etc.