1.1 Aussagenlogik:

Die Menge der aussagenlogischen Formeln ist induktiv definiert durch (1) 0, 1 ∈ AL (2) τ ∈ AL (3) ψ ∨ φ ∈ AL ⇒ ¬ψ, ¬φ, ψ ∧ φ, ψ ∨ φ, ψ ⇒ φ, φ ⇒ ψ ∈ AL (4) **induktives Beweisprinzip:** (1) Alle atomaren Formeln haben die Eigenschaft E (2) Haben ψ und φ ∈ AL die Eigenschaften E, so auch ¬ψ und (ψ ∨ φ) ... **Formeltiefe:** (1) d(ψ) := 0 für atomare ψ, (2) d(¬ψ) := d(ψ) + 1 und (3) d((ψ ∨ φ)) := 1 + max(d(ψ), d(φ)).

Koinzidenzlemma: Sei ψ ∈ AL eine Formel und seien I, I₀ zwei zu ψ passende Interpretationen, so dass I(X) = I₀(X) ∨ ¬X = τ(ψ). Dann ist I(ψ) = I₀(ψ). Eine Formel ψ ist genau dann erfüllbar, wenn X eine Tautologie ist. 0 : Λ 0 = 1 ; ∨ 0 = 0 **DNF:** V_{j=1}ⁿ Λ_{i=1}^m Y_{ij} **KNF:** Λ_{i=1}ⁿ V_{j=1}^m Y_{ij}

Hornformel: Eine al. Formel ist eine Hornformel ψ = Λ_{i=1}ⁿ V_{j=1}^m Y_{ij} in KNF, wobei jede Disjunktion U_{j=1}^m Y_{ij} höchstens ein positives Literal enthält. Anmerk. (1) ¬X₁ ∨ ... ∨ ¬X_k ∨ X = X₁ ∧ ... ∧ X_{k} → X Für k = 0 folgt 1 → X (2) ¬X₁ ∨ ... ∨ ¬X_k ∨ X₁ ∧ ... ∧ X_{k} → 0 **Erfüllbarkeits-Test:** Fange mit leerer Menge an. Nimm sukzessive alle X_i auf, die mit 1 interpretiert werden müssen. ... Ergebnis (falls erfüllbar) ist kleinstes Modell.}}

Semantische Folgerungsbeziehung: Ein Modell einer Formelmenge Φ ⊆ AL, ist eine Interpretation I, so dass I(φ) = 1 für alle φ ∈ Φ. Wir sagen φ ∈ Ψ gdw jede zu Φ ∪ {ψ} passende Interpretation, welche Modell von Φ ist, auch Modell von ψ ist.

Kompaktheitssatz: Sei Φ ⊆ AL, ψ ∈ AL. (i) φ ist erfüllbar ⇔ jede endliche Teilmenge ist erfüllbar (ii) φ ∈ Ψ ⇔ ∃Φ₀ ⊆ Φ, so dass φ₀ ∈ Ψ.

Zorn: Sei (A, <) eine nicht-leere partielle Ordnung in der jede Kette nach oben beschränkt ist. Dann besitzt A, < ein maximales Element.

König: Sei T ein endlich verzweigter Baum mit Wurzel w, in dem es beliebig lange endliche Wege gibt. Dann gibt es auch einen unendlichen Weg in T, der bei der Wurzel w beginnt.

Klauselmenge (KM): Klausel = endl. Menge von Literalen; □ = leere Klausel. ψ = Λ_{i=1}ⁿ V_{j=1}^m Y_{ij} und V_{j=1}^m Y_{ij} = C_i; endl. KM = K(ψ) := {C₁, ..., C_n}. Die leere KM ist erfüllbar. Wenn C ∈ K, dann ist K unerfüllbar. **Resolutionslemma:** Sei K eine Klauselmenge, C₁, C₂ ∈ K und C Resolvente von C₁ und C₂. Dann sind K und K ∪ {C} äquivalent. **Resolutionsatz:** Eine Klauselmenge K ist unerfüllbar gdw. Res^{*}(K). Res(K) = K ∪ {C : Resolvente zweier Klauseln aus K} / Res(0) = K / Resⁿ⁺¹(K) = Res(Resⁿ(K)) / Resⁿ(K) = ∪_{n∈N} Resⁿ(K)

1.2 Strukturen und Homomorphismen

Mächtigkeit: Jede abzählbare Menge ist entweder endlich oder gleichmächtig zu N. Keine Menge ist gleichmächtig zu Pot(A) := {B : B ⊆ A}

Strukturen: (Universum, Relationen, Funktionen) **Signatur:** τ := ∪ Rⁿ(τ) ∪ ∪ Fⁿ(τ)

Substruktur: Während beim Begriffspaar Substruktur/Erweiterung die Signatur fest bleibt und das Universum verändert wird, ist die beim Begriffspaar Redukt/Expansion, genau umgekehrt. Substrukturen müssen τ-abgeschlossen sein. Zu jeder nicht leeren τ-abgeschlossene Teilmenge A ⊆ B, gibt es genau eine Substruktur von B mit Träger A. (=einzudirekte Substr.)

Homomorphismen: (1) Für jedes Rel.symbol R ∈ Rⁿ(τ) und alle a₁, ..., a_n ∈ A gilt: (a₁, ..., a_n) ∈ R^A ⇒ (πa₁, ..., πa_n) ∈ R^B. (2) Für jedes Fkt.symbol f ∈ Fⁿ(τ) und alle a₁, ..., a_n ∈ A gilt: π f^A(a₁, ..., a_n) = f^B(πa₁, ..., πa_n). **starker ~:** (1') Für jedes Rel.symbol R ∈ Rⁿ(τ) und alle a ∈ Aⁿ gilt: a ∈ R^A ⇔ πa ∈ R^B. **Einbettung:** injektiver, starker Hom. **Isomorphismus:** bi-jektiver starker Hom. (A ≅ B). **Automorphismen:** π : A ≅ A

1.3 Syntax und Semantik der Prädikatenlogik

Signatur: τ, τ-Terme, τ-Formeln. **Alph(τ) := {Rel, Fktn, VAR = v₀, ..., v_n, =, ∧, ∨, ¬, ⇒, ⇔, ∃, ∀, (), ()}. τ-Terme := T(τ) := (1) VAR ⊆ T(τ) (2) wenn t₁, ..., t_n τ-Terme und f ∈ Fⁿ(τ), dann auch f t₁, ..., t_n. τ-Formeln := FO(τ) := (1) sind t₁, t₂ τ-Terme, dann ist t₁ = t₂ eine τ-Formel (2) sind t₁, ..., t_n Terme aus T(τ) und ist P ∈ ein n-stelliges Rel.symbol, dann ist P t₁, ..., t_n eine τ-Formel. (3) wenn ψ, dann auch ¬ψ. (4) wenn ψ und φ, dann auch (ψ ∧ φ), (ψ ∨ φ) und (ψ ⇒ φ). (5) wenn ψ und x ∈ VAR, dann auch ∃xψ und ∀xψ. **τ-Satz:** τ-Formeln ohne freie Variablen. **τ-Interpretation:** I = (A, τ) **Koinzidenzlemma:** Sei ψ ∈ FO(σ ∩ τ), (A, β) eine τ-Interpretation und (A', β') eine τ-Interpretation, so dass folgendes gilt: (i) A und A' haben dasselbe (σ ∩ τ)-Redukt: A[σ ∩ τ] = A'[σ ∩ τ]. (ii) Frei(ψ) ⊆ Def(β) ∩ Def(β') und β(x) = β'(x) für alle x ∈ Frei(ψ). Dann gilt: A ⊨ ψ[β] ⇔ A' ⊨ ψ[β'].**

Modellklasse von Φ: Mod(Φ) ... Axiomensystem. **semantische Folgerungsbeziehung:** Sei φ ∈ FO(τ) eine Formelmenge, ψ ∈ FO(τ) eine Formel. Wir sagen, ψ folgt aus φ (kurz φ ⊨ ψ) gdw, jede zu φ ∪ {ψ} passende Interpretation, welche Modell von φ ist, auch Modell von ψ ist. Wenn Φ = {φ} schreiben wir auch φ ⊨ ψ anstelle von {φ} ⊨ ψ. **Abschluss:** Sei ψ eine Formel mit freien Variablen x₁, ..., x_k. Dann nennen wir die Sätze ∃x₁, ..., ∃x_kψ und ∀x₁, ..., ∀x_kψ die existentiellen bzw. universellen ~ von ψ. **erfüllbar:** Formel ~ gdw, ihr existentieller Abschluss erfüllbar. **allgemeingültig:** Formel ~ gdw ihr universeller Abschluss allgemeingültig.

Substitutionslemma: Für jeden Term t ∈ T(τ), jede Formel ψ ∈ FO(τ), jede Substitution p : VAR → T(τ) und jede zu t [p] bzw. ψ [p] passende Interpretation I gilt: (i) [t] [p] = t [p] (ii) I ⊨ ψ [p] ⇔ I ∘ p ⊨ ψ. **Auswertungsspiel:** MC(I, ψ). Verifizierer V, Falsifizierer F. An Position (φ, ψ) geht Spiel abhängig von w wie folgt weiter: (für Literal) Spiel zuende. Falls A ⊨ φ [β] gewinnt V, sonst F. [θ ∨ η, β] V am Zug, entweder zu θ oder η ziehen. [θ ∧ η, β] F am Zug, entweder zu θ oder η ziehen. [∃xθ, β] V wählt ein a ∈ A und zieht zu θ, [β[x/a]]. [∀xθ, β] F wählt ein a ∈ A und zieht zu θ, [β[x/a]]. [V zieht bei ∃, V] [F zieht bei ∀, V]

1.4 Modallogik

ML: (mit Aktionen a ∈ A und atomaren Eigenschaften P_i für i ∈ I): (1) alle aussagenlog. Formeln mit AusVar P_i (2) ψ, φ ∈ ML ⇒ ψ ∧ φ, ψ ∨ φ, ψ ⇒ φ, φ ⇒ ψ ∈ ML dann auch (a)[a], wobei (a) für ∃ und [a] für ∀ steht > 1 es ex. Nachfolger, □ 0 es gibt keinen Nachfolger

Kripkestruktur: K = (V, {E_a}_{a∈A}, {P_i})_{i∈I}. ψ ist erfüllbar in K, wenn v in K ex., mit K, v ⊨ ψ. ψ, φ sind ψ ⇒ φ wenn [ψ]^K = [φ]^K für alle zu beiden passenden K.

Bisimulation: Seien K und K' zwei Kripkestrukturen. Eine Bisimulation zwischen K, K' ist eine Relation Z ⊆ V × V', so dass für alle (v, v') ∈ Z gilt: (1) v ∈ P_i, v' ∈ P'_i für alle i ∈ I (2) HIN: ∀ a ∈ A, w ∈ V mit v ^a w ex. ein w' ∈ V' mit v' ^a w' und (w, w') ∈ Z. HER: ∀ a ∈ A, w' ∈ V' mit v' ^a w' ex. ein w ∈ V mit v ^a w und (w, w') ∈ Z.

Bisimulationspiel: (1) Man spielt auf einem Pfad (2) Der Falsifizierer verliert, wenn er an einen Knoten gelangt, dem keine Transition ausgeht. Die Verifizierer verliert, wenn sie nicht mit der entsprechenden Aktion antworten kann (v ∈ P_i und v' ∉ P'_i). Sie gewinnt, wenn sie nie verliert, also wenn K, u ~ K', u'.

n-Bisimilarität: Das Spiel geht nur über n Züge. K, u ~_n K', u' Wenn K, u ~ K', u', dann ist auch K, u ~_n K', u'. Die Umkehrung gilt jedoch nicht. **Modaltiefe:** (1) md(ψ) = 0 für alle aussagenlogischen Formeln (2) md(¬ψ) = md(ψ) (3) md(ψ ∘ φ) = max(md(ψ), md(φ)) für alle ∘ ∈ {∧, ∨, ∧, →} (4) md((a)ψ) = md([a]ψ) = max(md(ψ), md(φ)) + 1

ML-Äquivalenz: (1) K, v ≡_M LK', v', wenn für alle ψ ∈ ML gilt: K, v ⊨ ψ, K', v' ⊨ ψ (2) K, v ≡_M LK', v', wenn für alle ψ ∈ ML md(ψ) ≤ n gilt: K, v ⊨ ψ, K', v' ⊨ ψ

Bisimulationsinvarianz: Modallogische Formeln können bisimulare Zustände nicht unterscheiden. (1) K, v ~ K', v' ⇒ K, v ≡_M LK', v' (2) K, v ~_n K', v' ⇒ K, v ≡_M LK', v' Wenn K, v ⊨ ψ und K, v ~ K', v' dann auch K', v' ⊨ ψ. Die Umkehrung gilt im Allgemeinen nicht. Sind K, K' endlich verzweigt,

dann: K, v ~ K', v', K, v ≡ MLK', v'. **Entscheidbarkeit:** Das Erfüllbarkeitsproblem für ML ist entscheidbar. **CTL:** (1) Alle aussagenlogischen Formel über {P_i : i ∈ I} gehören zu CTL. (2) CTL ist abgeschlossen unter ∧, ∨, →, ¬. (3) Wenn ψ, φ ∈ CTL, dann auch: EXψ, AXψ, E(ψ ∨ φ), A(ψ ∨ φ). **Semantik von CTL:** (1) ψ bedeutet: am nächsten gilt ψ (2) ψ ∨ φ bedeutet: es gilt solange ψ (auch am aktuellen Knoten) bis φ gilt (3) EXψ := ∃ψ (4) AXψ := □ψ (5) Fψ := (1U)ψ; irgendwann gilt ψ (6) Gψ := ¬F¬ψ; es gilt immer ψ

1.5 Definierbarkeit in der Prädikatenlogik

FO-axiomatisierbar: Eine Strukturklasse K heißt FO-ax'bar, wenn eine Satzmenge Φ ⊆ FO(τ) existiert, so dass K = Mod(Φ). Wenn Φ endlich für K, dann endlich axiomatisierbar.

Definierbarkeit (in einer Struktur): Sei ψ(x₁, ..., x_n) ∈ FO(τ) und A eine τ-Struktur. Dann definiert ψ in A die r-stellige Relation ψ^A := {(a₁, ..., a_n) : A ⊨ ψ(a₁, ..., a_n)} ⊆ A'. Eine Relation R ⊆ A' auf dem Universum einer τ-Struktur ist (elementar) definierbar in A, wenn R = ψ^A für eine Formel ψ ∈ FO(τ). Eine Funktion f : A' → A heißt elementar definierbar, wenn ihr Graph R_f elementar definierbar ist. FO-Formeln können nicht zwischen isomorphen Strukturen unterscheiden.

Isomorphielemma: Sei π : A ≅ B ein Isom. von τ-Strukturen. Dann gilt für alle ψ(x₁, ..., x_n) ∈ FO(τ) und alle a₁, ..., a_n ∈ A. A ⊨ ψ(a₁, ..., a_n) ⇔ B ⊨ (πa₁, ..., πa_n)

Theorie Th(=): Eine Th. ist eine erfüllbare Menge T ⊆ FO(τ) von Sätzen, die unter |= abgeschlossen ist, d.h. es gilt für alle t-Sätze ψ : T ⊨ ψ ⇒ ψ ∈ T. Eine Th. ist vollst., wenn für jeden Satz ψ ∈ FO(τ) entweder ψ ∈ T oder ¬ψ ∈ T. Th. einer festen Struktur (z.B. N) ist vollständige.

Quantorenrang qr(ψ): (1) = 0 für quantorenfreie ψ (2) qr(ψ) = qr(¬ψ) (3) qr(ψ ∘ φ) = max(qr(ψ), qr(φ)) mit ∘ ∈ {∧, ∨, ⇒, ⇔} (4) qr(∃xψ) = qr(∀xψ) = qr(ψ) + 1.

Ehrenfeucht-Fraisse (EF): Sei T endlich und relational, A, B τ-Strukturen. (1) Folgende Aussagen sind äquivalent: (i) A ≅ B (ii) Die Duplikatorin gewinnt das EF-Spiel G(A, B, B). (2) Für alle m ∈ N sind folgende Aussagen äquivalent: (i) A ≅_m B (ii) Die Duplikatorin gewinnt G_m(A, B)

axiomatisierbar: versuchen, Strukturen A ∈ K und B_m ∉ K zu finden, dann zeigen, dass die Duplikatorin G_m(A, B_m) gewinnt, dann kann A und B nicht durch FO-Sätze unterscheiden, also auch nicht endl. ax. werden.

1.6 Entscheidbarkeit

Sequenzkalkül (SK): Sei Φ ⊆ FO(τ) ein Menge von Sätzen. Ein Satz ψ ist ableitbar aus dem Axiomensystem Φ, (kurz Φ ⊢ ψ), wenn eine endliche Teilmenge Γ von Φ existiert, so dass die Sequenz Γ ⇒ ψ im SK ableitbar ist. Eine Sequenz Γ ⇒ ψ ist ableitbar aus Φ, wenn es eine ableitbare Sequenz Γ, Γ' ⇒ Δ gibt, mit Γ' ⊆ Φ.

Vollständigkeitsatz: Für jede Menge Φ ⊆ FO(τ) und jeden Satz ψ ∈ FO(τ) gilt: (1) φ ∈ Ψ ⇒ φ ⊢ ψ (2) φ erfüllbar ⇒ φ konsistent.

Löwenheim, Skolem: Jede erfüllbare, abzählbare Satzmenge hat ein abzählbares Modell.

Kompaktheitssatz der FO: Für jede Menge Φ ⊆ FO(τ) und jedes ψ ∈ FO(τ) (1) φ ∈ Ψ gdw. eine endl. Teilmenge Φ₀ ⊆ Φ existiert, so dass φ₀ ∈ Ψ (ii) φ ist erfüllbar gdw, jede endl. Teilmenge von Φ erfüllbar ist. FO(τ) = zu einer τ-Struktur passende FO-Formel.

Satz 6.15: Sei Φ ⊆ FO(τ) eine Satzmenge mit beliebig grossen endlichen Modellen (d.h. für jedes n ∈ N gibt es ein Modell A_n ⊨ Φ mit endlichem A_n und |A_n| > n). Dann hat Φ auch ein unendliches Modell. Die Klasse aller endlichen τ-Strukturen ist nicht axiomatisierbar in FO.

aufsteigender Satz v. Löwenheim-Skolem: Φ besitze ein unendliches Modell. Dann gibt es zu jeder Menge M ein Modell D ⊨ Φ über einem Universum D welches mindestens so mächtig wie M ist.

Satz 6.20: Sei A eine unendliche Struktur. Dann gibt es eine Struktur B mit A ≅ B, aber A' ≅ B'. Insbesondere ist die Isomorphieklasse {B : A ≅ B} von A nicht axiomatisierbar in FO.

1.7 SK(*) Wenn c in Γ, Δ, ψ(x) nicht vorkommt.

(¬)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma, \psi \Rightarrow \Delta}$	(⇒)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \psi}$
(∨)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \psi \vee \theta \Rightarrow \Delta}$	(⇒ ∨)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \psi \vee \theta}$
(∧)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \psi \wedge \theta \Rightarrow \Delta}$	(⇒ ∧)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \psi \wedge \theta}$
(⇒)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \psi \Rightarrow \theta \Rightarrow \Delta}$	(⇒ ⇒)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \psi \Rightarrow \theta}$
(=)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \psi \Rightarrow \theta}$	(⇒ =)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \psi \Rightarrow \theta}$
(S)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \psi \Rightarrow \theta \Rightarrow \Delta}$	(⇒ S)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \psi \Rightarrow \theta}$
(∃)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \exists x\psi \Rightarrow \Delta}$ (*)	(⇒ ∃)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \exists x\psi}$ (*)
(∀)	$\frac{\Gamma, \psi \Rightarrow \Delta}{\Gamma, \forall x\psi \Rightarrow \Delta}$	(⇒ ∀)	$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \forall x\psi}$ (*)

Eine Regel Γ' ⇒ Δ' ist unter Voraussetzung Γ ⇒ Δ korrekt, d.h. $\frac{\Gamma \Rightarrow \Delta, \Gamma' \Rightarrow \Delta'}{\Gamma \Rightarrow \Delta, \Delta'}$. Eine Sequenz Γ ⇒ Δ ist gültig, wenn es eine Interpretation gibt, die jede Formel auf der linken Seite und mind. eine auf der Rechten Seite wahrmacht. **1.8 Rechenregeln**

AL: (1) ¬(ψ ∧ φ) ⇔ (¬ψ ∨ ¬φ) (2) ¬(ψ ∨ φ) ⇔ (¬ψ ∧ ¬φ) (3) θ ∧ (ψ ∨ φ) ⇔ (θ ∧ ψ) ∨ (θ ∧ φ) (4) θ ∨ (ψ ∧ φ) ⇔ (θ ∨ ψ) ∧ (θ ∨ φ) **Normalformen:** (1) ∃x(ψ ∨ φ) ⇔ ∃xψ ∨ ∃xφ (2) ∀x(ψ ∧ φ) ⇔ ∀xψ ∧ ∀xφ (3) (6) falls x nicht frei in ψ vorkommt (3) ψ ∨ ∃xφ ⇔ ∃x(ψ ∨ φ) (4) ψ ∧ ∃xφ ⇔ ∃x(ψ ∧ φ) (5) ψ ∨ ∀xφ ⇔ ∀x(ψ ∨ φ) (6) ψ ∧ ∀xφ ⇔ ∀x(ψ ∧ φ) (7) ¬∃xψ ⇔ ∀x¬ψ (8) ¬∀xψ ⇔ ∃x¬ψ

2.1 Aussagenlogik

fullst. Vollst.: {∧, ∨, ¬}, {∧, ¬}, {∨, ¬}, {¬, ¬}, {¬, →}, {∧, ⊕, 1}

NICHT funkt. Vollst.: {∧, ∨, →}, {∧, ∨, 1, 0}

2.2 FO Arithmetik in FO: x Prim $\forall z(z|x \rightarrow (z = xvz = 1)) \wedge x \neq 1 \wedge x \leq y \Rightarrow \exists z(z \neq 0 \wedge (x+z=y)) \wedge x=0 \forall y(x+y=y) \wedge x=1 \forall y(x \cdot y = y) \wedge |x|y \exists z(x \cdot z = y) \wedge x \neq y \rightarrow \exists z \wedge x=2^n \forall y(y|pm \wedge y|x \rightarrow y=2) \wedge y=2 \exists z \exists z'(z < y \wedge z' < y) \wedge \exists z \exists z'(z < y \wedge z' < y \wedge z \neq z' \wedge z \neq z')$ **x, y gleiche Länge in 01** $\forall z(z=2^n \rightarrow (z > x > y))$

2.4 CTL EXψ ⇔ ∃ψ | AXψ ⇔ □ψ | U until | F finally | Fψ := (1U)ψ irgendwann gilt ψ (auf dem Pfad) | Gψ := ¬F¬ψ gilt immer ψ (auf dem Pfad) | AG¬(P ∧ Q) auf allen erreichbaren Zuständen schließen sich P und Q aus | AGψ auf allen erreichbaren Zuständen gilt ψ | AFψ auf allen Pfaden gilt irgendwann ψ | AGAFψ unendlich oft auf allen Pfaden ψ **Beispiele:** (1) AG([P → AX¬P] ∧ ¬(P → AXP)); auf allen Pfaden abwechselnd P und ¬P (2) AG(EXQ → P); vor jdm erreichbar. Q-Zustand liegt ein P-Zustand (3) AG(P → EFQ); von jdm erreichbar. P-Zustand wird ein Q-Zustand erreicht (4) EF(P ∧ EX(P ∧ EXP)); ∃ Pfad, auf dem 3 mal direkt hintereinander P gilt (5) AG(P → AGP) ⇔ AG(P → AXP); auf jedem Pfad gilt nur noch P, sobald ein P-Zustand erreicht wird (6) ¬EF(P ∧ ¬Q) ⇔ AG(P → Q); es wird nie ein Zustand erreicht, an dem P, aber ¬Q gilt (7) AG(P → AG ¬P); auf jedem Pfad, max ein Zustand an dem P gilt (8) ¬(E¬PQ); vor jedem erreichbar. Zustand, wo Q gilt, liegt ein Zustand wo P gilt

(9) AGEX1: nie ein Zustand erreichbar, der keinen Nachfolger hat **Folgende Formeln gehören nicht zu CTL:** FGP, A¬G¬P, F[PUQ], EF[QU], AEFP, AF([QU] ∧ (PUQ))

Außerdem gilt: ¬AFφ ⇔ EG¬φ | ¬EFφ ⇔ AG¬φ | ¬AXφ ⇔ EX¬φ | AGφ ⇔ φ ∧ AXAGφ | EGφ ⇔ φ ∧ EXEGφ | AFφ ⇔ φ ∧ AXAFφ | EFφ ⇔ φ ∧ EXEFφ | A[φ ∨ ψ] ⇔ ψ ∨ (φ ∧ AXA[φ ∨ ψ]) | E[φ ∨ ψ] ⇔ ψ ∨ (ψ ∧ EXE[φ ∨ ψ])

2.5 HowToZ

nicht endl. axiom. ⇒ EF: Wähle A ∈ Φ_K und B_m ∉ Φ_K (mit lediglich der m-fachen gewünschten Eigenschaft (von K)). Also wenn die Duplikatorin jedes Spiel G_m(A, B_m), ⇒ dass kein Satz ψ ∈ FO(τ) A und B_m trennen **Klasse nicht axiom. ⇒ KS:** Angenommen φ axiom. K. Bilde ψ := φ ∨ Gegenteil. Z.z.: jedes endliche ψ₀ ⊆ φ ∨ Gegenteil beschränkt bis n₀ ist erfüllbar. Nach KS wäre dann auch ψ erfüllbar, das geht aber nicht, da das Gegenteil auch drin ist.

Bisimulation: Stelle Tabelle mit Knotenmengen auf, die man in m Schritten unterscheiden kann (m=0, ...). Versuche diese durch ML-Formeln zu unterscheiden (z.B. A-Kanten-B-Kanten.a&B-Kanten) und damit aufzuspielen. **Gewinnstrategie:** Bei Formeln: (1) Feststellen, wer das Spiel gewinnt (2) mit → kennzeichnen, wie der GEWINNENDE Spieler zieht. **WICHTIG:** nur eine Strategie angeben!!!

Bei Spielgraphen: (1) Gewinnregionen „festlegen“ (2) Nur → da machen, wo der Spieler (in seiner Gewinnregion) ziehen kann **Resolution einer Folgerung:** φ ⊨ ψ ⇒ φ ⇒ ψ ⇒ φ ∨ ¬φ ∨ ψ ⇒ φ ∧ ¬ψ

Resolviere φ ∧ ¬ψ. Folgt □ so ist die Folgerung gültig, sonst nicht **Einheitsresolution:** Resolution einer Klausel (bel. Größe) mit einer Klausel, die nur ein Literal hat.

2.6 Axiomensysteme Formeln aus FO-Sätzen

Klasse der unendlichen Mengen: Φ_∞ = {φ_n : n ∈ N}, φ_n := ∃x₁ ... ∃x_n (Λ_{1 ≤ i < j ≤ n} x_i ≠ x_j)

Klasse der dichten Mengen: ∀x∀y (x < y ⇒ ∃z (x < z < y))

Klasse der dichten lin.Ord.: (1) lin.Ord. (2) dichte Menge

Klasse der lin.Ord.: φ < = {∀x¬x < x, ∀x∀y∀z (x < y ∧ y < z ⇒ x < z), ∀x∀y (x < y ∨ x = y ∨ y < x)}

Klasse der unendl. dichten lin.Ord.: Nicht def'bar, da jede abzählbare Satzmenge ein abzählbares Modell besitzt (LS)

Klasse der lin.Ord. mit größtem Element: (1) lin. Ord. (2) ∃x¬∃y(x < y)

Klasse der lin.Ord. über und unter jedem Element ⇒ volle EL: (1) lin.Ord. (2) ∀x∃y(x < y) (3) ∀x∃y(y < x)

Klasse der lin.Ord., wo nicht alle Intervalle dicht sind: (1) lin.Ord. (2) ∀x∀y∃z(x < y ⇒ x < z < y ≤ x ∧ y ≠ z) (x < z < y)

Klasse der disk.lin.Ord.: (1) lin.Ord. (2) ∀x(∃y x < y ∨ ∃y (x < y ∧ ∃z(x < z < y))) (3) ∀x(∃y x > y ∨ ∃y (x > y ∧ ∃z(x > z > y)))

Klasse der abzählbaren lin.Ord.: nicht axiomatisierbar nach aufsteigenden Satz von L-S, da eine Formelmenge, die das ax. würde mit einem unendlichen Modell auch ein überabzählbares großes Modell besitzen.

Klasse der überabzählbaren lin.Ord.: Diese Klasse ist nach Satz von LöwenheimSkolem nicht axiomatisierbar, da ein Axiomensystem Φ, welches diese Klasse axiomatisieren würde auch ein abzählbares Modell hätte.

Klasse der endl.lin.Ord.: ist nicht FO-ax. Annahme: Φ ist eine Menge von Sätzen mit A ⊨ Φ ⇔ A ist endl.lin.Ord.. Sei Ψ = Φ ∪ {φ_n : n ∈ N}. Beh.: Ψ erfüllbar. Sei P₀ ⊆ Ψ endl. H₀ ⊆ Ψ {φ₀, ..., φ_k} für ein k. Dann gilt, dass ((0, ..., k-1), c) ∈ P₀ also ist Ψ₀ erfüllbar. Nach KS ist damit auch Ψ erfüllbar, d.h. A ⊨ Ψ. Folglich gilt, dass A ⊨ Φ und A unendlich. **z**

Klasse aller unendlichen linearen Ordnung: nicht endlich axiomatisierbar (1) lineare Ordnung (2) φ_n n ∈ N

Klasse aller partiellen Ordnung, die jede endliche lineare Ordnung als Substruktur enthält: ist nicht endlich axiomatisierbar (1) ∀x¬(x < x) (2) ∀x∀y∀z (x < y ∨ x < z < y < z) (3) ∃x₁ ... ∃x_n (x₁ < ... < x_n)

Klasse der Gruppen: ∀x∀y∀z (x · y · z = x · (y · z)) (Assoziativität) und ∃e(∀x(x · e = e · x) ∧ ∀x∃y(x · y = e))

Klasse aller Strukturen mit Äquivalenz-Relation E:

{∀xExx, ∀x∀y(Exy ⇒ Eyx), ∀x∀y∀z(Exy ∧ Eyz ⇒ Exz)}

φ₂ := ∃x₁ ... ∃x_n (Λ_{1 ≤ i < j ≤ n} x_i ≠ x_j ∧ Λ_{1 ≤ i < j ≤ n} E x_i x_j ∧ 1 ∧ E x_n x₁)

Klasse aller Strukturen welche zu (Q, <) isom. sind: ist nicht FOax'bar

2.7 Normalformen AL / FO / ML (<=) NNF:

NNF: Neg. nur vor Literalen, kein →

PNF: keine Var kommt frei und gebunden vor, ∃ und ∀ nach vorne ziehen

SNF: alle ∃ durch entspr.-stellige Fkt. (Anzahl ∀ vor ∃, z.B. a → b(t) ersetzen

2.8 Logik-Übersicht AL < ML <= FO < FO

Erfüllbar-Prob.	Erreichbarkeit	EME/BM
FO	¬ formal. (in 4.5)	
ML	entsch. (4.24)	BM(4.19)
CTL	entsch. (in exp)	auch unendl. Pfade EME

2.9 Algorithmen

Erfüllbarkeitstest für Horn-Formeln (HF) [Markierungsalgorithmus]

Input: Eine aussagenlogische HF Ψ = Λ_i C_i