Aussagenlogik und Prädikatenlogik

Zusammenfassung

Fabian Damken

8. November 2023

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhaltsverzeichnis

1 Aussagenlogik

Die Aussagenlogik beschäftigt sich mit Grundlegenden Aussagen, in denen Fakten (Atome), denen ein Wahrheitswert zugeordnet werden kann, miteinander verknüpft werden.

1.1 Grundlegende Begriffe

Allgemeingültigkeit Eine FormEl φ ist allgemeingültig gdw. alle möglichen Interpretationen $\mathcal{J}$ die Formel φ erfüllen. Eine solche Formel wird auch Tautologie genannt.

Atome Eine Formel φ ist atomar (ist ein Atom) gdw. sie in der Form p oder der Form $\neg p$ vorliegt, wobei p eine eigenständige Variable darstellt. Diese werden auch Literale genannt.

Erfüllbarkeit Eine Interpretation $\mathcal{J}$ erfüllt eine Formel φ gdw. die Formel φ mit der Belegung $\mathcal{J}$ wahr wird. Diese Aussage ist äquivalent zu den folgenden Aussagen:

- φ ist wahr unter $\mathcal{J}$.
- $\mathcal{J}$ ist ein Modell von φ.

Formel Eine Formel ist eine Menge von Atomen, welche mit booleschen Junktoren und Klammern zu einer Aussage zusammengefasst werden. Formel werden häufg mit φ oder ψ bezeichnet.

Interpretation Eine Interpretation ist eine Belegung der Variablen in einer Formel mit Wahrheitswerten (wahr/falsch), sodass die Formel ausgewertet werden kann. Interpretationen werden häufig mit $\mathcal{J}$ bezeichnet.

Klausel Formeln in Klauselform ist eine Schreibweise für Formeln, in denen Atome ausschließlich mit Disjunktionen verknüft sind. Hierbei werden die Atome als Elemente einer Menge verstanden, was die Schreibweise ergibt.
Beispiel: $(a \vee b \vee \neg c)$ ist eine solche Formel, welche als Klauselform folgendermaßen dargestellt wird: $\{a, b, \neg c\}$.

Klauselmengen Eine Klauselmenge ist eine Menge von Formeln in Klauselform, welche bei der Umformung in einen logischen Ausdruck mit konjunktionen Verknüft werden.
Beispiel: $((a \vee b) \wedge (b \vee c) \wedge (a \vee \neg d \vee \neg e) \wedge d)$ ist eine Formel in konjunktiver Normalform, welche als Klauselform folgendermaßen dargestellt wird: $\{\{a, b\}, \{b, c\}, \{a, \neg d, \neg e\}, \{d\}\}$. Hierbei stellen die inneren Mengen die Klauseln dar und die äußere Menge die Klauselmenge.

Minimales Model Das minimale Modell $\mathcal{I}$ einer Formel φ ist das Modell mit der minimalen Anzahl an 1-Belegungen, um φ zu erfüllen.

Unerfüllbarkeit Eine Formel φ ist unerfüllbar gdw. die Formel φ unter allen möglichen Interpretationen $\mathcal{J}$ nicht wahr wird. Dies ist äquivalent dazu, dass das Negat der Formel φ (also $\neg\varphi$) allgemeingültig ist.

1.2 Notation

Sei $\mathcal{J}$ eine Interpretation, und seien φ, ψ Formeln.

$\models$ Das Symbol $\models$ kann folgende Dinge beschreiben:

- $\mathcal{J} \models \varphi$ bedeutet $\mathcal{J}$ ist ein Modell von φ
- $\varphi \models \psi$ bedeutet φ impliziert ψ
- $\models \varphi$ bedeutet φ ist allgemeingültig

$\equiv$ Das Symbol $\equiv$ kann folgende Dinge beschreiben:

- $\varphi \equiv \psi$ bedeuted φ ist logisch äquivalent zu ψ

1.3 Hornklauseln

Eine Hornklausel (oder auch Horn-Formel) ist eine Formel in Klauselform, welche maximal ein positives Literal hat.

Warning: Nicht jede Formel lässt sich als Hornklauselmenge darstellen!

Eine Hornklausel heißt

negativ wenn sie keine positiven Literale.

positiv wenn sie auschließlich positive Literale besitzt.

Werden meherere Horn-Formeln in einer Klauselmenge zusammengefasst, so spricht man von einer Hornklauselmenge.

1.3.1 Horn-Erfüllbarkeitstest

Der Horn-Erfüllbarkeitstest ist ein effizienter Test um zu prüfen, ob eine Hornklauselmenge erfüllbar ist.

Eingabe : Hornklauselmenge K
Ergebnis : K erfüllbar mit gegebenem minimalen Modell oder unerfüllbar

for $\psi \in K$ *der Form* $\psi = x$ **do**
  markiere x
for $\psi \in K$ *der Form* $\psi = \neg x_1 \vee \cdots \vee \neg x_n$ *(Typ 1) oder* $\psi = \neg x_1 \vee \cdots \vee \neg x_n \vee y$ *(Typ 2), wobei* $x_1, \cdots, x_n$ *markiert sind und, im Falle von Typ 2,* y *noch nicht markiert ist* **do**
  if ψ *ist vom Typ 1* **then**
    $\Longrightarrow$ unerfüllbar
  else if ψ *ist vom Typ 2* **then**
    Markiere y
$\Longrightarrow$ erfüllbar mit minimalem Modell gegeben durch die markierten Variablen

1.4 Kalküle

1.4.1 Resolutionskalkül

Das Resolutionskalkül ist ein Beweiskalkül, welches zum Beweisen der Unerfüllbarkeit von Klauselmengen.

Eine Klausel C heißt Resolvente von C_1 und C_2, wenn für ein Literal L gilt

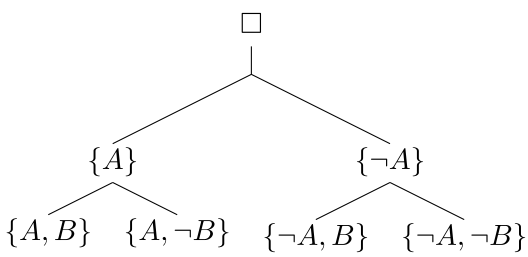
$$L \in C_1, \neg L \in C_1 \text{ und } C = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\neg L\})$$

Algorithmus Die Funktion Res(M) produziert alle möglichen Resolventen der Klauselmenge M.

Input : Klauselmenge K
Ergebnis : K erfüllbar oder unerfüllbar

while *Res*$(R) \neq R \wedge \Box \notin R$ **do**
  $R \leftarrow$ Res(R)
if $\Box \in R$ **then**
  $\Longrightarrow$ unerfüllbar
else
  $\Longrightarrow$ erfüllbar

Beispiel $K \coloneqq \{\{A, B\}, \{A, \neg B\}, \{\neg A, B\}, \{\neg A, \neg B\}\}$



Einheitsresolution Als Einheitsresolution wird eine Resolution benannt, bei denen ausschließlich Klauseln mit 1-elementigen Klauseln resolviert.

1.4.2 Sequenzenkalkül

Schlussregeln in $\mathcal{SK}$ für AL

Das Sequenzenkalkül $\mathcal{SK}$ definiert die Schlussregeln in Abbildung 1.1.

$$
\begin{array}{rlrl}
\text{(Axiom)} & \dfrac{}{\Gamma, \varphi \vdash \Delta, \varphi} & & \\[2ex]
\text{(0-Axiom)} & \dfrac{}{\Gamma, 0 \vdash \Delta} & \text{(1-Axiom)} & \dfrac{}{\Gamma \vdash \Delta, 1} \\[2ex]
(\neg L) & \dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma, \neg \vdash \Delta} & (\neg R) & \dfrac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi} \\[2ex]
(\vee L) & \dfrac{\Gamma, \varphi \vdash \Delta \quad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta} & (\vee R) & \dfrac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi} \\[2ex]
(\wedge L) & \dfrac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta} & (\wedge R) & \dfrac{\Gamma \vdash \Delta, \varphi \quad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \wedge \psi}
\end{array}
$$

Abbildung 1.1: Schlussregeln in $\mathcal{SK}$

Schlussregeln in $\mathcal{SK}^+$ für AL

Das Sequenzenkalkül $\mathcal{SK}^+$ erweitert $\mathcal{SK}$ (1.1) um Schnittregeln, unter anderem um modus ponens, wodurch sich eine vielzahl weiterer Regeln herleiten lässt. Die wichtigstens Regeln (inklusive der Kernschnittregeln modus ponens) sind in Abbildung 1.2 aufgezeigt.

$$
\begin{array}{rc}
\text{(modus ponens)} & \dfrac{\Gamma \vdash \varphi \quad \Gamma', \varphi \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta} \\[2ex]
\text{(Kontradiktion)} & \dfrac{\Gamma \vdash \varphi \quad \Gamma' \vdash \neg\varphi}{\Gamma, \Gamma' \vdash \emptyset} \\[2ex]
\text{(Widerspruch)} & \dfrac{\Gamma, \neg\varphi \vdash \psi \quad \Gamma, \neg\varphi \vdash \neg\psi}{\Gamma \vdash \varphi}
\end{array}
$$

Abbildung 1.2: Schlussregeln in $\mathcal{SK}^+$

1.5 Kompaktheitssatz

Eine (möglicherweise unendliche) Formelmenge Σ ist erfüllbar gdw. jede endliche Teilmenge von Σ erfüllbar ist.

1.6 Normalformen

In der Aussagenlogik existieren die disjunktive Normalform und die konjunktive Normalform, welche sich in ihrer Komplexität stark unterscheiden. Ebenfalls ist ein effizientes Umrechnen von KNV $\leftrightarrow$ DNF nicht möglich.

1.6.1 Disjunktive Normalformmodus ponens (DNF)

Eine Formel φ ist in disjunktiver Normalform gdw. sie die folgende Form hat (p_{ij} ist ein Atom):

$$\varphi = \bigvee_i \bigwedge_j p_{ij}$$

Beispiel Gegeben sei die Formel $\varphi := \neg(p \vee (\neg(p \wedge q) \wedge \neg r)) \vee s$. Diese ist logisch äquivalent zu folgender Formel in konjunktiver Normalform: $\varphi \equiv (\neg p \vee s) \wedge (p \vee r \vee s) \wedge (q \vee r \vee s)$.

1.6.2 Konjunktive Normalform (KNF)

Eine Formel φ ist in konjunktiver Normalform gdw. sie die folgende Form hat (p_{ij} is tein Atom):

$$\varphi = \bigwedge_i \bigvee_j p_{ij}$$

Beispiel Gegeben sei die Formel $\varphi := \neg(p \vee (\neg(p \wedge q) \wedge \neg r)) \vee s$. Diese ist logisch äquivalent zu folgender Formel in disjunktiver Normalform: $\varphi \equiv (r \wedge \neg p) \vee s$.

2 Logik erster Stufe

2.1 Grundlegende Begriffe

Formeln Eine Formel der Logik erster Stufe ist eine Kombination aus Quantoren, Variablen, Funktionen und Relationen, welche mit booleschen Junktoren Verknüpft werden. Typische Namen für Formeln sind φ und ψ.

freie/gebundene Variablen Eine freie Variable ist nicht an einen Quantor gebunden ((ab-) quantifiziert). Die Menge dieser Variablen wird mit der Funktion $\text{frei}(\varphi)$ rekursiv formalisiert:

$$\begin{aligned}
\text{frei}(t_1 = t_2) &:= \text{var}(t_1) \cup \text{var}(t_2) \\
\text{frei}(Rt_1 \cdots t_n) &:= \text{var}(t_1) \cup \cdots \cup \text{var}(t_2) \\
\text{frei}(\neg\varphi) &:= \text{frei}(\varphi) \\
\text{frei}(\varphi \wedge \psi) = \text{frei}(\varphi \vee \psi) &:= \text{frei}(\varphi) \cup \text{frei}(\psi) \\
\text{frei}(\forall x\varphi) = \text{frei}(\exists x\varphi) &:= \text{frei}(\varphi) \setminus \{x\}
\end{aligned}$$

Interpretationen

Quantoren In der Logik erster Stufe wird zwischen den folgenden Quantoren unterschieden:

Allquantor Der Allquantor ($\forall$) bindet (quantifiziert) die nachstehende Variable an sich und sagt aus, dass die folgende Aussage im Bezug auf den Quantor für alle Elemente aus der Trägermenge der Signatur gilt.
Beispiel: $\forall x(x > 0)$ ist semantisch äquivalent zu „alle Elemente sind größer als 0".

Existensquantor Der Existensquantor ($\exists$) bindet (quantifiziert) die nachstehen Variable and sich und sagt aus, dass die folgende Aussage im Bezug auf den Quantor für mindestens ein Element aus der Trägermenge gilt.
Beispiel: $\exists x(x > 0)$ ist semantisch äquivalent zu „mindestens ein Element ist größer als 0".

Quantorenrang Als *Quantorenrang* wird die Anzahl der Quantoren bezeichnet, was durch die Funktion

$qr(\varphi)$ rekursiv formalisiert wird:

$$\begin{aligned} \mathrm{qr}(\varphi) &:= 0 && (\varphi \text{ atomar}) \\ \mathrm{qr}(\neg\varphi) &:= \mathrm{qr}(\varphi) \\ \mathrm{qr}(\varphi \wedge \psi) = \mathrm{qr}(\varphi \vee \psi) &:= \max(\mathrm{qr}(\varphi), \mathrm{qr}(\psi)) \\ \mathrm{qr}(\forall x \varphi) = \mathrm{qr}(\exists x \varphi) &:= \mathrm{qr}(\varphi) + 1 \end{aligned}$$

Symbole In Formeln der Logik erster Stufe können die Folgenden Typen von Symbolen genutzt werden (sei A die Trägermenge):

Konstantensymbole Konstantensymbole, meist mit $a, b, c, \cdots$ bezeichnet, sind Elemente aus A, welche Konstant in einer Formel stehen und nicht von einer Interpretation belegt oder an Quantoren gebunden werden können.

Funktionssymbole Funktionssymbole, meist mit $f, g, h, \cdots$ bezeichnet, sind n-stellige Funktionen der Form $A^n \to A$.
Notation: Sei f ein 2-stelliges Funktionssymbol, so ist fxy als ein Aufruf dieser mit den Parametern x, y zu verstehen.

Relatonssymbole Relationssymbole, meist mit $R, L, P, \cdots$ bezeichnet, sind n-stellige Teilmengen von A^n.
Prefix-Notation: Sei R ein 3-stelliges Relationssymbol, so ist $Rxyz$ als ein test zu verstehen, ob x, y, z in Relation stehen.
Infix-Notation: Sei $\leq$ ein 2-stelliges Relationssymbol, so ist $x \leq y$ als ein Test zu verstehen, ob x, y in Relation stehen.

Sätze Eine Formel φ wird als *Satz* bezeichnet, wenn gilt $\mathrm{frei}(\varphi) = \emptyset$. In anderen Worten: Dass die Formel keine freien Variablen enthält.

$\varphi(t/x)$ besagt, dass die freie Variable x in φ durch c ersetzt wird.
Beispiel:

$$\begin{aligned} \varphi(x) :=& \forall n(n = x \vee n \neq x) \\ \varphi(c/x) =& \forall n(n = c \vee n \neq c) \end{aligned}$$

2.2 Herband

2.2.1 Herbrandstrukturen

2.2.2 Herbrandmodelle

2.3 Kalküle

2.3.1 Grundinstanzen-Resolutionskalkül

Die Grundinstanzen-Resolution ist eine Erweiterung des Resolutionskalküls der Aussagenlogik (1.4.1). Somit kann es ebenfalls genutzt werden, um die Erfüllbarkeit und Unerfüllbarkeit einer Klauselmenge Φ zu zeigen. Hierzu müssen alle Formeln, welche die Klauselmenge produzieren, in Skolem-Normalform vorliegen. Die eigentliche Formel (φ für $\forall x_1 \cdots \forall x_n \varphi$) entspricht dann einer Klausel der Klauselmenge. Die Klauselmenge wird nun für die Resolution vorbereitet, indem die folgenden Substitutionen vorgenommen werden können:

- Ersetzung einer Variable durch eine Konstante: $P(x) \to P(c)$
- Ersetzung einer Variable durch eine andere Variable: $P(x) \to P(y)$
- Ersetzung einer Variable durch eine Funktion: $P(x) \to P(fx)$

Hierbei müssen stets alle Variablen einer Unterklauselmenge, welche aus einer Formel entstanden ist, substituiert werden!

Beispiel Seien die folgenden Sätze in FO gegeben:

$$
\begin{aligned}
\varphi_1 :=& \forall x \forall y (Rxy \to (Px \leftrightarrow Qy)) \\
=& \forall x \forall y ((\neg Rxy \vee Px \vee \neg Qy) \wedge (\neg Rxy \vee \neg Px \vee Qy)) \\
\triangleq& \{\{\neg Rxy, Px, \neg Qy\}, \{\neg Rxy, \neg Px, Qy\}\} \\
\varphi_2 :=& \forall x \exists y (Rxy \wedge Py) \\
\simeq& \forall x (Rxfx \wedge Pfx) \\
\triangleq& \{\{Rxfx\}, \{Pfx\}\} \\
\varphi_3 :=& \exists x (\neg Px \wedge \forall y (Qy \wedge (Py \to Rxy))) \\
=& \exists x \forall y (\neg Px \wedge (Qy \wedge (Py \to Rxy))) \\
\simeq& \forall y (\neg Pg \wedge (Qy \wedge (Py \to Pgy))) \\
=& \forall y (\neg Pg \wedge Qy \wedge (\neg Py \vee Rxy)) \\
\triangleq& \{\{\neg Pg\}, \{Qy\}, \{\neg Py, Rxy\}\}
\end{aligned}
$$

Dies ergibt, nach überlegten Substitutionen, die folgende Grundinstanzen-Klauselmenge:

$$\begin{aligned}&\{\\&\ \{\neg Rgfc, Pg, \neg Qfc\}, \{\neg Rgfc, \neg Pg, \neg Qfc\},\\&\ \{Rgfc\}, \{Pfg\},\\&\ \{\neg Pg\}, \{Qfc\}, \{\neg Pfc, Rgfc\}\\&\}\end{aligned}$$

Wodurch eine Resolution möglich wird.

2.3.2 Sequenzenkalkül

Schlussregeln in $\mathcal{SK}$ für FO

Das Sequenzenkalkül $\mathcal{SK}$ für die Logik erster Stufe weitet das Sequenzenkalkül $\mathcal{SK}$ für die Aussagenlogik (siehe 1.1) auf die Logik erster Stufe aus. Es gelten somit auch alle Regeln aus dem Sequenzenkalkül $\mathcal{SK}$ der Aussagenlogik. Die Schlussregeln sind in der Abbildung 2.1 aufgeführt.

$(\forall L)$	$\dfrac{\Gamma, \varphi(t/x) \vdash \Delta}{\Gamma, \forall x \varphi(x) \vdash \Delta}$	$\forall R$	$\dfrac{\Gamma \vdash \Delta, \varphi(c/x)}{\Gamma \vdash \Delta, \forall x \varphi(x)}$
$(\exists L)$	$\dfrac{\Gamma, \varphi(c/x) \vdash \Delta}{\Gamma, \exists x \varphi(x) \vdash \Delta}$	$\exists R$	$\dfrac{\Gamma \vdash \Delta, \varphi(t/x)}{\Gamma \vdash \Delta, \exists x \varphi(x)}$

Abbildung 2.1: Schlussregeln in $\mathcal{SK}$
Achtung: Für die Regeln $\exists L$, $\forall R$ darf das c noch nicht in $\Gamma, \Delta, \varphi(x)$ vorhanden sein.

Schlussregeln in $\mathcal{SK}^=$ für FO

Das Sequenzenkalkül $\mathcal{SK}^=$ erweitert $\mathcal{SK}$ (siehe 2.1) um Schlussregeln für die Gleichheit $(=)$. Die Schlussregeln sind in 2.2 aufgeführt.

$(=)$	$\dfrac{\Gamma, t = t' \vdash \Delta}{\Gamma \vdash \Delta}$		
(Sub-L)	$\dfrac{\Gamma, \varphi(t/x) \vdash \Delta}{\Gamma, t = t', \varphi(t'/x) \vdash \Delta}$	(Sub-R)	$\dfrac{\Gamma \vdash \Delta, \varphi(t/x)}{\Gamma, t = t' \vdash \Delta, \varphi(t'/x)}$

Abbildung 2.2: Schlussregeln in $\mathcal{SK}^=$

2.4 Kompaktheitssatz

Entspricht dem Kompaktheitssatz der Aussagenlogik (siehe 1.5).
Daraus folgt, dass für jede Formelmenge $\Phi \subseteq FO$ und Formel $\psi \in FO$ gilt:

$$\Phi \models \psi \iff \text{es existiert eine endliche Teilmenge } \Phi_0 \subseteq \Phi, \text{ sodass } \Phi_0 \models \psi$$

2.5 Normalformen

2.5.1 Pränexe Normalform (PNF)

Eine Formel φ ist in der *pränexen Normalform*, wenn in dieser alle Quantoren vor der eigentlichen Formel stehen, das heißt, sie hat die Form $\substack{\forall\\\exists} x_1 \cdots \substack{\forall\\\exists} x_n \varphi$.

Um eine beliebige Formel φ in die pränexe Normalform umzuformen, müssen alle Quantoren nach vorne gezogen. Ist eine Variable nicht eindeutig an einen Quantor gebunden, so muss diese zunächst umbenannt werden.

Beispiel Sei $\varphi := \forall x \exists y(x < y \land \exists y(x > y))$.

Diese Formel wird nun schrittweise in die pränexe Normalform umgeformt:

$$\begin{aligned}
\varphi &= \forall x \exists y(x < y \land \exists y(x > y)) && \text{(Umbenennung)} \\
&= \forall x \exists y(x < y \land \exists z(x > z)) && \text{(Quantoren-Verschiebung)} \\
&= \forall x \exists y \exists z(x < y \land x > z) && \text{(Pränexe Normalform)}
\end{aligned}$$

2.5.2 Skolem-Normalform (SNF)

Eine Formel φ ist in der *Skolem-Normalform*, wenn in dieser ausschließlich Allquantoren existieren und diese vor der eigentlichen Formel stehen, das heißt, sie hat die Form $\forall x_0 \cdots \forall x_n \varphi$.

Um eine Formel in pränexer Normalform φ in die Skolem-Normalform umzuwandeln, müssen alle Variablen, welche an Existenzquantoren gebunden sind, in (neue!) n-stellige Funktionen geändert werden, welche von allen vorherigen freien und an Allquantoren gebundenen Variablen abhängen. Existiert keine solche vorherige Variable, so wird die Variable durch ein 0-stelliges Funktionssymbol substituiert, d.h. durch eine (neue!) Konstante. Dieser Prozess wird *Skolemisierung* genannt.

Beispiel Sei $\varphi := \exists x \forall y \exists z(x > y \lor y < z)$ in pränexer Normalform.

Diese Formel wird nun schrittweise in die Skolem-Normalform umgewandelt (skolemisiert):

$$\begin{aligned}
\varphi &= \exists x \forall y \exists z(x > y \lor y < z) && \text{(1. Existenzquantor)} \\
&= \forall y \exists z(c > y \lor y < z) && \text{(2. Existenzquantor)} \\
&= \forall y(c > y \lor y < fy) && \text{(Skolem-Normalform)}
\end{aligned}$$

2.5.3 Negationsnormalform (NNF)

Eine Formel φ ist in der *Negationsnormalform*, wenn sie ausschließlich aus atomaren und negiert atomaren Formel mit $\forall$, $\exists$, $\land$, $\lor$ aufgebaut ist.

Um eine beliebige Formel φ in die Negationsnormalform umzuwandeln, müssen die booleschen Gesetze angewandt werden und mögliche Negationen auf die unterste Ebene geschoben werden.

Beispiel Sei $\varphi := \forall x \exists y(\neg(x > y \rightarrow (x = y)))$.
Diese Formel wird nun schrittweise in die Negationsnormalform umgewandelt:

$$\begin{aligned}\varphi &= \forall x \exists y(\neg(x > y \rightarrow (x = y))) \\ &= \forall x \exists y(\neg(\neg(x > y) \vee (x = y))) \\ &= \forall x \exists y(x > y \wedge \neg(x = y))\end{aligned}$$

Dies entspricht der Negationsnormalform, da $\neg(x = y)$ als atomar gilt, da $(x = y)$ eine Relation darstellt.

2.5.4 Gleichheitsfreie Normalform (GNF)

Eine Formel φ ist in der *gleichheitsfreien Normalform*, wenn in dieser kein Gleichheitszeichen $=$ existiert.
Um ein beliebige Formel φ in die gleichheitsfreie Normalform umzuwandeln, müssen alle Gleichheitszeichen durch ein (neues!) Relationssymbol $\sim$ ersetzt werden, welches die Funktionalität des Gleichheitszeichens übernimmt.

Beispiel Sei $\varphi := \forall x \exists y(x = y)$.
Diese Formel wird nun in die gleichheitsfreie Normalform umgeformt:

$$\begin{aligned}\varphi &= \forall x \exists y(x = y) && \text{(Substitution)} \\ &= \forall x \exists y(x \sim y) && \text{(Gleichheitsfreie Normalform)}\end{aligned}$$

2.6 Spielsemantik

Die Spielsemantik ist eine Methode, um die Allgemeingültigkeit und auch die Unerfüllbarkeit einer Formel zu beweisen. Hierbei wird eine in Negationsnormalform vorliegende Formel φ von links nach rechts durchlaufen, wobei die in **??** gelisteten Spielregeln angewandt werden.

Semantik-Spiel $[\mathcal{A}; \mathrm{SF}(\varphi)]$:
Spieler: **V**erifizierer gegen **F**alsifizierer

Spielpositionen: $(\psi, a) \in \mathrm{SF}(\varphi) \times A^n$

Züge in Position $(\psi, a), a = (a_1, \cdots, a_n)$:

$$\begin{aligned}
\psi = \psi_1 \wedge \psi_2 &\quad \textbf{F} \text{ am Zug - zieht nach } (\psi_1, a) \text{ oder } (\psi_2, a) \\
\psi = \psi_1 \vee \psi_2 &\quad \textbf{V} \text{ am Zug - zieht nach } (\psi_1, a) \text{ oder } (\psi_2, a) \\
\psi = \forall x_i \psi_0 &\quad \textbf{F} \text{ am Zug - zieht nach einem } (\psi_0, a') \text{ mit } a' = (a_1, \cdots, a_i', \cdots, a_n) \\
\psi = \exists x_i \psi_0 &\quad \textbf{V} \text{ am Zug - zieht nach einem } (\psi_0, a') \text{ mit } a' = (a_1, \cdots, a_i', \cdots, a_n)
\end{aligned}$$

Spielende: In Positionen (ψ, a), wobei ψ atomar oder negiert atomar ist.

$$\text{Gewinner: } \begin{cases} \textbf{V}\text{erifizierer} & \text{falls } \mathcal{A} \models \psi[a] \\ \textbf{F}\text{alsifizierer} & \text{falls } \mathcal{A} \not\models \psi[a] \end{cases}$$

Abbildung 2.3: Spielsemantik

3 Entscheidbarkeit

- $\mathrm{SAT}(\mathrm{AL}) := \{\varphi \in \mathrm{AL} : \varphi \text{ erfüllbar}\}$
  entscheidbar
- $\overline{\mathrm{SAT}(\mathrm{AL})} := \{\varphi \in \mathrm{AL} : \varphi \text{ unerfüllbar}\}$
  entscheidbar
- $\mathrm{VAL}(\mathrm{AL}) := \{\varphi \in \mathrm{AL} : \varphi \text{ allgemeingültig}\}$
  enscheidbar
- $\mathrm{SAT}(\mathrm{FO}) := \{\varphi \in \mathrm{FO} : \varphi \text{ erfüllbar}\}$
  unentscheidbar, rekursiv aufzählbar
- $\overline{\mathrm{SAT}(\mathrm{FO})} := \{\varphi \in \mathrm{FO} : \varphi \text{ unerfüllbar}\}$
  unentscheidbar, rekursiv aufzählbar
- $\mathrm{VAL}(\mathrm{FO}) := \{\varphi \in \mathrm{FO} : \varphi \text{ allgemeingültig}\}$
  unentscheidbar, rekursiv aufwählbar
- $\mathrm{FINSAT}(\mathrm{FO}) := \{\varphi \in \mathrm{FO} : \varphi \text{ hat ein endliches Model}\}$
  unentscheidbar, nicht rekursiv aufwählbar
- $\mathrm{INFSAT}(\mathrm{FO}) := \{\varphi \in \mathrm{FO} : \varphi \text{ hat nur unendliche Modelle}\}$
  unentscheidbar, nicht rekursiv aufzählbar

4 Abkürzungen und Begriffe

Endlichkeitssatz Kompaktheitssatz.