Einführung in die Prädikatenlogik

DSAI - Dietmar Steiner

HTL Leonding
next level

Motivationsbeispiel

Die bisherigen Konzepte der Aussagenlogik sind beschränkt. Schauen wir uns dazu folgendes Beispiel an:

$$J \ldots \textbf{ Jane Doe studiert Informatik}$$
$$L \ldots \textbf{ Jane Doe mag Logik}$$
$$J \implies L$$

Wir wissen daher, wenn *Jane* Informatik studiert, dann mag sie auch Logik.

Motivationsbeispiel

Jetzt wollen wir zeigen, wenn *John* Informatik studiert, dann mag er auch Logik.

$$K \ldots \textbf{ John Doe studiert Informatik}$$
$$M \ldots \textbf{ John Doe mag Logik}$$

$$K \implies M$$

Während die gleichen Konzepte (*Informatik studieren*, *Logik mögen*) dargestellt werden, besteht keine Beziehung zwischen J, K, L, M.

Was ist ein Prädikat

- In der deutschen Grammatik entspricht das Prädikat eines Satzes dem Hauptverb (und seinen Hilfsverben). In den letzten Sätzen wären die Prädikate **studiert** und **mag**.
- In der formalen Logik und der traditionellen Grammatik ist ein Prädikat ein Ausdruck, der wahr sein kann. Es muss ein Verb enthalten und kann weitere Elemente umfassen, die Informationen über das Subjekt liefern, wie zum Beispiel, was das Subjekt tut oder wie das Subjekt ist: z. B. **studiert Informatik** oder **mag Logik**.

Prädikat Symbole

Um die gemeinsame Struktur von Aussagen herauszuarbeiten, können wir *Prädikatsymbole* verwenden, die in der Prädikatenlogik eingeführt wurden:

$$studiertInformatik(x)$$

$$magLogik(x)$$

Wir könnten nun unsere früheren Aussagen mit diesen Prädikaten neu formulieren, indem wir die Variable x mit *Jane* oder *John* instanziieren.

Prädikatensymbole

Anmerkungen:

- Ein *Prädikatsymbol* ordnet den übergebenen Objekten einen Wahrheitswert zu – z. B. $\boldsymbol{magLogik(x)}$ oder $\boldsymbol{isst(y, z)}$. Wir werden uns auf solche Symbole konzentrieren.
- Zusätzlich ordnen Funktionssymbole den übergebenen Objekten ein neues Objekt zu – z. B. $\boldsymbol{sin(x)}$, $+$ oder $\boldsymbol{\pi}$ in der Arithmetik
- Ein Symbol $\boldsymbol{p(t_1, t_2, \ldots t_n)}$ hat eine *Stelligkeit* $\boldsymbol{n}$.
  Die Stelligkeit entspricht der Anzahl an Argumenten.

Fortsetzung Motivationsbeispiel

Um zu zeigen, dass jeder, der Informatik studiert, auch Logik mag, können wir unsere neuen Kenntnisse der Prädikatenlogik einsetzen, indem wir Prädikatsymbole für die Aussagen verwenden.

$$studiertInformatik(JaneDoe) \implies magLogik(JaneDoe)$$
$$studiertInformatik(JohnDoe) \implies magLogik(JohnDoe)$$
$$\dots$$

Gibt es einen besseren Weg?

Quantoren

Ein weiteres neu eingeführtes Konzept in der Prädikatenlogik sind *Quantoren*, die es uns ermöglichen, Ausdrücke wie die folgenden zu verwenden:

- *Für alle Werte von* $\boldsymbol{x}$...
- *Es existiert mindestens ein Wert von* $\boldsymbol{x}$ *für den gilt...*

Der Allquantor

Das „für alle“ wird durch den Allquantor $\forall$ dargestellt, wie im folgenden Beispiel zu sehen ist:

$$\forall x : studiertInformatik(x) \implies magLogik(x)$$

Übersetzung: *Für alle* x, *wenn* x *Informatik studiert, dann mag* x *logik.*

Der Allquantor

Was, wenn jeder, der Informatik studiert, sowohl Logik als auch Algorithmen mag? Wir können die *logischen Verknüpfungen* verwenden, die wir bereits aus der Aussagenlogik kennen:

$$\forall x : studiertInformatik(x) \implies magLogik(x) \land magAlgorithmen(x)$$

Oder wenn jemand Logik *oder* Algorithmen mag:

$$\forall x : studiertInformatik(x) \implies magLogik(x) \lor magAlgorithmen(x)$$

Der Existenzquantor

Die *Es gibt mindestens eine(n)* -Phrase kann durch den Existenzquantor $\exists$ dargestellt werden. Nehmen wir an, dass es mindestens eine Person gibt, die Informatik studiert, aber keine *Regular Expressions* mag:

$$\exists x : studiertInformatik(x) \land \neg magRegEx(x)$$

Der Existenzquantor

Bitte beachten, dass wir eine *Konjunktion* statt einer *Implikation* mit dem Existenzquantor verwendet haben. Warum? Schauen wir uns das folgende unrealistische Beispiel an:

$$\exists x : studiertInformatik(x) \implies magBugs(x)$$

Auf den ersten Blick scheint es, als gäbe es mindestens eine Person, die Informatik studiert und Bugs mag. Nehmen wir an, wir instanzieren x mit jemandem, der nicht Informatik studiert. Da das Antezedens falsch ist, **wird die Formel wahr**, unabhängig davon, was das Konsequenz ist. Wir gewinnen dadurch keine sinnvolle Information über unsere Welt.

Negationen

Wir können auch *Negationen* vor unsere Quantoren setzen, um Phrasen wie *nicht für alle* oder *es gibt nicht* darzustellen. Zum Beispiel könnte unser vorheriges Beispiel auch so übersetzt werden: *Nicht jeder, der Informatik studiert, mag Regular Expressions.*

$$\neg\forall x : studiertInformatik(x) \implies magRegEx(x)$$

Negations

Ein weiteres Beispiel mit mehreren gültigen Übersetzungen wäre: *Niemand, der Informatik studiert, mag bugs*. Wir könnten dies wie folgt übersetzen:

$$\forall x : studiertInformatik(x) \implies \neg magBugs(x)$$

Oder:

$$\neg\exists x : studiertInformatik(x) \land magBugs(x)$$

Konstanten

Ein *Funktionssymbol* mit der *Stelligkeit* von 0 ist ein *Konstantensymbol* und wird üblicherweise mit einem großgeschriebenen Anfangsbuchstaben dargestellt. Wir können *Konstantensymbole* verwenden, um Formeln übersichtlicher zu schreiben. Anstatt:

$$\forall x : studiertInformatik(x) \implies codiertInJava(x) \lor codiertInCSharp(x)$$

Wir könnten schreiben:

$$\forall x : studiertInformatik(x) \implies codiert(x, Java) \lor codiert(x, CSharp)$$

Mehrere Quantoren

Wir können auch mehrere Variablen quantifizieren, zum Beispiel:

$$\exists x \forall y : studiertInformatik(x) \land codiert(x, y)$$

Übersetzung: *Es existiert mindestens eine Person, die Informatik studiert und alles programmieren kann.*

Mehrere Quantoren

Da wir unser Universum als die Gesamtheit aller Dinge in der Welt annehmen, kann es sinnvoll sein, unseren Bereich durch zusätzliche Prädikate weiter einzugrenzen:

$$\forall x, \exists y : studiertInformatik(x) \implies istProgrammiersprache(y) \land codiert(x, y)$$

Oder:

$$\forall x : studiertInformatik(x) \implies \exists y : (istProgrammiersprache(y) \land codiert(x, y))$$

Übersetzung: *Jeder, der Informatik studiert, kann in mindestens einer Programmiersprache programmieren.*

Beispiele

Every programmer drinks coffee but is tired anyways.

$$\forall x : isProgrammer(x) \implies \boldsymbol{drinks(x, Coffee)} \land \boldsymbol{isTired(x)}$$

Beispiele

Lemons and oranges are citrus fruits.

$$\forall x : isLemon(x) \lor isOrange(x) \implies isCitrusFruit(x)$$

Beispiele

No human being is perfect.

$$\forall x : isHuman(x) \implies \neg isPerfect(x)$$

Alternative:

$$\neg \exists x : isHuman(x) \land isPerfect(x)$$

Any questions?

d.steiner@htl-leonding.ac.at

HTL Leonding
next level