

Sprache B

Die Sprache B besteht aus Aussagen, wie sie unten beschrieben und erklärt werden. Diese Aussagen sind aus Schlüsselwörtern und den Namen der Individuen, Typen und Relationen der entsprechenden Mini-Welt aufgebaut. Die verwendeten Schlüsselwörter sind „HasType“, „SubTypeOf“, „HasDomain“, „HasRange“, „or“, „min“ und „max“.

Aussagen

Jede Aussage kann entweder wahr oder falsch sein. Jede Aussage der Sprache B hat die Form eines der vier Schemas, die hier beschrieben werden. Es ist zu beachten, dass Typen komplex sein können (siehe nächster Abschnitt).

HasType-Aussagen	
Schema:	<i>Individuum</i> HasType <i>Typ</i>
Beispiel:	John HasType golfer
Erklärung:	Eine HasType-Aussage verlangt ein Individuum und einen Typ und sagt aus, dass das gegebene Individuum zum gegebenen Typ gehört. Das obige Beispiel sagt aus, dass John ein Golfer ist.

HasDomain-Aussagen	
Schema:	<i>Relation</i> HasDomain <i>Typ</i>
Beispiel:	buys HasDomain golfer
Erklärung:	Eine HasDomain-Aussage verlangt eine Relation und einen Typ. Eine solche Aussage sagt aus, dass immer wenn ein Individuum die gegebene Relation zu einem anderen Individuum hat, dass dann das erste Individuum zum gegebenen Typ gehört. Das obige Beispiel sagt aus, dass jedes Individuum, das eine „buys“-Relation zu einem anderen Individuum hat, ein Golfer ist.

HasRange-Aussagen	
Schema:	<i>Relation</i> HasRange <i>Typ</i>
Beispiel:	loves HasRange woman
Erklärung:	Eine HasRange-Aussage verlangt eine Relation und einen Typ. Eine solche Aussage sagt aus, dass immer wenn ein Individuum die gegebene Relation zu einem anderen Individuum hat, dass dann das zweite Individuum zum gegebenen Typ gehört. Das obige Beispiel sagt aus, dass jedes Individuum, zu dem ein anderes Individuum eine „loves“-Relation hat, eine Frau ist.

SubTypeOf-Aussagen	
Schema:	<i>Typ1</i> SubTypeOf <i>Typ2</i>
Beispiel:	golfer SubTypeOf man
Erklärung:	Eine SubTypeOf-Aussage verlangt zwei Typen und sagt aus, dass jedes Individuum, das zum ersten Typ gehört, auch zum zweiten Typ gehört (aber nicht zwingenderweise umgekehrt). Das obige Beispiel sagt aus, dass jedes Individuum, das ein Golfer ist, auch ein Mann ist.

Typ-Operatoren

Jeder Typ (einfach oder komplex) steht für eine gewisse Gruppe von Individuen. Einfache Typen sind zum Beispiel „woman“ oder „golfer“. Neben einfachen Typen gibt es aber auch komplexe Typen, die durch die hier beschriebenen Typ-Operatoren zusammengesetzt werden. „sees **min** 2 golfer“ ist zum Beispiel ein komplexer Typ.

or-Operator	
Schema:	<i>Typ1</i> or <i>Typ2</i>
Beispiel:	woman or golfer
Erklärung:	Der or-Operator verlangt zwei Typen. Der entstehende komplexe Typ steht für alle Individuen, die zum ersten Typ oder zum zweiten Typ oder zu beiden gehören. Das obige Beispiel steht für alle Individuen, die Frauen oder Golfer oder beides sind.

min-Operator	
Schema:	<i>Relation</i> min <i>Zahl</i> <i>Typ</i>
Beispiel:	loves min 2 woman
Erklärung:	Der min-Operator verlangt eine Relation, eine Zahl und einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die entsprechende Relationen zu mindestens der gegebenen Anzahl von Individuen des gegebenen Typs haben. Das obige Beispiel steht für alle Individuen, die eine „loves“-Relation zu mindestens zwei Frauen haben.

max-Operator	
Schema:	<i>Relation</i> max <i>Zahl</i> <i>Typ</i>
Beispiel:	sees max 2 man
Erklärung:	Der max-Operator verlangt eine Relation, eine Zahl und einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die entsprechende Relationen zu maximal der gegebenen Anzahl von Individuen des gegebenen Typs haben. Dies schliesst auch alle Individuen mit ein, die gar keine entsprechende Relation haben. Das obige Beispiel steht für alle Individuen, die eine „sees“-Relation zu höchstens zwei Männern haben.