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“, „not“, „some“ und „only“.

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).

Positive einfache Aussagen	
Schema:	*Individuum1* *Relation* *Individuum2*
Beispiel:	John sees Mary
Erklärung:	Eine positive einfache Aussage besteht aus zwei Individuen und einer Relation und sagt aus, dass das erste Individuum eine entsprechende Relation zum zweiten Individuum hat. Das obige Beispiel sagt aus, dass John eine „sees“-Relation zu Mary hat.

Negative einfache Aussagen	
Schema:	*Individuum1* **not** *Relation* *Individuum2*
Beispiel:	Mary **not** helps Bill
Erklärung:	Eine negative einfache Aussage besteht aus zwei Individuen und einer Relation, wobei der Relation das Schlüsselwort „not“ vorangestellt ist. Eine solche Aussage sagt aus, dass das erste Individuum keine entsprechende Relation zum zweiten Individuum hat. Das obige Beispiel sagt aus, dass Mary keine „helps“-Relation zu Bill hat.

HasType-Aussagen	
Schema:	*Individuum* ***HasType*** *Typ*
Beispiel:	John **HasType** man
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 Mann ist.

SubTypeOf-Aussagen	
Schema:	*Typ1* **SubTypeOf** *Typ2*
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 **only** golfer“ ist zum Beispiel ein komplexer Typ. Es ist zu beachten, dass solche komplexe Typen auch ineinander verschachtelt sein können. In diesem Fall werden Klammern verwendet um die Struktur zu verdeutlichen, zum Beispiel „**not** (loves **some** woman)“.

not-Operator	
Schema:	**not** *Typ*
Beispiel:	**not** golfer
Erklärung:	Der not-Operator verlangt nur einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die nicht zum gegebenen Typ gehören. Das obige Beispiel steht für alle Individuen, die keine Golfer sind.

some-Operator	
Schema:	*Relation* **some** *Typ*
Beispiel:	loves **some** woman
Erklärung:	Der some-Operator verlangt eine Relation und einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die eine entsprechende Relation zu mindestens einem Individuum des gegebenen Typs haben. Das obige Beispiel steht für alle Individuen, die eine „loves“-Relation zu mindestens einer Frau haben.

only-Operator	
Schema:	*Relation* **only** *Typ*
Beispiel:	helps **only** woman
Erklärung:	Der only-Operator verlangt eine Relation und einen Typ. Der entstehende komplexe Typ steht für alle Individuen, die entweder gar keine entsprechende Relation zu einem anderen Individuum haben oder wenn doch, dann nur zu Individuen des gegebenen Typs. Das obige Beispiel steht für alle Individuen, die „helps“-Relationen (falls überhaupt vorhanden) nur zu Frauen haben. Das Beispiel beinhaltet also alle Individuen ausser jenen, die eine „helps“-Relation zu einem Individuum haben, das keine Frau ist.