Language B

The language B consists of statements of the forms described and explained below. These statements are composed of keywords and of the names of the individuals, types, and relations of the respective mini world. The used keywords are „HasType“, „SubTypeOf“, „HasDomain“, „HasRange“, „or“, „min“, and „max“.

Statements

Every statement can either be true or false. Every statement of the language B has the form of one of the four schemes described here. Note that types can be complex (see the next section).

HasType-statement	
scheme:	*Individual* ***HasType*** *Type*
example:	John **HasType** golfer
explanation:	A HasType-statement requires an individual and a type and it states that the given individual belongs to the given type. The example above states that John is a golfer.

HasDomain-statement	
scheme:	*Relation* **HasDomain** *Type*
example:	buys **HasDomain** golfer
explanation:	A HasDomain-statement requires a relation and a type. Such a statement states that whenever an individual has the given relation to another individual then the first individual belongs to the given type. The example above states that every individual that has a „buys“-relation to another individual is a golfer.

HasRange-statement	
scheme:	*Relation* **HasRange** *Type*
example:	loves **HasRange** woman
explanation:	A HasRange-statement requires a relation and a type. Such a statement states that whenever an individual has the given relation to another individual then the second individual belongs to the given type. The example above states that every individual to which another individual has a „loves“-relation is a woman.

SubTypeOf-statement	
scheme:	*Type1* **SubTypeOf** *Type2*
example:	golfer **SubTypeOf** man
explanation:	A SubTypeOf-statement requires two types and states that every individual that belongs to the first type also belongs to the second type (but not necessarily the other way round). The example above states that every individual that is a golfer is also a man.

Type Operators

Every type (simple or complex) stands for a certain group of individuals. „traveler“ and „aquarium“ are examples of simple types. Apart from simple types, there are complex types that are composed by the type operators described here. „buys **min** 2 presents“ is an example of a complex type.

or-operator	
scheme:	*Type1* **or** *Type2*
example:	woman **or** golfer
explanation:	The or-operator requires two types. The resulting complex type stands for all individuals that belong to the first type or to the second type or to both. The example above stands for all individuals that are women or golfers or both.

min-operator	
scheme:	*Relation* **min** *Number* *Type*
example:	loves **min** 2 woman
explanation:	The min-operator requires a relation, a number, and a type. The resulting complex type stands for all individuals that have the given relation to at least the given number of individuals of the given type. The example above stands for all individuals that have a „loves“-relation to at least two women.

max-operator	
scheme:	*Relation* **max** *Number* *Type*
example:	sees **max** 2 man
explanation:	The max-operator requires a relation, a number, and a type. The resulting complex type stands for all individuals that have the given relation to a maximum of the given number of individuals of the given type. This includes the individuals that have no corresponding relation at all. The example above stands for all individuals that have a „sees“-relation to at most two men.