ObjectLogic: Schema-level Statements

- Signature F-atoms define methods of classes
- Methods correspond to roles in Description Logic

Subclass Relation

```
woman::person
man::person
```

Every woman is a person
Every man is a person

Signature statements (methods, cardinalities and ranges)

```
person[has_father {1:1} *=> man].
person[has_mother {1:1} *=> woman].

person[has_son {0:*} *=> man].
person[has_daughter {0:*} *=> woman].
```

A person has exactly one father, which is of class man

A person can have daughters which are of class woman

ObjectLogic: Instance-level Statements

- The application of a method on an object is expressed by data-F-atoms which consist of a host object, a method and a result object
- Relating instances to classes

```
abraham:man
isaac:man
sarah:woman
```

Abraham is of class man
Isaac is of class man
Sarah is of class woman

- Method invocation: assigning values to methods

```
isaac[has_father -> abraham].
isaac[has_mother -> sarah].
abraham[has_son- > isaac].
```

Isaac has a father which is Abraham,
Isaac has a mother which is sarah,
Abraham has a son which is Isaac.

- Combining class associations and method invocations

```
isaac:man[has_father->abraham:man].
isaac[has_mother->sarah:woman].
```

Isaac is of class man and has a father who is Abraham, who is a man.
Isaac has a mother who is sarah, which is a woman

ObjectLogic: Rules

- Examples:

```
?X[ancestor -> ?Y] :- ?X[has_father -> ?Y].
?X[ancestor -> ?Y] :- ?X[has_mother -> ?Y].
?X[ancestor -> ?Y] :- ?X[has_father -> ?Z]  AND  ?Z[ancestor -> ?Y].
?X[ancestor -> ?Y] :- ?X[has_mother -> ?Z]  AND  ?Z[ancestor -> ?Y].
```

ObjectLogic: Further Rules

- Examples:

```
?X[has_son -> ?Y]  :- ?Y:man[has_father -> ?X].
?X[has_son -> ?Y]  :- ?Y:man[has_mother -> ?X].
?X[has_daugther -> ?Y]  :- ?Y:woman[has_father -> ?X].
?X[has_daugther -> ?Y]  :- ?Y:woman[has_mother -> ?X].
```

Further Queries

- Queries are rules without a head
- Example:

```
?- ?X:woman[has_son -> ?Y[has_father -> abraham]].
```