

FLogic

Noam Bercovici
Renata Dividino

- ◆ Native support for frame based language FLogic, which is declarative like Prolog, and object-oriented like Java,
- ◆ Relations and concepts as basis for describing instances
- ◆ Close World Semantics (like Databases)
- ◆ Logic Rules (like in logic programming)

F-Logic	Meaning
C1::C2	C1 is a subclass of C2
O:C	O is an instance of C
C1[A=>>C2]	For the class C1, the multivalued attribute A is defined which values are instances of the class C1.
C1[A=>C2]	For the class C1, the atomic attribute A is defined which values are instances of the class C1.
O[A->>{V1,V2}]	The values of the attribute A of instance O are V1 and V2
O[A->V]	Instance O has value V for the attribute A

Concepts, Concepts Hierarchy and Signature:

person::ROOT

man::person

woman::person[

name=> String,

phones =>> String,

children =>> person]

Instances:

Noam:man

Renata:woman [

name -> 'John Doe',

phones ->> {6313214567, 6313214566},

children ->> {bob, mary}]

Predicate:

killer(John, Marie)

location (UniKoblenz, Koblenz)

Rules

- Ancestor:

FORALL X,Y X[ancestor->>Y] <- X[father->Y].

FORALL X,Y,Z X[ancestor->>Y] <- X[father->Z] AND
Z[ancestor->>Y].

- All persons who are not-relatives

FORALL X,Y X[notrelated->>Y] <- X:person AND Y:person
AND NOT X[ancestor->>Y] AND NOT Y[ancestor->>X].

Queries

- Maximum

FORALL X <- p(X) AND FORALL Y (p(Y) -> lessorequal(Y,X)).

- The fathers of sons of Rebeca

FORALL X,Y <- X:man[son->>Y[mother->rebeca]].