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)
<table>
<thead>
<tr>
<th>F-Logic</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1::C2</td>
<td>C1 is a subclass of C2</td>
</tr>
<tr>
<td>O:C</td>
<td>O is an instance of C</td>
</tr>
<tr>
<td>C1[A=&gt;&gt;C2]</td>
<td>For the class C1, the multivalue attribute A is defined which values are instances of the class C1.</td>
</tr>
<tr>
<td>C1[A=&gt;C2]</td>
<td>For the class C1, the atomic attribute A is defined which values are instances of the class C1.</td>
</tr>
<tr>
<td>O[A=&gt;&gt;{V1,V2}]</td>
<td>The values of the attribute A of instance O are V1 and V2</td>
</tr>
<tr>
<td>O[A-&gt;V]</td>
<td>Instance O has value V for the attribute A</td>
</tr>
</tbody>
</table>
Syntax - Examples

Concepts, Concepts Hierarchy and Signature:

person::ROOT
man::person
woman::person[
    name=> String,
    phones =>>> String,
    children =>>> person]
Syntax - Examples

Instances:

Noam:man
Renata:woman [ 
  name -> ‘John Doe’,
  phones ->> {6313214567, 6313214566},
  children ->> {bob, mary}]

Predicate:

killer(John, Marie)
location (UniKoblenz, Koblenz)
Syntax - Examples

Rules

- Ancestor:

  FORALL X,Y X[ancestor->>>Y] <- X[father->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
  \[
  \text{FORALL } X \leftarrow p(X) \text{ AND FORALL } Y \ (p(Y) \rightarrow \text{lessorequal}(Y,X)).
  \]

- The fathers of sons of Rebeca
  \[
  \text{FORALL } X,Y \leftarrow X:man[son->]Y[mother->rebeca]].
  \]