

Cogui Tutorial : Rules

The vocabulary can be seen as a lightweight ontology. More expressivity is obtained by adding *rules* of form "if *hypothesis* then *conclusion*". Rules can be used to enrich facts. The mechanism that processes rules in this way is called *forward-chaining*.

1. Open the file [family-before-rule-tutorial.cogxml](#). Look at the rules in the set [Rules / family-relationships](#).

These rules represent the following knowledge about the family domain:

- R1 and R2: `parentOf` and `childOf` are *inverse* relations
- R3: `ancestorOf` is *transitive*
- R4: `siblingOf` is *symmetrical*
- R5: *definition* of `motherOf` (with one direction being already encoded in the vocabulary)
- R6 et R7: *definition* of `siblingOf`
- R8: *definition* of `brotherOf` (with one direction being already encoded in the vocabulary)

2. Exercise: Add the *uncleOf* relation in the relation hierarchy with the appropriate signature. Then add two new rules that define `uncleOf` as follows:
"x is an uncle of y if and only if there is z such that x is the brother of z and z is one of the parents of y"

Hints:

- Each rule corresponds to one direction of the "if and only if" statement, similarly to what we have done for the `siblingOf` relation.
- Draw one rule, and then copy/paste it to draw the second rule.

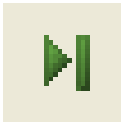
3. Open the forward chaining assistant (by the menu Reasoning/Apply rules) and use it to compute the `ancestorOf` relation between the individuals of the genealogical graph encoded in the fact [genealogy-fact](#). Run the assistant step by step to see each rule application. Store the fact obtained after saturation and name it [ancestors-fact](#).

Details:

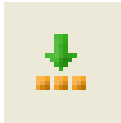
- Select the fact [genealogy-fact](#) and the single rule R3. Click on the next button: the initial fact is drawn.



- Apply the rule step by step (with the green triangle button): the image of the rule hypothesis in the fact is colored in gray and the modified part of the graph is colored in pink. More precisely, the pink part shows the new nodes as well as already existing nodes affected by the rule application, i.e. nodes whose label or neighbourhood has changed. Remember that Cogui forward chaining operates in a breadth-first manner: at each level, it collects all possible rule applications on the current fact, then perform them before looking for new rule applications. For each rule application, the assistant indicates the name of the rule applied and the current forward chaining level. In this example, forward chaining stops at level 2.
- Note that Cogui does not perform redundant rule applications: a rule application is not performed if the added information is already present (as such or in a specialized form). For instance, the relation ancestorOf is not added when parentOf is already present, because $\text{parentOf} \leq \text{ancestorOf}$. When no new rule application can be found, forward chaining stops.



- You can also compute the saturated fact directly with the "saturate graph" button. To test it, click the back button to come back to fact and rules selection, then click next and run the saturation process.



- To store the saturated fact, click on the "store resulting graph" button: in the Results window, a line beginning with "modified graph" appears. Drag-and-drop this line into the fact set. Rename the graph (right click on its name) with the name **ancestors-fact**. Arrange the saturated fact to make it as readable as possible.

4. Query **ancestors-fact** with the queries QancestorOf ("find all x such that x is ancestor of F") and Qancestors ("find all x and y such that x is ancestor of y")

5. Exercise:

(1) Saturate the genealogical graph **genealogy-fact** with all rules in the set **Rules/family-relationships** (step by step or directly) and store it with the name **all-deduced-fact**. To arrange this fact, try the automatic arrangement.

(2) Query this graph with a query asking for all nephews of B (that is all persons whose B is an uncle).

Remark: even if Cogui does not perform redundant rule applications, **all-deduced-fact** is redundant (you can check it with the "analyze facts" assistant). Understand why.

6. When rules produce new unknown individuals, forward chaining may not stop. It was not the case with previous rules, even if some of them introduce new unknown individuals (e.g. R7). The rule Rp (stating that "everyone has a mother") yields an example. Run it on the fact **A-fact** (with a single concept node A).

Note: when you use the "saturate graph" functionality, you are able to stop it whenever you want. You can even store the currently computed fact. However, be careful: this graph can be very large!