

## Jekejeke Minlog 0.5: First Extension Module

Date: July 24<sup>th</sup>, 2012

### 1 Introduction

Jekejeke Minlog is the first extension module for Jekejeke Prolog. The module has already been conceived in 2011 but it could be only completed now after a couple of months of development time. This module delivers a component for forward chaining. Based on it the module also delivers a chart parser and a constraint solver.

### 2 Forward Chaining

The Jekejeke Minlog is not simply a library of Prolog predicates. Moreover the module extends the Jekejeke Prolog interpreter by a new predicate property `delta/0`. This property is used to distinguish normal clauses from forward clauses. Forward clauses are translated in a special way to normal clauses so that they can be used to compute new facts from old facts.



Facts can be added via the new predicate `post/1` and the forward chaining component will automatically compute the closure. So that the forward chaining can be dynamically called during the execution of a Prolog program, the facts will only be temporarily stored in the knowledge base. For this purpose the changes on the knowledge base are kept in a list and automatically undone during backtracking. The mechanism of the `sys_unbind/1` predicate introduced in Jekejeke Prolog 0.9.3 was helpful for this purpose.

### 3 Chart Parser

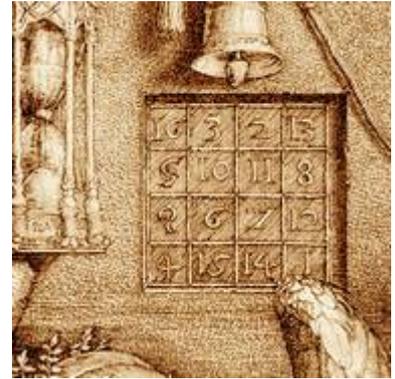
When we were integrating the chart parser, the question was whether a new operator should be introduced for the grammar rules. We decided also to test for the predicate property `delta/0`. A difficulty existed in tricking the existing DCG mechanism to accept additional rules. We decided to introduce the new directive `advice/1` which permits the late addition of clauses at the beginning of an existing predicate. The module extends the existing DCG mechanism this way.

The chart parser uses `'D'/3` facts for terminals with positions specification in resemblance of the `'C'/3` rule of the normal DCG parser. When working with the chart parser we learnt about the advantages of the asymmetric conjunction `(&)/2`. This conjunction calculates new facts only on the left side and it is sufficient to equip the first goal of grammar rule with it. If the terminals are supplied to the chart parser from right to left instead from left to right as usual, then one can find chart parsers that also allow semantic actions `{}/1` and which show a good performance.

## 4 Constraint Solver

We realized a constraint solver that keeps the constraints in the knowledge base as a further application domain of the forward chaining. For this purpose we had first to enhance the forward chaining by a mechanism to delete facts. Our inspiration here was the syntax of the push-back found in normal DCG rules. In the head of a forwards clause it is now possible to denote facts that will be removed separated by a comma.

By means of the delete mechanism the constraint solver is able to perform real simplifications in the knowledge base and thus for the constraints in that multiple constraints can be replaced by a new constraint. We then went on and tried different variants of CLP(FD) to finally choose a variant with directed interval consistency. During this work the forward chaining has been extended a second time by adding the possibility that a forward clause can indicate an abort. This is helpful when unsolvable constraints are detected.



## 5 Outlook

A first version of the chart parser and a variant of the constraint solver have been presented at the TaCoS 2012. There we also demonstrated the link to minimal logic, the name giver of the module. As a next step we will complete the tests for the application domain as noted at TaCoS 2012. We might also detect some performance improvements in the time to come. An evaluation version is available at <http://www.jekejeke.ch/> and can be tested by all interested parties.

Jan Burse  
Managing Director