tuProlog Eclipse plugin – version 0.5.0 (July 2011)

1. Installation

Currently, no automated installation support is provided. So, installation just amounts at physically copying the jar file in the "eclipse\plugins" folder (when Eclipse is not running); restarting Eclipse, the new toolbar and related menu items should be available.

2. Step-by-step user guide

As a first step, a new tuProlog project has to be created; then, a new Prolog file (*.pl) has to be added to the project. One can do so by just pressing the "New tuProlog Project" and "New tuProlog Theory" buttons in the tuProlog toolbar (**Figure 1**). More precisely

- pressing the "New tuProlog Project" a dialog appears (**Figure 2**), where the project name can be chosen (default: "My_Prolog_Project") and the desired Prolog libraries can be specified (the default libraries are proposed); alternatively, the same result can be obtained via menu selection (File > New > Other > tuProlog > tuProlog Project).

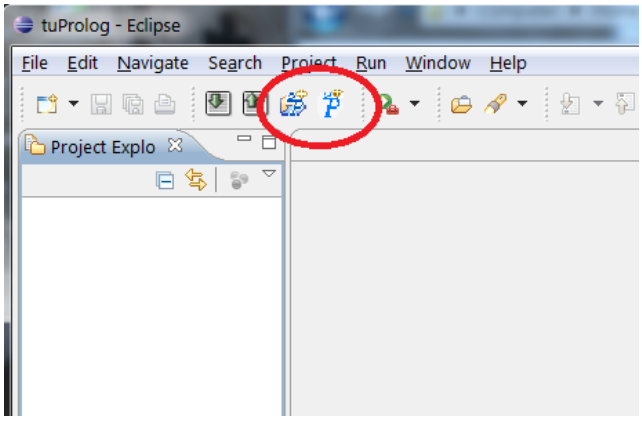

Figure 1: the tuProlog toolbar

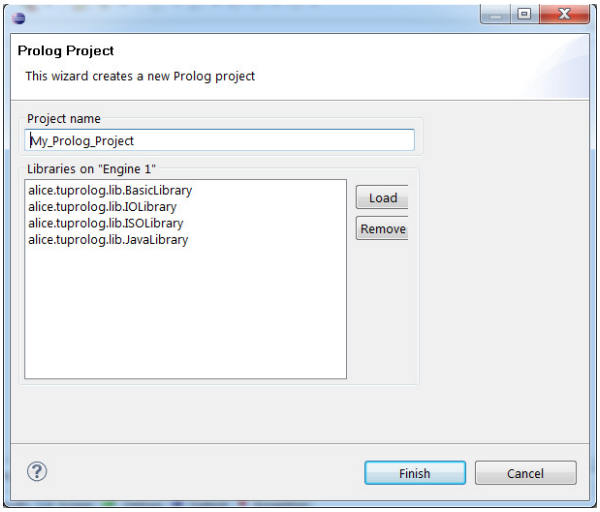

Figure 2: new tuProlog project

- pressing the "New tuProlog File" button, a dialog appears which asks the user to specify the theory name (default: "new_theory.pl") and the file *container*, i.e. the tuProlog project where the new file has to be added (**Figure 3**); this is a mandatory argument. Pressing the "Browse.." button, a new dialog proposes the currently available tuProlog projects (**Figure 4**); again, the same result can be achieved via menu selection (File > New > Other > tuProlog > tuProlog Theory); then, the new theory can be written in the editor.

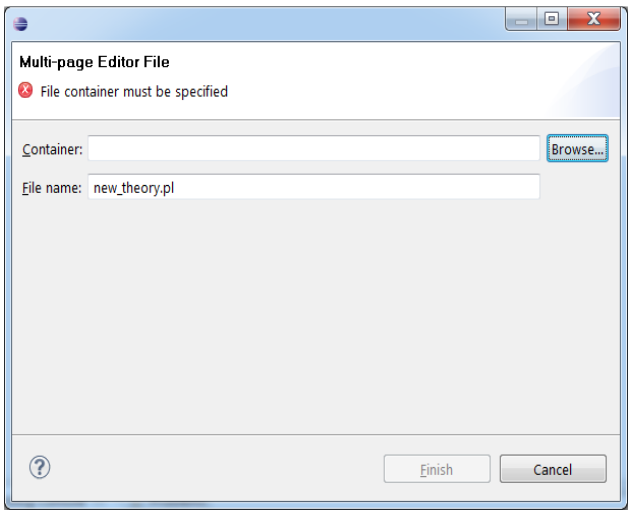

Figure 3: new tuProlog file

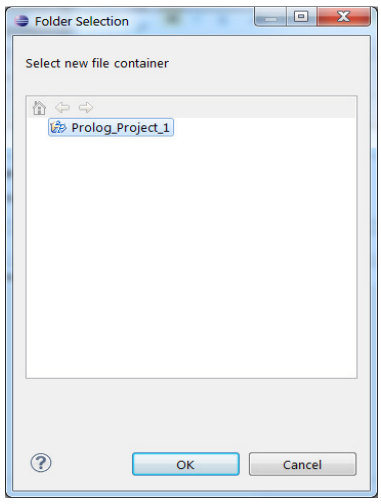

Figure 4: new tuProlog file > Browse..

This automatically opens the tuProlog perspective (**Figure 5** below).

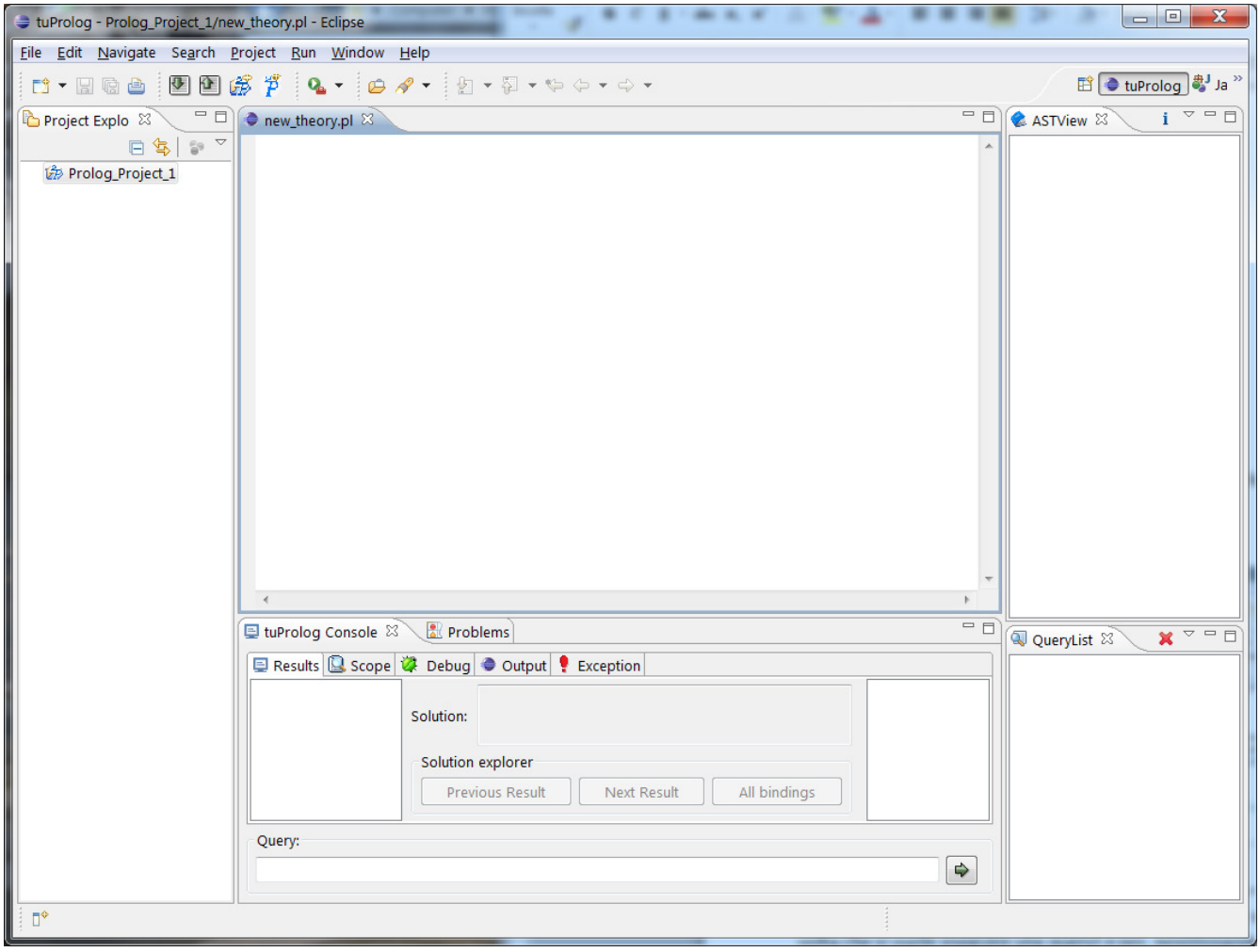

Figure 5: the tuProlog perspective

However, this perspective can also be opened via the Window >Open Perspective menu (**Figure 6**).

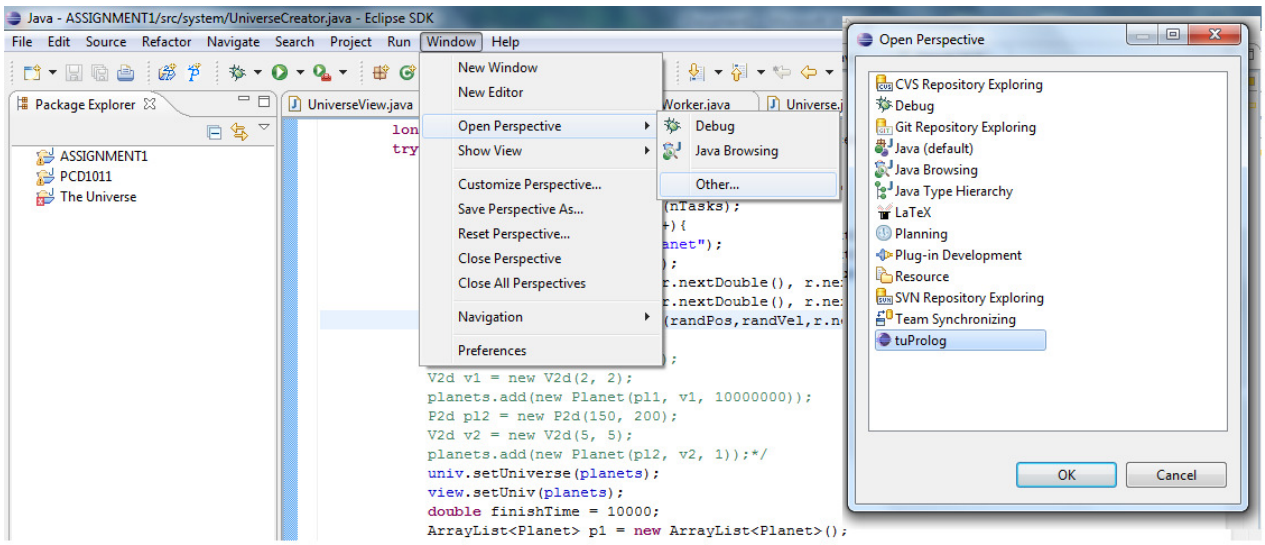

Figure 6: opening the tuProlog perspective

Once the theory has been written, the theory file must be saved (either clicking the save icon in the toolbar, or choosing the File > Save option, or by hitting CTRL+S on the keyboard): *this is mandatory before issuing any query*. The query can be written in the bottom of the tuProlog console, and is executed by either pressing the Enter key, or clicking the Solve button in the GUI.

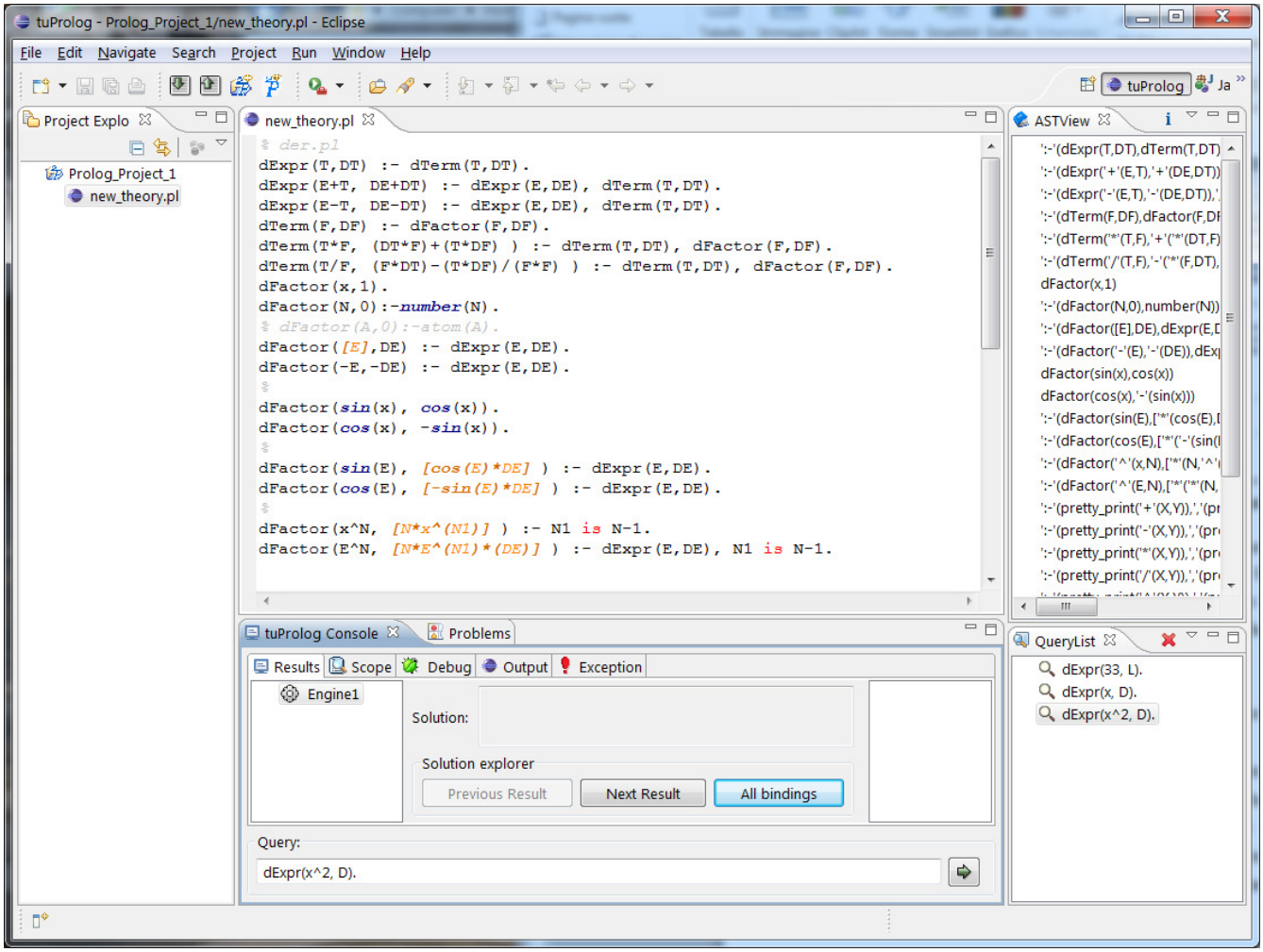

Figure 7: executing queries, available views

The query results are shown in different views (**Figure 7**):

- *the tuProlog Console view* reports the query results; the variable bindings are also available, pressing the "All bindings" button (**Figure 8**).
- *the Output view* shows the program output messages
- *the QueryList view* on the left side reports the list of all he executed queries, which can then be re-selected and re-executed in a click
- *the AST view* shows the (dynamic) set of current clauses: pressing the "i" icon, a graphical view of the Abstract Syntax Tree produced by the Prolog parser is shown (**Figure 9, Figure 10**).

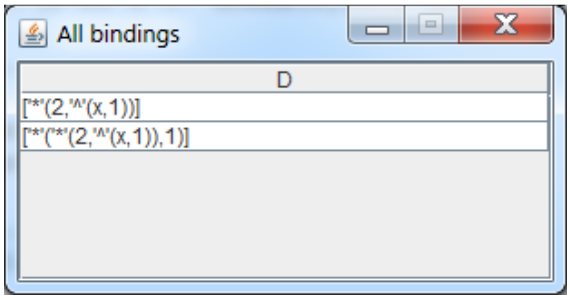

Figure 8: All bindings

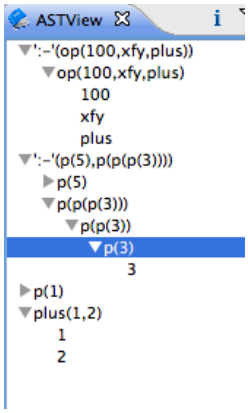

Figure 9: AST graphical view (expanded)

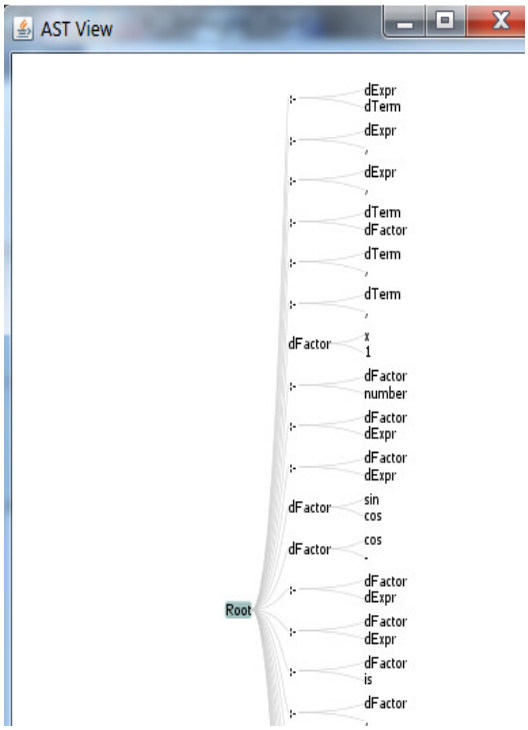

Figure 10: AST graphical view

3. GUI features

The plugin GUI provides several new features:

- error line number now shown in editor
- multiple errors now reported (not only the first)
- dynamic token colouring in syntax highlighting for operators and functors – in particular, syntax highlighting now supports also dynamically-declared operators
- direct warning in editor for undeclared terms (**Figure 11**)

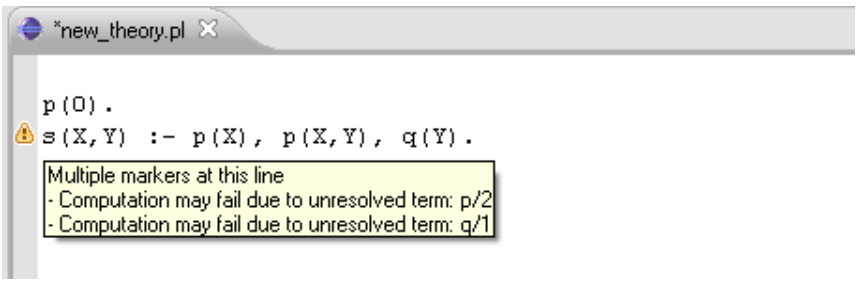

Figure 11: warning for undeclared terms

In addition, it is worth emphasising that multiple tuProlog engines can be handled simultaneously: each engine can be selectively loaded with each own set of libraries and theories, and can be separately queried.

4. Development information

In order to enable the plugin to work with any version of tuProlog, a separation layer was needed to decouple the plugin from the tuProlog engine. In fact, since the plugin needs to exploit the tuProlog engine's parser and interpreter, any change in these components would otherwise cause a refactoring of the plugin itself. This is why the development of this plugin was performed in parallel with a refactoring of tuProlog, where new interfaces [and factory classes] were introduced in version 2.4 ("alice.tuprolog.interfaces" namespace) precisely for this purpose. Such new classes include Parser, Prolog, Engine, OperatorManager, PrimitiveManager.

5. Known issues

- Performance problems in the IDE might be noted when the editor is reconfigured through some time-consuming methods (e.g. whenever an operator, defined with an op/3 directive, is defined/renamed/removed)
- Other (minor) performance problems in the IDE might be noted when modifying the current theory, since the current handling is quite trivial (namely, each change in the theory text causes a global comparison between the former and the actual theory texts)
- Terms asserted in queries are added twice (instead of once) to the dynamic theory

6. Acknowledgements

The following studens have strongly contributed to the development of this new version of the tuProlog plugin: *Andrea Mordenti, Marco Prati, Francesco Fabbri.*