LotrecScheme's manual

1 What is LotrecScheme?

LotrecScheme is a tool enabling to write graphically a tableau method. You can also execute the tableau method on a premodel in order to know if a formula is satisfiable for instance. The software is available at

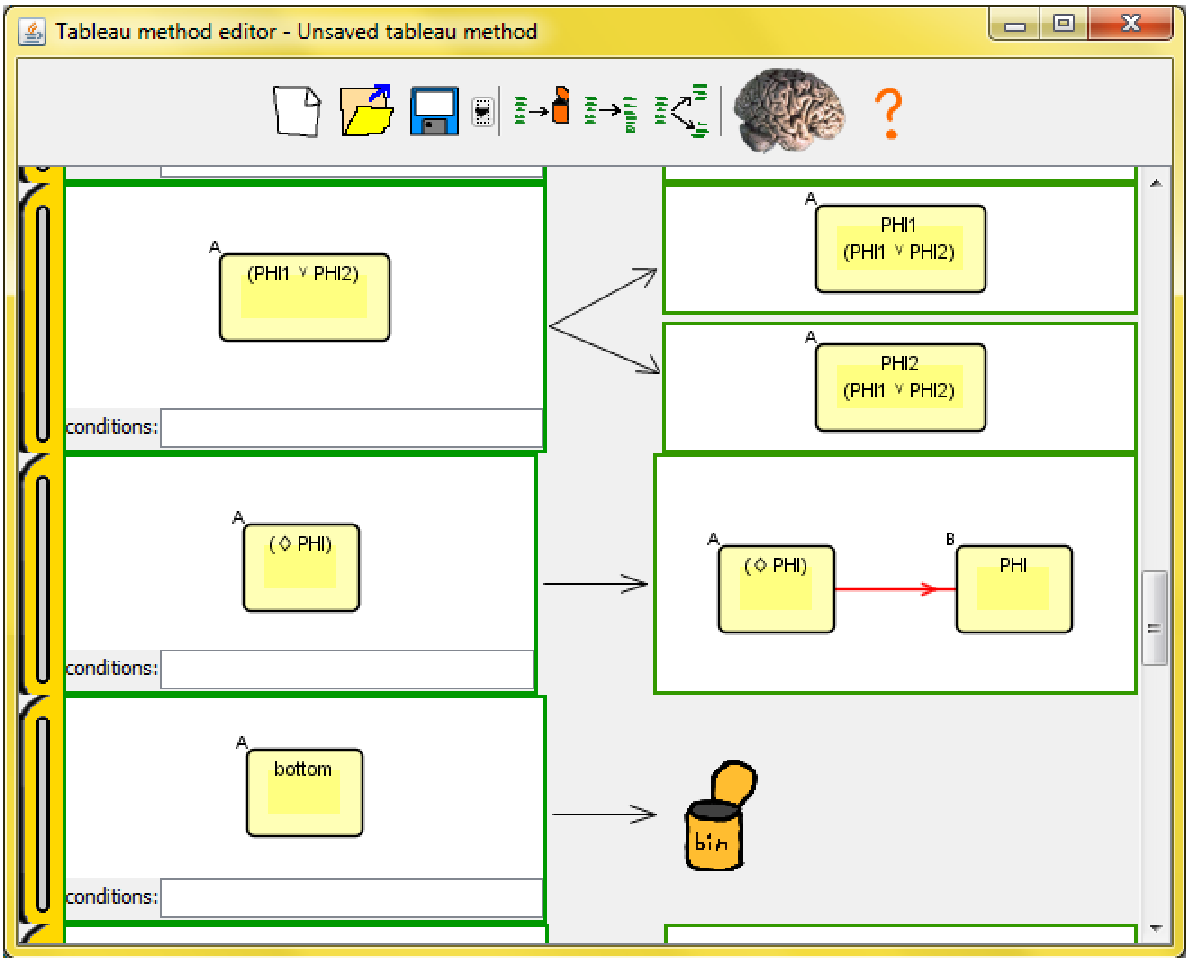
`http://people.irisa.fr/Francois.Schwarzentruber/lotrecscheme/`

You can also find files for tableau methods for logic K, KD, KT, S4, S5, S5$_n$-PAL, etc.

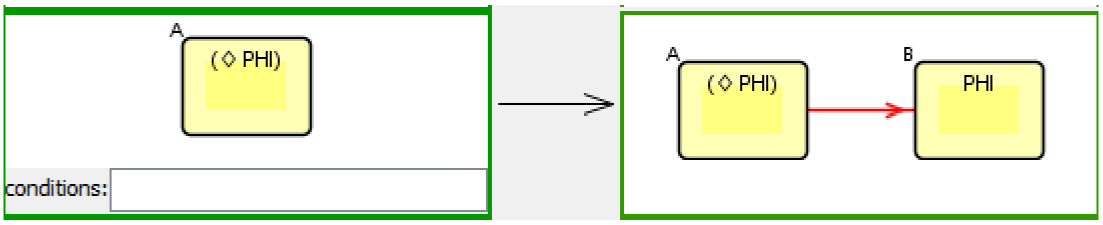
This software is inspired by Lotrec.

2 Editing the tableau method

The interface for writing the tableau method looks like:

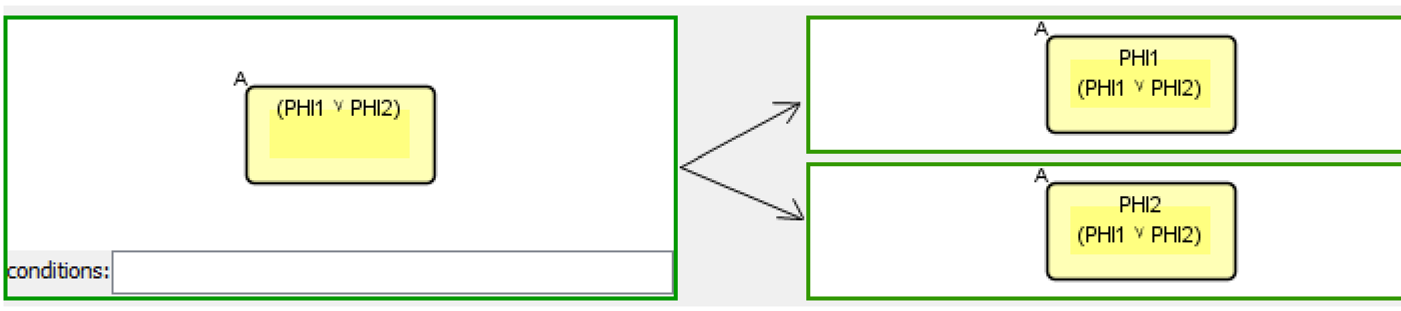


A tableau method is a list of tableau rules. A rule is a left hand-side and a right hand-side corresponding to a graph rewriting rule. For instance:



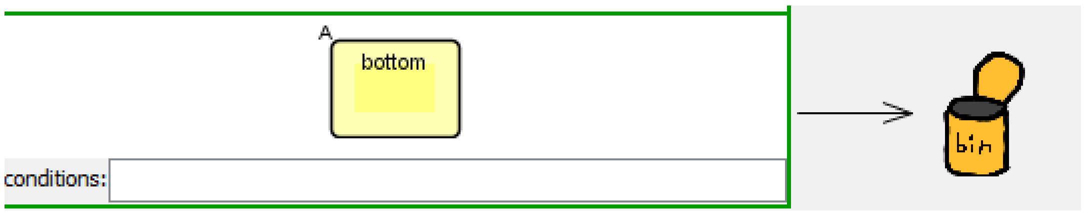
This rule means that if you have a graph in which there is a node containg a formula of the form $\Diamond\varphi$ then you will add to it a successor and add to this successor the formula φ.

A rule can also been non-deterministic: two futures are possible. For instance:



If a node contains $\varphi_1 \vee \varphi_2$, you can rewrite your graph in two ways: either you will add φ_1 or you will add φ_2. One of the execution has to succeed.

A rule can also been halting. For instance:



If a node contains "bottom" then you stop the execution. The tableau method will backtrack to the last possible execution due to non-deterministic rules.

2.1 Adding rules

You can add by clicking on buttons on the top of the screen.

2.2 Edit rules

You can edit rules by drawing graphs for left hand-side or right hand-side. To draw graphs, simply use the mouse to draw links. Use right button of the mouse to add nodes, edit nodes' contents, edit nodes' names, edit edges' contents, delete nodes, delete edges.

Formulas are directly written in a Scheme formula. For instance:

- p;
- (p or q);
- (p or (diamond e));
- (p and (atlbox (1 2) q));

- (diamond (p or q)).

The syntax is completly free except that the structure is given by parenthesis.

Do not write:

- diamond(p or q);
- p and (a or b).

Variables for pattern-matching are written in upper-case characters. For instance:

- (diamond PHI) means any formulas of the form (diamond ...) for instance (diamond (p or q)) or (diamond (p or (diamond q))).
- (diamond phi) means exactly the formula (diamond phi) and there is no pattern-matching.

2.2.1 Conditions of the left-hand side

In the field "conditions", you can write conditions in Scheme. For instance (proposition? P) means that the rule will be applied only if P matchs with a proposition.

For help on Scheme you can have help in the software. You can also read the book "How to Design Programs" by Matthias Felleisen, Robert Bruce Findler, Matthew Flatt and Shriram Krishnamurthi.

2.2.2 Right hand-side

Every new variable is instanciated with a new symbol. You can also execute Scheme code in the right hand-side of a rule. If you write (+ N 1) in the right hand-side then the system will add (+ ... 1) where you replace ... by the value of N. But if you write ,(+ 'N 1) then the system will compute the successor of N and add it to the node. (the comma says the system to eval the expression and to consider it as a term)

3 Execute the tableau method

3.1 Starting the tableau method

Before executing the tableau method, you can edit the initial premodel. Traditionaly, for the tableau method for K for instance, and if you want to test the satisfiability of a formula φ, you just have to add the formula φ in the unique node and press "start the tableau method".

You can also start the tableau method with a premodel and the tableau method may find a model in which the initial premodel is embedded.

3.2 Executing the tableau method

The software produces a ‘movie’ of the execution of the tableau method.