

INTRODUCTION

There are a great number of logics, different to themselves, and at times it's necessary to compare them or their results. So it come forth the idea of translate and consequently algorithms to make it.

Currently, the proliferation of logics used in philosophy, computer science, artificial intelligence, mathematics and linguistics makes a working reduction of this variety and urgent issue. The aim is threefold:

- To be able to use only one deductive calculus and a unique theorem prover for all of them.
- To avoid proving the metaproperties of the different existing logics by borrowing them from MSL.
- To compare to themselves the different logics, using the comparison between the heterogeneous theories that characterize them.

Modal logic is an essential component in every application that is related anyway with knowledge representation. In general, it's necessary when a non-static thing is analysed, that's to say, changes its status (situation, time instant, "world"), so that the sentences truth-value is subjected to those changes.

As an example of dynamic object, and related to computer science, we can use a program or a computer application. Dynamic logic, that is a modal logic extension, will give us a better study, since we consider the application like something that makes our computer to change its status.

This logic makes an association between a state and the computer memory registers contents in a time. As a program execution result, those contents are changed and the sentences that were truth before run the program could be false after it.

It would be very useful to logicians and scientists to have a tool that could translate the dynamic formulas into other kind of more versatile and effective logic that would be in many matters related to computer science. This logic is MSL.

The most important property is that it is logic unifier logic; that is, if we are able to translate formulas from other logics, as in this case dynamic formula, we can profit from the study, comparison and theorem proving from MSL.

DINAMUL

Dinamul is a logic translator that translates dynamic logic into MSL, in Spanish language. As this application is not for users with a great computer knowledge, it has an intuitive and easy to use user interface. The system lets with simple screens, dialog boxes, menus and shortcut keys, the navigation across the application, without need of computer knowledge, although it is necessary to know dynamic logic.

PROGRAM CHARACTERISTICS

When you start the application you must choose running way (viewer or translator) and change it later with only a mouse click.

Dinamul offers the ability of following translation step by step, with an edition environment, with a text box, where those steps are shown. Anyway watching only the final results, with a translate environment where appears an input line (that takes the string to translate) and an output read-only line (where appears the translated string).

Our application help system is clear and easy to use. It has the Windows help format because it is the user more familiarised can be with. This way, the user can access immediately to the desired help topic, user hyperlinks or the browser without

need to read all the help files. The help file includes some basic information about MSL and PDL, as well as a program description, its running way and an example.

It incorporates a store system using files, for not to translate formulas translated before. Those files opening and storage is made with a dialog box to make it easier.

You don't need to close current translation and open a new one each time you want to make a new translation. Every translations made during the same session can be stored in the same translate file. You can store each translation in a different file too.

Dinamul is a multi-document application; that is, you can have more than one translation file opened. This lets you have several translations on the screen at the same time.

The string input is made with buttons, whose icon represents the character that will be added to the input string. This avoids the dynamic logic alphabet characters lack in common keyboards problem. In other way, we can avoid the annoying key combinations to represent those characters.

You can write the variables for the translation or use the program ones to avoid their input. In this way the kind of variables to use are in the users hands.

During the string input you can use the clipboard functions, as cut, copy and paste, as well as deleting the last character or whole string. This avoids the string rewriting when a mistake occurs.

If cutting or copying actions induce to a string that does not match the dynamic logic syntactic rules, the system will advise it and erase the string, avoiding a wrong string.

If the clipboard format is wrong, that is it is not text, the system advises it and ignores the action to avoid a system failure.

To avoid a later translation string parsing to detect syntactic or semantic mistakes, right buttons are enabled or disabled on each click, that is after each click you will only can push the buttons whose character, put in the string, induces to a right sentence. This way, the dynamic logic syntax is kept.

After the edition operations described before, the system does a string parsing making the previous modifications in the buttons appearance according to the final input string status. This avoids forbidden operations according to the system status.

Finally, there is a printing option to can make a translation paper copy without have to hand-write it. This option is especially useful when you work with complex strings (those have many steps) or when you have made several translations in the same session.

The system also lets you to configure the printer, as well as choose one, if there are more than one printer, printing formats... All this is done using dialog boxes to make it easier to use.

This application has an installation program that leads the user, using screens that describes these process steps, for a correct program running.