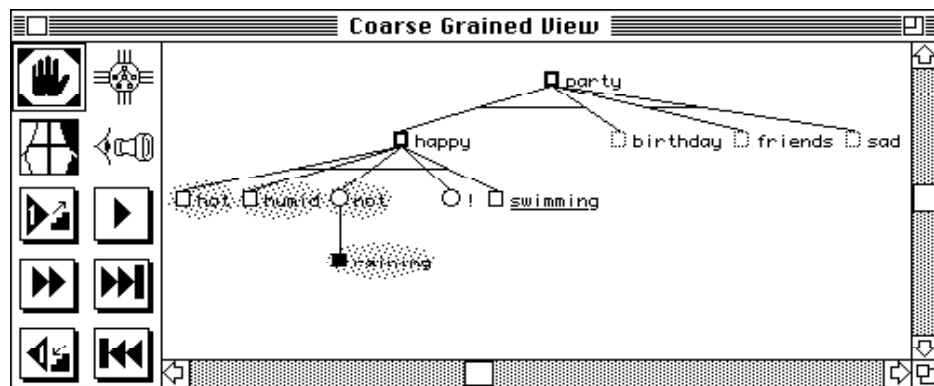


C

TPM for Macintosh *version 1.1, September 1991*

by

Fred Kwakkel



based on the Transparent Prolog Machine of
Marc Eisenstadt & Mike Brayshaw

Human Cognition Research Laboratory
The Open University
Milton Keynes MK7 6AA
UK

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

1. Introduction

1.1 Philosophy of TPM

A clear and consistent execution model of any programming language can lay the foundations not only for a good learning experience, but also for a smoother design/edit/run/debug cycle. The Transparent Prolog Machine was developed in attempt to construct precisely such a model for the logic programming language Prolog, based upon a notational extension of logic programming's traditional AND/OR trees. This extension, called the 'AORTA' diagram, is an **And/OR Tree**, **A**ugmented to include invocation history 'status boxes' at each node. This augmentation makes it possible to present a graphical view of Prolog execution which is very compact, yet which contains complete details of unification and control history, including multiple (backtracking) invocations and extra-logical features such as the cut. This fine-grained view can readily be integrated into a comprehensive coarse-grained view such as that required for understanding the execution of very large programs.

The TPM approach to Prolog debugging is simply this: you run a program using TPM, which stores a compact summary of the complete execution history (condensed, as appropriate, for speed); you then observe the coarse-grained view of the final (post-mortem) state of execution, use a collection of 'video replay' buttons and 'pause ports' to rewind or fast-forward the stored history of execution to somewhere interesting, then 'zoom in' to a fine-grained view to see precisely what happened. You can look at a variable in a fine-grained view, and ask to see the place at which its instantiation (binding) originated in the coarse-grained view. Both views are synchronized, and in general the fine-grained view gives you lots of detail at the expense of speed. A conventional textual trace can be displayed at the same time, and links to the source code and to a static 'call graph' are also provided.

TPM notation is already in use across a range of media, including textbook diagrams, video animations, and graphical tracing and debugging facilities running on modern graphics workstations. Section 5 contains pointers to detailed written descriptions of TPM, and section 6 tells you how to obtain the Open University's *Intensive Prolog* course materials, which rely extensively on TPM notation.

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

1.2 TPM for Macintosh

This implementation of TPM is for both novice and expert users. Novice users will probably profit most from the fine grained view in which they can see exactly how variables share and become instantiated and they can step through the execution trace to get a detailed understanding of how Prolog code is been executed. Expert users can turn off the AORTA mode (see TPM Preferences...) and use only the coarse grained view window together with

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

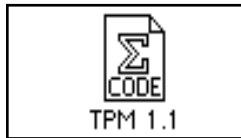
Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

the textual tracer to have a fast debugging system with all the advantages of having graphics to see the overall execution space of the program.

Because execution trees can grow very large, TPM has three features to trim the size: (i) The *trace depth cut-off* option provided by TPM Preferences... prevents the execution tree from growing too large downwards (but allows you to re-expand 'terminal nodes' later), (ii) the *compress nodes* feature lets you select those procedures for which you don't want to see the subtree (and likewise allows later expansion), (iii) the *selective trace* feature lets you place a spy point which signals where the detailed graphical trace should begin.

1.3 Getting started

To get TPM working, do the following:



Double click on the TPM icon to start both Demo MacProlog and TPM.

File

Open...

Load your own Prolog program to be debugged by selecting the Open... command from the File menu, or type fresh code into a newly created window. An Examples folder is supplied with the disk.

TPM

Traced Query...

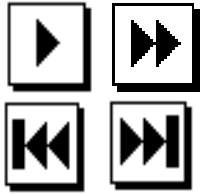
Select the Traced Query... option from the TPM menu and enter your query. The execution tree will come up automatically in the Coarse Grained View (CGV) window. When you see the tree displayed, *execution has already taken place, and a*

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

complete history has been stored internally.



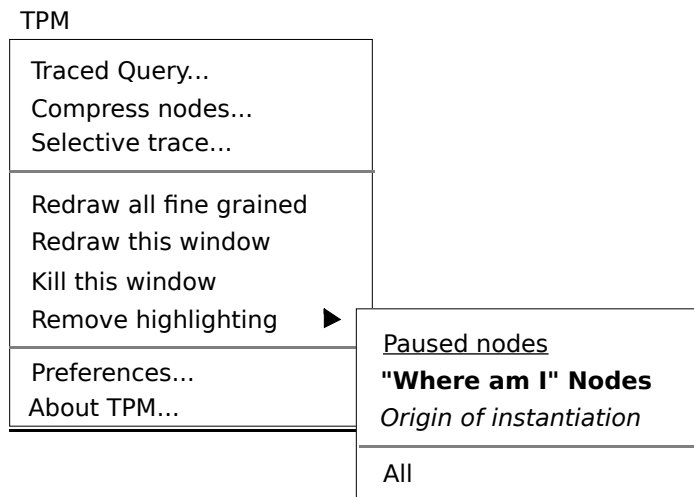
Use the ‘video replay panel’ buttons in the CGV window to replay execution. If you select Textual tracing: • Normal in the TPM Preferences... window, you will see a simultaneous textual trace of the kind familiar to users of other Prolog trace packages. This will help orient you to the CGV notation (see section 3 below).



You can now open Fine Grained View (FGV) windows for nodes by first selecting the ‘zoom in on node’ tool (shown at left) and then clicking on the CGV node you want to explore. FGV notation, based on ‘AORTA’ diagrams (And/OR Tree, Augmented) is summarized in section 4 below.

2. The TPM Menu

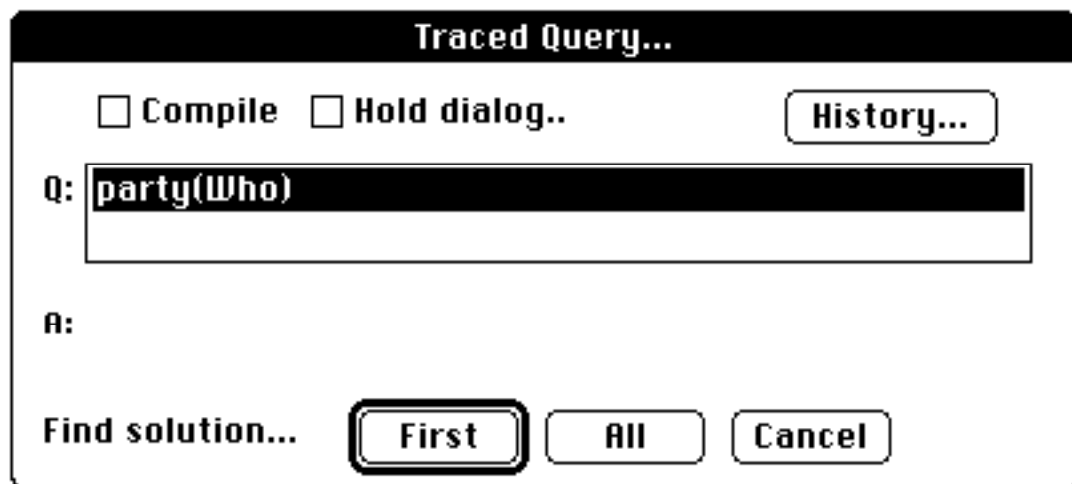
The structure of the TPM menu is as follows:



The next paragraphs explain all the menu items in detail.

2.1. Traced Query...

This brings up a dialog box in which you can initiate the trace of a query (this is how you 'run' TPM). The dialog box looks like this:



If the Compile check box is ticked, TPM will first compile all changed code and

then trace the query.

If the Hold dialog... check box is ticked, TPM will continue to show this dialog box after it has traced the query so that you can see the answer.

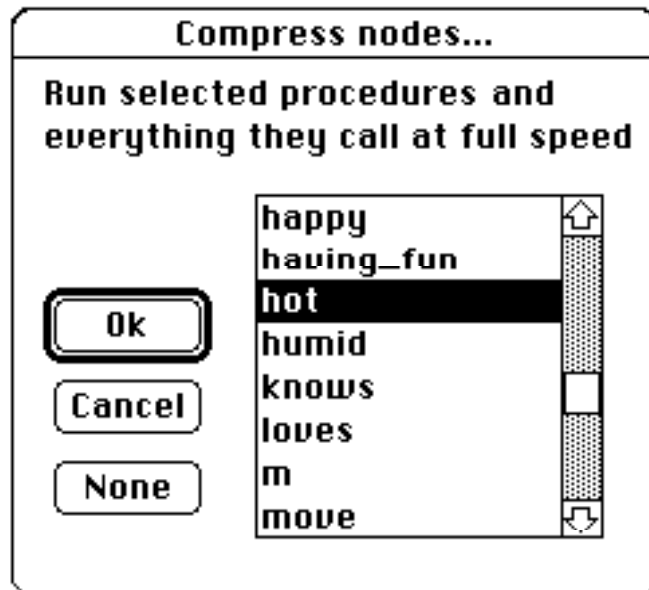
If you click on the History... button, TPM will show you a list of the previous queries. You can select one by clicking on it and selecting Ok or you can empty the list of previous queries by clicking on None

If you click on the First button, TPM will trace your query for only one solution and draw the tree in the CGV window.

If you click on the All button, TPM will trace your query for all solutions (by doing a fail after each solution) and draw the tree in the CGV window.

2.2. Compress nodes...

Compressed nodes are procedures which are executed at full (i.e. untraced) speed. They will be represented in the Coarse and Fine grained view windows as triangles, meaning that there is a whole compressed subtree underneath it (the 'refocus' tool, described in section 3, effectively 'uncompresses' such nodes). There are different ways of selecting what procedure(s) must be compressed. One is selecting the compress nodes... menu item in the TPM menu. This will bring up a dialog box where you can select the procedure(s) you want to see compressed (<SHIFT><CLICK> to select more than one). If, when you chose the compress nodes... option, your cursor was in a text window within the code for a compressible procedure (e.g. while performing ordinary editing) then that procedure will appear preselected at the top of the scrolling list.

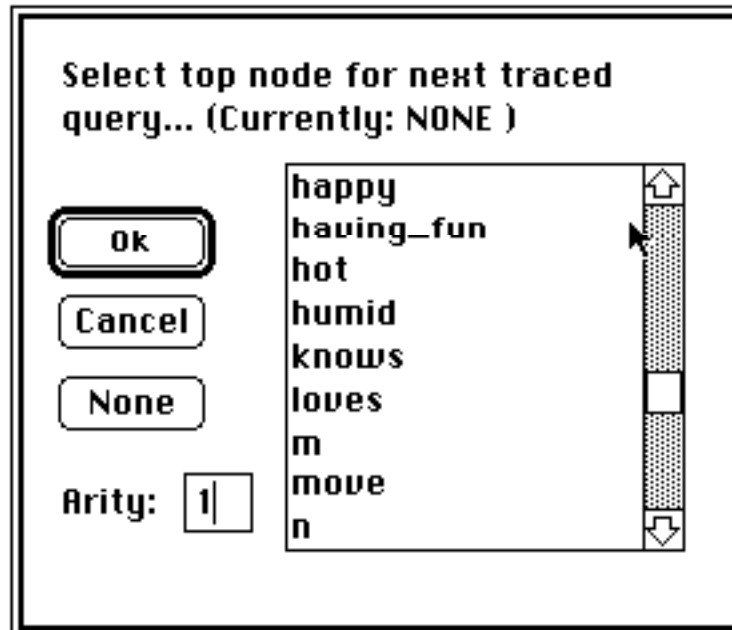


Another way of selecting a procedure to compress is in combination with the LPA MacProlog built-in 'callgraph'. When running TPM, all MacProlog callgraph windows have a new tool called TPM. When you select this tool and then click on a procedure name, a dialog box will pop-up offering you a choice of either compressing or selectively tracing that procedure.

2.3 Selective trace...

A selective trace of a procedure means that for the next query you pose, only the part of the execution tree that starts from the 'selectively traced' procedure will be traced by TPM (with the advantage that everything up until that point runs at full speed). There are two different ways of selecting which procedure must be selectively traced. One is by selecting the selective trace... menu item in the TPM menu. This will bring up a dialog box in which you can select the functor and enter the arity of the procedure you want to be

selectively traced. If, when you chose the selective trace... option, your cursor was in a text window within the code for a procedure (e.g. while performing ordinary editing) then that procedure will appear preselected at the top of the scrolling list.



Another way of selecting a procedure to selectively trace is in combination with the LPA MacProlog built-in 'callgraph'. When running TPM, all MacProlog callgraph windows have a new tool called TPM. When you select this tool and then click on a procedure name, a dialog box will pop-up offering you a choice of either compressing or selectively tracing that procedure.

2.4 Redraw all fine grained

This menu item will force all Fine Grained View windows to be updated. Normally, the graphics in an FGV window is always consistent with the current trace, but the text beside the procedure box occasionally becomes inconsistent (e.g. in certain cases of complex shared variables). Most tools in the CGV window will in fact update the FGV windows automatically and consistently, with the exception of the *play* tool and the *step back* tool. The reason is simply that total consistent updating of all FGV windows during *play* and *step back* is too time-consuming. Therefore, after using the *play* tool or the *step back* tool you can make sure that all FGV windows are consistent by selecting this menu option.

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

2.5 Redraw this window

As explained in the previous section, an FGV window's text can become inconsistent with the actual current state of the trace. If you want to update only one FGV window, bring it to the front (by clicking anywhere within it) and select this menu item. Note that 'this window' means 'currently active'

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

FGV window'. Another way of redrawing an FGV window is by holding down the mouse button within its viewing pane, at which point a special menu will pop up within which you can select redraw this window.

2.6 Remove highlighting

There are three types of highlighting for labels in the CGV window. Using sub-menus of the Remove highlighting option you can remove specific types of highlighting or all highlighting. The types (and how they appear) are as follows:

- Node labels will become underlined when there are one or more pauseports on the call/exit/redo/fail ports (see *set pause ports* tool in section 3).
- Node labels will become **bold** when you select the **Where am I** item from the popup menu in an FGV window (this allows you to identify where an FGV window is situated within the context of the overall CGV).
- Node labels will become *italic* when you ask for the *origin of instantiation* of a variable by selecting that item from the popup menu in an FGV window (this allows you to identify where and when a particular variable acquired its instantiation).

2.7 Preferences...

This menu option will give you a dialog box in which you can set and save your preferences:

Trace depth cutoff: indicates the maximum depth for the tree. All nodes that have children whose depth is higher than this depth cutoff will have a dot-dot-dot underneath their box meaning that there is a subtree that can not be shown. Use the *refocus* tool (section 3) to expand such 'terminal nodes'.

AORTA diagrams: there are three modes in which TPM can do its tracing for the FGV (or AORTA) diagrams. In the full mode all graphics and text will be shown in the FGV windows when you open them. In the No text mode only the graphics associated with the procedure box will be shown when you open an FGV window. When the AORTA mode has been turned off, no FGV window can be opened directly. If you do try to open an FGV window (by applying the *zoom* tool, as described in section 3), and the AORTA mode is off, you will get a dialog box in which you can choose if you want to

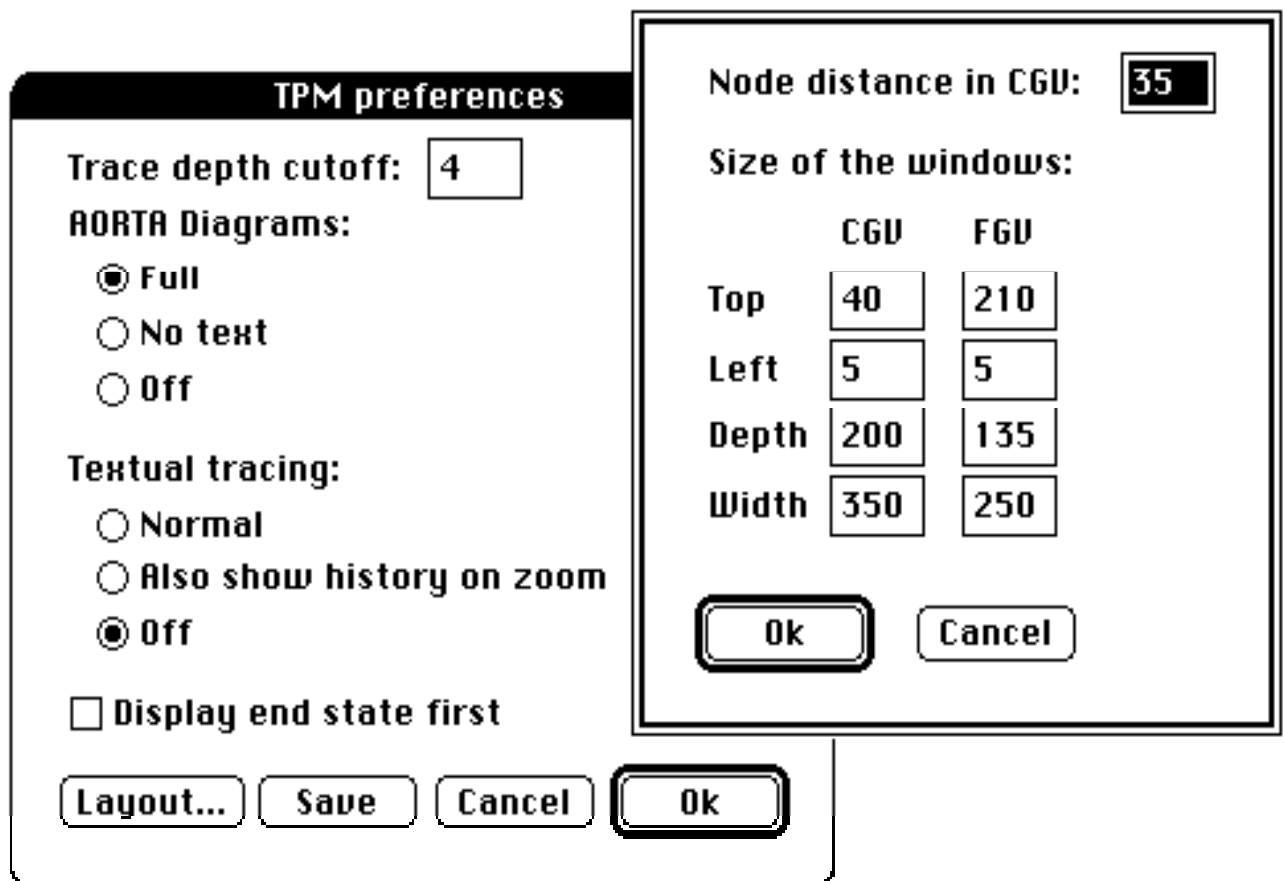
turn on the AORTA mode and then redo the tracing.

Textual tracing: there are three modes in which TPM can show the textual tracing. In the Normal mode a text trace window will be opened where you can see the call/unify/exit/redo/fail ports with all arguments when you step through the trace tree. When you select Also show history on zoom you will also see the whole history of a node when you try to zoom in on it. In the off mode the text trace window will disappear.

If the Display endstate first check box is ticked, the CGV tree will be displayed to depict the *end* of execution . If un-ticked, then the CGV tree will be displayed to depict the *beginning* of execution. In either case (ticked or un-ticked), the full history will already have been stored by the time you see the CGV tree, so that you can re-play it at will using the replay tools described in section 3.

The Save button will save all your preferences in a file called 'TPM preferences'.

The Layout... button will bring up another dialog box where you can set the sizes of the CGV and FGV windows and the vertical distances between the nodes in the Coarse Grained view window.



3. The Coarse Grained View

The coarse grained view window will show a global perspective of the AORTA

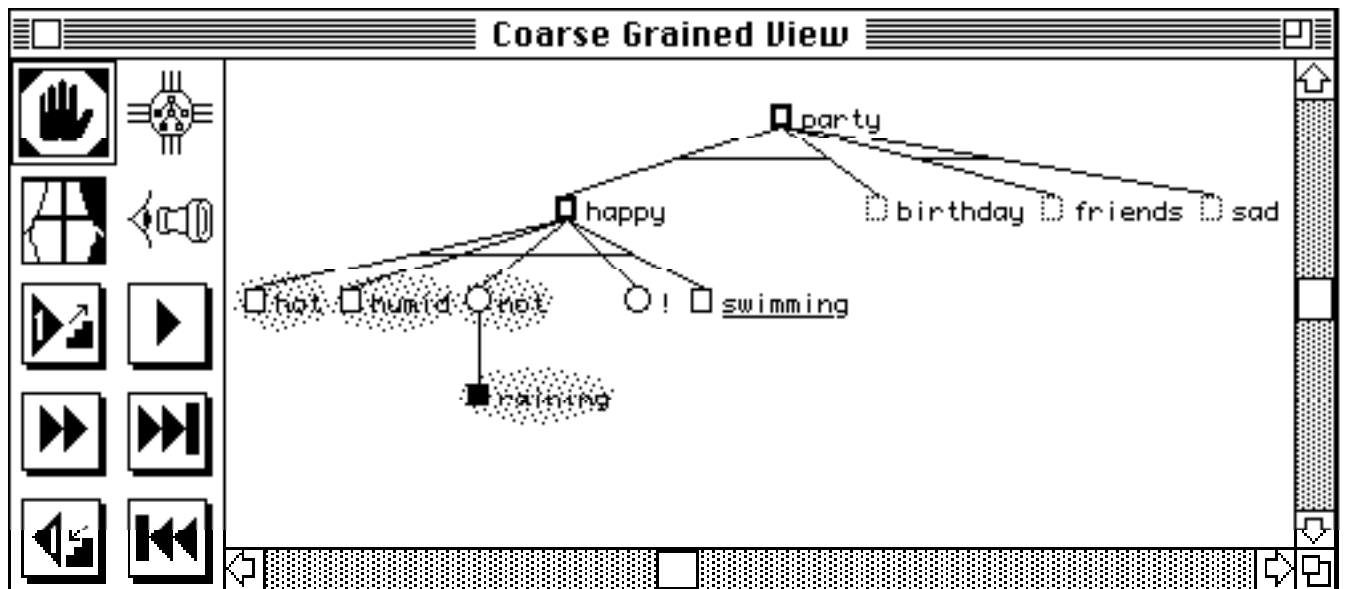
(And/OR Tree, Augmented) diagram.

To change the font of the labels of the nodes in the CGV select a new font and size from the LPA MacProlog Font menu and (re)do the query.

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

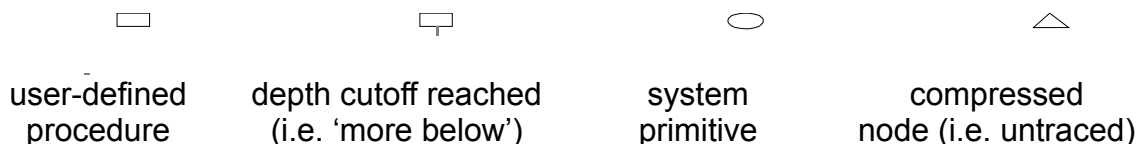
Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.



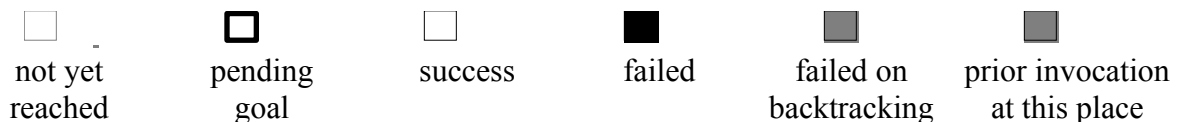
This screen snapshot is a typical situation somewhere in the middle of a trace. The node **happy** is the current pending goal (thick lines) and the clouds around the node to the left of the cut (!) indicate that these nodes are frozen (cannot be redone). The underlining of the node **swimming** means that there is a pauseport on this node. The nodes directly to the right of **happy** appear as dotted lines because they have not yet been reached during the current replay (although they were executed originally ‘behind the scenes’ in order to store the full history).

3.1 Notation conventions for the Coarse Grained View

Four types (‘shapes’) of node can occur in the CGV window.



Each node can be in one of the following six states:



For the relation between this representation and the AORTA diagrams, see the section

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
 LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
 Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

about the Fine Grained View (section 4).

3.2 Tools in the CGV window

The CGV window has the following tools:

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.



set pause ports



refocus



find procedure



zoom in on node



step



play



forward



to end



step back



to start



The first 4 tools can be used as follows: first select the tool and then click in the view pane (the CGV 'tree') directly on the node which you want the tool to act upon.

- *set pause port* tool: this will bring up a dialog box in which you can set pauses on call/exit/redo/fail ports for the node. These 'pause ports' will be operative when you select either the *play* or *forward* tool. The former (*play*) shows you every execution step in the CGV, pausing whenever a pause port is encountered. The latter (*forward*) takes you immediately to the next pause port.
- *refocus* tool: this will re-execute the last call-port of the node you click on and make that node the top node of a brand new CGV tree. This is a way of expanding nodes that were either compressed by you or were 'terminal nodes' (i.e. at the edge of the trace depth cutoff). *It is your responsibility to ensure that nodes re-executed in this way behave the same on re-execution as they did on the previous execution* (i.e. you might have to undo any tricky side-effects caused by **assert**, **retract**, etc.).
- *find procedure* tool: this will locate the text window and position of the source code for the node you click on, and bring that window up to the front, with the relevant procedure name highlighted.
- *zoom in on node* tool: this will open an FGV window if no FGV window for this node exists yet; otherwise it will bring the FGV window to the front.

The next set of 6 tools can be used for manouvering through the execution trace just by clicking on the desired tool.

- *step* tool: will step one step forwards in the execution trace. If there is an FGV window open for the current pending node, then this window will be updated accordingly.
- *play* tool: will step through the execution trace until it reaches a pauseport or you

click in the viewing pane of the CGV window (which is how you manually interrupt playback). Meanwhile the FGV will constantly be updated during the 'play'. See sections 2.4 and 2.5 for ways to ensure a consistent view within the FGV windows.

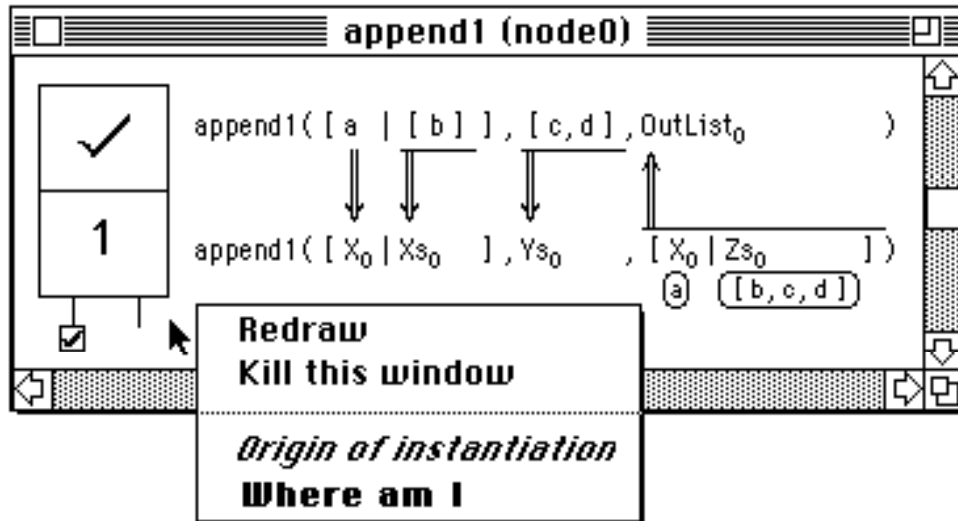
- *forward* tool: will bring you to the next pause port (or to the end, if no more pause ports are found). The CGV window and all FGV windows will be updated automatically.

- *to end* tool: will bring you to the end of the execution trace. The CGV window and all FGV windows will be updated automatically.
- *step back* tool: will step one step backwards in the execution trace. If there is an FGV window open for the current pending node, then only the graphics for that window will be updated. Because the variable instantiations are in an indeterminate state during ‘step back’, the text in all FGV windows will be removed, but can be reinstated in one of four ways: (i) choosing Redraw all fine grained from the TPM menu (section 2.4); (ii) choosing Redraw this window from the TPM menu (section 2.5); (iii) choosing Redraw from the FGV pop-up menu (see section 4.2); (iv) selecting the *to start* tool and then once again using the *play, step, forward, or to end* tools
- *to start* tool: will bring you to the beginning of the execution trace. The CGV window and all FGV windows will be updated automatically.

4. The Fine Grained View

The Fine Grained View windows show the actual AORTA diagram with the sharing/instantiation of variables and the whole procedure status box.

The arrows show the direction of instantiation and sharing: up-arrow depicts an output variable and down-arrow depicts an input variable. An FGV snapshot of the topmost goal at the end of an invocation (using the classic definition) of **append1([a,b], [c,d], OutList)** looks like that shown below.



In the above snapshot, the text on the top line shows a particular goal, while the text on the bottom line shows a matching database clause (clause number 1 in the definition of **append1**, in this case). The little 'lozenges' beneath X_0 and Zs_0 on the right show instantiations which have come 'from elsewhere', whereas the arrows depict instantiations which correspond directly. The vertical bars (e.g. underneath **[c,d]**) depict the full extent of the instantiated term. Additional notation is explained below.

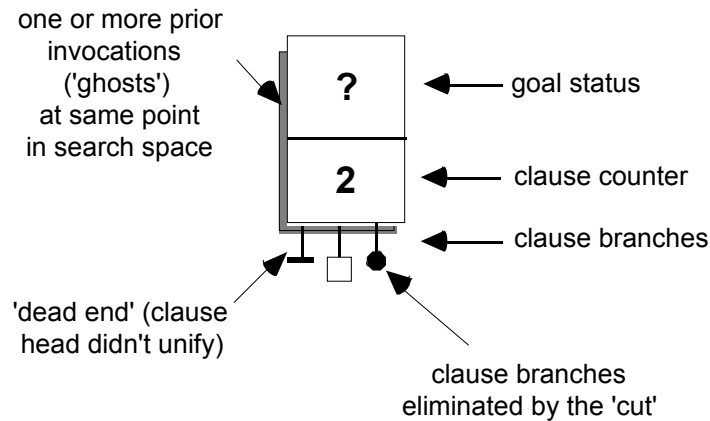
TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

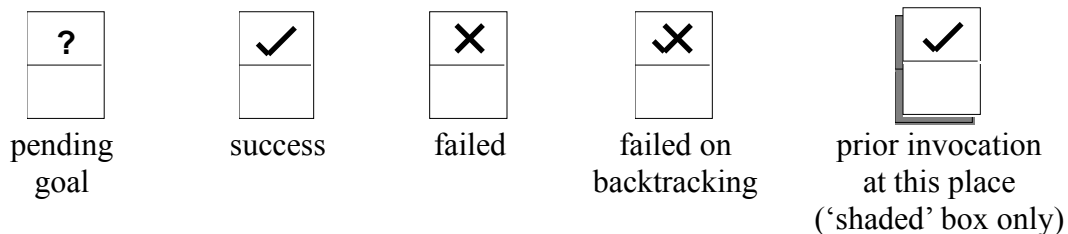
Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

4.1 Procedure status box conventions for the Fine Grained View

The meaning of all the graphical notations in the procedure status box is as follows:



The little square box beneath the '2' is a 'clause status' box, indicating the status of specific clauses during execution. The status in which a procedure box or clause box can be is one of the following:



4.2 The popup menu in the FGV window

When you click in the viewing pane of the FGV window you'll get a popup menu where you can select one of four items:

- Redraw will update and redraw the text of the current FGV window (see sections 2.4 and 2.5 for the reasons you need this).
- Kill this window will kill the current FGV window.
- *Origin of instantiation* will bring you to a dialog box where you can select the variable(s) for which you want to see the origin of instantiation. The nodes where the variable is instantiated will be displayed in the CGV tree with *italicized* labels.

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

- **Where am I** allows you to identify where an FGV window is situated within the context of the overall CGV by making the label of the corresponding node in the CGV window appearing in **bold**.

5. References to publications about TPM

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

The second article listed below is provided on the TPM disk as the file 'TPM Journal Article' in Microsoft Word version 4.0 format.

Eisenstadt, M., & Brayshaw, M. The transparent Prolog machine (TPM): an execution model and graphical debugger for logic programming. *Journal of Logic Programming*, 5 (4), 1988, pp. 277-342.. [also available as Technical Report No. 21A, Human Cognition Research Laboratory, The Open University, Milton Keynes, UK]

Eisenstadt, M., & Brayshaw, M. A fine-grained account of Prolog execution for teaching and debugging. *Instructional Science*, 1990, 19(4), pp. 407-436.

Brayshaw, M., and Eisenstadt, M. A practical graphical tracer for Prolog.. *International Journal of Man-Machine Studies*, (1991, in press). [also available as Technical Report No. 41, Human Cognition Research Laboratory, The Open University, Milton Keynes, UK]

For a full list of publications about TPM see the About TPM... menu item in the TPM menu.

6. The Open University's *Intensive Prolog* course

The notation developed for the Transparent Prolog Machine serves as the basis for the Open University's course *Intensive Prolog*. This course involves between 80 and 100 hours of study, and is available in both assessed ('course') and non-assessed ('study pack') versions. The assessed version offers 1/4 credit in the Open University's 'Computers for Commerce and Industry' profile, which allows you to accumulate credits towards a postgraduate diploma (2 full credits required), and an M.Sc. degree (postgraduate diploma followed by approved individual project).

Intensive Prolog comprises a set of course notes, a workbook, a set book (Bratko's *Prolog Programming for Artificial Intelligence*), a 120-minute video cassette, and floppy disks containing a Prolog interpreter, examples, exercises and answers (for IBM-PC compatibles as well as for Apple Macintosh). The main study material is 160 pages in length, and is accompanied by a workbook of exercise material, 136 pages in length. The course materials rely extensively on the fine-grained view of execution (AORTA diagrams) used by the Transparent Prolog Machine. For example, many of the programs developed in Bratko's book are illustrated through a carefully-worked sequence of TPM/AORTA diagram screen snapshots.

Further details, including prices, application and order forms may be obtained as follows:

DM864 (*Intensive Prolog* 1/4 credit course, including tutorials, assessment, and examination) is available from:

Associate Student Courses Office (ASCO)
P.O. Box 76
The Open University
Milton Keynes MK7 6AN, U.K.
Phone: +44 908 653449
Fax: +44 908 653744

The Associate Students Courses Office can also supply information regarding the full Computers for Commerce and Industry course profile.

PD622 (*Intensive Prolog* study pack with no assesement) is available from:

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.

Learning Materials Sales Office
The Open University
P.O. Box 188
Milton Keynes MK7 6DH, U.K.
Phone: +44 908 653338
Fax: +44 908 653744

TPM for Macintosh Copyright © 1991 The Open University (U.K.)
LPA MacProlog Copyright © 1984-1991, Logic Programming Associates, Ltd. (U.K.)
Apple Macintosh™ is a registered Trademark of Apple Computer, Inc. (U.S.A.)

TPM for Macintosh and LPA Demo MacProlog 3.5 may be copied and distributed free of charge, provided that all files on the supplied disk are copied and distributed with them. TPM for Macintosh is intended for educational purposes, and may not be sold as or incorporated in a commercial product without written permission from: The Copyrights Officer, Open University, Milton Keynes MK7 6AA, U.K. The Open University accepts no responsibility for any legal or other consequences which may arise directly or indirectly as a result of the use of all or parts of the contents of this program.

Fred Kwakkel spent 6 months at The Open University on a work placement programme as part of his studies at the Hoger Informatica Onderwijs, Enschede, The Netherlands, for which we gratefully acknowledge support.