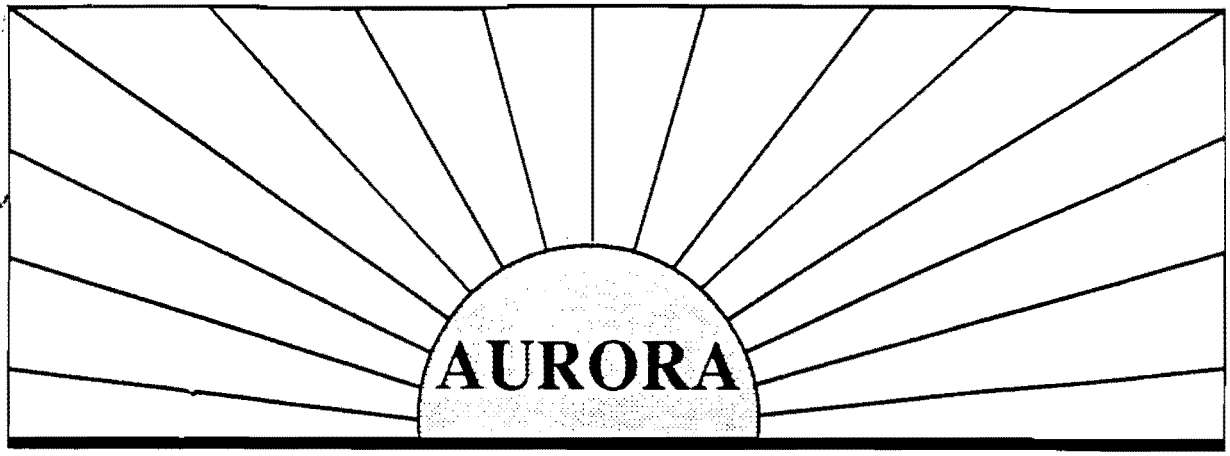


User's Guide
to

AURORA 2.0

A Discovery System

A Personal Inference System for Automated Rule Learning, Conceptual Data Analysis and Building Advisory or Expert Systems

International Intelligent Systems, Inc.

9711 W. Maury Road
Fairfax, VA. 22030

Copyright

Limited Warranty on Media and Manual

Even though International Intelligent Systems, Inc. has tested the software and reviewed the documentation, International Intelligent Systems, Inc. makes no warranty or representation, either express or implied, with respect to this software, its quality, performance, merchantability, or fitness for a particular purpose. As a result, this software is sold "as is", and you the purchaser are assuming the entire risk as to its quality and performance.

In no event will International Intelligent Systems, Inc. be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in the software or its documentation. The warranty and remedies set forth above are exclusive and in lieu of all others, oral or written, express or implied. No International Intelligent Systems, Inc. employee is authorized to make any modification, extension, or addition to this warranty. This warranty gives you specific legal rights, and you may have additional rights which may vary from state to state.

Trademarks

IBM AT is a trademark of the International Business Machines Corporation. MS-DOS and DOS are trademarks of Microsoft Corporation.

AURORA FACTS

Minimum configuation:

IBM AT™
640 KB RAM
Hard disk, 1.5 MB free space
CGA or EGA video card

AURORA was developed in Pascal and requires approx. 1.1 MB of disk space.

INTERNATIONAL INTELLIGENT SYSTEMS, INC.
9711 Maury Road
Fairfax, VA 22030

Thomas W. Farmanian, Ph. D.
Executive Director
R.R. 5 Box 25B
Mahomet, IL 61853
(217) 586-5761

PREFACE

AURORA is a revolutionary new concept in intelligent programs for personal computers. It is an integrated inference system that allows a user:

- To search for unknown patterns in data
- To automatically build or improve decision rules using examples of decisions
- To test the rules on new examples
- To develop personal expert systems for various applications.

AURORA uses new ways of conducting inference using its knowledge base, so that it can quickly determine one or more most credible suggestions or advise.

In contrast to a more conventional style of building expert or advisory systems in which a knowledge engineer creates the system based on an interaction with a domain expert, AURORA allows the expert himself/herself to build a system. By eliminating a technical person from the process of building a system, AURORA makes it possible to quickly and efficiently build new expert systems. Moreover, since the user himself is building the system, this allows for a totally secure system, whose operation, decision making rules and important details are known only to the initial creator. The above ability has been created by very carefully designing the interaction between the user and system, and introducing very powerful learning capabilities into the system. AURORA and in particular the learning capabilities are based on many-years of research conducted by Professor Michalski and his research group at the University of Illinois. This group is a leading research group in the field of machine learning.

AURORA was therefore, designed to be a personal system. It may be installed on a IBM AT on your desktop and does not require the services of a knowledge engineer to build advisory systems. AURORA is also useful for recognizing patterns in supplied examples. This may help you quickly discover a consistent problem area or pattern of success.

AURORA represents the collective efforts of many individuals to develop a system which is both powerful in its ability to build advisory systems and to learn concepts from examples and yet still easy to use. We believe we have achieved these goals in AURORA 2.0. While there were many more features which might be useful, they could not be included in this version. Some of the fea-

tures planned for future versions include incremental learning with new examples, use of real numbers, and knowledge base testing facilities.

Items within each topic in this user's guide are presented in the order they appear on menus. Chapters are arranged in their order of increasing complexity and detail. A novice should be able to use AURORA after reading chapters 1- 4. Chapters 5-10 are intended as both an introduction to the potential of AURORA and as a reference to different components of the system.

Symbols Used in this Manual

- Text appearing as **BOLD SMALL CAPS** indicates text you should type.
- *italicized* words or phrases are copies of ones found on the screen.
- The "enter ↵ " or "return" key is represented as <CR>.
- The "control" key is represented as <Ctrl>.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE
AN INTRODUCTION TO AURORA

What is AURORA?

AURORA is a revolutionary new concept in intelligent programs for microcomputers. It is an integrated inference system capable of learning and reasoning about its knowledge. In addition, it allows the creation of an expert system (ES), with this knowledge. It can take the guise of an expert system builder (ESB). It is, however, far more versatile, as it also serves in the role of conceptual data analysis by providing a means for indicating a set of characters which best describes a concept. This is accomplished through examining a set of examples provided by you.

What is an ES builder?

An ESB is tool used to build an ES. An ES is a computer system that achieves a high level of performance in task areas that, for human beings, required years of special education and training. EBS's are not limited to giving advice in one subject area or domain rather, it can become an expert in any domain, depending on the knowledge provided to it. Typically, this knowledge is in the form of IF-THEN rules, which are created by the developer of the system. One major difficulty with ES's is that domain experts often have difficulty putting their knowledge into this form. For example, try describing all the things you know about driving a car in IF-THEN rules, and you'll see what we mean. Knowledge acquision and representation is often the most time consuming part of building an ES.

How does AURORA overcome the knowledge acquisition bottleneck?

AURORA takes the creation of ESs out of the hands of the experts in knowledge engineering and puts it back where it belongs, in the hands of domain experts. It does this by allowing one to construct a rule base in four possible ways. These are:

Direct editing.

Your favorite editor may be used to type in the knowledge of the system. As alluded to above, this requires an in-depth understanding of the domain, as well as a knack for expressing this knowledge in the appropriate format. Nevertheless,

there are many situations in which this type of knowledge acquisition is most appropriate.

Learning from examples

A powerful way to learn rules, this method permits the encoding of knowledge into rules by first entering appropriate examples of expert decisions. The knowledge gained by the system through this route may outperform a human expert in a given domain.

Optimizing existing rules.

Often rules created by human experts are good if taken as rough guides, but when taken literally contain insufficient or poorly expressed information. AURORA has the ability to take these rules and optimize them according to various criteria.

It should also be stressed that you can use any combination of the above methods. For example, you could first learn rules from examples, then augment them by hand, and finally optimize them. Such a spectrum of possibilities is not available on any other system.

What makes AURORA different than a typical ES builder?

All ESs are limited by the knowledge that is given to them. What is often not stated, however, is how difficult it is to provide them with this knowledge. AURORA reduces this difficulty, by providing the three methods described above for knowledge acquisition. These same capabilities permit the system to act as a tool for discovering regularities and hitherto unknown relationships in data given to AURORA.

What Sorts of Data Analysis Can AURORA Do?

AURORA can analyze data conceptually to discover new relations within the data. The induction (learning) module that comes with AURORA may be used to discover unforeseen relations among data. Learning in AURORA is not another implementation of the ID3 algorithm but a new concept of "learning by example" based on the AQ11 algorithm. Unlike traditional statistical analysis programs, which are limited to correlating "numerical" variables, AURORA uncovers regularities in "logical" variables. Such variables can be of three types, nominal, such as color or location; linear, as in the case of size, which might for example take the ordered values of small, medium, or large; or structured values, declared by the user. An example of a structural variable may be a list of Ford automobiles (Mustang, Fairlane, etc.) for the variable car. (Please refer to appendix B for a full explanation of variables and their types.)

A banker may use AURORA to discover why certain loans fail while others are successful, a discrimination that may require non-numeric factors such as type of business receiving the loan. An attorney could use AURORA to figure out which types of arguments have failed in the past for a particular type of case. Baseball managers could figure out whether one should bunt or not in a given situation. For the farmer, AURORA could be used to determine the reasons for one year's success compared with the less abundant yield of another year. The possibilities are virtually without limit.

Structuring of Knowledge?

Unlike some ESB's, AURORA encourages one to structure knowledge bases in some powerful ways. This is done by placing the action, or THEN part of a rule, in the condition, or IF part of another rule. For example, one can have three rules, the first stating that one should buy a used car only if it is a reliable model, and if the dealer appears to be honest. One could then have two additional rules, the first enumerating which models tend to make a reliable used car, and the last stating which dealers are likely to be honest. These rules could be learned, created by hand, or some combination of the two. For example, AURORA could be led to learn that all honest used car dealers have been in business for at least 10 years and don't advertise on UHF television late at night, while the other two rules might be handcrafted for entry directly into the system.

What About Uncertainty?

Not to worry. The real world is not cut and dried, and AURORA does not expect its knowledge to come in neat but simplistic forms, in which every statement can be said to be either true or false. In AURORA, one is allowed to insert a number, known as the distinctiveness weight, after every condition (variable - value set) in a rule. The distinctiveness weight indicates the relative importance of a condition in the rule when all other conditions are not know. For instance, one might have this rule:

It will rain if:	*DW*
1. The barometer is falling,	60%
2. The sky is cloudy,	30%
3. My dog Fido is howling.	10%

The numbers to the right, the distinctiveness weights, are the estimates of the designer of the rule that the indicated action, in this case the appearance of rain, will take place if the condition is satisfied, and no other conditions are known. So the falling barometric pressure is an important piece of evidence in predicting rain,

while Fido's howling, which may occur for a number of reasons, is a small but contributory factor in the prediction.

In the case of a hierarchical rule base, these weights are propagated throughout the knowledge base in a manner that simulates the way in which people reason with uncertainty. Thus, through relatively small changes in the way knowledge is represented, AURORA may gain as a sophisticated reasoning tool.

CONCLUSION: What are the exciting new features of AURORA?

- AURORA can perform conceptual data analysis.
- AURORA can handle uncertainty in its knowledge representation.
- All screens are menu-driven and come with on-line help.
- Event and feature editors make entering examples simple.
- Rules are easy to read, understand and create.

As you read through this manual, and create ES's on you own, the superiority of AURORA will become apparent.

CHAPTER TWO
GETTING STARTED WITH AURORA

AURORA comes on two 1.2 MB diskettes. One contains the advisory module and the rule learning module, as well as additional system files. The second diskette contains two sample advisory systems. A full list of the files and their description appears at the end of this chapter.

How to Install and Run AURORA.

1. You may wish to create a separate directory for AURORA. To do so, enter the following after booting up the machine.

```
C>MKDIR\AURORA
C>CD\AURORA
```

The first command creates a new directory named AURORA, while the final command places you in the new directory. If you wish to store all the files in AURORA in the root directory, you may ignore this step and proceed to step 2.

2. Insert the first diskette into drive A: and enter the following:

```
C>COPY A:*.* C:\AURORA
```

This command copies the entire contents of the disk onto the hard drive.

3. Repeat step 2 with the second diskette.

4. Enter AUR and the program will come up. Entering AUR DEMO will produce a short animation before displaying the main menu. All AURORA functions will be accessible from the main menu.

Generally, a Q may be typed from any menu to exit the program. Typing <CTRL> - C at any time will also return control to the operating system.

Warning! Due to large memory requirements, AURORA is not totally compatible with Terminate and stay resident programs (e.g. sidekick, Lightning, etc.). Unpredictable results may occur.

Contents of diskette 1

Main system files for executing AURORA

aur.com	advv.com	aur.com	gem_comp.chn	data.com
edit.chn	infpar.chn	create.chn	gem.chn	gem7a.chn
varb.chn	varbdata.chn	s1.chn	tut.chn	mkb.chn
padvmod.chn	param.chn	rev.chn	progmod.chn	comp_dis.chn

Auxiliary files for holding default values

draw.aur	edit.aur	systems.aur	learn.aur

Files for on line help

dev.hlp	gem.hlp	instr.hlp	inf.hlp	ldmenu.hlp
mkb.hlp	aur.hlp			

Contents of diskette 2

Diskette 2 contains all the files necessary for two exemplary advisory systems. PCPICK was designed to assist in selecting a personal computer. LET'S_EAT assists the weary traveler with chosing a restaurant in ChampaignUrbana. Advisory system files have the following standard extensions:

<SYSTEM>.adv	compiled knowledge base - executable copy
<SYSTEM>.cin	index file of compiled rules
<SYSTEM>.crl	uncompiled rules - display copy
<SYSTEM>.dta	contents of example editor
<SYSTEM>.ind	index file for most recent learned rules
<SYSTEM>.inf	inference parameters
<SYSTEM>.lng	text file contain a long description of the system
<SYSTEM>.log	history of last advisory session
<SYSTEM>.lrn	current learning parameters
<SYSTEM>.rle	uncompiled rules
<SYSTEM>.var	contents of variable editor

CHAPTER THREE
THE MAIN MENU OF AURORA

Assuming you have booted up your system and accessed the AURORA directory or diskette. You will next enter AUR and the program will start. The first menu you see is the main menu. (Figure 1) Your options are as follows:

- Get advice from an advisory system already in existence
- Develop a new or improve a current system
- Enter the learning and discovery module
- Review an advisory system (without editing it)
- Go through the tutorial to learn how to use AURORA
- Quit

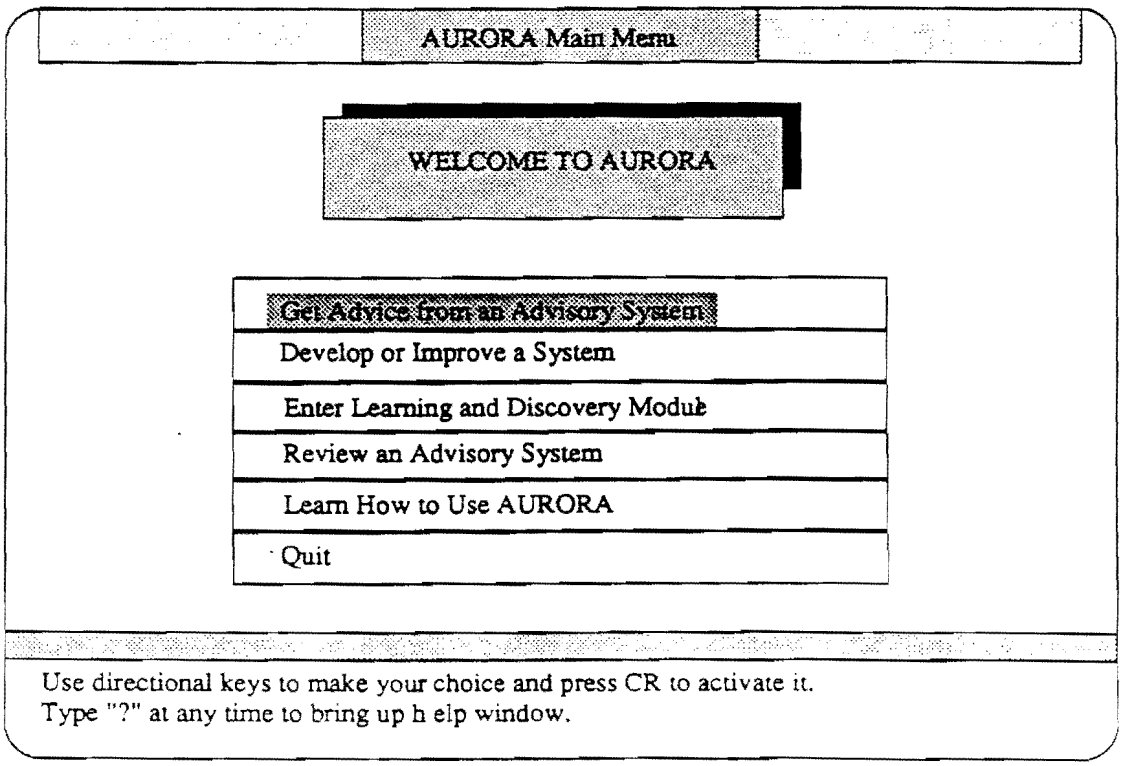


Figure 1. AURORA Main Menu

After positioning the inverse video line (highlighted area) on your screen over your chosen option—you do this with cursor keys (directional arrows) on the keyboard—press the <RETURN> key (hereafter, <CR>) to activate your choice.

The following chapters will explain the use of each possible choice from the main menu. Typing a ? will provide a help window at any time. For example, when you are going to design an expert system you just type in ? from the screen you currently working (suppose you are working with the Develop or Improve an Advisory System screen. See Figure 15, Chapter 4.). After you type in ?, you will have the following screen:

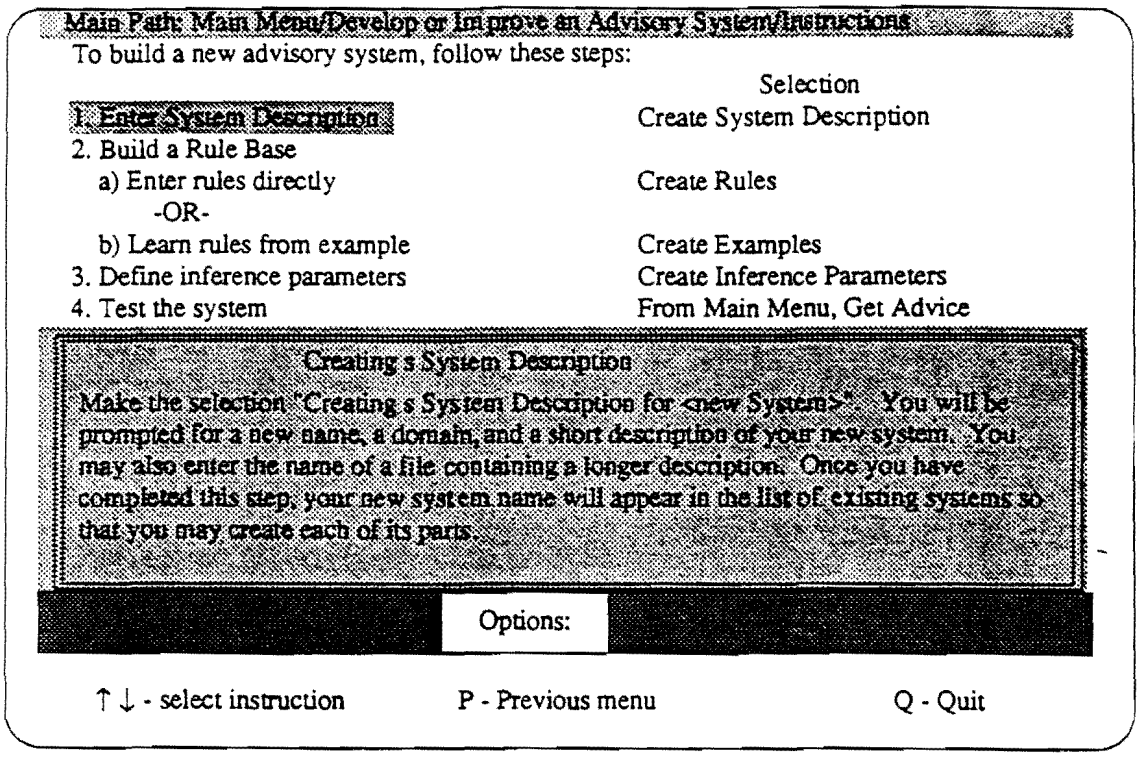


Figure 2. Help for Development of an Advisory System

The up and down arrow keys help you to move from one item to another. The explanation in the lower box corresponds with the highlighted item you chose.

CHAPTER FOUR
A TUTORIAL - PCPICK AN ADVISORY SYSTEM FOR CHOOSING A PERSONAL COMPUTER

In this tutorial, you may choose to learn:

- How to run an advisory system
- How to build an advisory system
- More about AURORA's Learning and Discovery Module.

In this chapter you will be shown how AURORA can provides advice in the selection of a personal computer. The tutorial will also cover the development of the PCPICK system, showing how learning by examples helps to create rules, how the rules were compiled and prepared for the advisory system. Finally, an example will be provided, how to enter the various modules of AURORA directly through the learning and discovery module.

After starting the computer and changing to the AURORA directory, type AUR and press the <CR> key. You will be presented with the AURORA main menu as shown in Figure 1. Next select *Get advice from an advisory system.* A second menu as shown in Figure 3 will now appear on the screen. Many of AURORA's screens have a similar format with three areas to each screen which are consistent throughout AURORA. The top line, called the Menu Path, will provide information on the current pathway taken to arrive at the menu on the screen. Each previous menu will be separated by a / character to show its unique path. On the bottom of the screen will be an area labelled as *Options:*, which will present additional options if the Escape (Esc) is pressed. The center portion of the screen will change from module to module, but will provide the operating information necessary for that area. As you can see in Figure 3, the Advisory Systems are listed in the left-hand column, with a short description of the system to the right. Hilight the system PCPICK and press <CR>.

In the development of an advisory system, it is possible to provide a text file, which is an extended description of the system. An example of this long description file is seen in Figure 4. This file can be more than one screen in length and is used to provide the novice user of the system with beginning information on its use and purpose. The options area for this screen has automatically provided additional options. To take advantage of an option, simply press the character shown for its action.

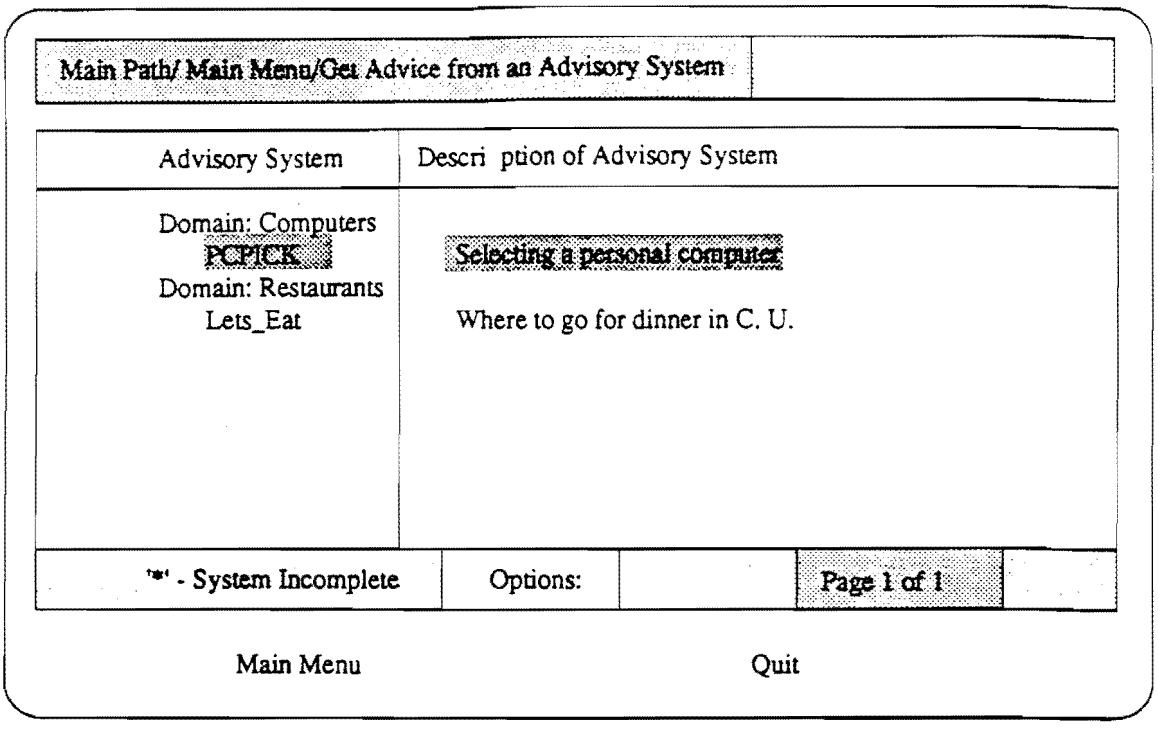


Figure 3. Advisory System Selecti on Menu

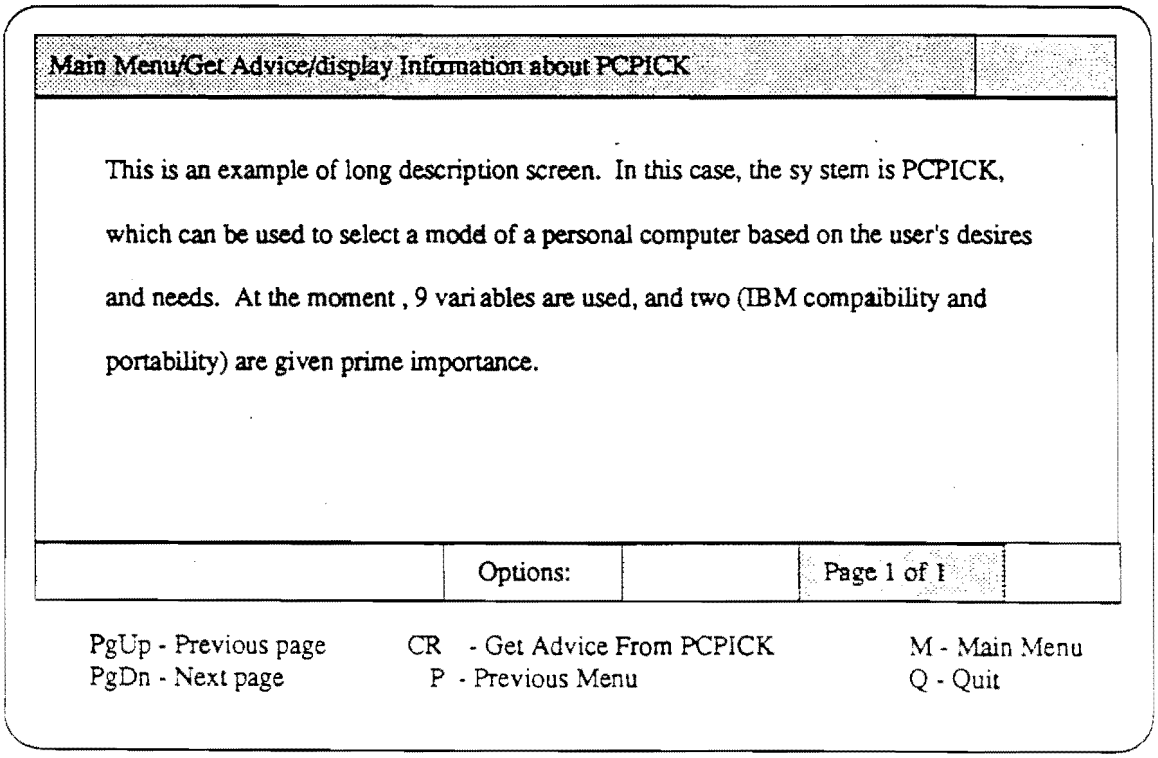


Figure 4. Information about PCPIC K

Now press <CR> to move to the first advisory screen for PCPICK. The format of advisory screens is shown in Figure 5. While the information within each window will change as new evidence is provided to the system, the windowing structure will stay the same.

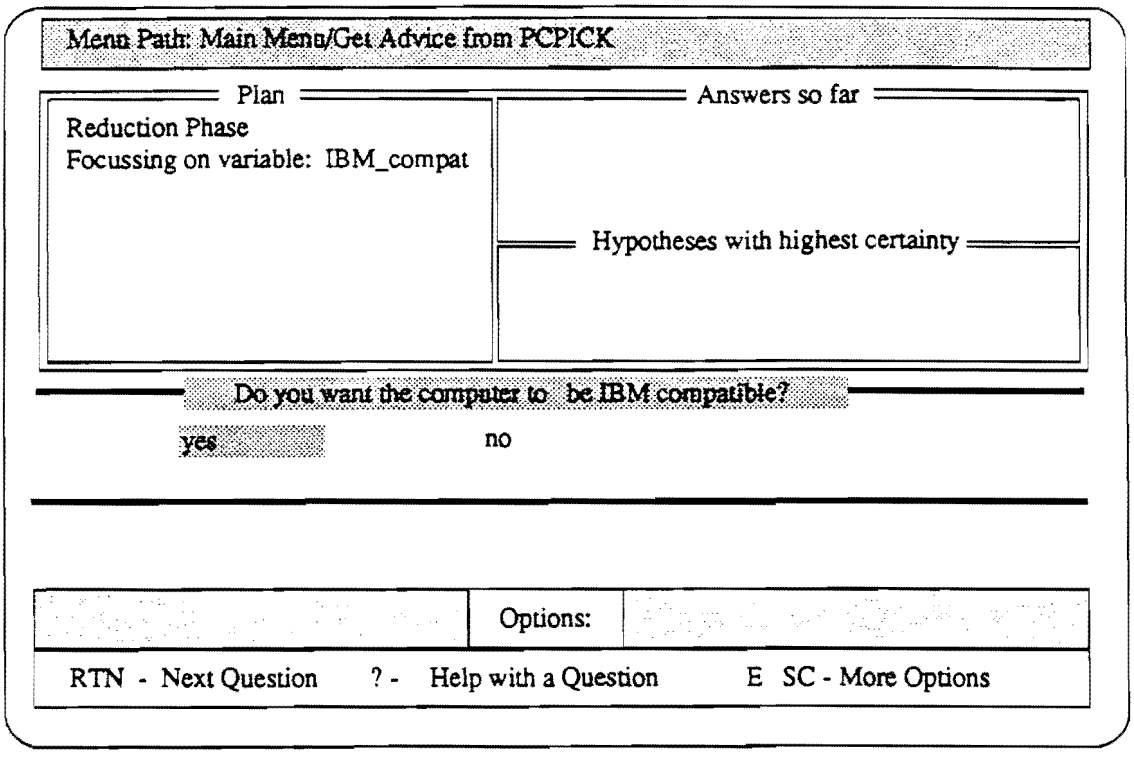


Figure 5. Getting Advice from PCP ICK

In addition to the menu path and options areas of the screen, there are four additional windows. The *Plan* window will provide information on current system strategy for arriving at the appropriate advice. Answers that have been provided will be listed in the *Answers so far* window to the right. When sufficient evidence is presented to the system, rules which have the greatest support will be listed in the *Hypothesis with highest certainty* window. The rule or hypothesis with the greatest potential will be listed first, with the next three possible hypotheses following. Additional hypotheses can be viewed through the options area. Immediately below the top three windows is the query window, where questions will be asked of you to help build evidence for the final answer.

Since PCPICK has a high priority on IBM compatibility, the initial question *Do you want the computer to be IBM compatible?* will be asked first. At this time, select yes and press <CR>. As in the previous screen,

the Plan window will indicate that the inference system is in a reduction phase (Figure 6). In this phase, the system is attempting to eliminate as many hypotheses as possible before focussing on the final confirmation of the remaining hypotheses. In order to reduce hypotheses, the system must focus on key variables, which will help to segregate the knowledge base.

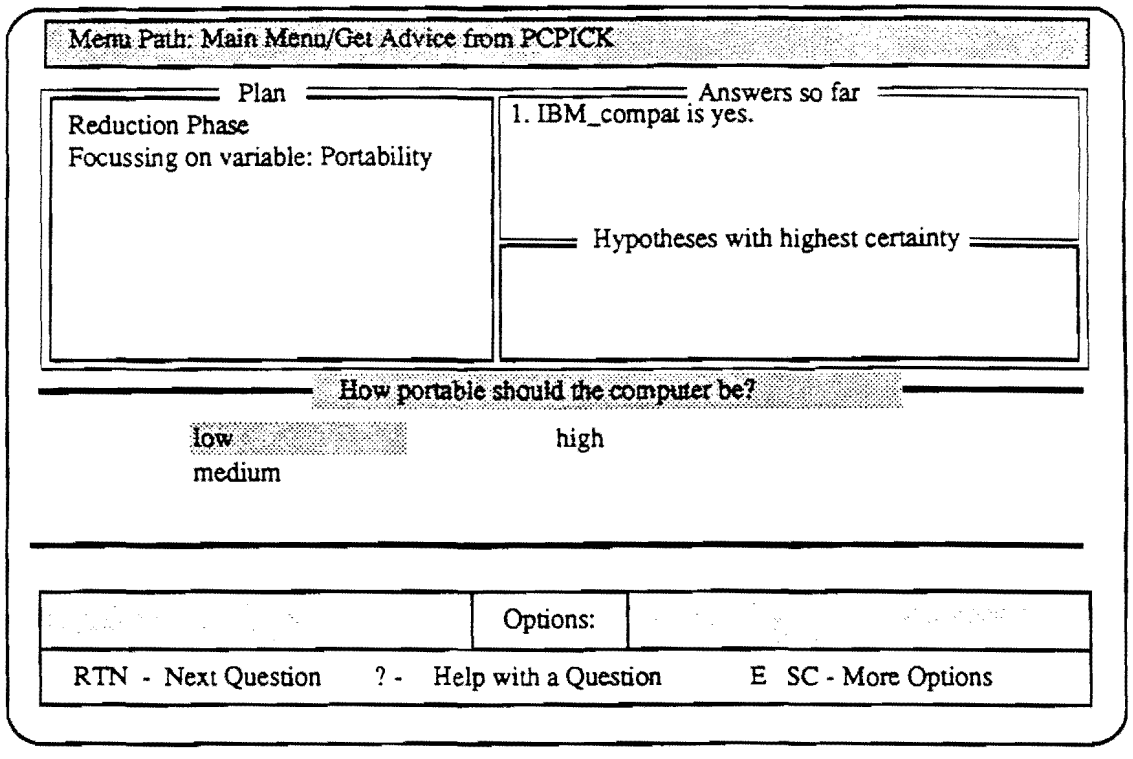


Figure 6. Beginning of "Reduction Phase"

A question on the portability of the computer is now being asked. The previous answer *IBM compatibility is yes*, now appears in the *Answers so far* window. During the reduction phase, the certainty of each hypothesis is not changed, therefore, no listing appears in the *Hypotheses with highest certainty window*.

Select the value *low* for the question *How portable should the computer be?*. With this second piece of evidence, the reduction phase has been completed (Figure 7). As indicated in the *Plan* window, rules have been reduced from a possible 15 to a subset of 5. The top four hypotheses or rules, are listed in the *Hypotheses with highest certainty* window. Since they all have the same degree of certainty, they are listed in the order in which they appear in the rule base.

The *Plan* window also indicates that the discrimination phase of the inference mechanism has begun. During this phase, focus will be placed upon variables which provide the discrimination of a single hypothesis

from the set of five. The second answer *portability is low* has been added to the *Answers so far* window. A question asking your need for processor speed for a computer is now presented in the center portion of the screen.

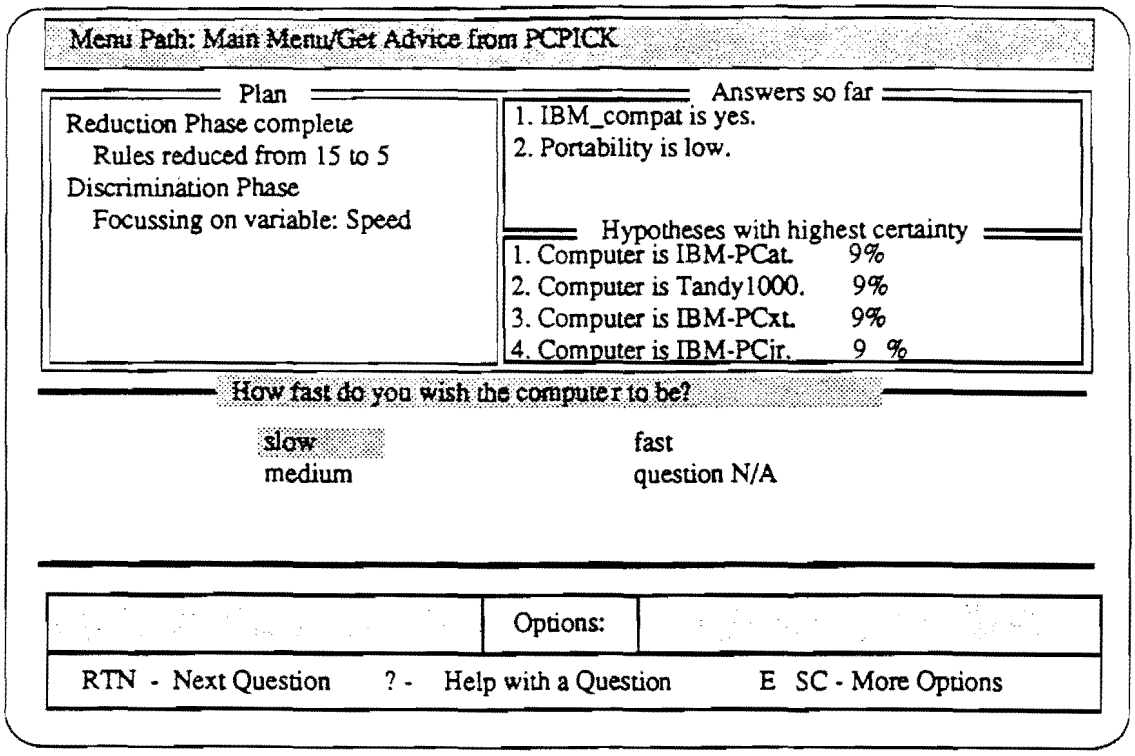


Figure 7. Beginning of "Discrimin ation" Phase

Assuming that process speed is a high priority, answer *fast* and you will be presented with the screen in Figure 8. Notice that the three pieces of evidence that you have provided the system are listed in the *Answers so far* window. With this evidence comes the completion of the discrimination phase. The third and final phase of inference, the confirmation phase, is now being engaged to confirm that the best advice is the purchase of an IBM-PC AT. In order to confirm this hypothesis, additional evidence will be needed.

Now answer the question *What price are you interested in?* with the value *high*. Next, you will be asked whether or not you require a hard disk drive. Answer *yes*, doesn't everyone! This fifth piece of evidence is enough to confirm the candidate hypothesis. A final screen (Figure 9) is provided. In addition to providing the final advice from the system, any additional information associated with the answer will be displayed. Additional options are available in the options area of the screen.

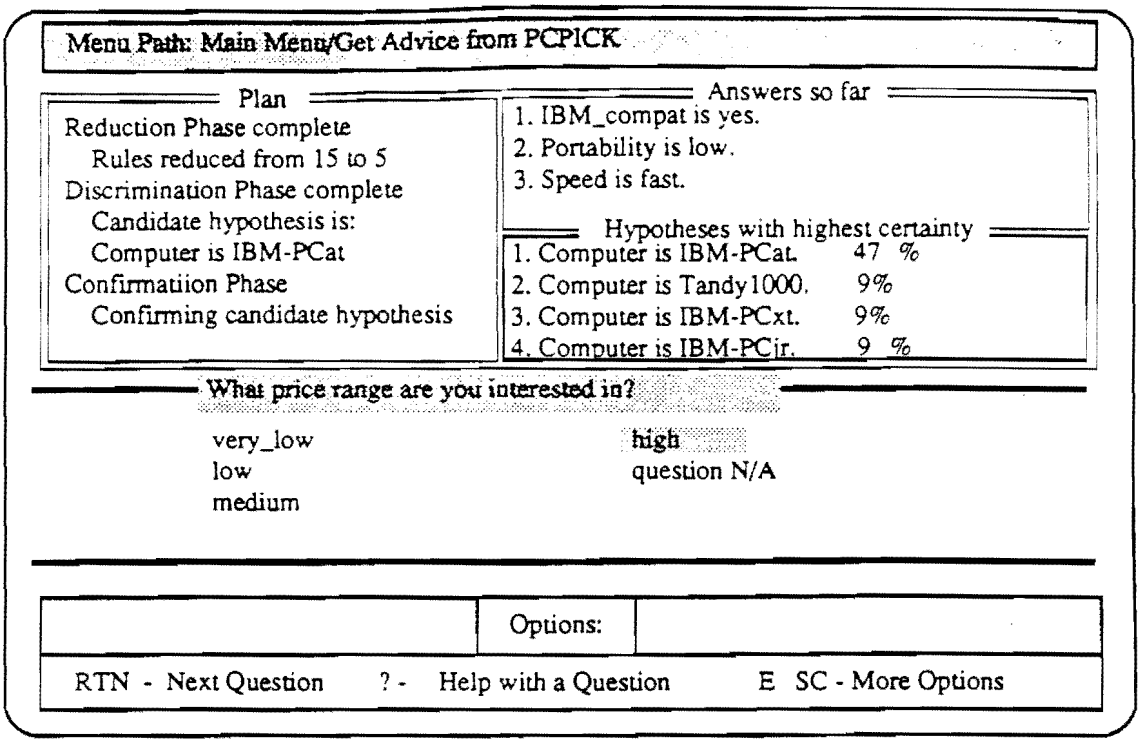


Figure 8. Beginning of "Confirmat ion Phase"

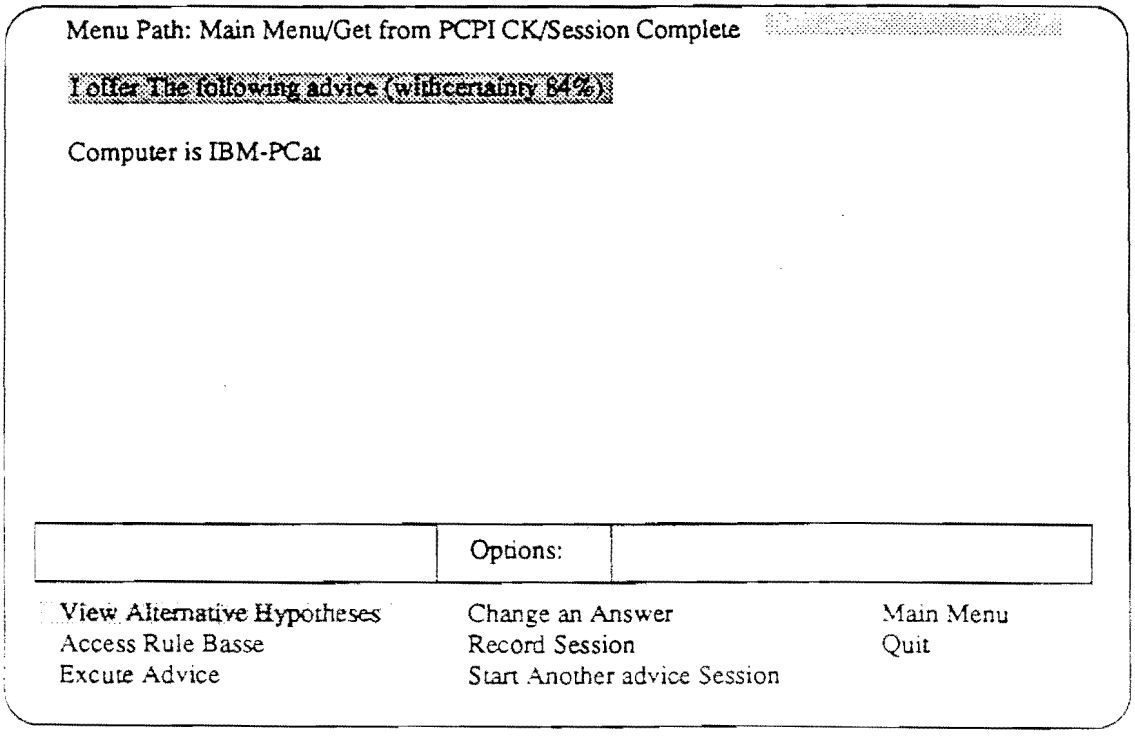


Figure 9. Final or "Advice" Screen

Select *View alternative hypothesis* in the *Options:* window. Five potential hypotheses as determined in the reduction phase will be listed on the screen (Figure 10). As shown, they are grouped into those meeting the criteria for candidate hypothesis and all other non-selected hypothesis. The final distinctiveness weight or total evidence is listed on the far right-hand side, both as number and as a visual bar graph. This screen allows you to see the relative weight the evidence has provided for all hypotheses.

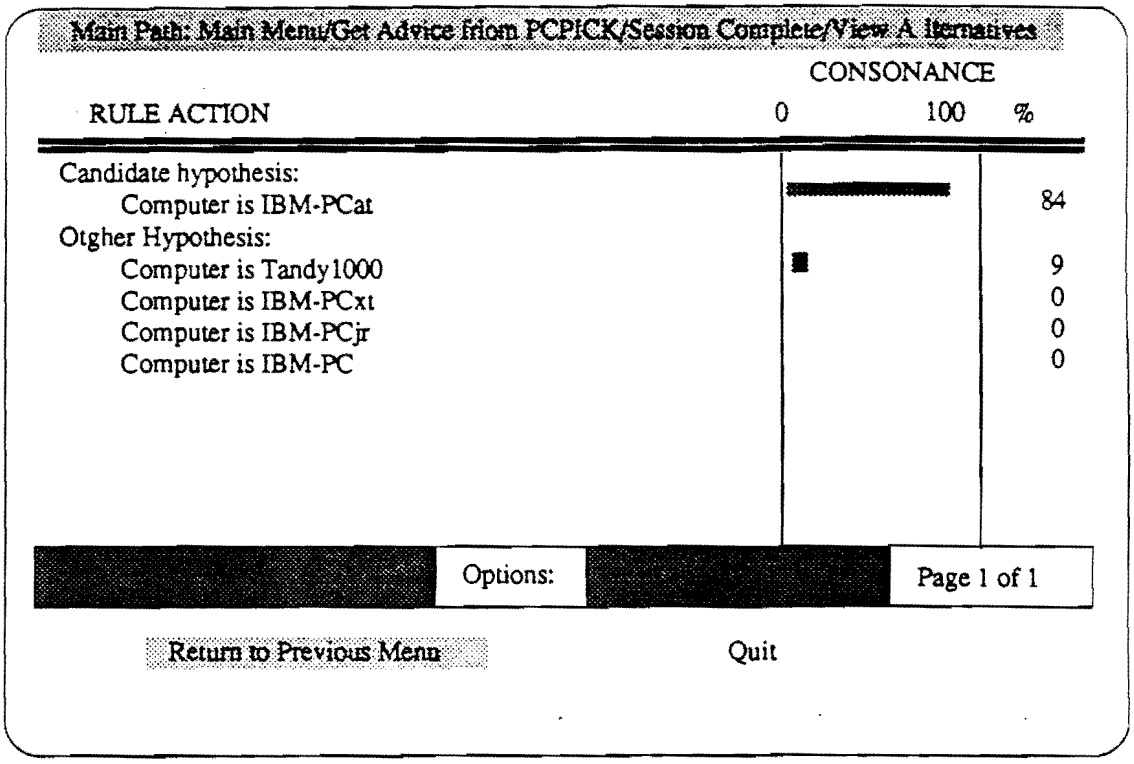


Figure 10. View Alternative Hypotheses

Choose *Return to Parent Menu* and press **<CR>**. Now select *Access Rule Base*, a list of all of the rules in the knowledge will be provided as in Figure 11.

Choose the rule *IBM-PC AT* in the right-hand column and press **<CR>**. The rule which concludes by advising to select an IBM-PC AT is shown in Figure 12. For each selector, or variable, in the condition portion of the rule, an indication (shaded values) is shown for variables which have been used by the system. Values which were chosen are listed on the right-hand side. If the value matches the answer given in the advisory session, it is hilight in both areas. For answers which differ from those provided during the advisory session or were not asked, the value is not hilight. The distinctiveness and communality weights assigned to each variable and value set is also listed on the right.

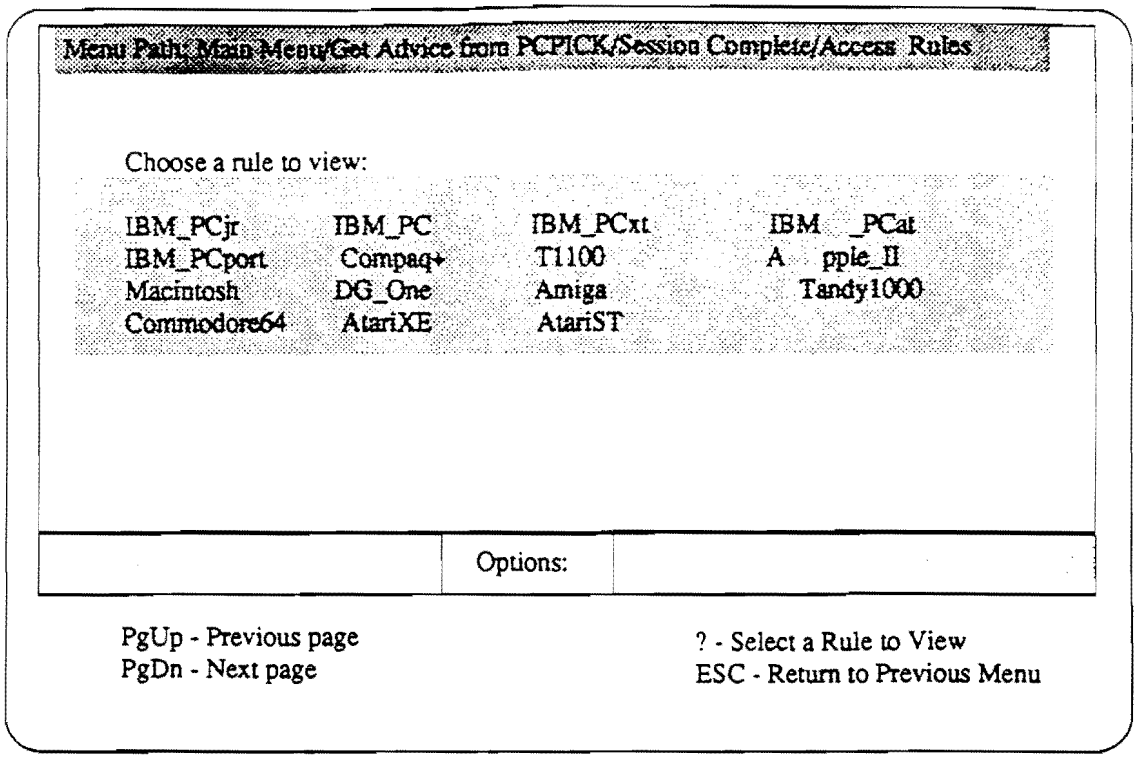


Figure 11. Access Rules in PCPICK

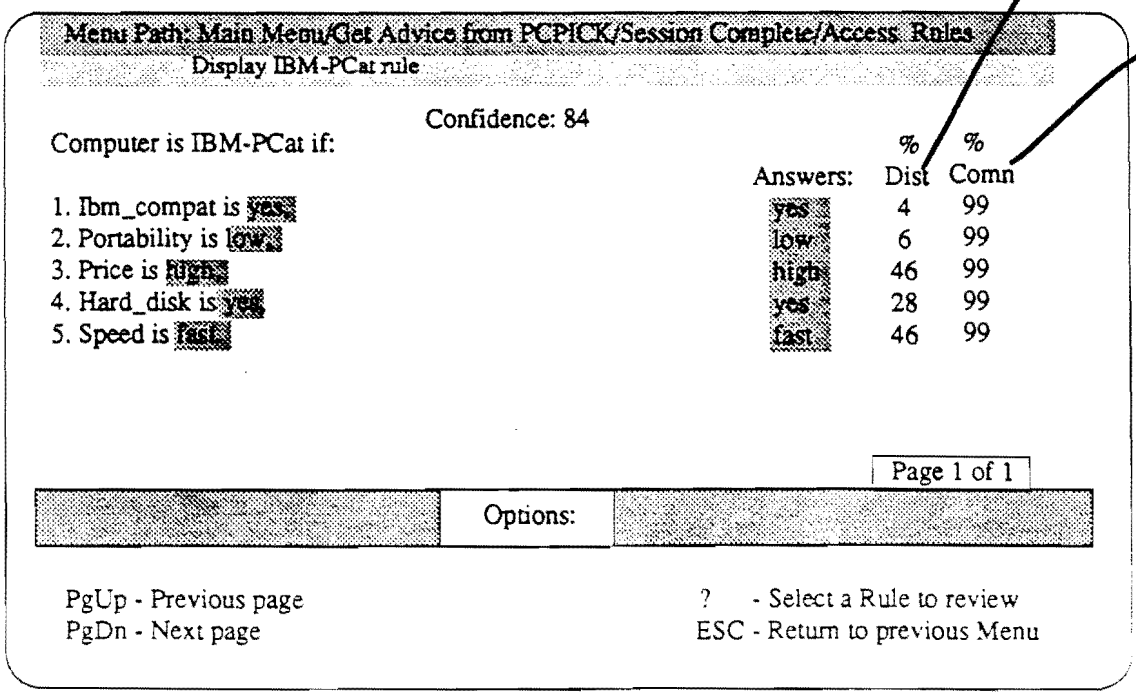


Figure 12. IBM-PCat Rule

For a comparison of a rule which was selected for the discrimination phase but not confirmed, type a *?* then select IBM-PCjr from the menu of Figure 11. The displayed rule for the IBM-PCjr (Figure 13) indicates a zero confidence or distinctiveness was achieved by the system. You will also notice that in the case of the third variable *Price is low*, since the answer provided is *high*, it was not used in building the final confidence level.

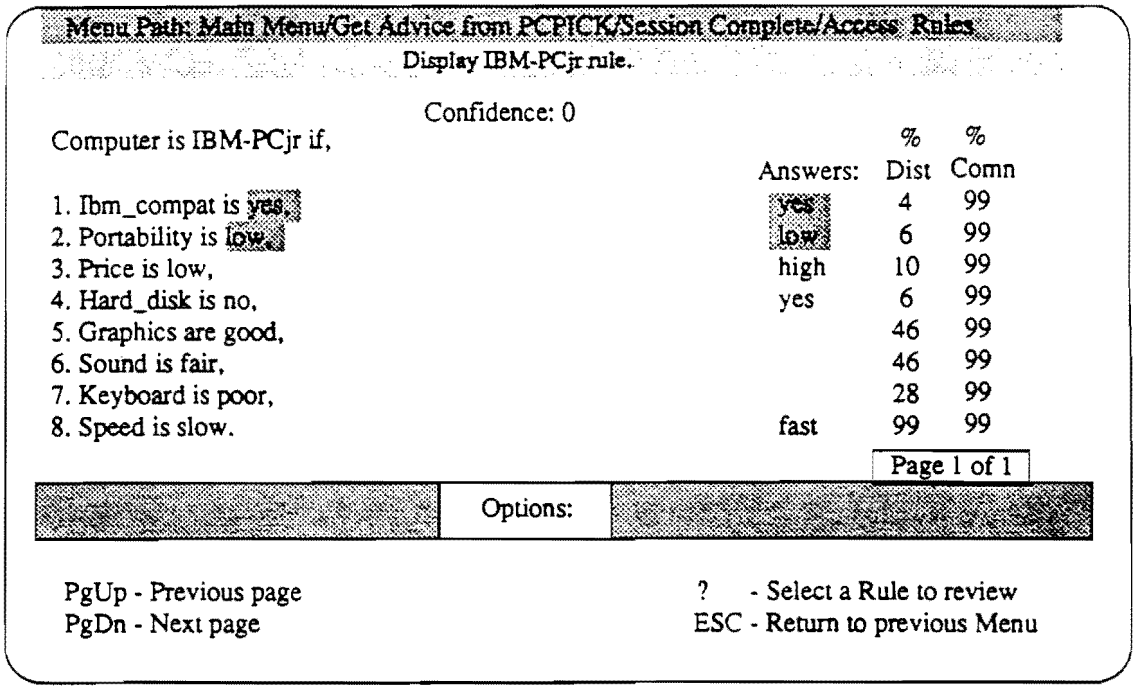


Figure 13. IBM-PCjr Rule

An example of a rule rejected in the reduction phase can be seen in Figure 14, illustrating the Macintosh rule. Type a ? and select the value *Macintosh* from Figure 11. *Rule has been eliminated* appears in the area where the confidence value would normally be listed.

Now press ESC to return to the final menu of the advisory session (Figure 9.). Additional options are available to change an answer or value for any of the questions. This might be useful if after receiving the final advice, you feel an inappropriate answer was selected.

You may also directly start another advisory session from this screen. If you wish to access a portion of AURORA other than the advisory module, select *Main menu* or type an M. To quit AURORA altogether select *Quit* or type a Q.

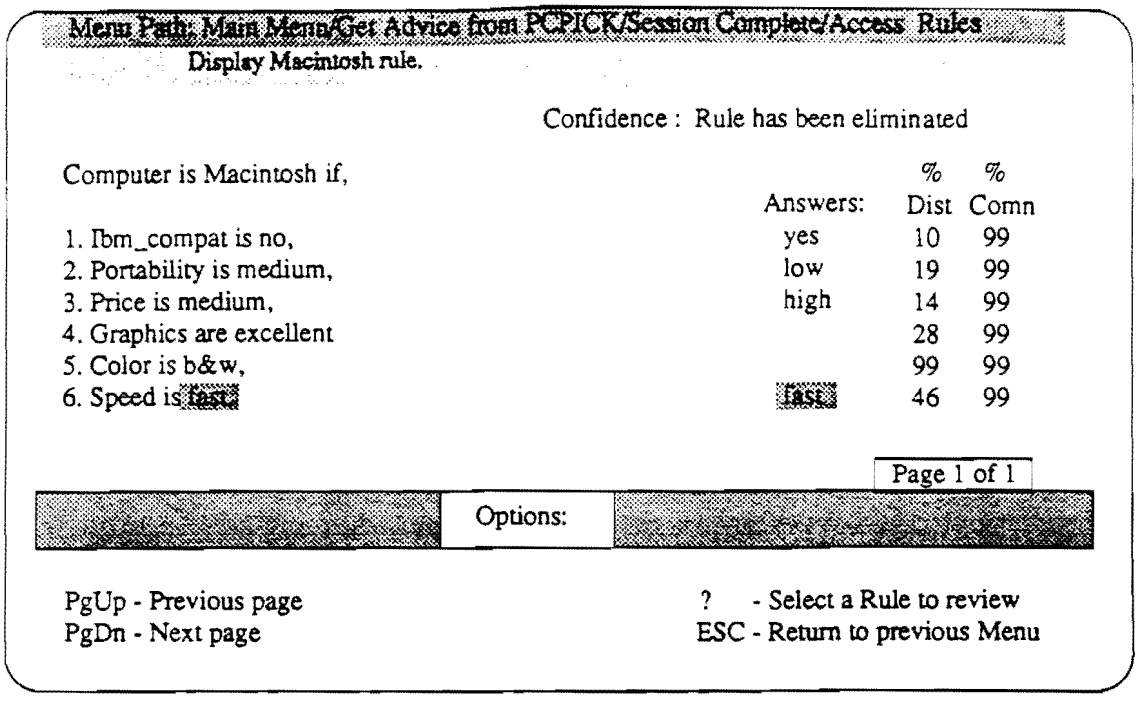


Figure 14. Macintosh Rule

Developing or Improving a System

The Develop or Improve an Advisory System menu (Figure 15) is presented after selecting *Develop or Improve a System* from the AURORA Main Menu (Figure 11). The Develop or Improve an Advisory System menu will require you to make a selection from each of the three columns shown. The left-hand column designates the advisory system to work with or the creation of a new system. The type of operation and item to operate on are selected from each of the two remaining columns. To move across columns, use the right or left arrow key, and to move up and down rows within a column use the up or down arrow key.

Now select *New System* from the *Select Advisory System* column, *create* from the *Select Operation* column and *System Description* from the *Selected Item* column. After making a selection from all columns, invoke the next menu by typing a <CR>. The System Description menu (Figure 16) will appear.

To move throughout the menu, use the up or down arrow keys. This menu allows the selection of a value for four parameters. The first, *Expert System Domain* can be selected from current domains as shown in Figure 15. You may additionally enter a new value from 1 to 18 characters. Enter COMPUTER to create a second personal computer selection advisory system. Now move to the *Expert System Name* row and enter the name PCPICK2. Once again, move the cursor down to the short system description. At this point a description of up to 50 characters can be added to appear in the advisory system menu (Figure 3). Type in SELECTING A PER-

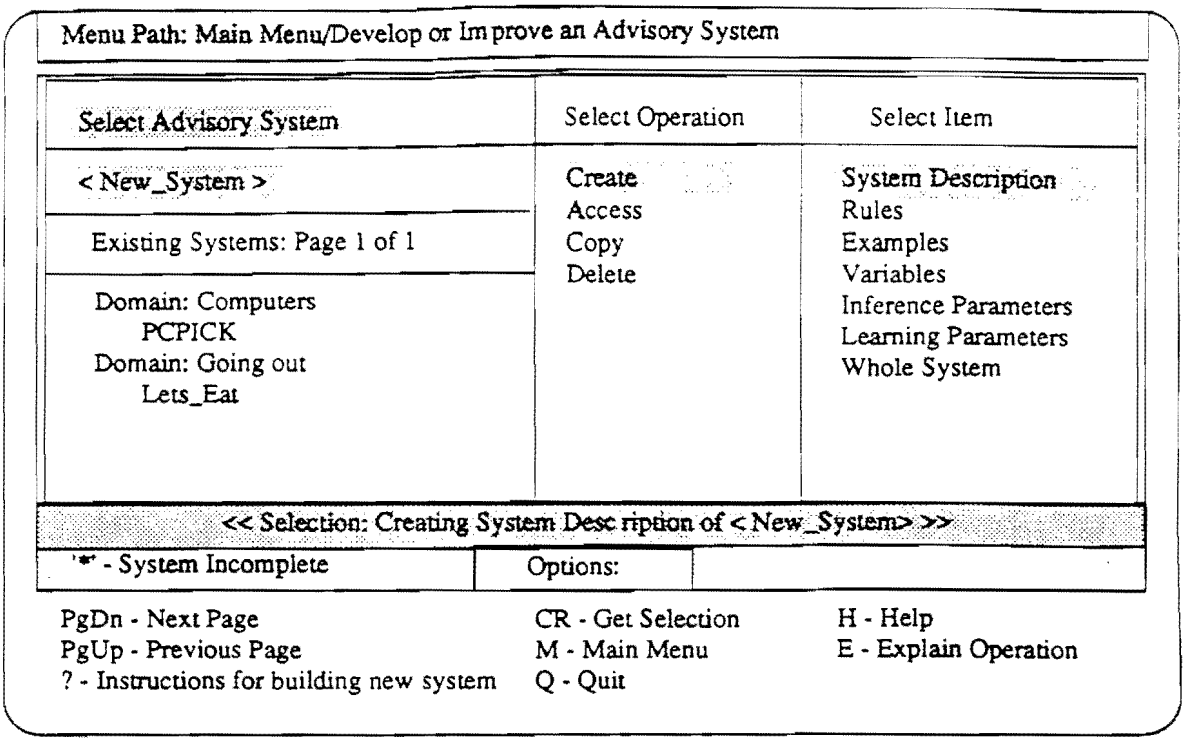


Figure 15. Develop or Improve an Advisory System Menu

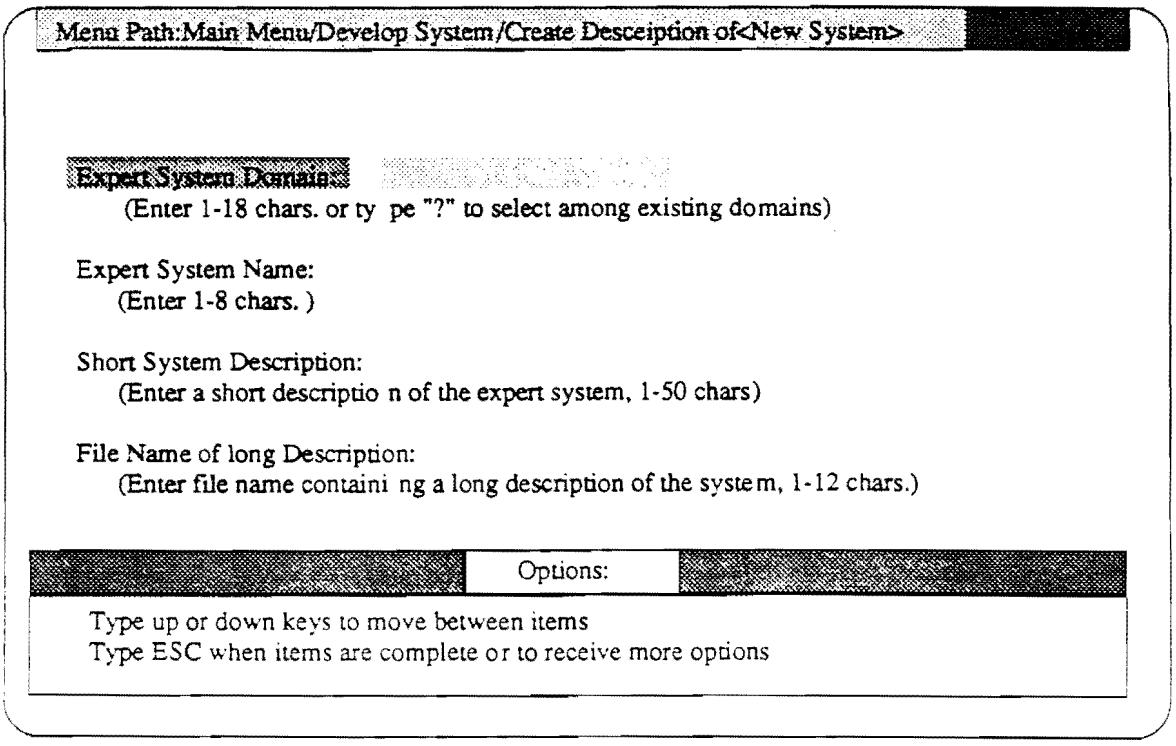


Figure 16. System Description Men u

SONAL COMPUTER. Finally, select *filename of long description*. For this row, you may add the name of a file which will display an extended description upon the start of an advisory session. You will be given the opportunity to either create the new system you have just described or return to the Develop or Improve an Advisory System menu without creating. To return to the Develop and Improve menu, type an Esc. Choose the second and return to the Develop or Improve an Advisory System menu. Now select the advisory system *PCPICK*, select the operation *Access* and the item *Example*. Press **<CR>** to move to Figure 17, the Example Editor. There will be a momentary pause while the examples are being loaded into the Editor.

Menu Path: Main Menu/Develop System/Access Examples

	DECISION	EXAMPLES		
VAR	Computer	1. Ibm_compat	2. Por tability	3. Price
TYPE	nominal	nominal	li near	li near
# 1	IBM-PCat	yes	low	very_low
2	IBM-PCat	yes	low	medium
3	IBM-PCxt	yes	medium	low
4	IBM-PCxt	yes	high	low
5				
6				
7				
8				
9				
10				

Options:

PgUp - Page up
PgDg - Page down
Ctrl → Page right
Ctrl ← Page left

F1 - Insert row
F2 - Insert column
F3 - Grab word
F4 - Grab row

F5 - Grab column
F6 - TYPE/COST
F7 - Jump to decision
F8 - Jump to variable

F9 - Edit domain
F10 - Legal values
Esc - More options

Figure 17. Example Editor

The Example Editor resembles a spread sheet in many ways. Each column represents a single variable and its value for each example. Either a selection variable to describe the conditions of an example, or a decision variable in the left-hand column to designate the final decision or results of the example. Each numbered row represents a single example with its single value for the decision variable and single value for each selection variable. There are currently four examples entered for the PCPICK system. The first two examples are descriptions of PC-AT systems, and examples three and four describe PC-XT systems.

Let's add one more example of an IBM-PC XT in order to see how the Example Editor functions. Move the hilight to the decision column if it is not already there, and then downward to Row 5, or the first blank

row available. You may simply enter the value by typing it. You may also use the **F10** key to move through the set of legal values. If the value is not in the list of legal values, you'll be required to enter the variable editor to add it to the list. Using the F10 key, place IBM-XT in the cell. Next move the cursor to the right in the same row to Variable 1, *IBM Compatibility*. Select YES for the value. Once again, move right to Column 2, *Portability*, and select LOW for the value. Finally move to Column 3, *Price*, and enter MEDIUM for the value.

Up to 50 columns of selection variables for an example can be entered. In order to add values to the additional columns, move the cursor once more to the right. The first column is shifted off the screen, while the fourth column appears. This feature is also available for rows to enter examples. Up to 500 examples may be added by scrolling through the Editor. To speed up the scrolling, both the PGUP and PGDN keys and <CTRL> RIGHT and <CTRL> LEFT key combinations are used to move one full page at a time.

Now that you have added an additional example, save all the examples, then move to Learning. Type an **Esc** for additional options. Select *Save Changes* and press **<CR>**. You may save this example set to an additional system by changing the name or to PCPICK by typing a **<CR>**. Now move to the *Learning Menu* option to move to the *Learn Rules* menu (Figure 18).

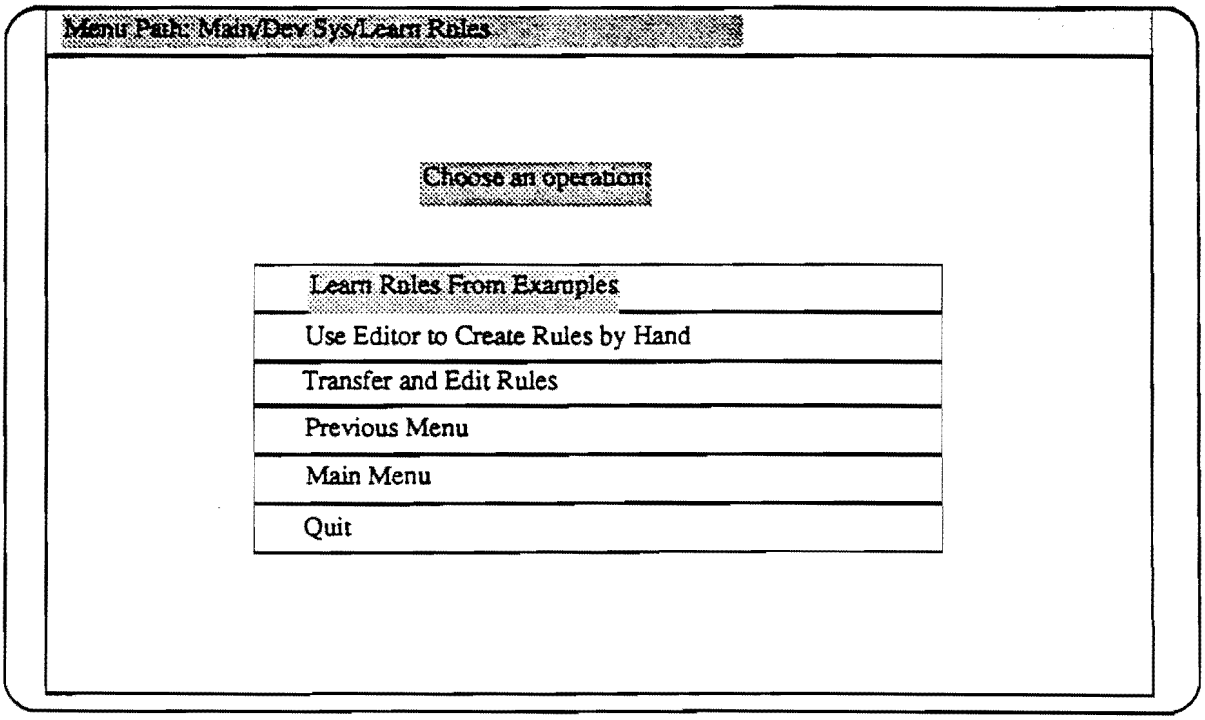


Figure 18. Learning Rules Menu

An external editor may be used to create rules by hand in the *Transfer and Edit Rules* option which will be discussed later. Now select *Learn Rules from Example* and press **<CR>**. A second menu (Figure 19)

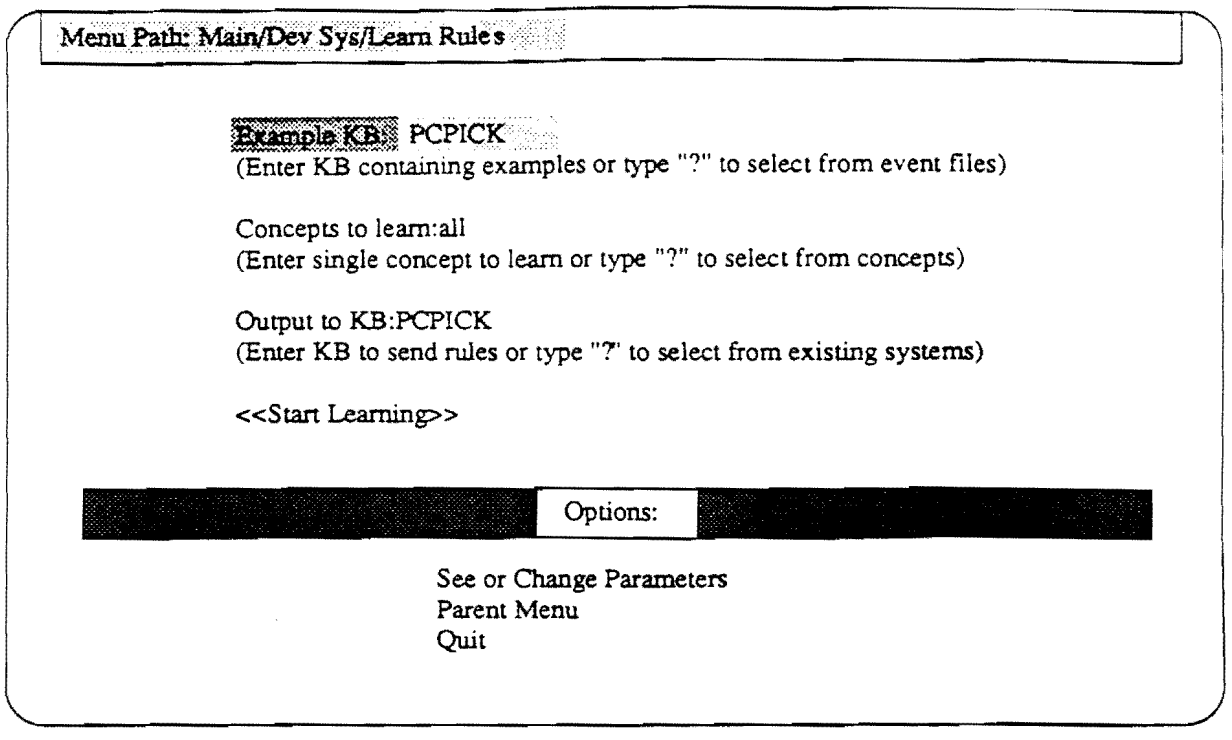


Figure 19. Learning Rules, Second Menu

will be presented to allow you to name the example file and a file to contain the output of learning. Since we came from the advisory system PCPICK, the name PCPICK is already given for the example KB and for the *Output to KB:*. The middle row, *Concepts to Learn*, allows for the formation of a single rule rather than all the rules necessary to cover each value unique example. For now, let the value remain at *All* and move to the cursor to *Start Learning*. Before pressing <CR>, notice that in the options section, there is a selection to change the learning parameters. This option will present a menu discussed later which will allow many kinds of rules to be developed.

Now press **<CR>** to start learning. Figure 20, the Reading/Learning screen will be presented. It is simply a visual prompt to show that learning has been initiated and that various components of the selected system are being read by the Learning program. If any errors or inconsistencies in the example table or variable file exist, an error message will show on the screen.

Upon successful completion of the reading of all the files necessary for learning, the Learning, then Learning Complete screens (Figure 21) will appear. The top row will designate whether the system is currently in the process of learning a rule or displaying a previously learned rule. Columns on the right-hand side will show the level of confidence or weight determined in learning and number of events or examples covered for the rule displayed. Positive events are found in examples which match the variablevalue pair displayed in the

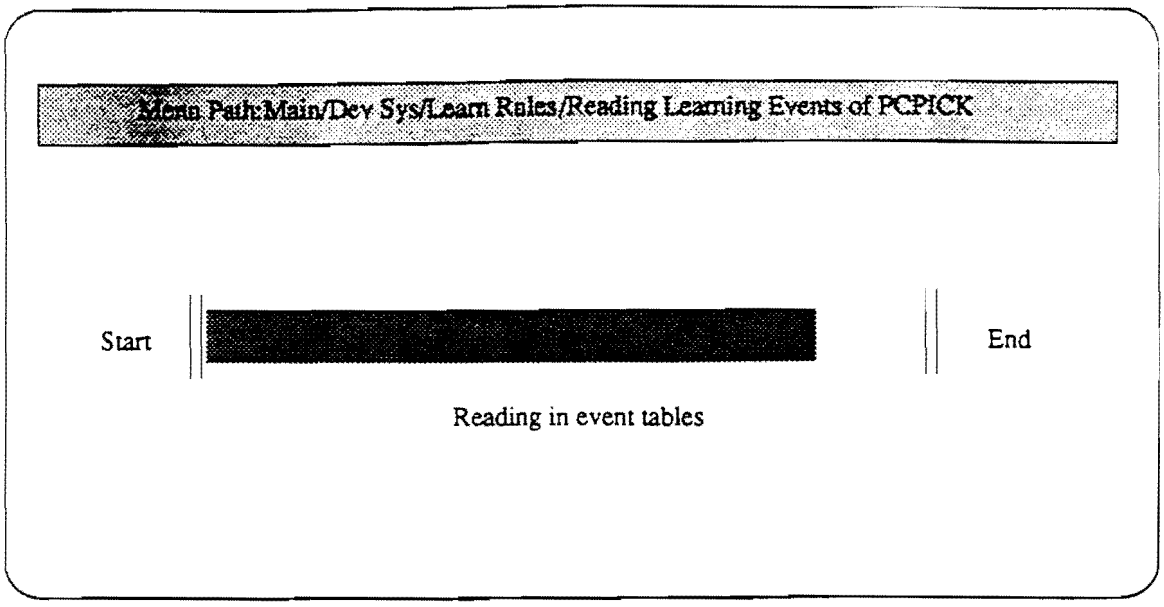


Figure 20. Reading Learning Events Screen

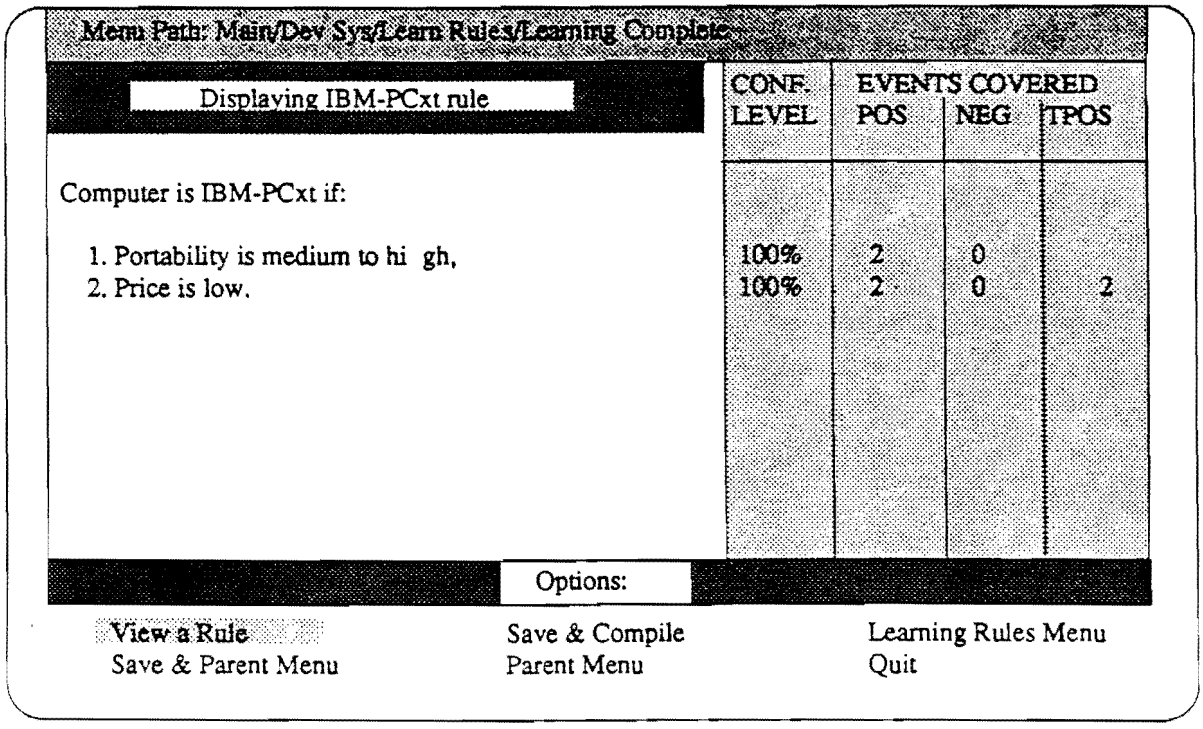


Figure 21. Learning Complete

rule. Negative events, are those in which the same value of the variable is found in other examples. The level of confidence is a measure of the relative difference in these two values. The column labeled *TPOS* provides a single value for each complex as an indication of the number of examples which are exactly covered by all the variables in the complex. Upon the completion of learning, the hilight will move to the options area where newly constructed rules may be viewed, saved or compiled. Now move the hilight to the option *save or compile* and type a **<CR>**. When prompted for the expert system to save the rules, type a **<CR>** to select *PCPICK*.

It will take a few moments for the compiler to load from the disk. Compilation will then proceed automatically. Any errors in rule format with be discovered by the compiler and a message will be displayed on the screen. For greater detail on rule format and the operation of the compiler see pages 81-83. After the successful completion of the compiler you may review the system from the advisory module available from the Main menu.

CHAPTER FIVE
GETTING ADVICE FROM AN ADVISORY SYSTEM

Overview

In an advisory session, AURORA asks you a sequence of questions, then evaluates the answers to determine the appropriate advice. Each question has a list of answers displayed on the screen (multiple choice); you select the best response. AURORA asks the minimum number of questions required to choose the correct hypothesis (i.e. advice or diagnosis) and to reject all others. At the end of a session, AURORA states which hypotheses has been confirmed, and displays additional information about the confirmed action. If, for example, AURORA confirms the hypothesis "The disease is a common cold", it can then display to you, pages of text on the common cold, such as modern remedies, the cause of the cold, folk remedies, and so on. AURORA also has the capability to execute another program that may be triggered by a confirmed hypothesis, if desired.

Choosing the Source of Advice

When you choose the option, *Get Advice from an Advisory System*, from the main menu the next screen you will see is something like this:

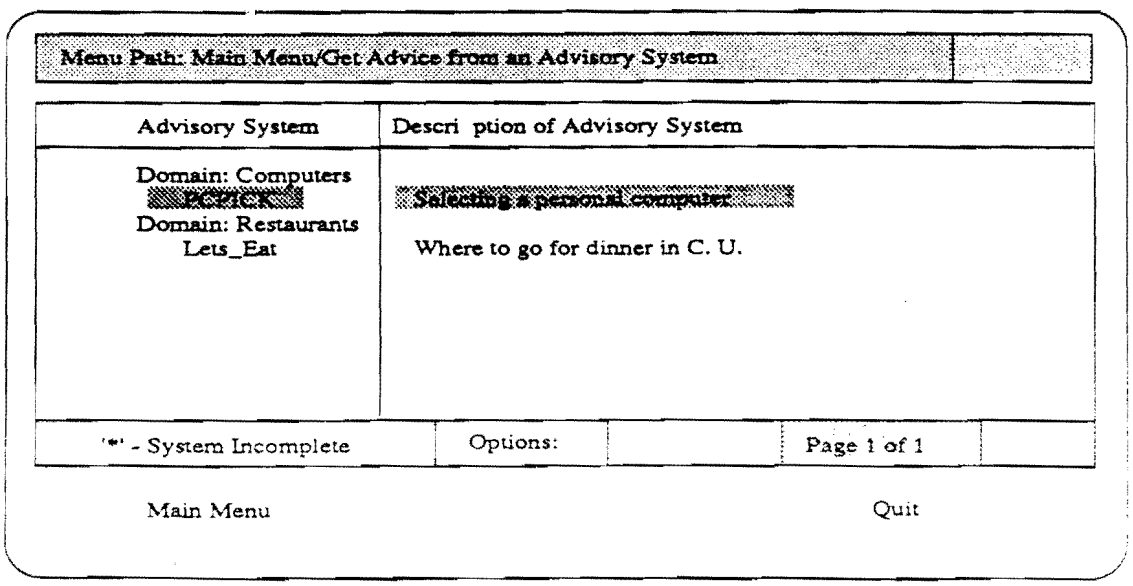


Figure 22. Menu for Choosing an Advisory System

A *domain* in the Advisory system column is the general problem area included advisory systems are designed to address: PCPICK is one system dealing with the problem of selecting a personal computer. There can be more than one system in a given domain . To the right of each system name is a brief description.

By moving the highlighted line (via directional keys) to a system name and pressing **<CR>**, you proceed to enter an advice-giving session with the system. First, however, you are shown a screen with information about the system selected. After reading this, you can still quit or return to the main menu without actually starting the session if you desire. If a long description file (the source of this text) has not been assigned to the particular system by its creator, the advisory session starts without this preliminary screen.

Entering the Advisory Session

The next screen (Figure 23) is labelled with the path name:

Main Menu/Get Advice From PCPICK

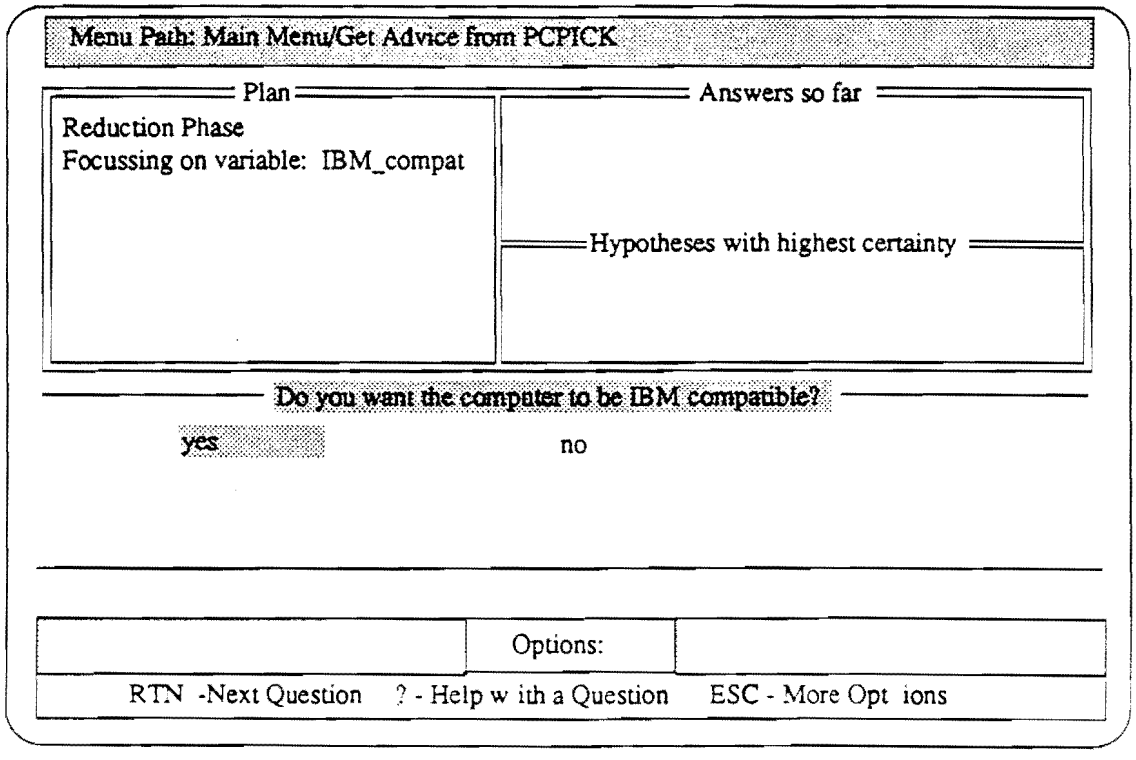


Figure 23. Sample Advisory Sessio n Screen

where the system name (*PCPICK*) was chosen. The series of questions asked by the advisory system begins here. A sample screen is shown below, where the PCPICK system has already been selected for the session.

As seen above, this screen is divided into five areas or advisory system windows:

1) Plan
2) Answers so far
3) Rules with highest certainty
4) The question being asked, and possible answers
5) and Options:

To answer the current question, the user moves the highlighted area (via directional keys) to the value desired, and presses **<CR>**. The highlight bar always remains in the question window. This enters the selected value, and brings up the next question.

Other Concurrent Windows

While in the advisory session, the windows above the question show the status of the evaluation as it proceeds. The following describes each window:

Plan Window

The "Plan" window displays status messages for the Reduction phase, the Discrimination phase, and the Confirmation phase of the evaluation. For a more detail explanation of the inference mechanism see the section "AURORA Inference Engine" (chapter 10). Samples of messages that might appear, based on an advisory session with the PCPICK system, are shown below in quotes.

a. During Reduction Phase:

Reduction Phase

Focussing on variable: Ibm_compat

At Completion of Reduction Phase:

Reduction Phase complete

Rules reduced to 15 from 5

b. During Discrimination:

Discrimination Phase

Focussing on variable: Speed

At Completion of Discrimination Phase:

Discrimination Phase Complete

Candidate Hypothesis is:

Computer is IBM-PCat

c. During Confirmation Phase:

Confirmation Phase

Confirming Candidate Hypothesis

If Candidate disconfirmed:

Candidate Hypothesis Disconfirmed

Focussing on new candidate:

Computer is IBM-PCjr

Answers So Far Window

The *Answers so far* window displays the answers given for the last four questions asked. For example, in PCPICK, the *Answers so far* Window displays the answers:

1. Ibm_compat is yes.
2. Portability is low.
3. Price is high.
4. Speed is fast.

The fifth oldest answer will scroll off the top of the screen with each new answer.

Hypotheses with Highest Certainty Window

The *Hypotheses with highest certainty* window displays the four rules with the highest certainty. Note that this is only applicable after the Reduction Phase. See the chapter "AURORA Inference Engine" for an explanation of how certainty is determined. Only four rules are displayed at any time To view the certainty of any other rules select *View Alternative Hypotheses* in the *options:* window.

Options Window

Note that this section is generally found on each screen, though with differing contents. A couple of notes for this particular screen:

? - Help

A help screen is presented for the currently highlighted value.

ESC - More options

This brings up the following options:

View Certainty Values (applicable after rule reduction phase)

The option "View Certainty Values" leads the user to a screen showing, the certainty levels of all rules not rejected during the reduction phase of evaluation.

View Rules

"View Rules" leads the user to screens showing any given rule (you select) including the distinctiveness and commonalty weights as they appear when the rule was created(Figure 24). (For more on distinctiveness and commonalty, see chapter 10.

Resume Advisory Session

Allows the resumption of the Advisory session were it was interrupted.

End advisory session

Ends current advisory session and returns user to the Main menu.

Quit

Ends AURORA and returns to DOS.

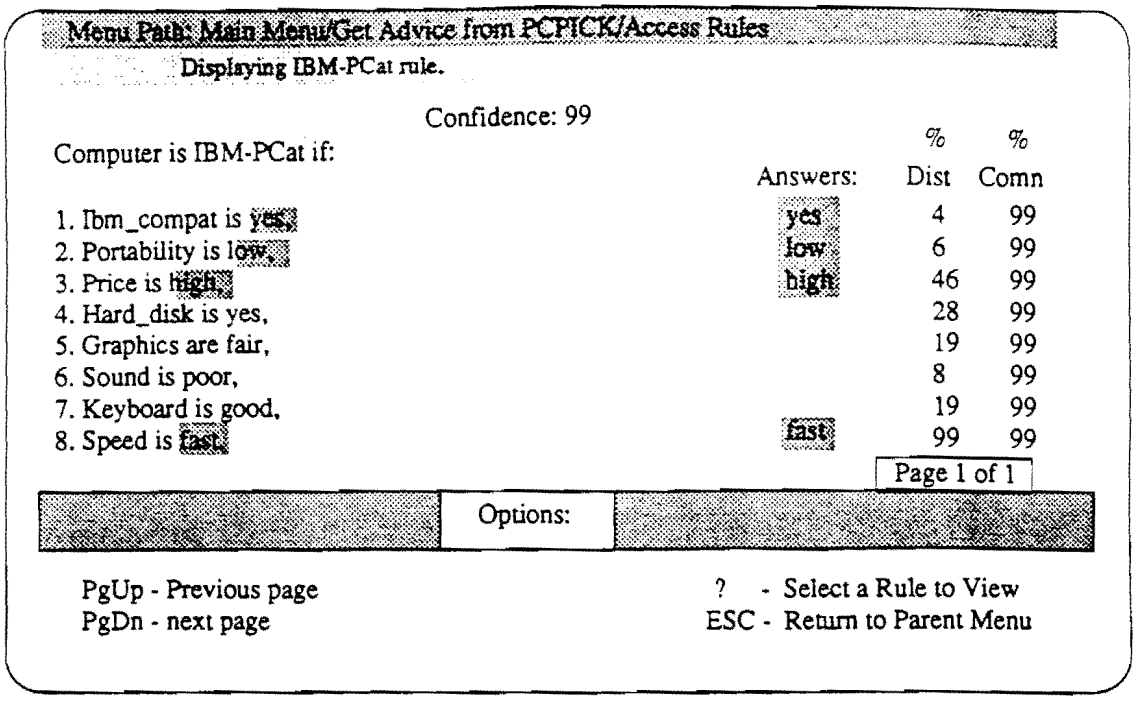


Figure 24. Access to Rule Base

Completion of the Advisory Session

When the evaluation by the system is finished,you see a screen with a conclusion (Figure 25). The second row on the screen is used to declare the certainty which has been generate through the advisory session. Below this row is the answer and any text that might be associate with that particular rule. The file is entered into the system through the variable editor, as explained in Chapter 6, Page 59. There are six selections in the options window that you may choose. A brief description of their function is below.

View Alternative Hypothesis

All possible hypotheses may be viewed upon selecting alternative hypotheses. A screen (Figure 26) is displayed. The purpose of this screen is to provide you with the relative confidence levels developed for each potential rule or hypothesis through the session. In this way, you will be able to determine how well the given advice could be separated from other possible choices. As you can see from Figure 26, the confidence level is given both in visual and numeric values. If more than one screen is required to display all the rules, the page down and page up key will allow you to move back and forth through the displayed rules.

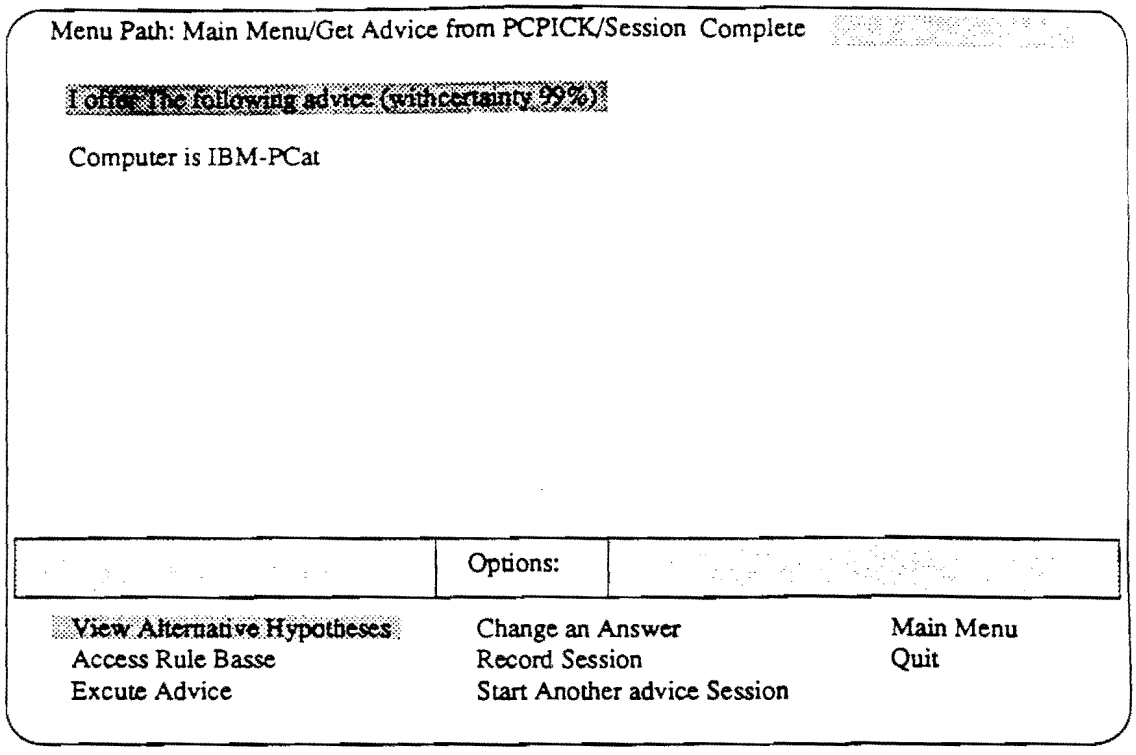


Figure 25. Final Screen From "PCP ICK" Advisory Session

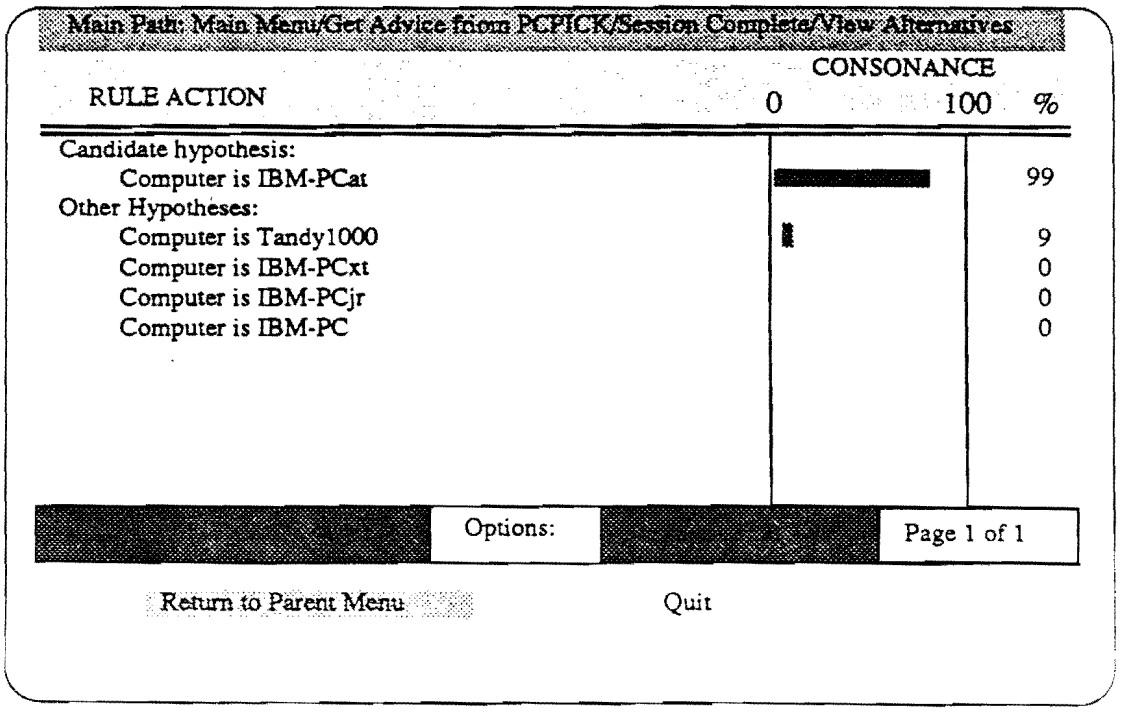


Figure 26. View Alternative Hypotheses

Access Rule Base

If access rule base is selected, an additional screen (Figure 27) is displayed. Use the directional keys to select the appropriate rule for viewing. An example of a rule can be seen in Figure 28. With this option, you may determine how useful or close the match was in the evidence given versus the criteria set for confirming the rule. Each piece of evidence given is written in the answer column with its value highlight if it was present in the condition. The last two columns on the right show the assigned distinctiveness weight and commonalty weight, which were present at the time the system was developed. For more information on the use of these weights, see the section on Uncertainty in Chapter 6.

Change an Answer

To change an answer, one selects the option so named, which displays a screen where you type in the name of a

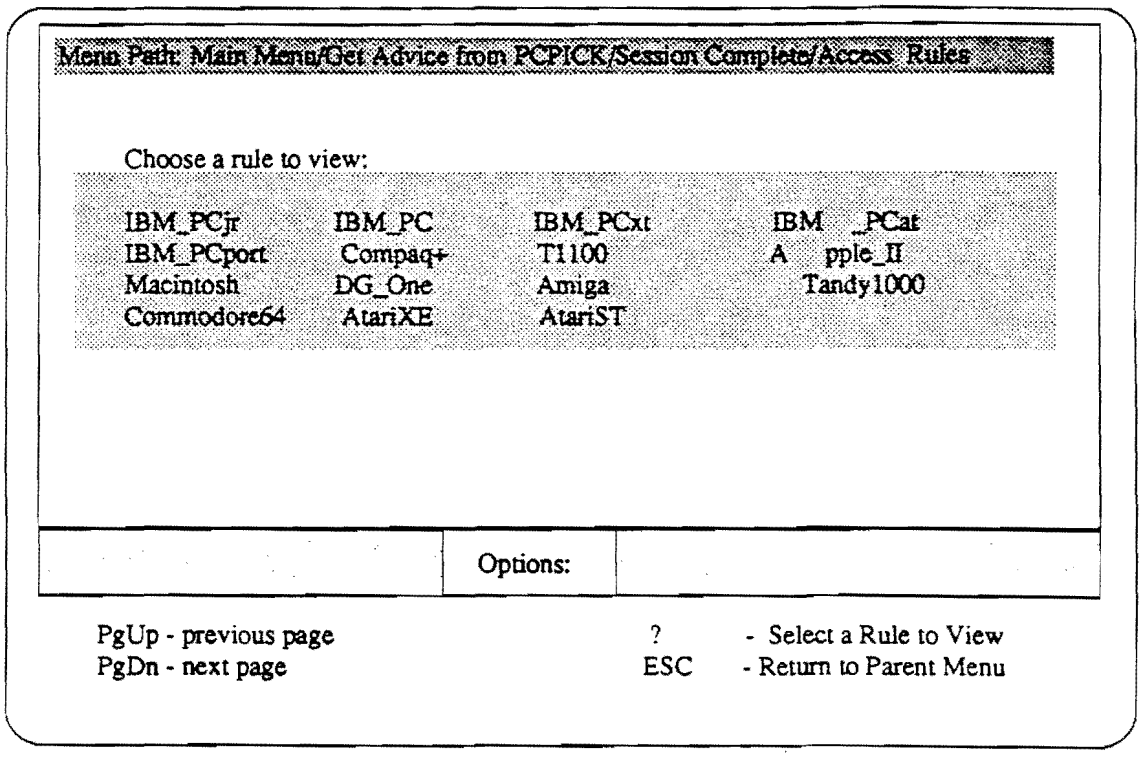


Figure 27. Access Rule Base

question whose answer is to be changed, or type a ? then select from the list of questions displayed. After selecting the appropriate variable or question, and entering the return, the screen clears and just the question portion of the advisory system is written back to a new screen. You may then change the answer from its current value. This often necessitates re-answering other questions because it might not eliminate a rule which was eliminated in the reduction phase.

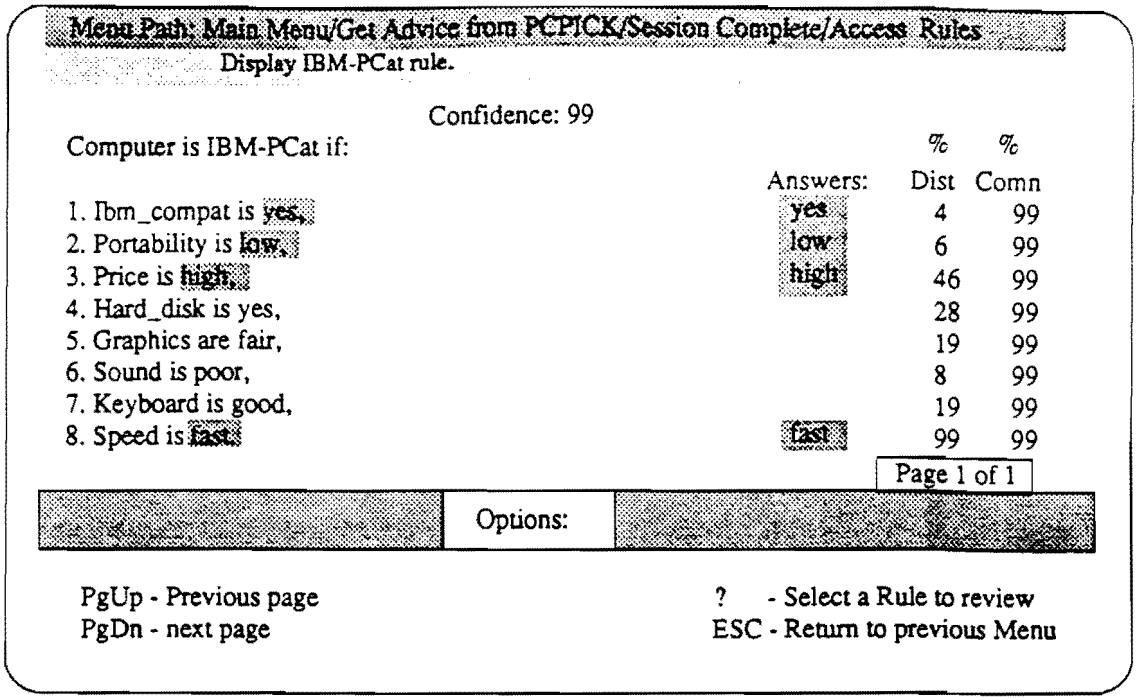


Figure 28. IBM-PCat Rule

The session, therefore, must begin over with additional questions being asked.

Record Session

To record an advisory session, you select the option *Record Session*. The following Figure shows that the *PCPICK.log* file is going to be recorded. This will save the values of variables presented in the session along with the final confidence level of all rules remaining after the reduction phase.

Start Another Session

Allows the user to clear AURORA advisory system memory and initiate a new advisory session.

Main Menu

Returns the user to the top level or Main Menu.

Quit

Quits AURORA and returns you to DOS.

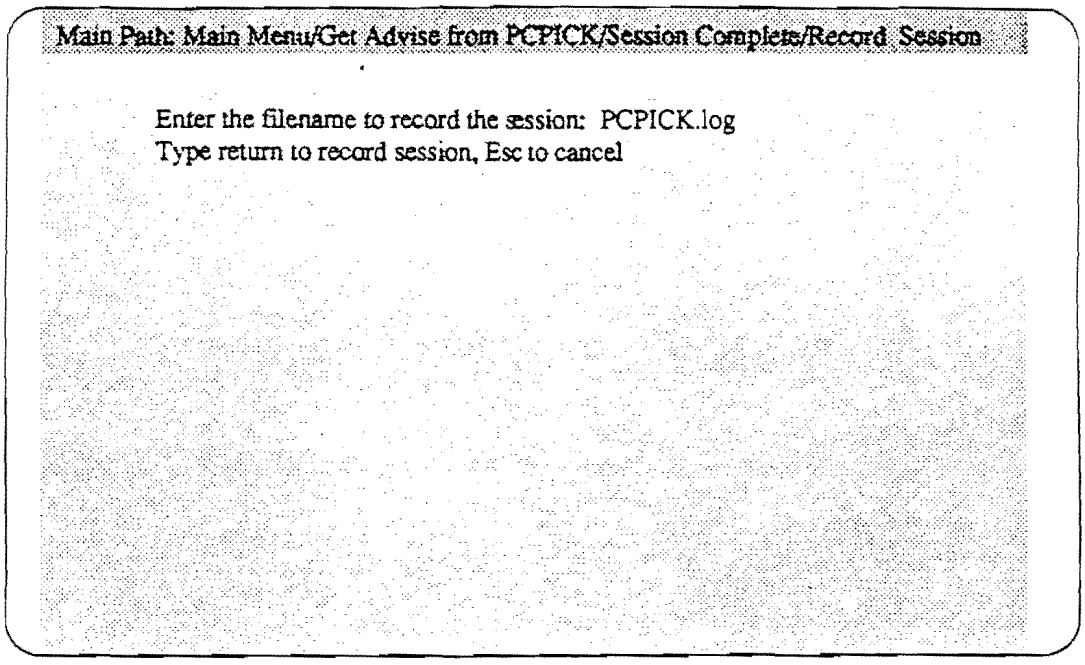


Figure 29. Record Session

CHAPTER SIX
DEVELOPING A NEW OR IMPROVING AN OLD ADVISORY SYSTEM

Without the ability to change or modify an advisory system, a tool such as a AURORA would be of minimal value. This chapter describes AURORA's advanced tools to perform these operations. From the develop or improve an advisory system menu (Figure 30), any operation which creates, modifies, or deletes any part of an advisory system can be undertaken. This menu is presented in a three column format allowing the exact selection of the advisory system you wish to operate on, the operation of choice, and the item of the system that you wish to work with.

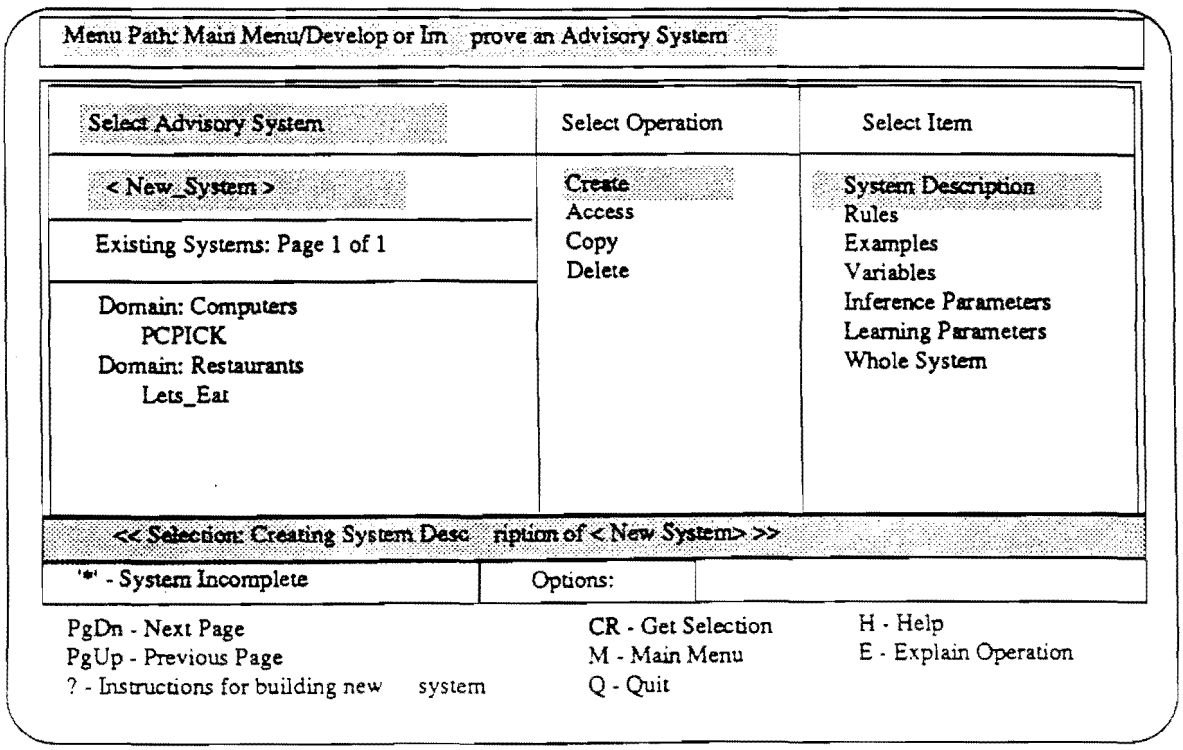


Figure 30. Develop or Improve an A dvisory System menu

Making a selection

The *Develop or Improve an Advisory System* menu is selected from AURORA's main menu (Figure 1). Once the menu is fully displayed, you will notice its three column format, at the top of each column is a header phrase labelling the operation for that column. Only one column header may be highlighted at a time, indicating which column is currently active. You can make a selection in the active column by moving the up or down direction keys to highlight the choice. In order to switch to another column, use the right or left direction key until the header of the column of your choice is highlight. You can then proceed to use the up or down direction keys for a selection in that column.

Left Column - Selecting an advisory system

The selection of an advisory system takes place in the left column. If more advisory systems have been created than will fit on a single screen, the up or down direction key may be selected to scroll through additional systems. In addition, the PGDN and PGUP keys can be used to scroll an entire screen at a time, either backward (PGUP) or forward (PGDN).

Notice that the first selection in Figure 30., titled *New Systems*, is separated from the others. When it is selected only the *Create* Operation, and 'Whole System' selection in the Item column can be chosen. This is the only mechanism for creating a new Advisory system (page 40). A different selection in the left hand column will pick one of the existing advisory systems to further develop or improve it. If an asterisk (*) is shown in front of a system name, it indicates the system has not been compiled (see page 52) and therefore not ready for use. Advisory systems are grouped within domains due to the similarity in variables used for selection (see creating a new system, page 40)

Middle Column - Selecting an operation

Once the advisory system is chosen, with the right direction key, move the header highlight to select the Operation (middle column.)

Create

Will set up an empty file for the selected item allowing you to add new information. If a file already exists for the the selected item an error message will appear suggesting you select the *Access* operation.

Access

Allows editing of any item already created. If the item in question has not been created, it prompts you to change to the *Create* Operation.

Copy

The copy selection will allow the input transfer of the currently selected item in another advisory systems. When this option is selected, the following will appear at the bottom of the screen:

Copy <item> from <system> to:

(Enter an Expert System Name or type Esc to return to Developing ES)

Delete

This option will delete the current selection from the selected advisory system. If delete is selected, the following will appear at the bottom of the screen:

Are you sure you want to erase the <item> of <system>? (Y/N)

In the case of deleting an entire system:

"Are you sure you want to erase the entire <system>expert system? (Y/N)

Right Column - Selecting an item

After selecting an operation, use the right direction key, to move to the right column, select an item to operate on. The items can range from individual components of a system to a whole system, its description, or the parameters for either an advisory session or for learning.

System Description

This item selects the system description menu (page 41).

Rules

An ASCII formatted file (<system.rle>) with the name of the selected advisory system will be selected from the current drive. Rules in AURORA syntax which have not been complied are contained in this file.

Depending on the selected operation additional menus will be displayed.

Examples

An ASCII formatted file (<system.dta>) with the name of the chosen advisory system will be selected. This file contains examples of previous expert decisions. If the *Create* or *Access* operation was chosen the example editor is displayed. (for more detail, see page 55)

Variables

An ASCII formatted file (<system.var>) with the name of the chosen advisory system will be selected. Variable files contain a list of all declared variables and their set of values. They are used in conjunction with example files to build new rules. If the *Create* or *Access* operation was selected the variable editor will be displayed, (page 60)

Inference Parameters

An ASCII formatted file (<system.inf>) with the name of the chosen advisory system will be selected. This file contains the most recently saved inference parameter settings selected from the inference parameter table (page 64), which will be displayed.

Learning Parameters

An ASCII formatted file (<system.lrn>) with the name of the chosen advisory system will be selected. Learning parameters selected from the learning parameters table (page 67), will be displayed.

Whole System

Selects all files associated with an advisory system for deletion or copying. If the operation Access is also selected a new window (Figure 31) appears in the center of the screen. Previously created files for the selected advisory system are indicated.

Additional options

Additional options are at the bottom of the screen in the options window.

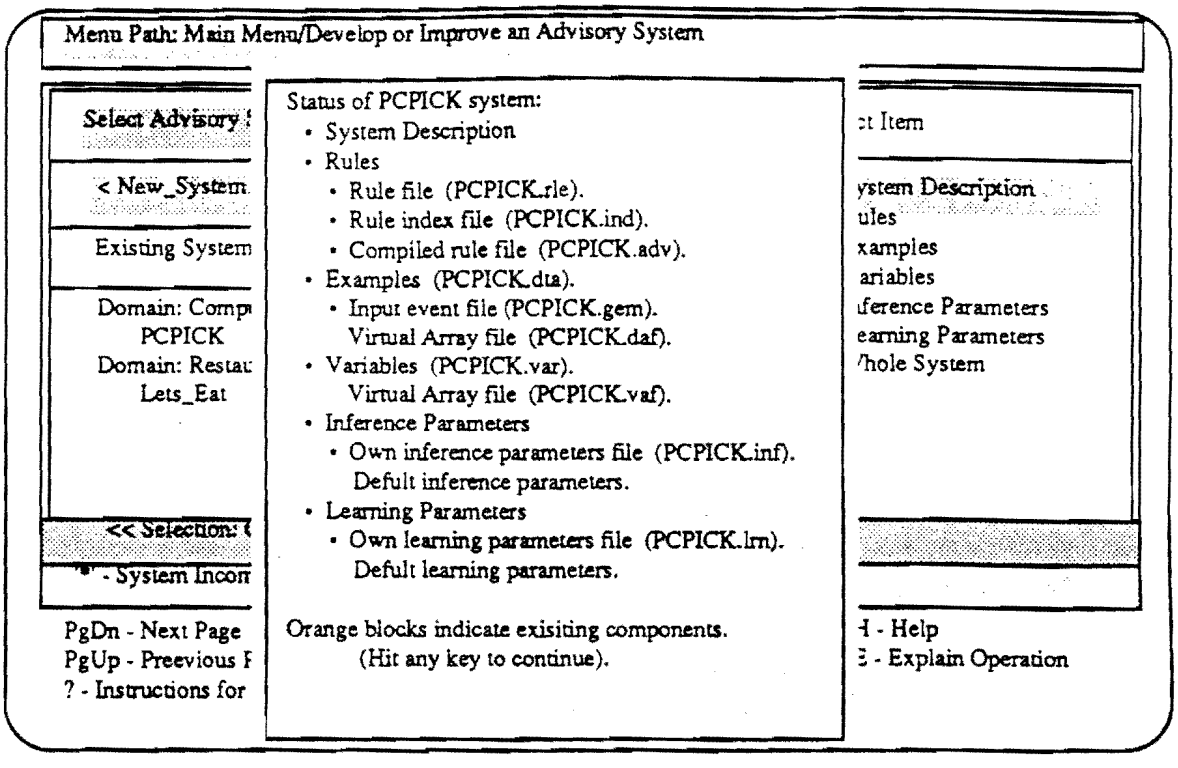


Figure 31. System Status Window

PgDn - Next page

List the next screen of Advisory system names. The page number listing in the lower left hand corner of the screen will be updated. PGDN only affects the left column.

PgUp - Previous page

List a previous screen of Advisory system names. The page number listing in the lower left hand corner of the screen will be updated. PGUP only affects the left column.

CR - Get Selection

After making a choice in each column, typing a <CR> will conclude the selection process and present the next relevant screen.

M - Main menu

Will select and display the AURORA main menu (see Chapter 3).

H - Help

A short message (single screen) to provide assistance in using the *Develop or Improve an Advisory System* menu.

E - Explain operation

If you have questions on the extent or the method of using an operation (center column), information will be provided on the highlight operation.

? - Instruction for building a new system

Presents a single screen (Figure 2) with two windows. The top window presents four steps to constructing a new advisory system. An expanded explanation of each step is displayed in the lower window. The up and down arrow keys may be used to select one of the four steps. As each new step is selected the message changes in the lower window.

Q - Quit

To quit AURORA and return to DOS.

Description of combinations of operations and items

After a selection is made from each of the columns and a <CR> is pressed, another menu may be activated and presented to you. The following section has been organized into groups representing combinations of operations and selected items.

Creating a New System

In order to create a new system, you must select *New System* in the left column. Selections for an operation and an item have been limited to *Create* and *System Description*, respectively. After striking the carriage return, the system description menu will appear. (Figure 32) Four items can potentially be added or modified in this menu.

Expert System Domain

The domain will default to the first domain listed in the systems description file (system.aur). This is also the domain listed first in the *Select Advisory System* column below the *new systems* selection. A new domain can

be entered. (1 to 18 characters) You may also type a ? to select among the list of existing domains. When the domain is selected, variables that are characteristic to that domain will be used as default variables for the new system and will appear in the example or variable editors when first selected.

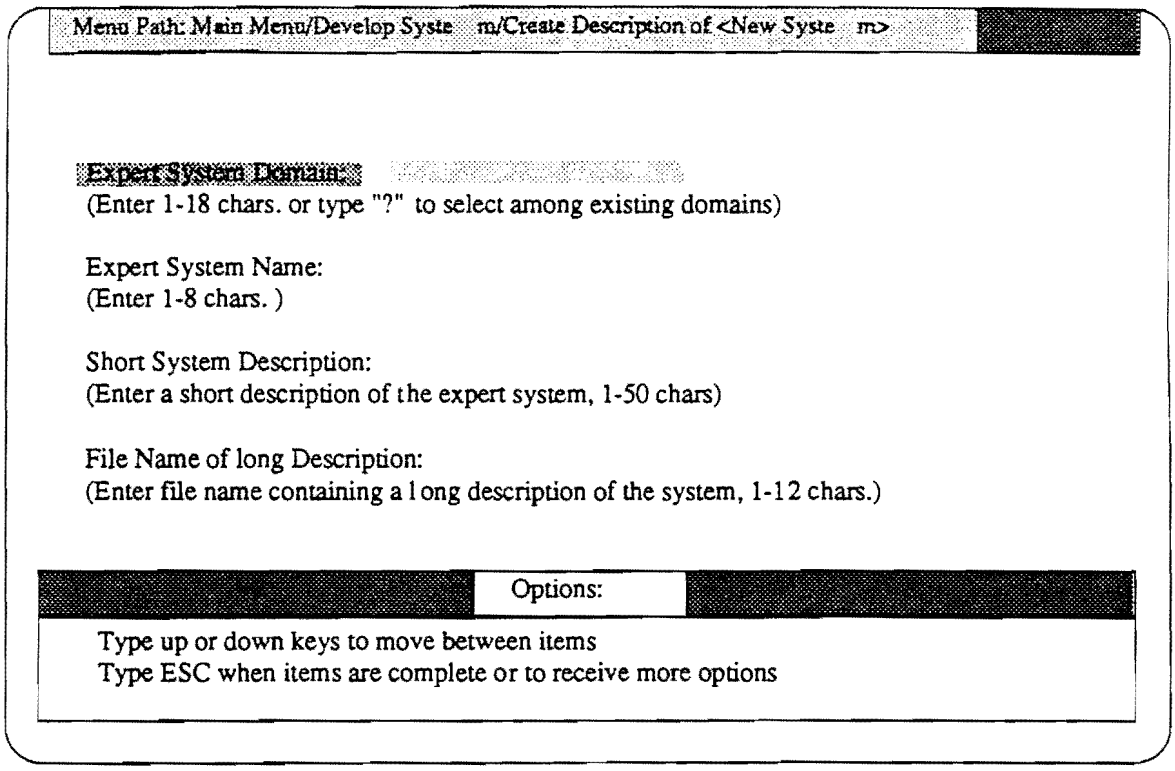


Figure 32. System Description Menu

New Expert System name

Provide a name (1 to 8 characters) for the newly created advisory system. No extension to the name is necessary or permitted, but use normal DOS file naming syntax. The selected name will also be used to label all associated files (example, variable, parameter, etc.)

Short System description

Short System Description will be displayed on the right hand side of the advisory selection menu (1 to 50 characters.) It provides a brief description of the system.

File name of long description

For a greater description of the domain, enter the name of a file and its extension in the space at the right hand side. This file will be placed in the Help window when help is asked to explain the advisory system.

Options

Pressing the Esc key will display additional options at the bottom of the menu. You will be presented with the choices of:

Create new system and return to developing ES menu

This will save the indicated changes or additions and return you to the *Develop or Improve an Advisory System* menu. You will also be asked if you desire to inherit variables from an existing advisory system. If you answer No, the system description will be saved without the development of a variable file for that system. If you respond Yes to the question, you will be presented with the opportunity to either enter the system name or to type a ? to bring up the names of additional systems. This option will build a variable table with the variables and their values from the selected advisory system.

Return to the developing ES menu without creating

As above, however any changes made will not be saved.

Resume entering items

The cursor will be moved to the *Expert System Domain* field at the top of the menu.

Quit

To quit AURORA and return to DOS.

Access or Create Rules

Creating Rules

If the *Create* operation is selected with the item *rules*, a menu similar to the Access Rules operation is displayed. (Figure 33) There are five possible selections on the Create Rules Menu: *Learn Rules From Example, Use*

Editor to Create Rules by Hand, Transfer and Edit Rules, Parent Menu, and *Quit*. If the advisory system selected for the creation of rules has existing rules, a warning window (Figure 34) will appear on the lower third of the screen. By pressing any key you will be placed back at the *Develop or Improve an Advisory System* menu in which you can select *access* for the operation.

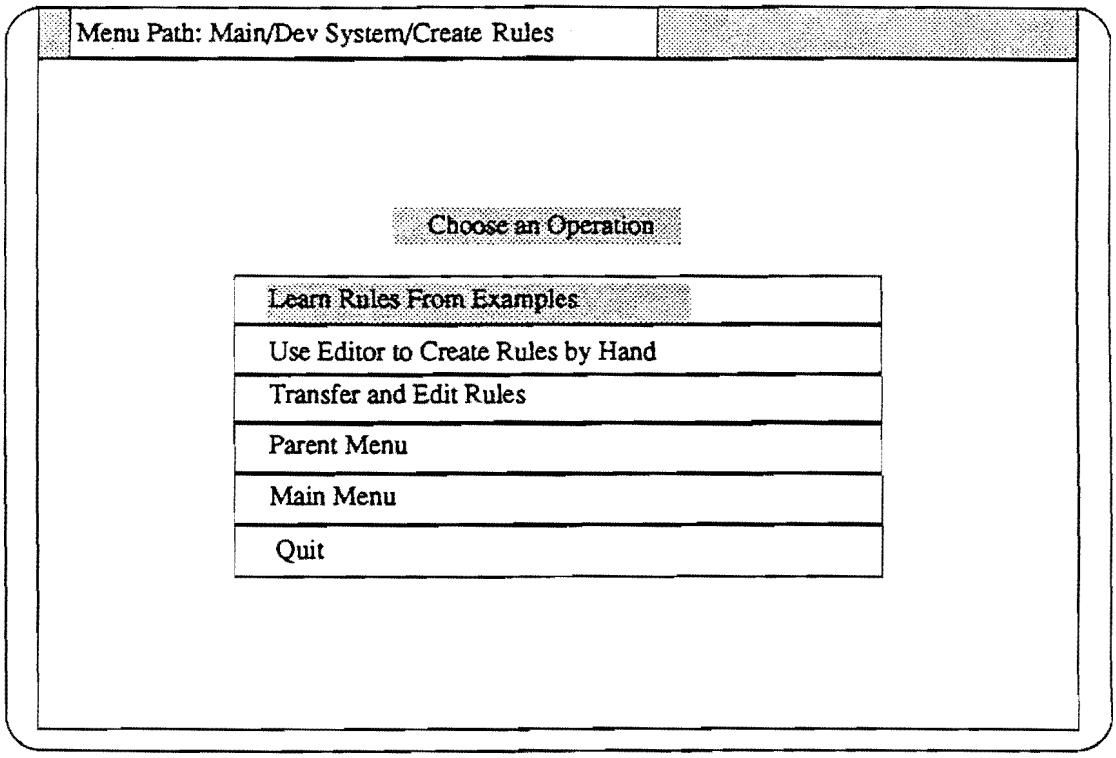


Figure 33. Create Rules Menu.

Learn Rules from examples

Selection of the Learn Rules From Examples line will allow you to move through the steps necessary to prepare rules through inductive learning from examples. This procedure is covered in detail in chapter 9, page 75. The screen shown in Figure 19 will appear.

Example KB:

The top line is the example base which should contain the new set of examples. You can, however, type a ? to list all of the potential example bases to choose from, or type in the name of any other existing example base.

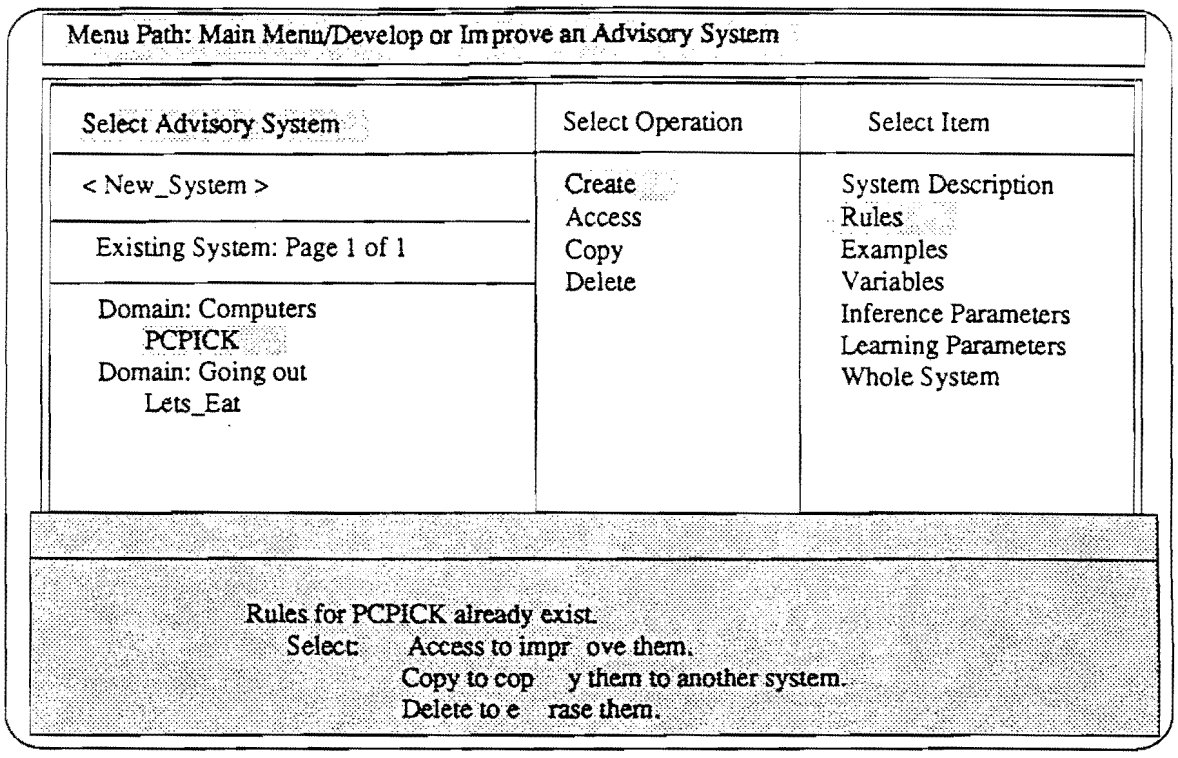


Figure 34. Warning for Previously Created Rules

Concepts to learn:

This will allow you to select a single event or decision variable value to learn from rather than from all the examples, which is the default value. To provide a list of the concepts or variables available for learning, type a ?. A list of all concepts available will appear just under the title of the menu. The right or left direction keys can then be used to select a concept to be learned.

Output to KB:

A third option which must be set or left at its default value is the output to knowledge base line, which indicates the name of a file which will contain the rules as they are developed by the learning module. This name will default to the name of the advisory system selected in the *Develop or Improve an Advisory System* menu.

Start Learning

When the options have been modified as required, learning is started by selecting *<<Start Learning>>*. Most often you will bypass changing any of the default levels asked for on the menu and proceed with

down arrow key directly to the last selection, <<*Start Learning*>> to begin the learning process.

This will momentarily blank the screen, bringing up the Reading Learning Events screen (Figure 35.) which provides a graphic bar indicating each item as it is read. As items are read and processed, the bar will increase in length from left to right.

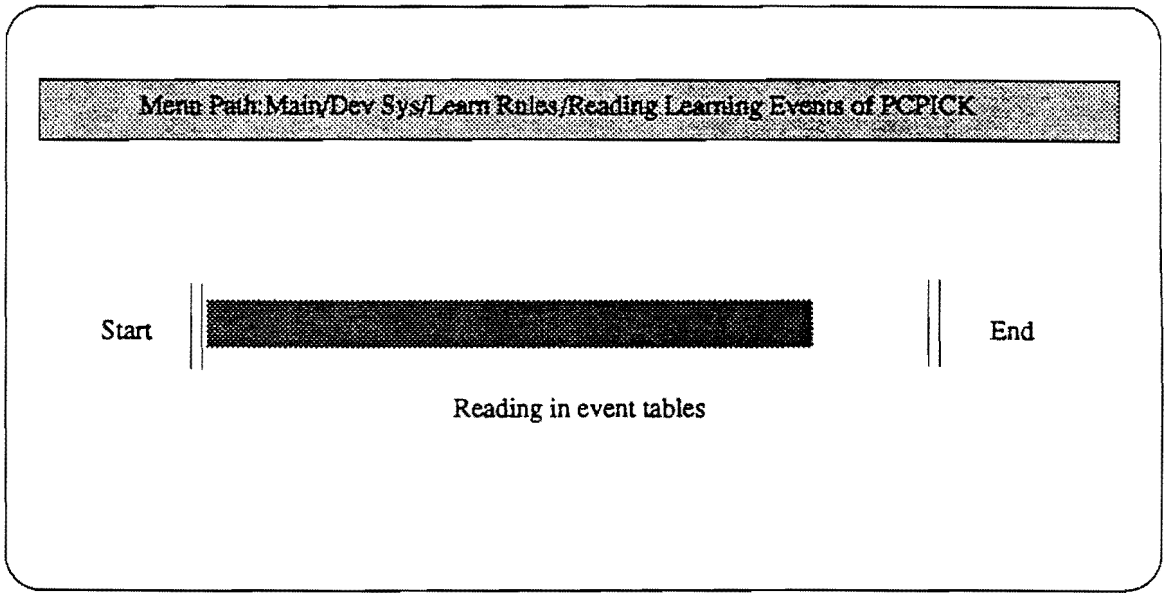


Figure 35. Reading Learning Events Screen

As each rule is created, it will be displayed on the screen. (Figure 36) As rules are being constructed, asterisks (**) will appear to the right of *Creating IBM-PCxt rule* to indicate that learning is in process. When the rule is displayed, the type of rule which was developed will be listed in the second line. Also on the top line will be the labels Confidence level and Events covered (positive, negative and total positive).

The rule will be listed in AURORA rule syntax with each variable being numbered in consecutive order. The confidence level will be listed as they have been computed in the learning process. In the right-most column, the number of positively covered events or negatively covered events will be listed. And at the bottom of the rules, the total positive events covering all variables displayed in a complex will be listed.

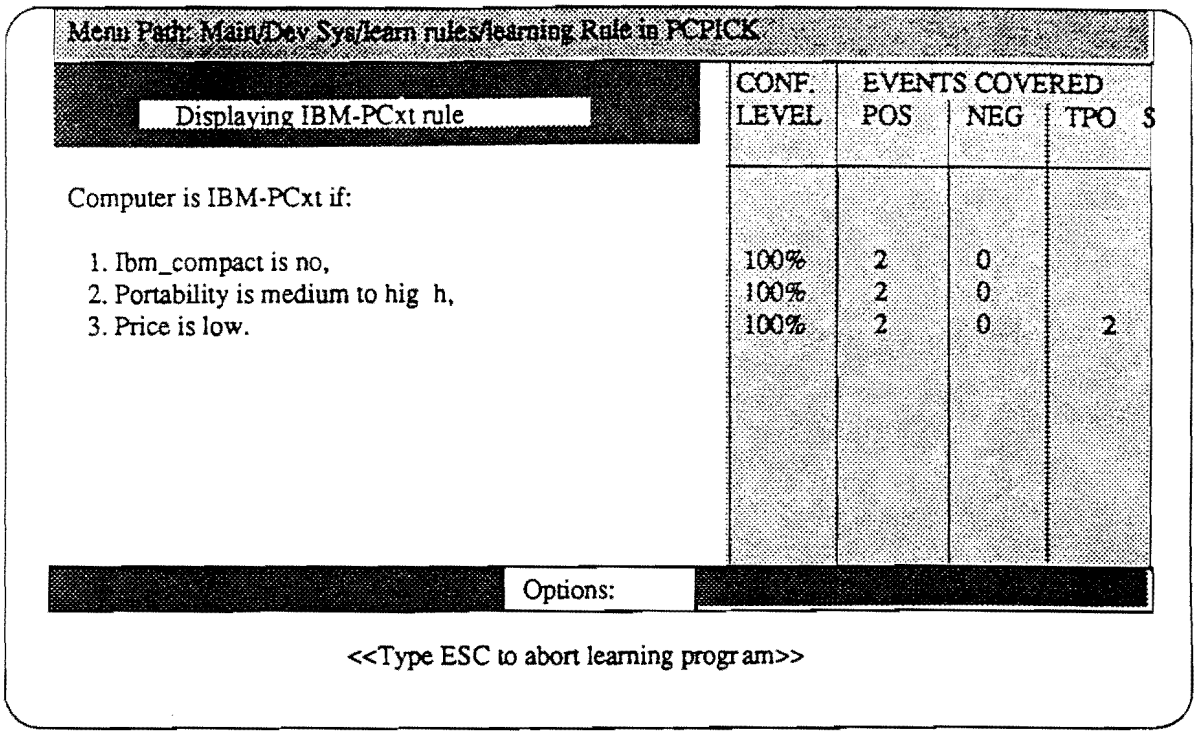


Figure 36. Learning PCPICK Rules

Options

If the user should decide to stop or interrupt the learning process, type the Esc key.

Options on completion of learning

When the last rule is learned and displayed (Figure 37), the options section or lower portion of the screen will change, providing several options to the user.

View a Rule

This option presents you with the following three additional options for examining the rule base.

? - Select a rule to view

The name of a rule may be entered on the line requesting rules to view or a rule may be selected from those listed on the screen. If multiple complexes are available for the selected rule, they can be brought to the screen by using the PGUP or PGDN. Figure 38 presents a sample of a rule to view.

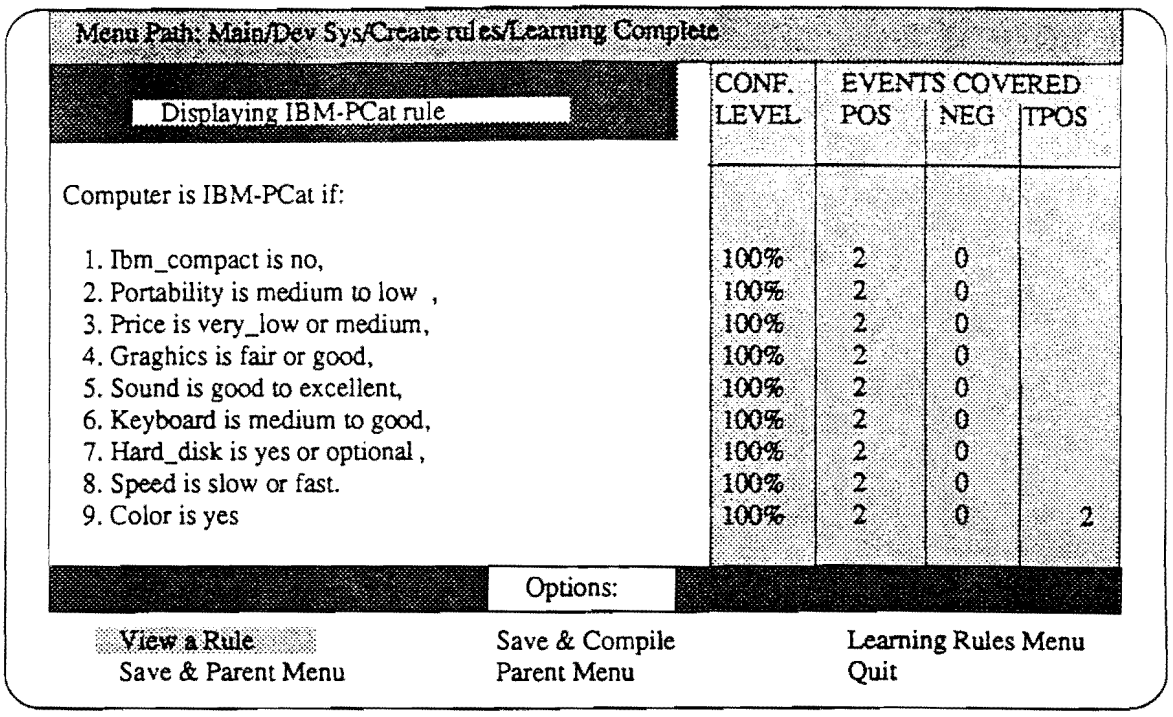


Figure 37. Completion of Learning PCPICK Rules

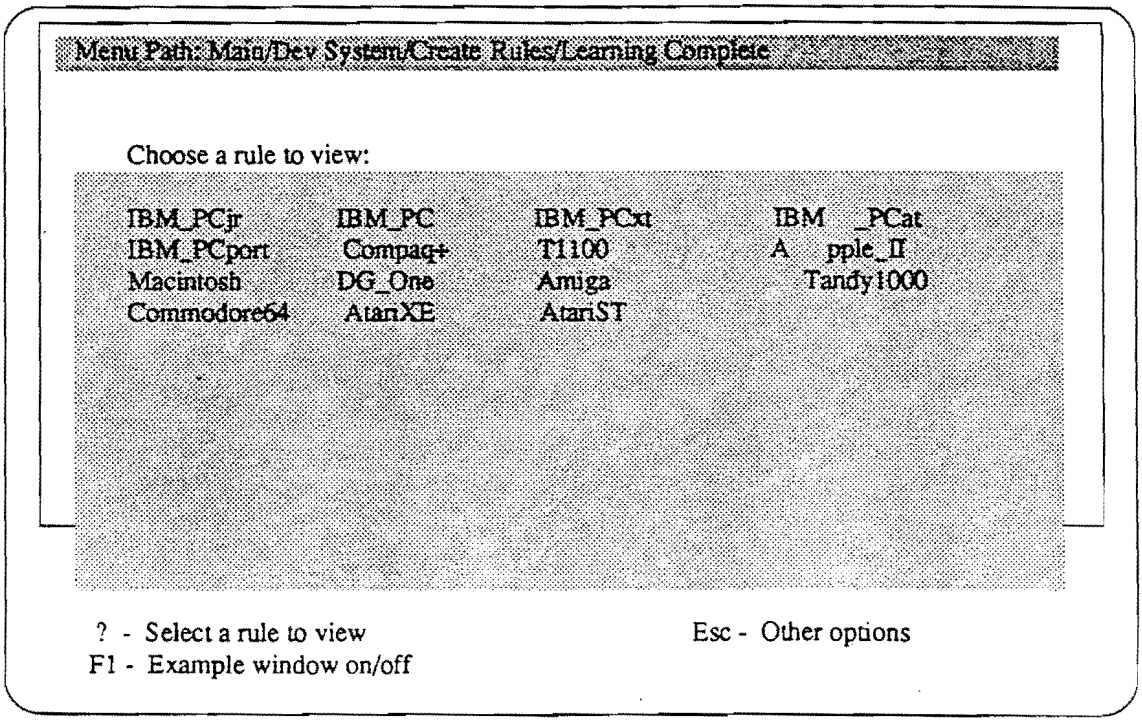


Figure 38. Select a rule to view

F1 - Example window on

If F1 is selected, the screen is split with a second window displaying the example table, allowing the user to scroll through the table to look at the examples which were used to develop the rule. (Figure 39) Selecting the F1 key again will close this window. Selecting F2 will allow the selection of a different rule or return you to the example table.

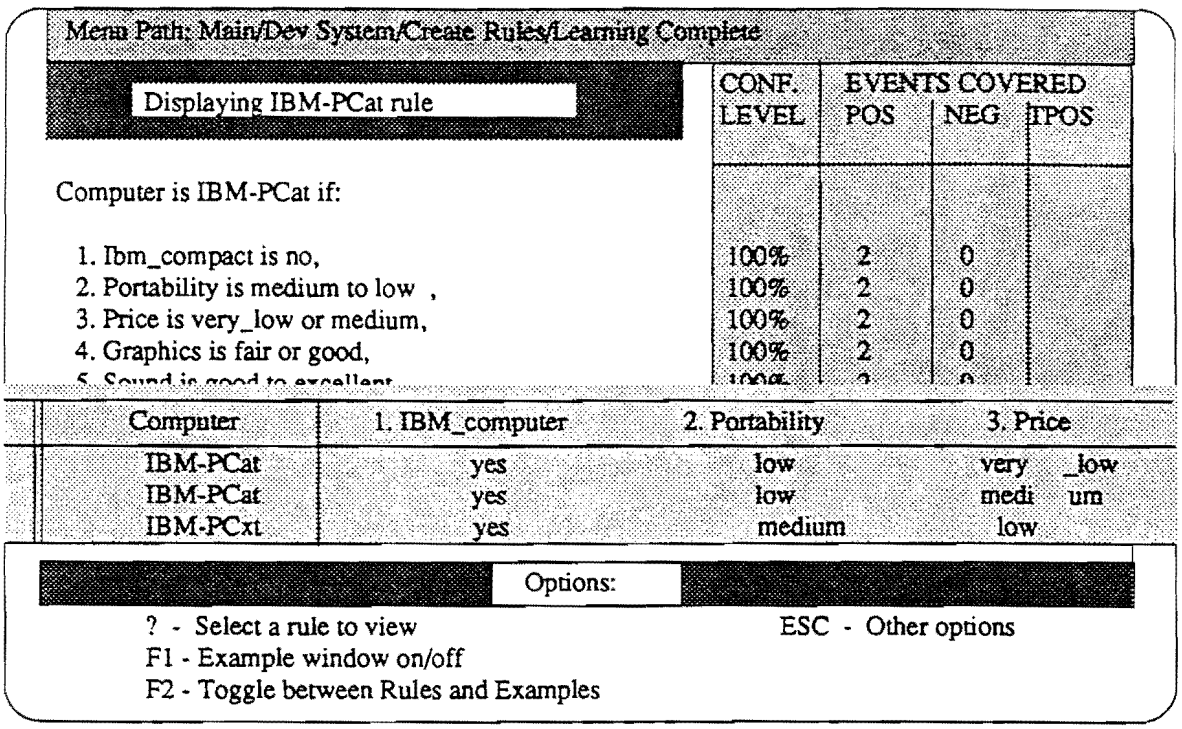


Figure 39. View a Rule with Examp le Window On

ESC - More options

Selecting the Esc key will return you to the set of previous options.

Save & Previous Menu

After you are prompted for a file name, the rule set is saved and the parent menu is presented.

Save & Compile

After you are prompted for a file name, the rule set is saved then compiled.

Parent Menu

You are returned to the parent menu without saving the newly created rule set.

Learning Rules Menu

You are returned to the Learning Rules Menu which is generally the parent menu without saving the newly created rule set.

Quit

Stops AURORA and returns you to DOS.

Using an external editor to create rules by hand

After selecting *Use Editor to Create Rules by hand*, a window requesting the name of an editor to use will appear (Figure 40). After adding the name and pathway of your editor of choice and typing a <CR> the editor will be presented. For more detail on rule construction see page 81.

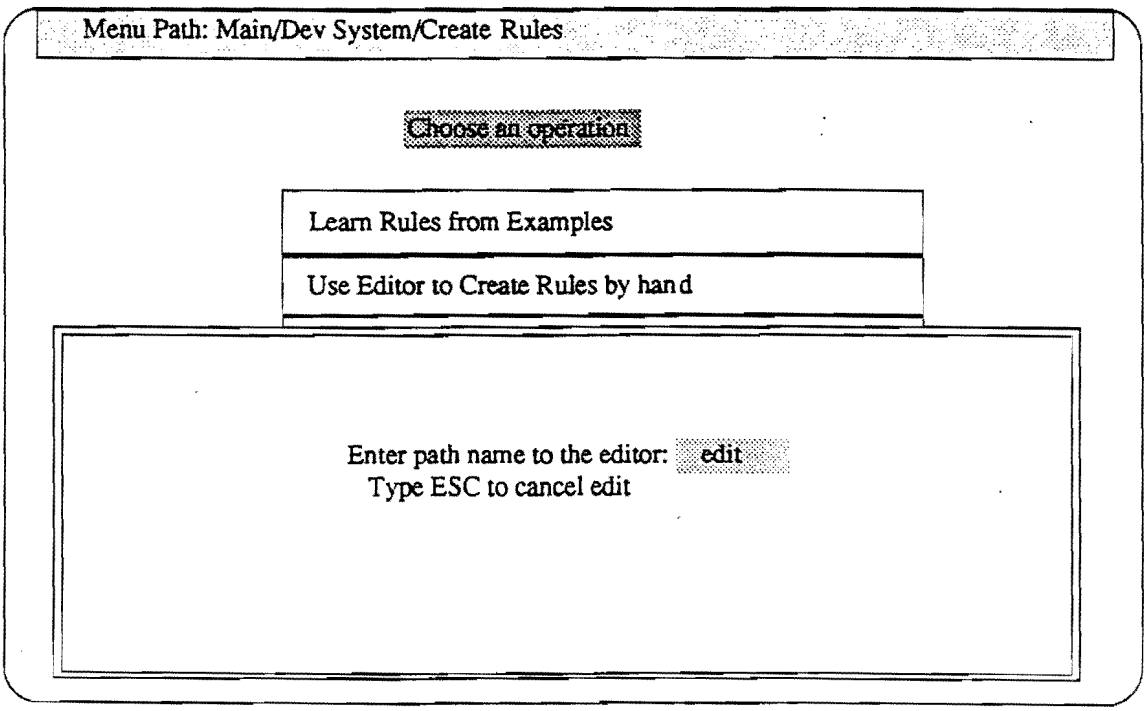


Figure 40. Request for an Editor

Transfer and edit rules

When selected, a message appears at the bottom of the screen, prompting you for the name of an advisory system containing rules to be copied. If the system entered is available, the rules from that system are copied to the current system, and your chosen editor.is invoked. This option may be useful in transferring rules from one advisory system to another. This may be the initial set of rules for a new system or additional rules from another existing system to add to the current system. The Esc key may be pressed to return you to the *Create Rules* menu.

Parent menu

This option will exit the *Create Rules* menu and return to the *Develop or Improve an Advisory System* menu.

Quit

Quit AURORA and return user to DOS.

Accessing Rules

Some of the selections available on the *Access Rules* menu are similar in function to those available on the *Create Rules* menu. After selecting the appropriate advisory system, *access* for the operation and *rules* for the selected item, a tap on the enter key <CR> will send you to the . If no previous rules exit, a warning window (Figure 34.) will appear to allow you to change the operation to create. The following selections are available in the access rules menu shown in Figure 41.

Edit Rules

Allows you to modify or add rules with the aid of any external editor or word processor. You may choose the name of any editor currently available to DOS. A window similar to the one shown in Figure 40 will appear. Due to AURORA's large memory requirement, it may be only possible to use an editor with a minimal memory requirement. When this option is chosen, a file with the name of the selected knowledge base and an extension ".rle" is loaded. Complete control is then given to the editor which may be used in the same fashion as if it were executed by itself. When modification of the rule base have been completed, exiting from the editor will then return you to the *Access Rules* menu. As with all modifications to rules, they must be compiled before they can be used in an advisory system.

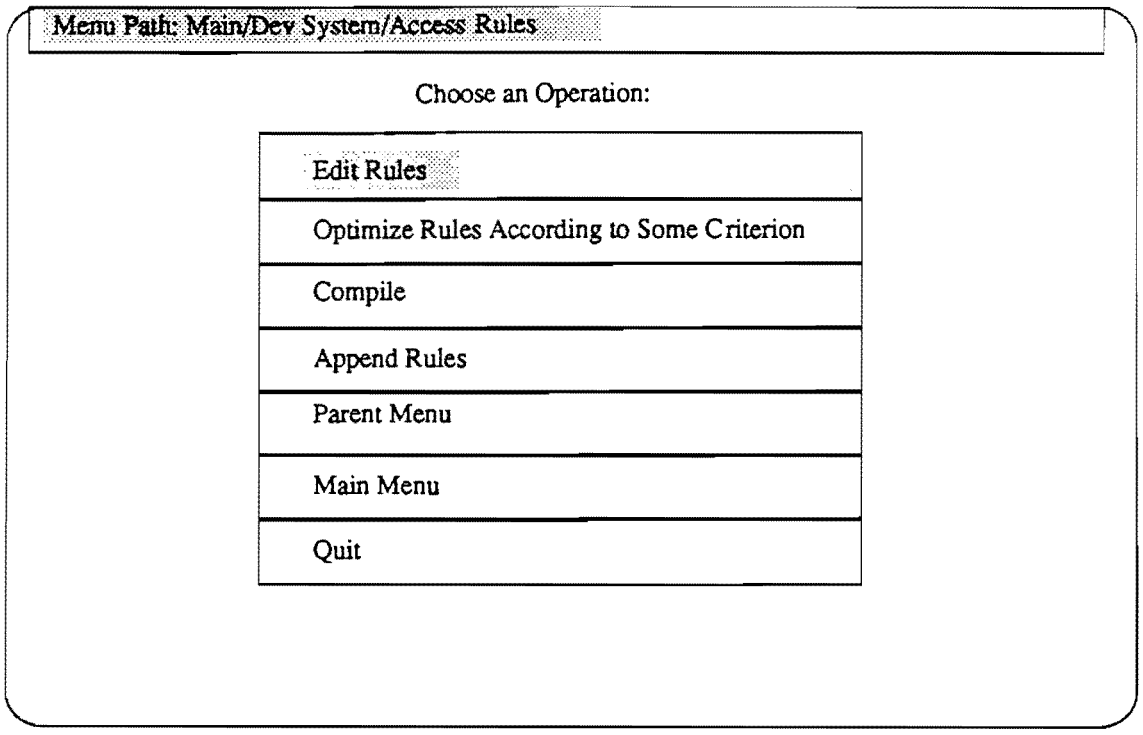


Figure 41. Access Rules Menu

Optimize rules according to some criterion

This procedure is used to change the current values and variables of all rules according to a new set of learning parameters (For greater detail see page66 and chapter 10). The current rule type setting is display at the bottom of the *Learning parameters* menu. Unlike Learn rules by example, which requires an example set, the optimization of rules only requires a set of rules. You may want to optimize rules by redefining *characteristic* rules to a more discriminate set. When the selection is made, an Optimize Rules menu will appear. (Figure 42)

Input AS

You may change the currently listed files to provide rules from another advisory system for optimization.

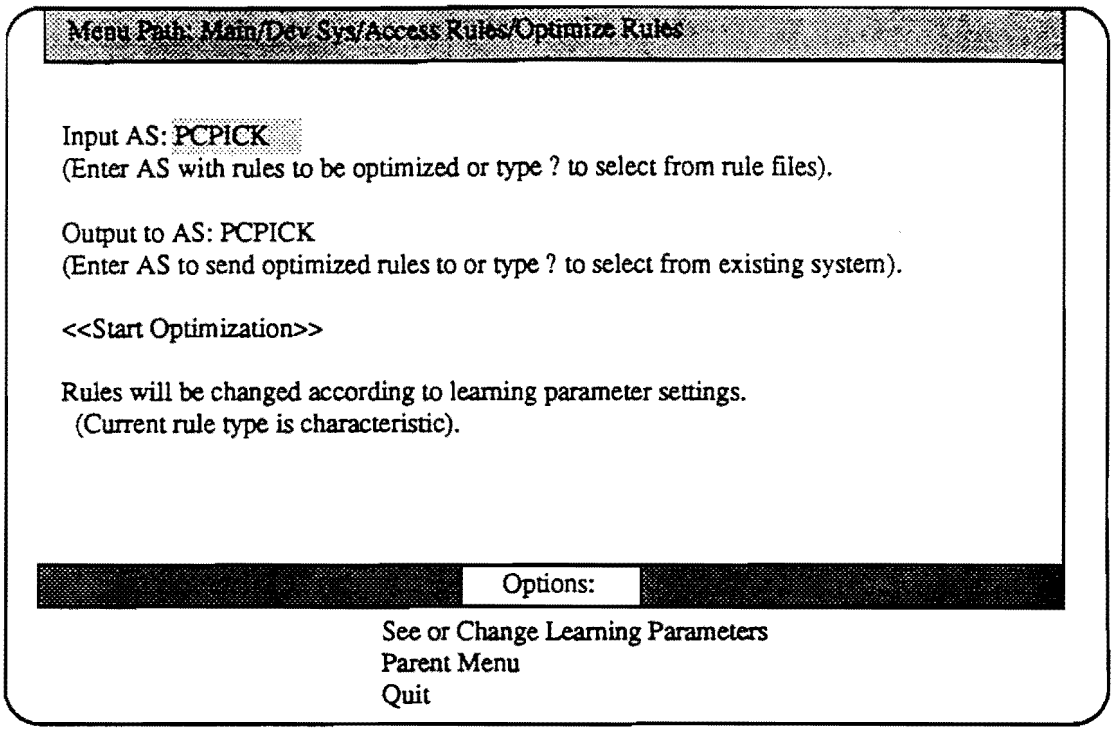


Figure 42. Optimize Rules Menu

Output to AS

Changing the currently selected advisory system to a new name will allow you to save the optimized rules to a second knowledge base, giving you the opportunity to save both sets of rules.

Start optimization

Learning will proceed as described in chapter 9. In order to check on the planned optimization, the current rule type to be used in optimization is listed in a statement just below start optimization.

Options:

Reached by moving the down arrow key to highlight any of the options.

Seeing or changing learning parameters

This will provide the learning parameters menu (Figure 47) for changing of the rule type or other parameter settings. Greater detail on the function of these settings can be found on page 66 and chapter 9.

Parent menu

This option will return you to the access rules menu.

Quit

Stops AURORA and provide the DOS prompt.

Compile Rules

Selecting this option will blank the current screen and begin the compilation procedure. It will take a while for the system to load the compiler, which is held in a separate file. Details of rule compilation can be found on page 82.

The compiling rules menu (Figure 43) will appear and the outcome of the system's attempt to compile each rule will be displayed in the order which it occurs. A tap on the escape key will stop the computation. When complete the computation results are written to a file with the advisory system name and the extension ".adv". The compiler uses information from files with the advisory system name and extensions, ".var" or ".rle".

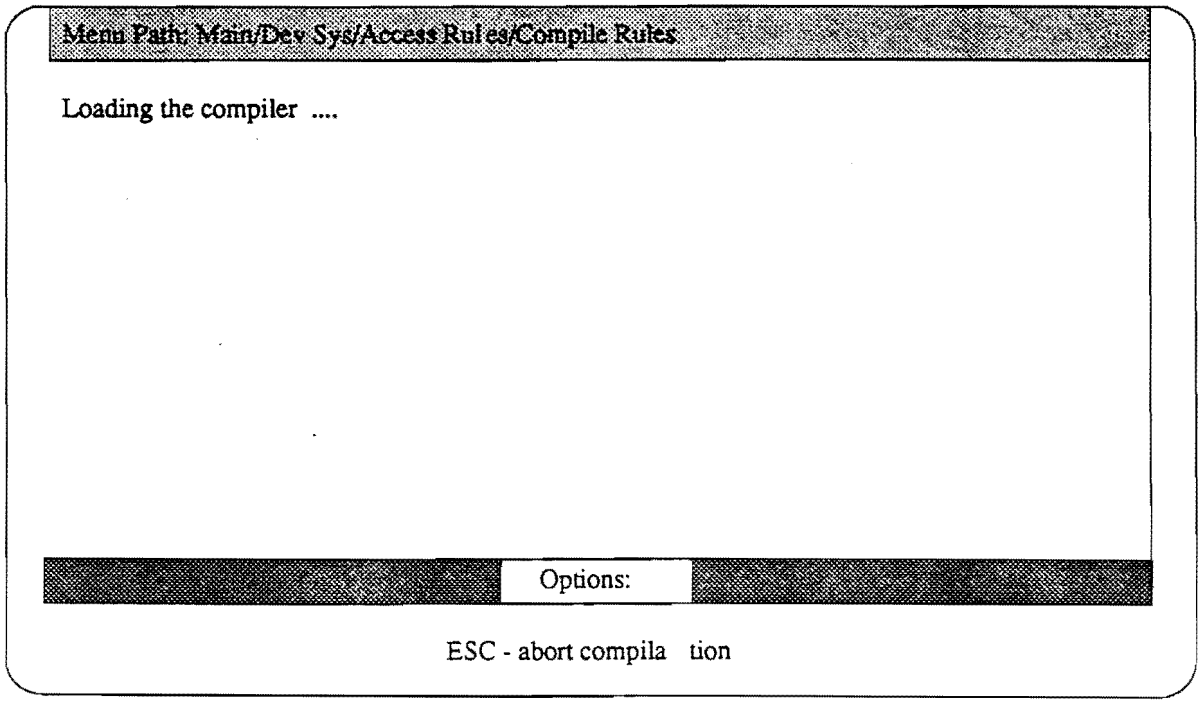


Figure 43. Compiling Rules

Options

When computation is complete, four options will appear, allowing you to move back to the *Access Rules* menu, the *Develop or Improve an Advisory System* menu, the *Main* menu, or to *Quit* AURORA altogether.

At any time during the computation, errors may be encountered and will be listed on the screen. These errors are listed in Chapter 10 and can be either fatal which will stop the system and allow the user to examine the error statement, or warnings which will not stop the system but which may not provide an optimum rule.

Comments

Any portion of a rule enclosed in brackets will be ignored in the compilation, so comments can be added to the rule file. In addition, at the end of the compilation period, the rules utilized in the compilation are copied to a file of the same name with the extension ".crl", which is used by other portions of AURORA to provide consistency between the compiled and uncompiled version of the rules.

Append Rules

Instructions will be provided at the bottom of the screen asking you which system to append the rules from.

What rules do you wish to append to the existing _____ rules?

(Use CR to enter system, ESC to return to previous menu, ? to select system)

A <CR> will append the rules selected. Esc may be selected to ignore this option or a ? can be selected to provide a listing of current systems with rules.

Parent menu

Moves you back to the *Develop or Improve an Advisory System* menu.

Quit

Stops AURORA and returns the DOS prompt.

Access or Create Examples

Accessing previously created examples or creating new examples proceeds in a similar manner. If an attempt is made to access examples which have not been previously created, a message (Figure 34) will appear at the bottom of the screen allowing you to change the operation to *Create*. Likewise, if you try to create examples that already exist, a message will appear providing you with a means to change the operation to *Access*.

Once the appropriate operation is selected, the example editor (Figure 44) is displayed. The general operation of this editor is very similar to many spreadsheets, except the column width is fixed. The leftmost column represents the decision class for each group of examples. A rule will be produced for each unique decision class value. All columns to the right of the leftmost column represent various selectors or variables which make up the set of examples provided. You should start a new set of examples by declaring the name of the decision variable or goal of the advisory system. This is placed in the upper left-hand cell. You then list the names of the variables to the right, in the order their values will be listed for each example (row). These names must follow the standard syntax of AURORA using 12 or less non-blank, characters.

	DECISION	EXAMPLES		
VAR	Computer	1. Ibm_compat	2. Portability	3. Price
TYPE	nominal	nominal	linear	linear
1	IBM-PCat	yes	low	very_low
2	IBM-PCat	yes	low	medium
3	IBM-PCxt	yes	medium	low
4	IBM-PCxt	yes	high	low
5				
6				
7				
8				
9				
10				

Options:

PgUp-Page up	F1-Insert row	F5-Grab column	F9-Edit domain
PgDn-Page down	F2-Insert Column	F6-TYPE/COST	F10-Legal values
CTRL →Page right	F3-Grab word	F7-Jump to decision	ESC-More options
CTRL ←Page left	F4-Grab row	F8-Jump to variable	

Figure 44. Example Editor

The second row, labelled *TYPE*, allows you to enter the variable type for the set of values associated with the variable name above. If no value is entered, a nominal variable type is automatically entered. For non-nominal types (see page 88 for a discussion of variable types), you are quarried to determine if you wish to edit the set of values for the variable in the variable table. The variable type may be changed by typing the first letter representing the various variable types (L, I or N).

The same row is also used to provide a cost value for the variable. Applying a cost to a variable will help to designate the relative importance or strength that variable has to all examples. It can be used to help minimize or promote one or more variables over the remainder. First strike the **F6** key to change the label *TYPE* to *COST* then enter a number from 0 to 100 with 100 indicating the greatest importance.

Options:

Many controls exist for entering and editing examples in the example editor.

Right or Left direction keys

The right or left arrow keys can be used to move the cursor from cell to cell to input examples. If more than four selector variables are to be used, the right arrow key will shift the displayed examples one column to the left.

PgUp or PgDn keys

Ten examples can be displayed at one time, use either the PGUP or PGDN keys to display the next or previous ten examples.

Control right or control left keys

The display can be shifted by four columns either right or left by using the <CTRL> RIGHT or <CTRL> LEFT keys.

F1 - Insert Row

A blank row will be inserted above the current (highlight) row.

F2 - Insert column

A blank column will be inserted to the left of the current column.

F3 - grab word

F3 will provide operators to the currently highlighted cell. At the bottom of the screen after selecting F3, the user will have the ability to:

- copy the contents of that cell to the cell below
- copy the contents of that cell to all rows (examples) with the same decision variable value (class).
- delete the contents of the cell
- abandon the operation and return to the example set.

F4 - Grab Row

If F4 is selected, the user is given the following options at the bottom:

- insert a copy of the row above the selected row
- delete the currently selected row
- resume the original edit without making any changes.

These three options will require the user to answer the number of the row to be operated on.

- copy to an another row
- move to an another row
- Switch row with another row

F5 - Grab Column

Will provide the same options as F4 but pertaining to columns.

F6 - TYPE/COST

The F6 function key serves as a toggle to shift the third row between designating a type for each variable and its relative cost as compared to other variables in the example set. A more detailed discussion on variable types may be found in Appendix B. When the row is toggled to the Cost Factor, the default value of zero will appear for each variable. You may then provide a number between 0 and 100 to show the relative weight that that particular variable has in providing evidence for the example. It is suggested to use the cost to set either all of the variables at a various costs or to leave the values at the default zero.

F7 - Jump to decision

Choosing F7 will move the current column highlight to the decision column remaining in the same row.

F8 - Jump to variable

Choosing F8 will jump to the variable name or top row, remaining in the same column.

F9 - Edit domain

Choosing F9 will provide the variable editor as described on page 59.

F10 - Legal Values

Each time this key is depressed a new value, which has been entered in the variable editor for the selected variable is presented in the current cell. You may cycle through the entire set of possible values until you find the one of interest. This may be an aid in entering values in a large example set and might also provide a mechanism for rapidly searching if a value has already been used.

ESC - More Options

If the Esc key is selected, the option fields will change, listing the following new options:

Save Changes

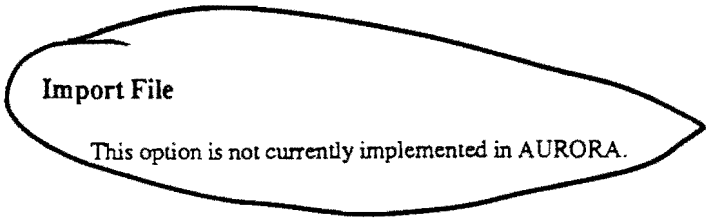
You will first be asked to either verify the knowledge base name for the current example set or to enter a new file to save the example. After saving the file you will be returned to the example editor.

Learning Menu

The Learn option will move you directly to the Learning Menu (Figure 18) without saving any current changes in the example set.

Append File

This option will allow you to add examples developed for a different advisory system. All the examples in that system will be appended to the end of any examples in the current system.



Parent menu

You will first be prompted to save the current example set, then the *Develop or Improve an Advisory System* menu (Figure 15) will be presented.

Main menu

You will be moved to the AURORA main menu without saving the current examples set.

Resume edit

The example file will not be saved, but you may continue editing in the current example set.

Quit

The AURORA system will end and you will return to the DOS prompt.

If you have not saved the example set you will be prompted with a statement:

Do you want to save recent changes? (y\n)

Access or Create Variables

The variable editor (Figure 45) can be entered by one of two methods. Select the **F9** key while in the example editor or select the combination of either the *Access* or *Create* operation and the item *Variable* from the *Develop or Improve an Advisory System* menu. As with the Example editor, if the *Access* or *Create* operation is selected and either the variables have not been created or they already exist, respectively, then a warning message will appear in a window in the lower portion of the screen allowing you to choose another appropriate operation to continue.

This editor is similar to the example editor in appearance. The upper left-hand cell contains the name of the decision variable provided by you or found in the example table. Immediately below it in the *TYPE* line is the default value *nominal* or the value selected by you in the example table. Each selector variable is listed in columns to the right with appropriate type and cost values provided in the example editor or by you. All current

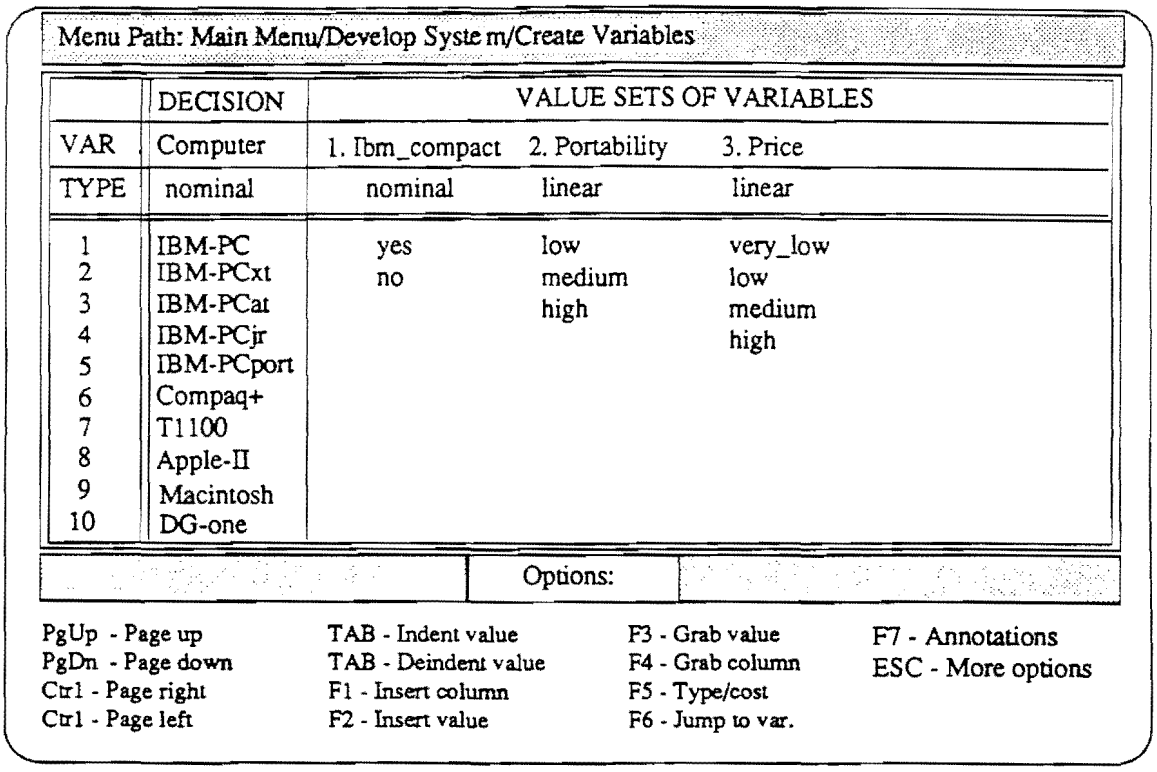


Figure 45. Variable Editor

declared values for each variable are listed in the columns in the remainder of the window.

The control of these values is similar to that used in the example editor. In addition, if a structured variable type is selected, the tab or backtab key can be used to indent or unindent the value to show its relationship to other values in the set. An indentation of a value would indicate it is a child or subvalue of any less indented value.

Backspace

When a new cell is highlighted, the cursor is placed to the right of the last non-blank character in the cell. The backspace key may be used to delete single characters.

Right or Left direction keys

The right or left direction keys may be used to move the cursor from cell to cell to input variable names or values. If more than four variables are used, the right arrow key will shift the display variables to the left by one column.

Page Up or Page Down keys

Ten values (rows) may be displayed at one time. The PGUP or PGDN keys will allow you to display the next or previous ten values. Fifty values per variable are currently permitted.

<Ctrl> Right arrow or <Ctrl> Left arrow keys

The display can be shifted by four variable values either right or left by using the <CTRL> RIGHT arrow or <CTRL> LEFT arrow key.

Tab

Used to indent a value for structured variables. An indentation of a value would indicate that it is the child or subvalue of any less indented value.

Back Tab

Used to unindent a value for structured variable type, principally used to correct mistakenly indented values.

F1 - Insert Column

A blank column will be inserted at the left of currently selected column.

F2 - Insert Value

A blank value will be inserted above the currently selected value.

F3 - Grab Value

F3 will provide options for operations on the currently highlighted cell. After selecting **F3**, you will have the ability to:

- move the value to a new row
- delete the *value* of the grabbed cell
- switch with another *value*
- abandon the operation and return to the example set

F4 - Grab Column

If **F4** is selected, you are given the following options which are displayed at the bottom of the screen:

- insert a copy of the selected column to the left of the current column.
- delete the currently selected *column*
- switch with another *column*
- copy the current column to another *column*
- move the current column to another *column*
- resume the original edit without making any changes.

F5 - TYPE/COST

Toggles the second row to receive either the variable *TYPE* or its *COST* value.

F6 - Jump to var.

Choosing **F6** will jump to the variable name in the top row, remaining in the same column.

F7 - Annotations

If **F7** is selected a number of annotations can be provided by filing in the corresponding areas to answer questions about the display. The position of the currently highlite cell will determine which question is asked.

Decision or selector variable name

If a variable name is highlighted when **F7** is entered, a prompt for a question to be associated with that variable is listed. This question can consist of 1 to 55 characters. In addition, a prompt for the importance of the value, an integer between 1 and 10 can be entered to be used in determining the inferences made on that variable.

Value of a decision variable

If the value for a decision variable is highlight, when F7 is selected, you will be asked to supply the name of a file with its extension to be displayed if the value is highest supported (confirmed) at the conclusion of an advisory session. In addition, the user may also enter the name of an executable file to be run when the value is confirmed.

Value of a selector variable

If a value of any selector variable is highlight when the F7 key is struck, a prompt for a help file of 1 to 8 characters (including the extension) is displayed. Help files may be prepared with any external editor capable of producing an ASCII file.

Esc - Options

If the <CR> key is selected, four additional options are provided.

Save and Resume Edit

All variables and values in the editor may be saved to a file which is associated with the advisory system name or another file may be created to save the current variables.

Resume Edit

You will be returned to the variable editor without saving any variables or their values.

Developing ES

You will be returned to the *Develop or Improve an Advisory System* menu.

Quit

To quit AURORA and return to DOS.

Access Inference Parameters

In order to provide additional control over the advisory system, you may provide values to the inference engine to change its criteria for selecting of rules. This can be done through accessing the inference parameter table by selecting the *access* operation and the *inference parameter* item from the *Develop or Improve an advisory system* menu. In response to this selection, the inference parameter menu (Figure 46) will be displayed. Chapter 10 provides greater detail on the AURORA inference mechanism.

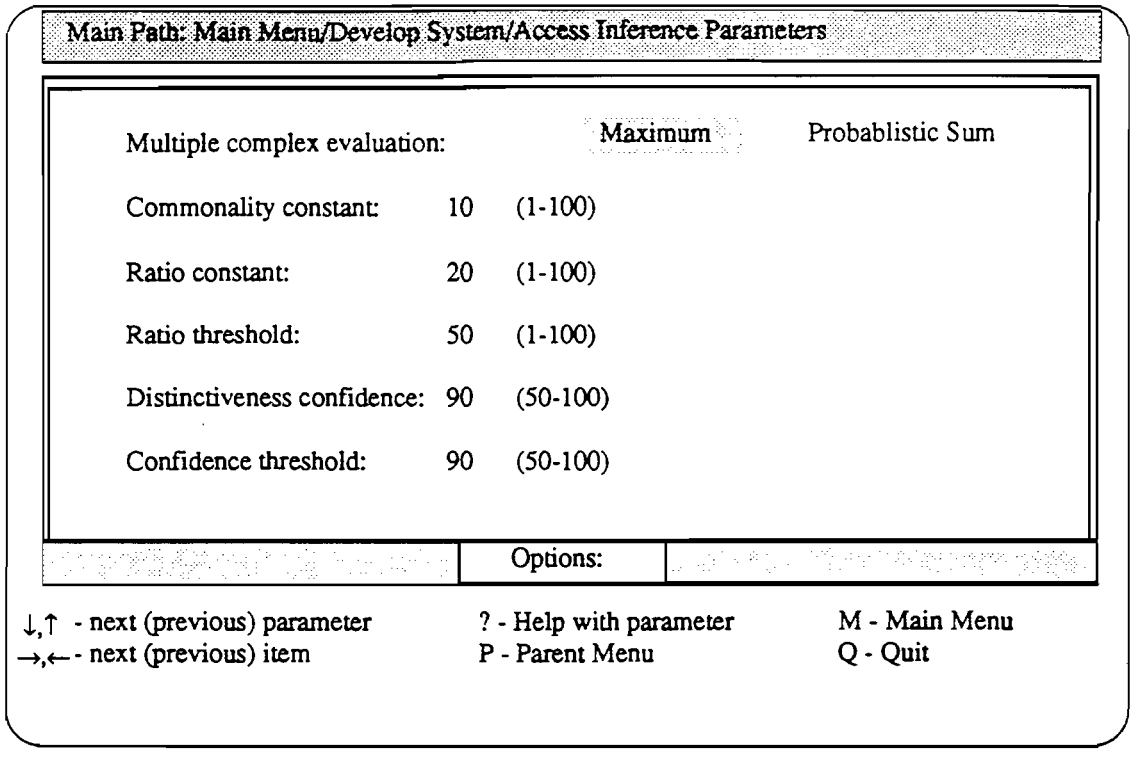


Figure 46. Inference Parameter Screen

Multiple complex evaluation Default: Maximum

Multiple Complex Evaluation - this parameter determines how a degree of match is calculated when a rule consists of several complexes. When maximum is chosen, the degree of match for the rule is the maximum of each of the degrees of match for each complex.

When probabalistic sum is chosen, two degrees of match are combined using the following formula:

$$\text{(Total Match)} = \text{(Degree of Match [complex 1] + Degree of Match [complex 2])} - \text{(Degree of Match [complex 1] * Degree of Match [complex 2])}$$

Commonality constant Default: 10

During the confirmation phase, the degree of match for each complex is updated for each question asked as follows:

$$\text{Degree of Match [current question]} = \frac{\text{Degree of Match [last question]}}{\text{Commonality constant * commonality (feature)}}$$

Ratio constant Default: 20

During the confirmation phase, the relative difference between the candidate hypothesis and the next best hypothesis is an indication of the degree of confidence in the candidate hypothesis. During this phase, the degree of confidence for the candidate is calculated from the degree of match as follows:

$$\text{Degree of Certainty (best)} = \frac{\text{Degree of Match [best hypothesis]} + \text{Ratio Constant} * (\text{Degree of Match [best hypothesis]} - \text{Degree of Match [next best hypothesis]})}{\text{Degree of Match [best hypothesis]}}$$

Ratio threshold Default: 50

The discrimination phase continues until one candidate stands out far above the others, or until all relevant questions are exhausted. The ratio threshold parameter defines how high above the others a candidate must be in order to complete discrimination. Discrimination continues until:

$$\frac{\text{Degree of Match [best hypothesis]} - \text{Degree of Match [next best hypothesis]}}{\text{Degree of Match [best hypothesis]}} > \text{Ratio threshold}$$

Distinctiveness confidence Default: 90

The distinctiveness values supplied by learning may be associated with some error. This error is roughly related to the number of examples supplied. The distinctiveness values will be adjusted according to the level of confidence specified by the user.

Confidence threshold Default: 90

The confirmation phase will continue until all relevant questions have been asked, or until the confidence level for the candidate hypothesis reaches the confidence threshold.

OPTIONS:

Next (Previous) parameter

The Up or Down direction keys may be used to move from one parameter to the next.

Next (Previous) Item

The Right or Left Arrow keys can move within a parameter from one item to another. Only valid for the parameter *Multiple complex evaluation*.

? - Help with parameter

Depressing the ? key will provide information on the currently selected parameter.

P - Parent Menu

Typing a **P** will move you from *Access inference parameter* back to the *Develop or Improve an Advisory System* menu.

M - Main Menu

Typing a **M** will present you with the AURORA Main Menu.

Q - Quit

A typed Q will quit AURORA and present you with the DOS prompt.

Access or Create Learning Parameters

When either operation, *Access* or *Create*, is selected and the item, Learning parameters, is selected, the learning parameters menu is displayed (Figure 47) or a warning message (Figure 34) is displayed. This menu provides you with an ability to set the value of parameters controlling the learning procedure. The various parameters in the menu can be accessed by using the directional keys. The default level is the value listed first for each parameter.

Description of Learning Parameters

The first parameter which may be selected is in the first row below the menu path on the top line. The space bar or right and left arrow keys may be used to move between items within each parameter. In order to select a new parameter, the up and down arrow keys should be used.

Upon leaving the learning parameters menu, all the current settings will be saved to be used in future learning in the current advisory system.

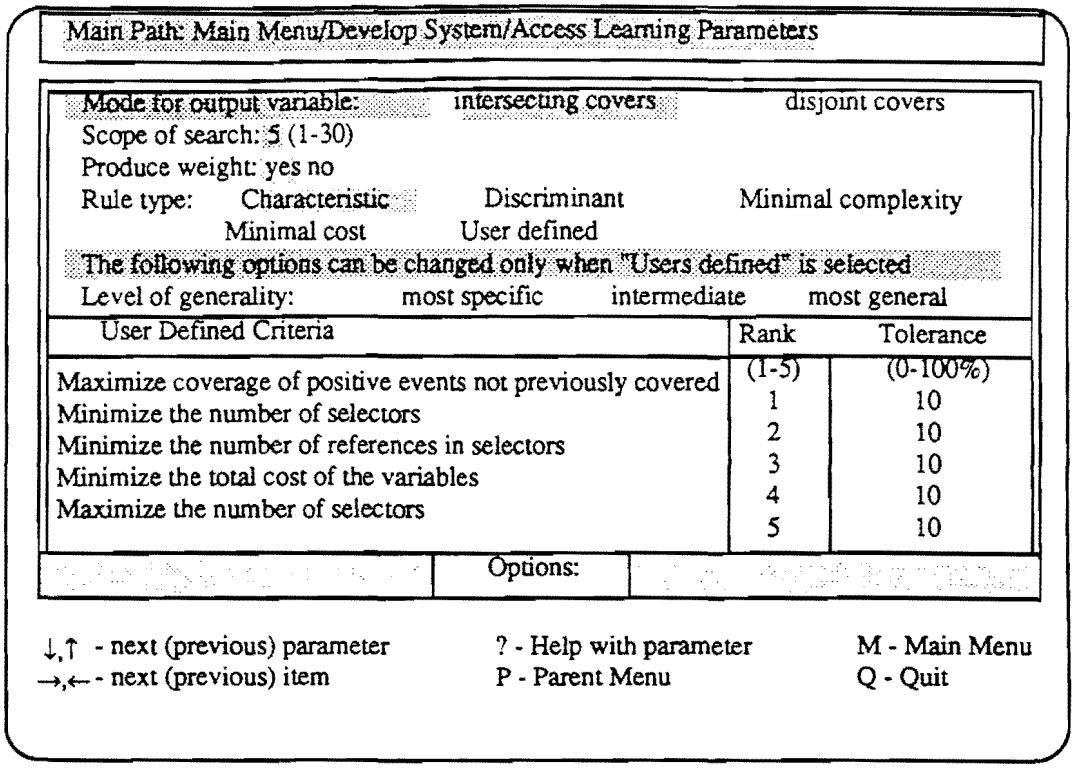


Figure 47. Learning Parameters Screen

Mode for Nominal Output Variable Default: intersecting covers

This parameter controls whether intersecting covers or disjunctive covers will be used to produce rules. If the intersecting covers mode is selected, then rules may overlap in a space where there are no learning events. With disjunctive covers, no two rules will cover any one possible example. For more detail on "Learning by Example" see Chapter 9.

Scope of the search Default: 5

This parameter controls the extent of the search for additional positive covers. Raising the value of this parameter may increase the quality of the generated rules, however, it will also increase the time required to complete the learning process.

Produce weights Default: yes

You may select to produce and record weights by indicating yes. This parameter specifies whether the weights or confidence values associated with each selector will be saved in the created rules. If it is set to no, weights will be displayed during the learning process but will not be permanently stored with the

rule base. Any weights produced during learning may be modified or deleted in an external editor after leaving the learning environment. Weights produced during learning are for your information only and are not used in the inference system.

Rule type Default: Characteristic

This parameter is, perhaps, the most important control for determining the nature of the produced rules. Four basic types of rules may be produced. The names listed indicate the level of generality of produced rules. When the user moves from one rule type to another, the level of generality and criterion listed below the rule types will also change.

Characteristic rules

Rules which describe most or all of the known features (variables or selectors) of a class of objects. They are the most specific rules, relating closed to the examples, and usually contain more information than is needed to discriminate between two or more classes. They typically include a maximum number of selectors with a minimum number of corresponding values.

Discriminate rules

Discriminate rules, on the other hand, contain only those variables necessary to discriminate one class from another. Thus, they are the most general descriptions of a set of learning events. They typically include a minimum number of variables but with a large set of values.

Minimal complexity

Minimal complexity rules are in between characteristic and discriminate rule types. They are never the most general nor the most specific rule that can be produced. They may be the simplest rule type with a minimal number of both variables and their values.

Minimal Cost

Rules favor variables that are associated with the lowest cost (Setting of the cost of a variable is described on page 62). Criteria for minimizing the cost of the variable, is ranked the highest when working with this type of rule.

User Defined

May be selected to allow you to customize the type of rule to be produced by allowing access to the *user define criteria* section. The level of generality ranges from most specific to most general. There are five criteria that may be ranked. They are ranked by assigning a number from 1 to 5 for each of the criteria. Ties in rank are permitted. You may optionally specify a tolerance for each criterion; this is a number between 0 and 100% which implies the uncertainty you may have in applying the criterion. Specifying a high tolerance will soften the distinction between complexes of roughly similar values as evaluated by the given criterion. Rules are first compared with respect to the criterion ranked #1. If the results are within the tolerance, they are considered to be tied. If there is a tie for the best rule, the second criterion is evaluated, etc. until one rule is determined to be better than the others.

Options:

The directional keys will allow you to move either between items for a given parameter or between different parameters.

? - Help with parameter

If help is needed with a parameter, a ? can be struck and help information will be provided.

P - Parent menu

Will bring up the *Develop or Improve an Advisory System* menu.

M - Main menu

Will take you to the AURORA main menu.

Q - Quit

To quit AURORA all together and return to DOS.

Access or Create System Description

In order to create a new advisory system, *new system* must be selected in the left hand column on the *Develop or Improve an Advisory system* menu (see page 35). There might be other times, however, when you may wish to change the name, description, or file containing a long description of any current advisory system. This can be accomplished by selecting the advisory system name, then selecting the *access* operation and then finally selecting

System Description for the item. Figure 32, will appear (see page 41). This is the same menu that appears during the creation of a new system. Moving the highlight section by use of the direction keys will allow the user to then change any settings for either the domain, the advisory system name, the short system description, or the file name of the long description.

After the satisfactory revision of the menu, typing an Esc will bring up a additional options menu, which will allow the user to save the changes and return to the *Develop or Improve an Advisory system* menu, ignore the changes and return to the *Develop or Improve an Advisory system* menu, or resume entering or changing information.

Copy or Delete

The final two operations available from the develop or improve advisory system menu are copy and delete.

Copy

If the copy operation is selected with any item, the following line appears in the bottom options window:

"Copy <item> from <system> to: _____"
(Enter an Expert System Name or Type Esc to return to Developing ES)

This allow you to input the name of a system to receive the copied information. You can either enter a system and then a <CR> to complete the operation or type Esc to ignore the operation and return to the *Develop or Advisory an Advisory System* menu.

Delete

If delete is chosen, the following message will appear in the options window at the bottom of the screen:

Are you sure you want to erase the <item> in <system> (y\n)?

Or in the case of deleting the whole system:

Are you sure you want to erase the entire <system> expert system(y\n)?

CHAPTER SEVEN
LEARNING AND DISCOVERY MODULE

If the *Enter Learning and Discovery* module option is chosen from the Main menu, the "learning and discovery" submenu (Figure 48), will be presented to you. This menu presents a list of operations available for building and improving rules in an advisory system.

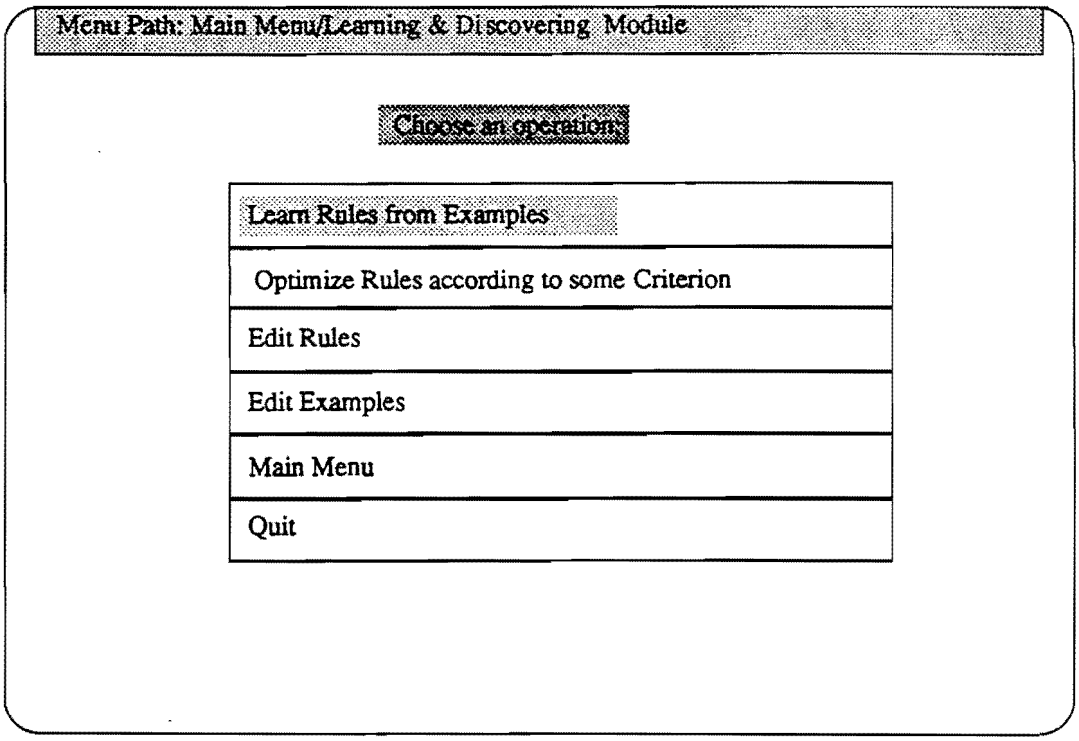


Figure 48. Learning and Discovery Module Menu

All options on the learning and discovery module menu operate as explained in earlier chapters. The main exception for any operation within this menu is that you must supply the name of a previously defined advisory system.

Note that in this module, AURORA goes to any option chosen without a system name being specified. You enter a name in the interactions that follow. After choosing an option, a list of the currently available

advisory systems will be written to the screen in a separate window. Choose the desired advisory system by moving the highlight area with the directional keys. Then enter a <CR>.

Learn Rules from Examples

Presents you with the *create rules from example* menu as explained in chapter 9. Any newly created rules will replace all existing rules for the knowledge base.

Optimize Rules According to Some Criterion

Presents the *Optimize Rules According to Some Criterion* menu described on page 52. You must, however, input a valid advisory system name when prompted.

Edit Rules

Presents the selected external editor with a blank screen.

Edit Examples

Presents the selected example editor (page 55) with a blank screen.

Main Menu

Returns user to the Main Menu (page 7).

Quit

Returns user to DOS.

CHAPTER EIGHT
REVIEW AN ADVISORY SYSTEM

Often it is helpful to look at or examine portions of an existing advisory system without the possibility of modifying or changing it. This can provide a quick method to see which system contains the knowledge you might be interested in. This operation can be conducted by choosing the Review an Advisory System selection from the AURORA main menu.

After the choosing the review an advisory system selection, the review an advisory system (figure 49) menu will appear. This menu looks quite similar to the *Develop or Improve an Advisory System* menu. For the most part, both the left and right hand columns on the menu are exactly as presented in the *Develop or Improve an Advisory System* menu presenting to you advisory systems which have been previously installed and letting you select any item as in the *Develop or Improve an Advisory System* menu. The *whole system* and *<new system>* items are exceptions.

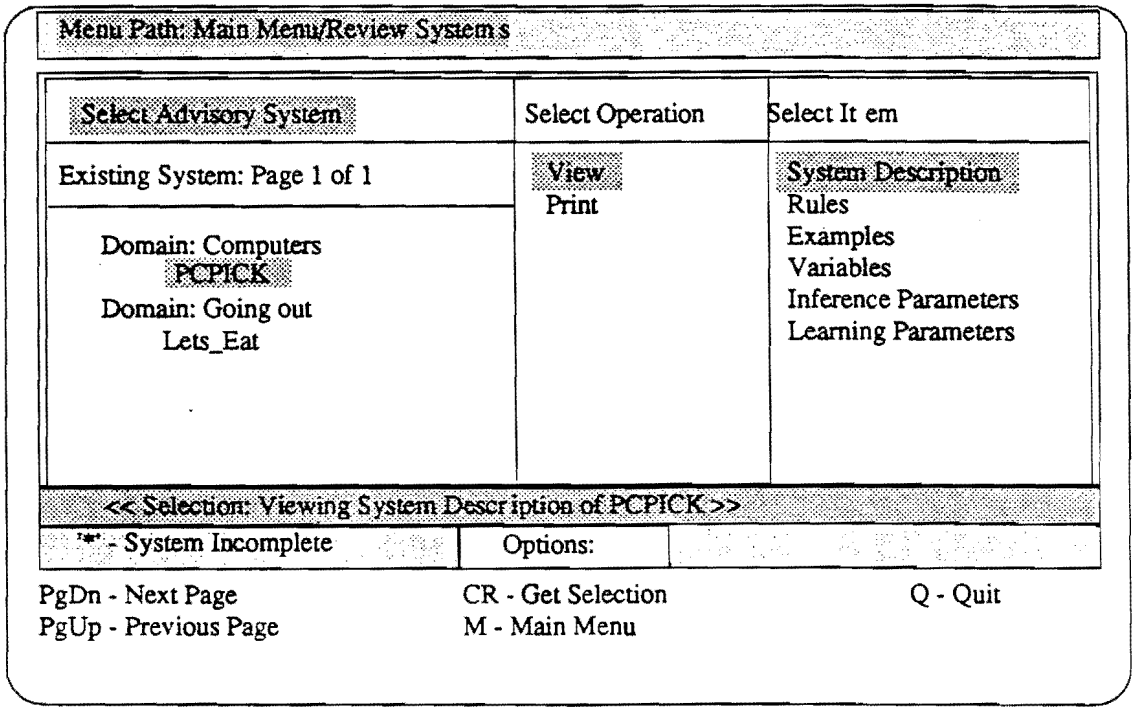


Figure 49. Review an Advisory System menu

The real difference in the two menus is in the scope of operations. The review menu was designed to allow you to only view or print any selected item from a advisory system. Modifications are not allowed from this menu. Selecting view will bring the selected item to the screen. Selecting print will print the same information on the default printer. A new advisory system cannot be developed from this menu.

CHAPTER NINE
LEARNING BY EXAMPLE A HELPFUL TOOL

One of the more powerful tools available in Aurora is the learning module. This module incorporates the AQ algorithm in a recent modification GEM (Generalization of Example by Machine). Machine learning is based on inductive inference in contrast to the more common place deductive process found in other expert system shells. Inductive inference is a process of preventing falsity in a set of provided examples. Deductive inference, on the other hand, is truth preserving. More specifically, learning through GEM can be considered learning from example. In general, examples of expert level decisions are drawn from the domain and provided to the system. The examples must have some structure to provide the basis for the incremental generalization of their components to provide the most concise representation or rule. In GEM each example must be represented as a collection of evidence (VariableValue pairs) which supports a single result or conclusion. This conclusion represents one value in a set of values for the goal of the learning process. The goal is known as the decision variable with each of its values being represent in the form of a rule. For each example, in addition to the value stated for the decision variable, you may list a value for each variable designated as a selector variable. Selector variables provide the evidence for each example. Examples which have the same value for the decision variable are said to be in the same "class". In simplified form, the AQ algorithm can be described as follows:

Description of AQ

The set of all examples is also referred to as the learning space. A seed or single example (in the case of Aurora, one row from the example editor) is selected from the learning space. All members in the class of this seed example are considered to be positive examples of the class concept, while members of all other classes are considered to be negative examples. A list of all the sets of variablevalue pairs found in the positive examples and not found in any negative example is developed and is called a partial star. The partial star is then generalized to form a rule for the class according to the value of parameters set in the "learning parameters" table. This process is repeated for each of the remaining classes. The following is a description of each parameter and how it would affect rules produced through learning:

Mode for nominal output variable

The intersecting covers mode produces rules that may intersect over areas of the learning space where there are no

learning examples. An example of this would be a value provided in the variable table, but which is not represented by an example in the example table. Selection of intersecting covers will allow rules to be produced with complexes that cover the same variables for a given class. Selecting disjunct covers will ensure that all complexes in a rule use exclusive variables.

Max Star (scope of search)

This parameter represents the maximum number of stars that may be produced in the learning procedure. Potentially setting the value higher may produce better solutions, however, larger resources are needed. This improvement in quality will only increase until the value of max star is equal to the number of selector variables. Due to the memory limitations of the DOS operating system, it is generally useful to start at the default level and increment the scope of the search value only when required.

Produce Weights

Weights are always produced in the learning algorithm. The selection Yes/No simply indicates whether weights will be added to the final output set. The weight associated with each generated selector in a complex is determined by:

$$\text{weight} = \frac{pe}{(pe + ne)}$$

where pe is the number of positive examples covered, and ne is the number of negative examples covered by the selector. Thus, the weight produced represents the probability that the given decision class is indicated given that the selector is satisfied.

Rule Type

Rule Type is perhaps the most important feature of the "learning parameters" table. While in learning the partial star is extended against all negative examples and variables from all positive examples are retained. The final rule, however, will not contain all of these variables. Upon completion of a star a lexicongraphicical evaluation functional (LEF) is used to trim the star to provide a rule with predetermined criteria.

If *characteristic* is selected, rules are as specific as possible, that is, they include maximum number of selector variables, each with a minimum number of values. If the *discriminate* selection is made, induced rules are as general as possible, that is, they involve the minimum number of selector variables, each with a maximum number of values. If *minimal complexity* is chosen, rules are as simple as possible, with the minimum number of selector

variables and each with a minimum number of values. Redundant values are removed from selector variables sequentially for each complex after all complexes for a rule are sorted in descending order of the total positive examples covered. A value is removed if it does not appear in any example covered by the current complex. Variables that appear in the rule are not changed. *Minimal cost* may be selected if cost values had been added to selectively weight the variables.

User Define

Selection of *User Define* will provide access to a lower portion of the "learning parameter table". User Define criteria may be ranked and the tolerance may be added to provide an indication of the uncertainty associated with the ranking. While changes in the ranking of these defined criteria can have dramatic affect on rules produced, there is no universal strategy for their ranking. If one of the particular criteria is of high importance in the resulting rules, it should be ranked as one of the initial or first selection. Those criteria that are not as critical should receive a lower ranking. Rules are first compared with respect to the criterion ranked #1. If the results are within the tolerance, they are considered to be tied. If there is a tie for the best rule, the second criterion is evaluated, etc. until one rule is determined to be better than the others.

CHAPTER TEN
AURORA INFERENCE ENGINE

Inference Algorithm

This chapter contains a brief outline of the inference algorithm used for the advisory session. The advisory module, conducts a question and answer session with you, giving advice based on the rule base. The inference algorithm attempts to ask as few questions as possible in order to find the appropriate advice. The process consists of three phases: reduction, discrimination, and confirmation.

Reduction

During the reduction phase, all primary variables are queried for. Primary variables are those with importance 10 as defined by you in the variable editor by selecting F7 - annotations. All rule complexes that are inconsistent with the information received during this phase are removed from consideration. If a complex does not contain one of the primary selectors, it is considered consistent as long as the primary selector variables that are present match the answers given.

Discrimination

The purpose of discrimination is to ask relevant questions until one hypothesis stands out as the most likely candidate. First, some definitions.

Distinctiveness:

You may assign each selector variablevalue(s) pair a distinctiveness value. If a distinctiveness value is not assigned, the compiler will calculate one from the rules. The compiler calculates a distinctiveness for each value of the selector, rather than for the selector as a whole:

$$\text{Distinctiveness value} = \frac{(N/n) - 1}{N-1}$$

where N = total number of rules and n = the number of rules which contain the value.

Selector variablevalue(s) pair utility:

The utility of a selector is a measure of the information gained, if the given answer should match the selector. The utility of a selector is the average distinctiveness of all of the values in the selector.

Variable utility:

The variable utility is obtained by first summing the selector utilities for each appearance of the variable. Each of these sums is then normalized by dividing by the maximum sum. The variable utility is calculated within the context of the complexes that are left at the end of reduction. The variable utility is calculated only once at the beginning of the discrimination to determine the order in which questions will be asked.

During discrimination, variables are selected from the set of non-primary variables on the basis of the sum of the variable utility and the variable importance (a value of 1 to 9 assigned by pressing F7 in the variable editor). These measures are given equal weight, however, if the utility is 0, the question will not be asked regardless of the assigned importance, since the variable is not relevant to any of the rules that are still under consideration. In the case of ties, the variable that appears first in the table will be asked first.

After each question is answered, the degree of match (DM) and degree of certainty (DC) is calculated from each rule as follows:

If the answer matches a selector in the complex then:

DM(complex) = previous DM(complex) + distinctiveness - previous DM(complex) * distinctiveness

If the answer does not match the selector:

DM(complex) = previous DM(complex)

DM(rule) = probabalistic sum of DM of the complexes or maximum of DM of the complexes

DC(rule) = DM(rule) * distinctiveness certainty

The calculation of DM(rule) uses probabalistic sum or maximum, depending on a parameter set by you. The distinctiveness certainty is also determined by you in the inference parameters screen (Figure 46). This parameter represents your confidence in the distinctiveness values specified. When rules are learned by example, this confidence should be proportional to the number of examples that have been provided.

The discrimination phase continues until one of the following conditions has been met:

1. There are no more relevant questions to ask.
2. The best candidates ratio is greater than the ratio threshold set by the user.
   The best candidates ratio is defined as:

$$\frac{\text{DC (best hypothesis)} - \text{DC (next best hypothesis)}}{\text{DC (best hypothesis)}}$$

Confirmation

When the ratio threshold is exceeded, control switches to the confirmation phase. The relevant variables not previously asked for which are mentioned in the candidate hypothesis are now asked in order of their utilities. This phase continues until the questions are exhausted or the confidence in the candidate exceeds the confidence threshold set by you in the inference parameters screen (Figure 46). If one of the other hypotheses becomes greater than the candidate, it becomes the new candidate. As each question is asked, degree of match and degree of certainty are updated as follows:

DM (complex) = previous DM (complex) commonality constant * commonality (feature)

DM (rule) = probabalistic sum of DM of the complexes or maximum of DM of the complexes

For the candidate hypothesis:

DC = DM * distinctiveness confidence + ratio constant * best candidates ratio

For all others:

DC = DM * distinctiveness confidence

Notice that the degree of confidence in the highest rule is dependent upon the distance of that rule from its next closest competitor (best candidates ratio). This reflects the idea that if only one candidate stands out distinctly from others, it should be given high confidence even if the candidate does not represent an exact match. You determine the numerical significance assigned to the best candidates ratio by assigning a value between 0 and 99 to the ratio constant parameter in the inference parameters screen (Figure 46)

Commonality:

When rules are learned from examples, the commonalty of a selector value can be calculated as follows:

$$\frac{\text{number of examples containing the value for the given class}}{\text{total number of examples in that class}}$$

Using a similar algorithm, you may assign commonalties for hand-written rules. The commonality can also be calculated directly from the rules, each complex is considered as one example of the concept. This gives the formula:

$$\frac{\text{number of complexes containing the value for the given rule}}{\text{total number of complexes in the rule}}$$

Backtracking

The advisory session has the ability to handle background rules, i.e. rules that trigger variables other than the decision variable. The decision variable is focussed on first. If the user answers "don't know" to one of the questions, and this variable appears as an action in another rule, the system will backtrack to that rule. All questions appropriate to the background rule will be asked, if they were not asked previously.

Constructing Rules by Hand in an Editor

Figure 50 below contains the complete grammar for rules in the system. Summarizing this table, we see that a rule consists of a condition part and an action part, or part to be executed if the condition holds. A condition consists of the disjunction of a set of complexes, which in turn consist of the conjunction of selectors. A selector consists of a variable, a relation, and either a value, a disjunction of values, or a range of values, plus an distinctiveness weight and a commonalty weight.

<rule>	:=	<action> <condition>
<action>	:=	<variable> <relation> <value> "if:"
<condition>	:=	<complex> "OR" <complex>
	:	<complex>
<complex>	:=	<selector> <selector> <selector> <selector> •••
	:	<selector>
<selector>	:=	<variable> <relation> <value-list> <terminator>
	:	<a-expression> <a-relation> <a-expression> <terminator>
<variable>	:=	string of letters
<relation>	:=	string of letters
<value-list>	:=	<value> "or" <value-list>
	:	<value> "to" <value>
	:	string of letters
<terminator>	:=	<termchar>
	:	<termchar> <distinctiveness weight> <commanality weight>
<termchar>	:=	"."
	:	","
<distinctiveness weight>	:=	integer 0-99
<commanality weight>	:=	integer 0-99
<value>	:=	string of letters

Figure 50. Grammar for Rules

Spacing is not critical. You may indent parts of a rule to make it more readable. Figure 51 shows a typical hand built rule. Each selector, except for the last one in the last complex of a rule, must end with a "," after the last value. You may add only a distinctiveness weight or both distinctiveness and commonality weights, but not just a commonality weight. Any number of blank lines may appear between rules. Finally, the file must be saved as a ASCII only file with the name of the advisory system and the extension ".rle".

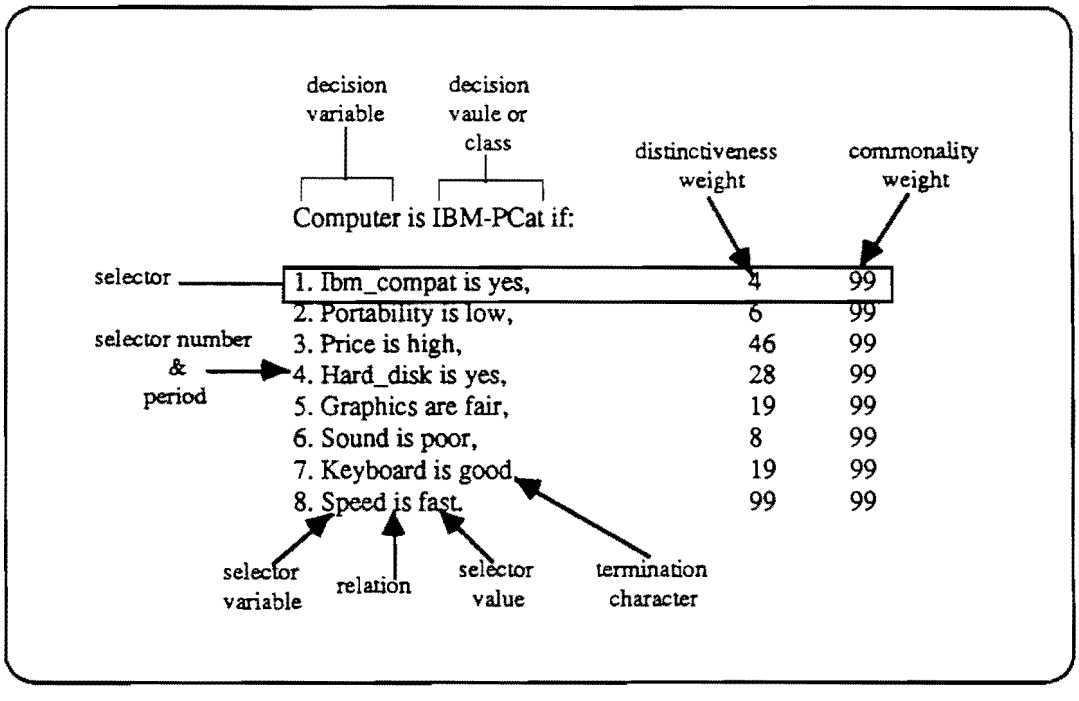


Figure 51. Typical rule constructed in an editor

Rule Compilation

Once acquired or constructed, rules must be compiled if they are to be used in an advisory capacity. Essentially, the compiler has two chores. One is to check the syntax of the rules and provide you with an appropriate error messages if the rules do not compile. (A complete listing of error messages is provide below). If the rules compile successfully, the compiler will then create an file suitable for asking questions and giving advice.

Compiler Error Messages

Compiler error messages are divided into those that are fatal, which prevent a working advisory system, and

warnings, which inform the user of certain problems during the compilation, but will result in a new system.

Fatal Errors

- "IF" not found. Compilation aborted.

  The word "IF" is not in first line of the rule.

- Number not found in above selector. Compilation aborted.

  A number was not found before the selector.

- "." not found after number. Compilation aborted.

  A period was not found after the number for a selector.

- Variable not found in above selector. Compilation aborted.

  A variable was not found (in the variable editor) after the number at the beginning of a selector.

- Relation not found in above selector. Compilation aborted.

  A relation ("is")was not found following the variable.

- Value not found in above selector. Compilation aborted.

  A value was not found following the relation or a comma or period is before the value.

- "or" or "to" not found between values. Compilation aborted.

  All values must be in a disjunctive set or in a linear range. "or" or "to" are only valid connectors

- Final Value missing. Compilation aborted.

  A value was expected after "to".

- "," or "." not found. Compilation aborted.

  Each selector must be followed by a comma or a period. A period is only used after the last selector of the last complex in a rule.

• Rule name not found. Compilation aborted.

A value for the decision variable was not found in the first line.

• "to" construction used with non-linear variable. Please declare <variable> as a linear variable.

The variable was not declared as linear, thus it cannot be associated with a range.

• Linear value <value> was not found in domain.

All linear values must be predeclared in order that the compiler can construct the correct range.

• Structured value <value> was not found in domain.

All Structured values must be predeclared in order that the compiler can construct the correct relationship for all values of that variable.

Warnings

• Variable too long. Truncating to 12 characters.

Variable names are limited to 12 characters. It is truncated if too long.

• Relation too long. Truncating to 12 characters.

A relation (usually "is") must be 12 or less characters.

• Value too long. Truncating to 12 characters.

A value must be 12 or less characters.

• <variable> not declared. Assuming nominal type.

The variable was not found in the variable editor. Its type is assumed to be nominal.

APPENDIX A
GLOSSARY

Backtracking:	Ability to recursivly search through rules for values of a decision variable.
Class:	A group of examples with the same value for the decision variable.
Commonality:	A measure of the degree to which a particular selector is associated with one particular class. This value, in the range 1 to 100, may be assigned by the developer, or calculated from learning examples.
Complex:	A group of variables and their values connected by the conjunctive operator "AND". A rule can consist of one or more complexes and one corresponding decision variable value.
Condition variable:	A variable whose value in whole or part determines the validity of a decision variable.
Confirmation:	The final process of inference to determine whether the indicated rule fits the required goal.
Cover:	A logical expression which matches all variable values for a given set of examples.
Decision variable:	A variable whose satisfaction presents the user with a value for the goal of the session.
Discrimination:	A middle or secondary phase of inference which ultimately selects one candidate rule for confirmation.
Distinctiveness:	A measure of the degree to which a particular selector is able to distinguish one class from all other classes. This value, in the range 1 to 100, may be assigned by the developer or calculated automatically from the rules during compilation.
Domain:	A limited area of knowledge or expertise in which experts provide advice.

Event:	A single example of an expert decision consisting of one decision variable value and any number of selector variables to describe the values present at the time the event was developed.
General rule:	A rule with only enough condition variables to distinguish it from all other rules in the knowledge base. This type of rule can be applied to the most general type of situation.
Improving Rules by Examples:	A learning process which incorporates the knowledge of new examples into previously developed rules.
Inference engine:	A portion of Aurora which attempts to determine which part of the knowledge base is appropriate to satisfy a desired goal.
Inference parameters:	Values set in the inference parameter screen which provide control during an advisory session.
Inheritance:	The ability to provide similar values and domain parts (variables) to a new advisory system from a previous one in the same domain.
Integer variable:	A variable whose set of values are whole numbers between 0 and 50.
Intermediate rule:	A rule with the number of conditioned variables lying somewhere between the number found in specific rules and general rules, having some degree of generality.
Learning parameter:	A variable whose current value is utilized in determining the extent and breadth of a learning session.
Learning:	In Aurora, implemented through the AQ algorithm of learning by example of previous expert decision.
Level of Generality:	The level of, or the number of variables and values available in a rule.
Linear variable:	A variable whose set of values can be ordered in a single relationship from a lowest point to a highest point. For example, a variable, computer speed, with values of low, medium and high.

Negative cover:	Values of a selector variable found in a class other than the one under consideration.
Nominal variables:	A variable whose set of values has no special relationship to each other. It can be a set of names. For example, a set of values of car color, such as red, blue, yellow so on.
Numeric variables:	A variable whose set of values are real numbers. Only numeric variables can be used in arithmetic expressions.
Optimization of Rules:	Converting specific rules into more generalized rules.
Parsing:	The conversion of Aurora rules in their English-like syntax to a compiled format which is utilized by the inference system.
reduction:	The first stage of the inference mechanism which tries to minimize the number of rules to be searched for the answer or goal.
rule:	A complex statement consisting of a condition component and a decision or conclusion portion. If the condition component is satisfied, then the decision section is passed on to another rule or the user. There may be more than one condition component with the satisfaction of any one satisfying the rule.
Specific rule:	A rule in which multiple condition variables are used to describe a unique occurrence.
Structured variable:	A variable whose set of values has some relationship to other values in a hierarchical structure (Figure 52.). For example, a variable of computer may have a set of values of supercomputer, mainframe computer, minicomputer and microcomputer. Under microcomputer there may be some values like IBM-PC, Macintosh so on.
Tolerance:	The degree of uncertainty in applying a particular learning criterion. Rules are first compared with respect to the criterion ranked #1. If the results are within the tolerance, they are considered to be tied. If there is a tie for the best rule, the second criterion is evaluated, etc. until one rule is determined to be better than the others.

APPENDIX B
DESCRIPTION OF VARIABLES AND THEIR VALUE TYPES

Variable Types

Variables in the AURORA fall into four categories: nominal, linear, integer and structured. The type of the variable and its domain are defined in the Variable editor (see section Chapter 6). Below each variable type is described in more detail.

Nominal variables

Nominal variables have values which do not have any explicit structure. Examples of this type include boolean variables which may take the values of true or false, a variable such as color, which may take the values red, green, blue, etc., and all variables the values for which cannot be ordered in any reasonable fashion.

Linear variables

Linear variables are those with values that can be ordered in some fashion. Examples of this type are variables such as size, taking on the values of small, medium, and large, and age, which could consist of the ordered domain infant, child, adolescent, young-adult, middle-age, and old-age. Linear variables can appear in rules with the "to" separator. For example, a condition in a rule might be that "Age is child to young-adult". The system understands this to include the value "adolescent"

Integer variables

Integer variables, pre-defined as a convenience, are linear variables with a range that extends from the integer 0 to the integer 50. Thus, a valid condition involving an integer variable might be, "No of flowers is 29 to 33".

Structured variables

Structured variables have values that be ordered in a hierarchical manner. An example of the structure of the variable shape appears below.

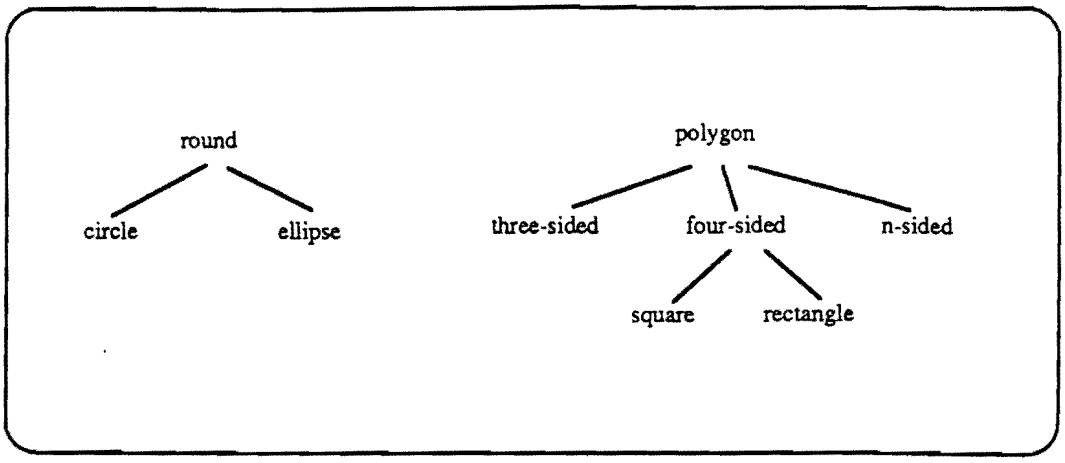


Figure 52. Structure for the variable "shape"

The rule learning module will attempt to climb the tree in producing descriptive rules. For example, in the above figure, if two examples for one decision class have the values for shape of square and rectangle, and these values did not appear in any other decision class, then the condition produced will not be "Shape is square or rectangle", but "Shape is four-sided". Additionally, the advisory system makes use of the variable structure in its attempt to satisfy rule conditions. For example, if one answers the question "What shape does the object have" with the value "ellipse", then the system understands that "Shape is round"is also satisfied.