

LPL Software Manual

LPL Software Manual

Dave Barker-Plummer, Jon Barwise, John Etchemendy
Albert Liu, Michael Murray, Emma Pease

Copyright ©1999, 2000, 2005, 2007, 2008, 2009, 2011, 2013, 2015,
2017

CSLI Publications
Center for the Study of Language and Information
Stanford University

Contents

1	Introduction	1
1.1	System requirements	2
1.2	Obtaining the applications	2
1.3	Installing the applications	3
1.4	Getting Help	4
2	Using Submit	5
2.1	Getting started	5
2.2	Choosing files to submit	7
2.3	How you know your files were received	9
2.4	Preferences and User Data	10
2.4.1	Proxy Server Settings	11
2.4.2	Preferred Host Settings	11
2.4.3	Timeout settings	12
2.4.4	Personality	12
2.4.5	Check for Updates	12
2.4.6	Use Native File Chooser	12
3	Using Tarski's World	13
3.1	Getting started	13
3.1.1	Opening saved files	14
3.1.2	Starting new files	15
3.1.3	Saving a file	15
3.1.4	Closing Tabs	16
3.1.5	Reverting a File	16
3.1.6	Printing	17
3.1.7	Quitting (Exiting) Tarski's World	17
3.2	The World Panel	17
3.2.1	Adding blocks	17

3.2.2	Selecting blocks	17
3.2.3	Moving blocks	18
3.2.4	Sizing and shaping blocks	18
3.2.5	Naming blocks	18
3.2.6	Deleting blocks	19
3.2.7	Cutting, copying, and pasting blocks	19
3.2.8	Hiding labels	19
3.2.9	2-D view	19
3.2.10	Rotating Worlds	20
3.3	The Sentence Panel	20
3.3.1	Writing formulas	20
3.3.2	Commenting your sentences	21
3.3.3	Creating a list of sentences	21
3.3.4	Moving from sentence to sentence	21
3.3.5	Deleting sentences	22
3.3.6	Typing symbols from the keyboard	22
3.3.7	Cutting, copying, and pasting	23
3.4	Verifying syntax and truth	23
3.5	Playing the game	24
3.5.1	Picking blocks and sentences	25
3.5.2	Backing up and giving up	25
3.5.3	Controlling the interaction in the game	26
3.5.4	When to play the game	26
3.6	Preferences	27
4	Using Fitch	29
4.1	Getting started	29
4.1.1	The menus	29
4.1.2	The proof toolbar	31
4.1.3	The FOL toolbar	31
4.1.4	The proof window	31
4.2	Creating and editing proofs	32
4.2.1	Step numbers	33
4.2.2	The current focus	33
4.2.3	Moving the focus	33
4.2.4	Adding new steps	34
4.2.5	Entering sentences into steps	34
4.2.6	Deleting steps	34
4.2.7	Specifying a step's rule	35
4.2.8	The Lemma Rule	36

4.2.9	Changing a step's rule	37
4.2.10	Specifying a step's supports	37
4.2.11	Checking steps and verifying proofs	37
4.2.12	Rule Defaults	38
4.2.13	Add Support Steps	38
4.2.14	Starting and ending subproofs	38
4.2.15	Boxed constants in subproofs	39
4.2.16	Collapsing subproofs	39
4.2.17	Deleting subproofs	39
4.3	Goals	40
4.4	Copying and pasting	40
4.4.1	Copying and pasting sentences	40
4.4.2	Copying and pasting goal sentences	41
4.4.3	Copying and pasting ranges of steps	41
4.5	Printing proofs	42
4.6	Setting up exercises	42
4.6.1	Adding and deleting premises	43
4.6.2	Adding and deleting goals	43
4.6.3	Saving new problems	44
4.7	Preferences	44
5	Using Boole	46
5.1	Getting started	46
5.1.1	The menus	46
5.1.2	The toolbars	47
5.1.3	The assessment panel	48
5.1.4	The table panel	48
5.1.5	Adjusting the table panel	49
5.2	Writing and editing tables	49
5.2.1	Entering target sentences	49
5.2.2	Creating reference columns	50
5.2.3	Filling in truth values	50
5.2.4	Using the Column Ordering Panel	51
5.3	Specifying your assessment	52
5.4	Verifying your table	52
5.5	Saving or printing your table	53
5.6	Preferences	53
6	Languages	55
6.1	Blocks	55

6.2	Pets	56
6.3	Sets	57
6.4	Arith	57
Index		58

Chapter 1

Introduction

This manual contains instructions for the applications Submit, Tarski's World, Fitch, and Boole. These four applications are designed to be used with the textbook *Language, Proof and Logic*. We refer to these applications collectively as “the LPL software”. This manual simply tells you how to use the software, not about the logical concepts behind the software. For that you need to read, and complete exercises from, the text.

This manual assumes that you know the basics about using your computer, things like plugging it in, starting applications, opening and saving files, using menus, basic editing, muttering quietly when things go wrong, and quitting. If you don't, you should ask a friend, consultant, your instructor or a younger sibling to get you started. We will focus on what is special about the packaged applications.

Tarski's World, Fitch, and Boole are used for completing exercises interspersed through the text. Submit, by contrast, is used for submitting the exercise files thus completed to an Internet-based grading service called the Grade Grinder. The Grade Grinder receives the files sent by Submit, and then returns an assessment of those files to you (and to your instructor, if you so choose). To submit exercises, you will need the unique Book ID that comes with the package and your computer must be able to access the Internet. (If you can browse the Web from your computer, then you will also be able to submit files.)

This manual should be used as a reference when you are working through the textbook. When you start using one of the applications, you should read through enough of the corresponding

chapter of the manual to get started. Then, if you have trouble with some feature, or can't figure out how to do something, you should try to look it up in the manual.

1.1 System requirements

All four of the applications run on

- Macintosh computers running Mac OS 10.7 or later,
- computers equipped with Microsoft Windows 7, Windows 8, Windows 10 and,
- 64 bit versions of the most recent long term support releases of Fedora and Ubuntu Linuxes.¹

Consult the **ReadMe** file in the installation if you have problems getting the applications to run.

To submit files to the Grade Grinder using the Submit application, your computer must have access to the Internet. If you are using a computer at a college or university, the computer is probably connected to the campus network and thus to the Internet. If you are using a computer at home, you will have to connect first to your Internet Service Provider (ISP). The rule of thumb is that if your browser can access the World Wide Web, then Submit should be able to find the Grade Grinder.

1.2 Obtaining the applications

You will obtain the software package, including the text book and this manual, by downloading an installer from our web site, and running the installer to install the software.

- If you purchased the paperless package from our online store, an account is automatically created for you at our web site <https://www.gradegrinder.net>.

¹Older versions of our software have, and current software may still, run on Windows back to 98 and MacOS back to 8.5 but we no longer support these configurations. The software may run on other flavors of Linux, but again we do not support these configurations. You can obtain these older versions from our web site.

- If you purchased a physical text book, you can register and obtain the software using information provided with your text book. First locate the pouch attached to the inside back cover of this book. Once this pouch is detached from the cover or opened, you will be unable to return the book. Open the pouch and turn over the card to reveal your registration information.

On the card you will see the URL of the registration page on our web site, a registration id beginning with the letter L, and also a Quick Response (QR) code. If you have a QR code reader (software that is available, free for most computers, tablets and phones) you can simply scan this code. Alternatively, you can enter the URL manually into your web browser. In either case, you will see a form which requires you to enter your name, email address and registration code, which you should fill out to complete your registration.

Once you are registered, you can log into this account to download an installer appropriate for your computer. You may download any installer, as many times as you would like, and can install the software on as many different computers as you like, provided they all belong to you.

1.3 Installing the applications

As a first step, you should install the software on your computer's hard disk. Just launch the installer that you downloaded and follow the prompts. If you are on a Linux machine, the installers are shell scripts that you will need to execute as a privileged user (typically using the `sudo` command).

If you are not permitted to install software on the computer that you are using, and that computer is an Apple Macintosh, you can install the applications onto a USB stick. Just choose the USB stick as the target drive in the installer. On a Windows machine it is possible to run the software directly from the CD-ROM.

You will need to save your homework files on a writable disk (such as a hard disk or USB stick). If you are working on a public computer, you should be careful not to leave your homework files on the computer once you are finished, since the files may be erased or copied by another user.

You can register at

<https://www.gradegrinder.net/gradegrinder/login>

(select ‘register as a student’) using the Book ID number (see page 6), login, and download the software.

If you buy online, you will be registered automatically and taken to the page where you can download the software.

1.4 Getting Help

If you encounter problems with the software, you should consult the help pages on the LPL web site. The address of this web site is:

<https://www.gradegrinder.net/>

The web site also contains the latest information about the software, and downloadable updates for registered users. You can also visit the application-specific help pages by clicking on the help icon in the applications. This is a small question mark found in the top right corner of each of the applications main screen.

If you discover a bug in the software, first consult the web site above for any information that we might have about the issue. In particular you should consult the list of Frequently Asked Questions (FAQ) at the web site to see if we have advice on fixing your problem. If you don’t find the information that you need, you can contact us either by filing a bug report, or sending a more general comment from the web site, or by email to lpl-bugreports@lists.stanford.edu.

Chapter 2

Using Submit

Submit is a computer program that allows you to submit your homework exercises over the Internet to the Grade Grinder, a grading server that checks your homework and returns reports to you and, if you ask, your instructor. In this chapter we describe how to use Submit.

2.1 Getting started

The computer you use to submit homework to the Grade Grinder must be connected to the Internet. Submit uses the same form of communication used by web browsers, so if you can access the Internet with your web browser, you should be able to submit files to the Grade Grinder.

To submit files to the Grade Grinder, you need to have all of the following ahead of time:

1. **The solution files you want to submit.** You might want to collect together all the files you want to submit in a single folder. Remember that the files must be named exactly the way you are asked to name them in the book. Submit will only send files whose names begin with **World**, **Sentences**, **Proof**, **Table**, or **Solution** and that are of the appropriate type (Tarski's World, Fitch, Boole, or text files). If you try to submit a file with an incorrect name, it will give you a chance

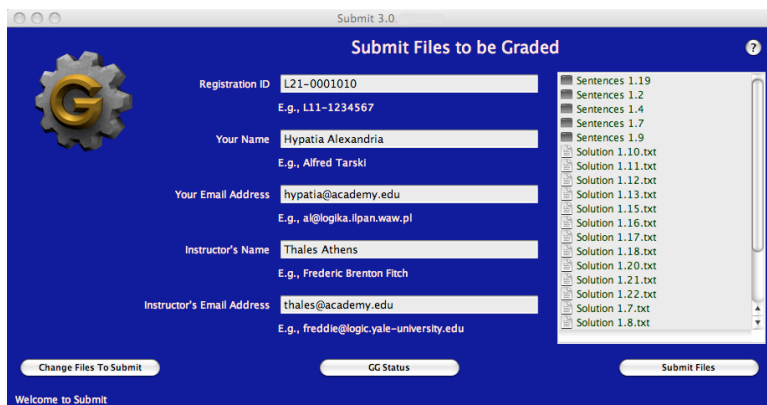


Figure 2.1: Main window of Submit.

to correct the name. If you try to submit a file with an incorrect exercise number (e.g., **World 1.1** rather than **World 10.1**), then Submit will send it but the Grade Grinder will tell you that it doesn't know how to grade it or grade it as the wrong exercise. Be careful when naming your solution files!

2. **Your Book ID number.** This is a unique ID number that is included in the LPL package. It is of the form L21-1234567, that is, a letter, followed by two digits, a dash, then seven more digits. Do not let anyone else use your ID number, since the number is how the Grade Grinder associates your homework exercises with you.
3. **Your name and full email address.** The name you enter should be sufficient for your instructor to identify you. You must use the same email address throughout the course, so make sure you choose the right one and enter it correctly.
4. **Your instructor's name and full email address.** If you want your results to be sent to an instructor as well as to you, you will need his or her name and full email address. The instructor's email address must match one of the instructors in the Grade Grinder's database, so make sure you find out what it is. If you do not want results sent to an instructor, you won't need this information.

Launching Submit

To launch Submit double-click on the application. Its icon shows a cogged wheel with a gold G in the middle.¹ After a moment, Submit's main window will appear on your screen. You'll know it by the spinning cog.

Your goal is simply to fill in the various parts of this window by typing in the information requested and specifying the list of files to be submitted. Once that is done, you will simply press the **Submit Files** button in the lower right of the window.

Start by filling in the information requested (Book ID, your name, etc.). Read about this information above if you haven't already. Remember to use your full email address and to spell it correctly. Once you have submitted files, your Book ID will be associated with the email address you type in, so that no one can use your Book ID to submit bogus homework in your name. In later submissions, you will have to use the exact same email address with your Book ID, so if you have more than one email address, remember which one you used.

2.2 Choosing files to submit

There are several ways to choose the files you want to submit. The most common is to click on the button **Choose Files to Submit** in the lower left corner of the main Submit window. This will open another window showing two file lists. The list on the left shows all the files in the current folder (directory). The list on the right will be built by you as you choose files to submit. The goal is to find the names of your solution files on the lefthand list and move them to the righthand list.

To find your solution files, you will have to navigate around the folder structure of your computer in the lefthand list. To move to "higher" folders, those containing the folder whose contents is currently shown in the list, click on the folder name that appears above the list. A menu will pop up and show all the folders (and volume) that contain this folder. Choose the folder whose contents you want to view. To move to "lower" folders, those contained in-

¹If you installed Submit on a Windows machine, you can also launch the program from the **Start** menu by choosing **All Programs/LPL Software/Submit**.

side the folder whose contents you are viewing, choose those folder names from the list and click **Open**, or simply double-click on the folder names. Using these two techniques, you will be able to find any file located on your computer's hard disk or on any disk inserted into one of the computer's drives.

Once you have found the file(s) you want to submit, select the file name in the lefthand list and click the **Add>>>** button to add the name to the righthand list. If you want to add all files in a folder, just select the folder and use the **Add>>>** button. Keep doing this until the righthand list contains all the files you want to submit. If any of the files are of the wrong type or have names of the wrong form, Submit will let you know before putting them on the list. It will give you a chance to correct the names of files that are of the right type, but not named correctly. (This does not change the names of the files on your computer, only the name sent to the Grade Grinder.) You will also be able to skip files that are not appropriate for submission. When you are finished choosing files, click the **Done** button under the righthand file list.

Another way to specify files to submit is by choosing **Choose File to Submit...** from the **File** menu while you are at the main Submit window. This gives you the standard file open dialog box. If you choose a file of an appropriate type (e.g., a Fitch file), it will be added directly to the list of files to submit. This takes longer if you have more than one file to submit.

The fastest way to specify the files to submit is to drag the files (or a folder containing them) to the Submit application icon. This will launch Submit (if it is not already running) and put the file names directly onto the list of files to submit.

Submitting the files

Once you have entered all the information on the main Submit window and have constructed the list of files to submit, click the **Submit Files** button under the list of files. Submit will ask you to confirm that you want to submit the files on your list, and whether you want to send the results just to you or also to your instructor. When you are submitting finished homework exercises, you should select **Instructor Too**, but if you just want to check to see if you've done the problems right, select **Just Me**. One of these boxes must be chosen before you click the **Proceed** button, which sends your submission.

The first time you make a submission, you are registered as a user of the Grade Grinder system. This registration associates your book ID number with your email address. Because it is so important to enter the email address correctly, you will be asked to confirm your email address the first time you make a submission. On subsequent submissions, you will only have to enter your address once.

If you wish, you can use the **Add Text Message** button to send a text message with your submission. This message will appear in the text of the grade report. It is useful for complaining to your instructor about how long you spent on the exercises.

After a moment, you will get a notice back from the Grade Grinder telling you which files it received and which of them it knows how to grade. (If you misnumbered a solution, it won't know how to grade it.) You can save this notice as a receipt to prove that the files got to the Grade Grinder.

What Submit sends

When you submit files to the Grade Grinder, Submit sends a copy of the files. The original files are still on the disk where you originally saved them.

2.3 How you know your files were received

If you receive the notice back from the Grade Grinder described above, then you know your files were received. If you receive an error message, or if nothing at all happens when you try to submit your files, then the Grade Grinder has not received them. If your submission does not get through, it is probably a problem with your Internet connection. You should try submitting the files again, perhaps from another computer. If you are certain that you are connected to the Internet, you may be able to solve this problem by modifying the proxy server settings in the preferences dialog (see section 2.4.1.)

A second confirmation that your submission was received is the email message that the Grade Grinder will send you with the results of its grading. This will arrive a while after you make the submission, depending on how large the submission was, how many

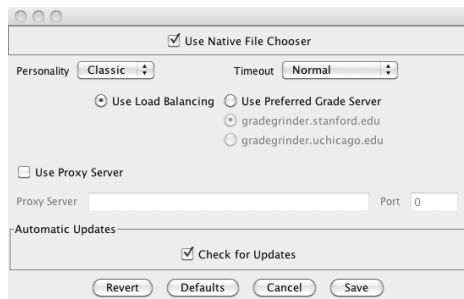


Figure 2.2: Submit Preferences Window

other submissions the Grade Grinder is checking, and how long it takes email to reach you. Generally, you will receive the email message within minutes of submitting your files.

Logging into at our web site allows you to check all your successful submissions.

You can check on a submission by clicking on the **GG Status** button at the bottom of the Submit window. If the Grade Grinder was unable to grade your submission, it will tell you which of your submissions have been delayed and for what reasons.

2.4 Preferences and User Data

The information that you enter into the main Submit window, other than the files to submit, is known as the user data. All of this information except the Book ID is saved on your behalf, so that you can avoid typing your name, email address, etc., each time you run Submit. This information will automatically be entered into the appropriate fields when you launch the program.

Submit also has preference settings which may be changed from the Preferences window, which is obtained by using the Preferences command from the Application menu (Macintosh) or Edit menu (Windows). If you use this command, a window similar to that shown in figure 2.2 will appear. You can change a number of aspects of the way in which Submit behaves using the controls in this window.

2.4.1 Proxy Server Settings

In some cases, you may not be able to connect directly to the Internet to make web connections, usually because your machine is located behind a firewall. In such a situation, there will be a computer that you are allowed to access, and this machine makes connections to the Internet on your behalf, returning the results to your machine. A computer configured in this way is called a “proxy server”. If you need to use a proxy server, you should check the box marked **Use Proxy Server** and enter the name of the server, and the port to connect to, in the text fields below. You will be able to obtain the correct information from the corresponding proxy settings in your web browser, or from a local computer expert.

The default setting is not to use a proxy server, and unless it is necessary to use one, these settings should normally be left unchanged.

2.4.2 Preferred Host Settings

The Grade Grinder is implemented by multiple computers called grade servers. By default, when you make a submission, Submit will determine which server is least busy, and send your submission to that server. This is called “load balancing”. However, if you wish you can ask Submit to prefer one of the servers over the other(s). To do this, check the radio button marked “Use Preferred Server”, and then the radio button that corresponds to your choice of server.

The default setting is to use load balancing, and we do not recommend changing from the default, since you run the risk of unnecessarily delaying your own submissions. The only situation in which selecting a preferred server might be beneficial is if problems with one of the Grade Grinder servers, causes it to accept submissions but be delayed in grading them. If this happens, then preferring to send submissions to the other server would be appropriate.

It is important to know that selecting a preferred server does not guarantee that your submissions will always go to that server. It is a preference, not a guarantee. Submit will try to send your submission to your preferred server first, but if that server is unavailable then the other server(s) will be used instead.

2.4.3 Timeout settings

Submit waits a period of time called the “timeout” when connecting to a grade server. If the server does not respond within this time, the next server is tried, until either one accepts the submission or all have been unsuccessfully tried, in which case you will be notified of the problem. You can modify the length of the timeout value using the timeout choice menu. If you are routinely having difficulty connecting to a grade server within the timeout period, perhaps because you are connecting over a slow dialup line, or sending very large submissions, then you may wish to change the setting to Patient or Very Patient in order to instruct Submit to wait longer for a response. You might like to experiment with the (Very) Impatient settings if you think that you will routinely get very fast responses from the server (perhaps because your connection is very fast due to being on the Internet backbone.)

2.4.4 Personality

The Grade Grinder generates congratulatory messages in the grade report when you get an exercise completely correct. By default, the “classic” behavior, these are randomly selected. You can control the exclamations that are generated by using the Personality setting, or switching all exclamations off using the setting “None” if you prefer.

2.4.5 Check for Updates

There is a global preference which controls whether all of the applications check for updates when they are launched. If this box is checked, the application will determine if an update is available, and ask if you want to download and install it.

2.4.6 Use Native File Chooser

Use Native File Chooser (only visible in MacOS) preference is shared with the other Language, Proof and Logic applications; in almost all cases you should leave it checked.

Chapter 3

Using Tarski's World

Tarski's World lets you represent simple, three-dimensional worlds inhabited by geometric blocks of various kinds and sizes, and test first-order sentences to see whether they are true or false in those worlds. We begin with instructions on how to start and stop Tarski's World, and explain the basic layout of the screen.

3.1 Getting started

The Tarski's World application is contained inside the folder called Tarski's World Folder. Also in this folder is a folder called TW Exercise Files, in which you will find the Tarski's World exercise files referred to in the book.

When Tarski's World is running you will see a large window divided into three sections. The first narrow gray tool bar is the *Tarski toolbar*.

Second (either on the left or in middle depending on window layout, see section 3.6) is the *world panel* containing a checkerboard on which blocks are placed, called a world, and a toolbar above for manipulating the content of this world, which we call the *world toolbar*. In the area between the toolbar and the world is a tab which contains the name of the world. Initially this is **Untitled World**.

The third section has the *sentence panel* which is the white panel to the right or bottom of the window (depending on window layout) with the FOL *toolbar* above it. At first the sentence panel

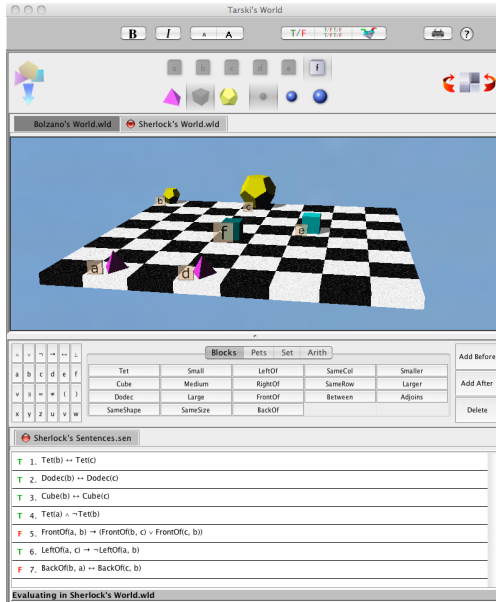


Figure 3.1: Main window of Tarski's World.

contains only the numeral “1” inside. This is where sentences are entered and evaluated to see whether they are true or false in the world represented in the world panel. Feel free to type something in the sentence panel, say, “I’d rather be in Philadelphia.” Immediately above the sentence panel is a tab which contains the name of the collection of sentences. Initially this is **Untitled Sentences**. The FOL *toolbar* above the sentence panel is used to enter sentences of first-order logic. Feel free to play around by clicking on the buttons in the FOL toolbar.

3.1.1 Opening saved files

Both worlds and sentence lists can be saved as files on your disk. Indeed, many prepackaged world and sentence files come with Tarski's World. To open a saved file, you use the **Open...** command on the **File** menu.

To open a file, pull down the **File** menu and choose **Open...** A file dialog will appear which allows you to navigate to the file that

you wish to open. You will have to navigate to the right folder to find the prepackaged files, which are in **TW Exercise Files**. Find this folder, select it, and then click **Open**, or simply double-click on the name. Feel free to open one of the files you see, say, **Ackermann's World**, but if you make any changes to the world, don't save them.

When you open a file, a new tab will be created above the new sentence or world panel.¹ This tab will contain the name of the file that you opened. To return to viewing any other world or sentence file, just click on its tab, and it will reappear.

3.1.2 Starting new files

If you want to start a new world or sentence file, choose **New** from the **File** menu. You may then specify whether you want a new world or new sentence file from the menu which appears. The **New World** and **New Sentences** commands create a new empty world or sentence panel as appropriate. These are created as new tabs within the collection of worlds or sentences.

The command **New Random World** on the **New** menu creates a new world, and populates it with randomly chosen blocks.

The **New Window** item on the **New** menu creates a new window identical to the initial main window.

You may have noticed that there is another **New** command on the **File** menu. depending on which panel is active, this reads **New Sentences** or **New World**, and is equivalent to the corresponding item on the **New** submenu. This item also has a shortcut.

3.1.3 Saving a file

If you want to save a file, use the **Save** submenu from the **File** menu. There are items here which allow you to save the current world, **Save World** or **Save World As...**, the current sentences, or all worlds and sentences in all tabs.

If the file has never been saved before, a dialog box will appear giving you the option of naming the file you are about to create. If you were to hit the return key, or click the **Save** button, the file would be saved with the default name. You should type in some other name before hitting the return key or clicking **Save**. You

¹There is one exception to this rule, and that is when the current tab is one of the "Untitled" tabs that has not been changed. In this case the old tab will be replaced by the new one.

should also make sure you are saving the file where you want it. Check the directory name at the top of the save dialog box. If you're not in the folder where you want to save the file, navigate to the right one by clicking on this name.

You may have noticed that there is another save command on the **File** menu. depending on which panel is active, this reads **Save Sentences** or **Save World**, and is equivalent to the corresponding item on the **Save** menu. This item also has a shortcut.

Once a file has been saved, the name of the file appears in the corresponding tab. If you are working on a named file, the **Save** and **Save As...** commands behave differently. The first will save a new version of the file under the same name, and the old version will be gone. The second gives you a chance to create a new file, with a new name, and keeps the old file, with its name. For this reason, **Save As...** is the safer of the two options.

You can also access the save commands by right-clicking or control-clicking (Macintosh) on the corresponding tab to get the tab popup menu.

All files created by Tarski's World can be read by either the Macintosh or Windows version of the application.

3.1.4 Closing Tabs

When you are done with a world or sentence file, you can close it using the **Close** commands on the **File** menu. As usual, there is a command which closes the active tab whether it is a world or sentence, and a submenu which allows you to close the tab of your choice. The close commands can also be accessed from the tab popup menu or by clicking on the close icon (a red dot) on the tab itself.

3.1.5 Reverting a File

If you want to reload a tab from its corresponding file, you can do so using the **Revert** submenu on the **File** menu. You will be asked first whether you want to save the changes that you have made to the file (to a different file), and then the content of the current tab will be replaced from the file. This command can also be accessed from the tab popup menu.

3.1.6 Printing

To print your sentences or world, choose the appropriate **Export HTML** command from the **File** menu, or from the tab popup menu. This will export the sentences and/or world to a new tab in your default browser. From there you can print. Note that you should print with background colors and images turned on.

3.1.7 Quitting (Exiting) Tarski's World

Eventually you will want to leave Tarski's World. To do this, choose **Quit** from the application menu (**Exit** from the **File** menu on Windows). If you've made any unsaved changes to the files, Tarski's World will give you a chance to save them.

3.2 The World Panel

In this section we explain how to create, edit and save worlds. Worlds can be edited using a variety of different commands. Each of these can be undone using the **Undo** command from the **File** menu, and redone using the **Redo** command, on the same menu.

3.2.1 Adding blocks

To put a block on the grid, simply click the **New** button on the world toolbar. This is the leftmost button on the toolbar, and looks like an arrow pointing at the world, with blocks above it. Try this out. The size and shape of block that is created can be controlled by setting a preference (see section 3.6). A small cube is created by default.

3.2.2 Selecting blocks

A block can be selected by clicking on it. The block will change color to indicate its selection. To deselect a block, click elsewhere in the world panel.

To select more than one block, hold down the command key (control key on Windows) while clicking on the blocks. If you want to deselect a block, repeat the command-click.

3.2.3 Moving blocks

To move a block, position the cursor over the block and drag it to the desired position. (That is, move the mouse's arrow over the block and then, with the button depressed, move the mouse until the block is where you want it.) If multiple blocks are selected, they will all move. There is one exception to this. Large blocks are so big that they overlap onto the adjoining squares. Consequently, it is impossible to move blocks so that a large block adjoins a square that is occupied.

If you move a block (or blocks) too close to the edge it will fall off and be deleted.

3.2.4 Sizing and shaping blocks

To change a block's shape, select it and click on one of the shape buttons on the world toolbar. These display a tetrahedron, cube and dodecahedron and change the shape appropriately. If multiple blocks are selected all will changed to the new shape.

Similarly, to change a block's size, select it and click on one of the size buttons on the world toolbar. These display circles of small, medium and large sizes. If multiple blocks are selected all will changed to the new size. There is one exception to this. Large blocks are so big that they overlap onto the adjoining squares. Consequently, it is impossible to make a block large if it adjoins any square which has a block on it.

3.2.5 Naming blocks

When a block is selected, the name checkboxes on the world toolbar are activated. To add a name to the selected block, click on the appropriate name button, which looks like a cube with a name on it. If the box is already selected, the name will be removed from the block.

In first-order logic, one object can have several names, but two objects cannot share the same name. Hence Tarski's World lets you give a block more than one name, but once a name is used, that name cannot be assigned to another block. So if one block is named **a** and you want a different block to be named **a**, you must remove the name from the first block before adding it to the second.

3.2.6 Deleting blocks

To delete a block, drag the block off the edge of the grid and drop it. Alternatively, select the appropriate block or blocks and hit the **Delete** key.

3.2.7 Cutting, copying, and pasting blocks

If you want to copy some blocks from one file to another, use the cut, copy, and paste functions.

If you select blocks and then choose **Cut** or **Copy** from the **Edit** menu, the blocks are stored on the computer's clipboard. The difference between the two commands is that **Cut** deletes the blocks from their present position, while **Copy** leaves them in place. You can't see the contents of the clipboard, but the blocks will be there until you cut or copy something else to the clipboard.

Once some blocks are on the clipboard, they can be pasted into a different (or the same) world. Just select the relevant tab and choose **Paste** from the **Edit** menu. A copy of the blocks on the clipboard will be inserted.

You can paste several copies if you want to, even into the same world. Tarski's World will attempt to paste the blocks in the same configuration as they were cut, but will need to move them if there are already blocks in any of those positions. Because two blocks cannot have the same name, pasted blocks will have their names removed.

3.2.8 Hiding labels

Whenever you name a block, Tarski's World labels the block with its name. Of course, in the real world we only wear name tags at unpleasant social occasions. Like us, blocks in Tarski's World can have names without wearing labels. To hide the labels, simply choose **Hide Labels** from the **World** menu. To redisplay the labels, choose **Show Labels** from the **World** menu.

This command toggles the display of labels in all open worlds.

3.2.9 2-D view

Labels aren't the only things that can hide. Sometimes a small block can be obscured from view by another block in front of it. To get a bird's eye view of the world, choose **2-D View** from

the **World** menu. To get back to the usual perspective, choose **3-D View** from the **World** menu. These commands can also be accessed from the world toolbar using the button which looks like a small version of the checkerboard.

Blocks can be moved, selected, and changed from the 2-D view in exactly the same way as the 3-D view. You can even change to the 2-D view in the middle of playing the game (see section 3.5); sometimes you will have to in order to pick an appropriate block, or to see what Tarski's World is referring to.

3.2.10 Rotating Worlds

To rotate a world by 90 degrees in either direction, choose **Rotate World Clockwise** or **Rotate World Counterclockwise** from the **World** menu. Such a rotation counts as a change to the world and will be saved when you save the world.

You can also rotate the world from the world toolbar using the arrow buttons.

3.3 The Sentence Panel

There are two ways to enter formulas into the sentence panel, from the FOL toolbar or from the keyboard. Most people find it easier to use this toolbar than the keyboard.

All sentence edits can be undone and redone using the **Undo** and **Redo** commands from the **File** menu.

3.3.1 Writing formulas

Tarski's World makes writing first-order formulas quite painless. As you may have noticed while playing with the FOL toolbar, when you enter a predicate, like **Tet** or **BackOf**, the insertion point locates itself in the appropriate position for entering "arguments"—variables (u, v, w, x, y, z) or individual constants (a, b, c, d, e, f).

What this means is that a sentence like **BackOf(a,b)** can be entered into the sentence list with three mouse clicks in the toolbar: first on the **BackOf** button, then on the **a** button, then on the **b** button. To enter the same thing from the keyboard would require 11 keystrokes.

In order to allow you to write more readable formulas, Tarski's World treats brackets (" $[]$ ") and braces (" $\{ \}$ ") as alternatives to

parentheses when grouping formulae. Thus, for example, you could write $[\text{LeftOf}(a, b) \wedge \text{Large}(a)]$ and Tarski's World will read this sentence as $(\text{LeftOf}(a, b) \wedge \text{Large}(a))$. But you have to type brackets and braces from the keyboard, and you must use parentheses to indicate the arguments in atomic sentences.

3.3.2 Commenting your sentences

You can add comments to your sentences in a way that will be ignored by the program when it is checking to see if they are well formed or true. You do this by prefacing each line of text you want ignored by a semicolon (;). This will cause Tarski's World to ignore anything that follows on the same line. Tarski's World displays all of the characters in the comment in green to remind of their (in)significance.

3.3.3 Creating a list of sentences

To create a whole list of sentences, you first enter one sentence, and then choose **Add Sentence After** from the **Sentence** menu. You are given a new, numbered line, and can then enter a new sentence. If you hit the Return (or Enter) key, this will *not* start a new sentence, but will simply break your existing sentence into two lines. Use **Add Sentence After**!

Instead of choosing **Add Sentence After** from the **Sentence** menu, you can do this from the FOL toolbar by clicking the Add After button or you can do it directly from the keyboard in two ways. You can type Shift-Return (that is, hit the Return (or Enter) key while holding the shift key down) or use the keyboard equivalent shown in the menu.

To insert a new sentence in your list *before* the current sentence, choose **Add Sentence Before** from the **Sentence** menu, or using the **Add Before** button on the FOL toolbar.

3.3.4 Moving from sentence to sentence

You will often need to move from sentence to sentence within a list of sentences. You can move the insertion point with the up and down arrow keys (\uparrow , \downarrow) on the keyboard or by clicking on the sentence of interest with the mouse. The left and right arrow keys (\leftarrow , \rightarrow) on the keyboard also move the insertion point, but only within a single sentence.

Table 3.1: Keyboard equivalents for typing symbols.

Symbol	Key	Symbol	Key
\neg	\sim	\neq	$\#$
\wedge	$\&$	\vee	$ $
\rightarrow	$\$$	\leftrightarrow	$\%$
\forall	$@$	\exists	$/$
\subseteq	$—$	\in	\backslash

If you hold down the Option key, the up arrow takes you to the first sentence of the list, the down arrow takes you the last sentence of the list, and the left and right arrows take you to the beginning and the end of the current word. Holding down the command key while using the left and right arrow keys will move to the beginning or end of the current sentence.

3.3.5 Deleting sentences

To delete a whole sentence and renumber the sentences that remain, choose **Delete Sentence** from the **Sentence** menu. First make sure the insertion point is somewhere in the sentence you want to delete.

Note that you cannot highlight parts of two different sentences and then delete them. If you want to delete a sentence boundary, you must use the command **Delete Sentence** from the **Sentence** menu.

3.3.6 Typing symbols from the keyboard

Sentences can be entered into the sentence panel by typing them on the physical keyboard. When typing predicates in the blocks language, you must be sure to spell them correctly and to capitalize the first letter (since otherwise they will be interpreted as names, not predicates). You also have to insert your own punctuation: parentheses after the predicate, and commas to separate multiple “arguments” (as in **Between(a, x, z)**). To get the logical symbols use the keyboard equivalents shown in Table 3.1.

The sentence panel must be “active” before typing on the physical keyboard will have any effect. If you type and nothing shows up, that’s because the world panel is currently the active panel.

To activate the other panel, just click in it somewhere.

You can change the size of the font used to display sentences using the **Text size** submenu on the **Sentence** menu.

3.3.7 Cutting, copying, and pasting

If you want to change the order of the sentences in a list, or copy a sentence from one file to another, use the cut, copy, and paste functions.

If you highlight a string of symbols and then choose **Cut** or **Copy** from the **Edit** menu, the string of symbols is stored on the computer's clipboard. The difference between the two commands is that **Cut** deletes the highlighted symbols from their present position, while **Copy** leaves them in place. You can't see the contents of the clipboard, but the symbols will be there until you cut or copy something else to the clipboard.

Once something is on the clipboard, it can be pasted anywhere you want it. Just put the insertion point at the desired place and choose **Paste** from the **Edit** menu. A copy of the string of symbols on the clipboard will be inserted. You can paste several copies at several different points, if you want to.

You can copy sentences out of Tarski's World and paste them into Fitch or Boole, and vice versa.

3.4 Verifying syntax and truth

As you will learn, only some strings of symbols are grammatically correct, or well formed, as we say in logic. These expressions are usually called *well-formed formulas*, or *wffs*. And only some of these are appropriate for making genuine claims about the world. These are called *sentences*. Sentences are wffs with no free variables. You will learn about these concepts in the text.

To see if what you have written in the sentence panel is a sentence, and if so, whether it is true in the world currently displayed, click on the Verify button in the Tarski toolbar. This is the leftmost of the group of three colored buttons on the toolbar. Alternatively, you can type Command-Return (Control-Enter on Windows). If you want to check a whole list of sentences, choose **Verify All Sentences** from the **Sentence** menu. Alternatively, use the **Verify All** button on the toolbar, which is in the center of the group..

When you verify a sentence, the results are displayed in the margin to the left of the sentence number: “T” or “F” indicates that the sentence is true or false in the world, “*” indicates that the formula is not well-formed or not a sentence, while “+” indicates that the formula is a sentence of first-order logic, but not evaluable in the current world. When you enter something that is not well-formed, Tarski’s World will display the portion of the expression after the error in red. If you are unsure why a sentence is not evaluable, verifying the sentence again will result in a dialog explaining the reason.

The evaluations are removed when the sentence or world is changed.

3.5 Playing the game

When you stake out a claim about a world with a complex sentence, you are committed not only to the truth of that sentence, but also to claims about its component sentences. For example, if you are committed to the truth of a conjunction $A \wedge B$ (read “A and B”) then you are also committed both to the truth of A and to the truth of B. Similarly, if you are committed to the truth of the negation $\neg A$ (read “not A”), then you are committed to the falsity of A.

This simple observation allows us to play a game that reduces complex commitments to more basic commitments. The latter claims are generally easier to evaluate. The rules of the game are part of what you will learn in the body of this book. Here, we will explain the kinds of moves you will make in playing the game.

To play the game, you need a guess about the truth value of the current sentence in the current world. This guess is your initial commitment. The game is of most value when this commitment is wrong, even though you won’t be able to win in this case.

Clicking on the **Game** button, the rightmost of the group of three colored buttons on the Tarski toolbar, will start the game. Tarski’s World will begin by asking you to indicate your initial commitment. At this point, how the game proceeds depends on both the form of the sentence and your current commitment. A summary of the rules can be found in Table 9.1 in Chapter 9 of the textbook.

3.5.1 Picking blocks and sentences

As you see from the game rules, at certain points you will be asked to pick one sentence from a list of sentences. You do this by clicking on the desired sentence and then clicking **OK**.

At other points in the game, you will be asked to pick a block satisfying some formula. You do this by moving the cursor over the desired block and selecting it. Then click **OK**. If necessary, Tarski's World assigns a name to the chosen block, for example *n1*, and labels it.

3.5.2 Backing up and giving up

Tarski's World never makes a mistake in playing the game. It will win if it is possible for it to win, that is, if your initial commitment was wrong. However, *you* may make a mistake, and so lose a game you could have won. All it takes is some bad choices along the way. Tarski's World will take advantage of you. It will not tell you that you made a bad move until it has won, when it will inform you that you could have won. What this means is that there are two ways for you to lose: if you were wrong in your initial assessment, or if you make a faulty choice in the play of the game. To put this more positively, if you win a game against the computer, then you can be quite sure that your initial assessment of the sentence, as well as all subsequent choices, were correct.

To make up for the edge the computer has, Tarski's World allows you to retract any choices you have made, no matter how far into the game you've gone. So if you think your initial assessment was correct but that you've made a bad choice along the way, you can always retract some moves by clicking on the **Back** button. If your initial assessment really *was* correct, you should, by using this feature, eventually be able to win. If you can't, your initial commitment was wrong.

The **Back** button undoes the last step of the game, while the **Reconsider** button undoes all of the moves since the last time that you made a choice.

If, halfway through the play of the game, you realize that your assessment was wrong and understand why, you can stop the game by clicking the **End Game** button. This ends the game, but does not shut down Tarski's World.

3.5.3 Controlling the interaction in the game

There are three checkboxes which can be used to control the detail with which the game is presented. Some moves can be carried out completely automatically rather than require you to click a confirmation button. We recommend leaving these in the checked (most verbose) position until you gain some experience with the game. The function of these checkboxes are described below

- **Rewrites:** At some points in the game one formula is replaced by another equivalent formula. This is how implications and biconditionals are handled. Since there are no choices to be made at this point, you can make the game proceed with these replacements without requiring interaction by switching off this checkbox.
- **Formula Choices:** At some points in the game Tarski's World must choose one formula from a range of choices. Since the are choice is out of your control, you can make the game proceed with these choices without requiring interaction by switching off this checkbox. Of course, when you are required to make a choice of formula, the game will ask you what to do regardless of the state of the checkbox.
- **Block Choices:** At some points in the game Tarski's World must choose a block from the world. Since the are choice is out of your control, you can make the game proceed with these choices without requiring interaction by switching off this checkbox. Of course, when you are required to make a choice of block, the game will ask you what to do regardless of the state of the checkbox.

3.5.4 When to play the game

In general, you won't want to play the game with every sentence. The game is most illuminating when you have incorrectly assessed a sentence's truth value, but are not sure why your assessment is wrong. When this happens, you should always play the game without changing your commitment. Tarski's World will win, but in the course of winning, it will usually make clear to you exactly why your assessment was wrong. That's the real value of the game.

You might wonder what happens when you play the game with a correct assessment. In this case, if you play your cards right, you

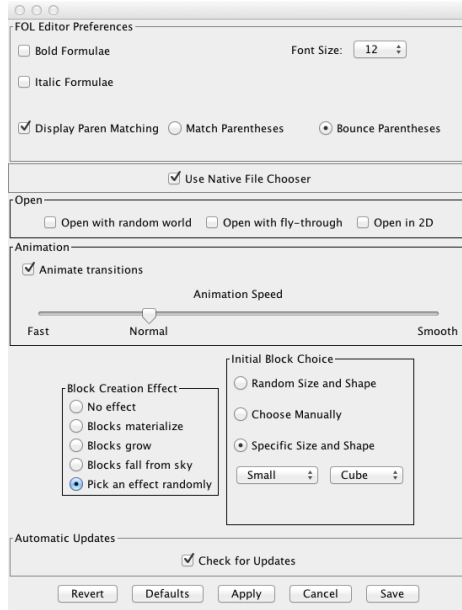


Figure 3.2: Tarski's World Preferences Dialog

are guaranteed to win. But Tarski's World does not simply give up. At those points in the game when it needs to make choices, it will make them more or less randomly, hoping that you will blunder somewhere along the line. If you do, it will seize the opportunity and win the game. But, as we have noted, you can always renege by backing up.

3.6 Preferences

Some aspects of the behavior of Tarski's World can be controlled using the preferences dialog. This can be accessed by choosing the **Preferences...** command from the application menu (**Edit** Menu on Windows). The preferences dialog is shown in figure 3.2.

The preferences allow you to control various aspects of the applications. The **FOL Editor Preferences** are shared by Fitch, Tarski's World, and Boole and control the style of the text you type such as bold/italic, size, and whether parentheses give hints

on how they are paired. **Use Native File Chooser** (only visible in MacOS) is also shared; in almost all cases you should leave it checked. **Automatic Updates** is also a global preference which controls whether the applications check for updates when they are launched. If this box is checked, the application will determine if an update is available, and ask if you want to download and install it.

Preferences specific to Tarski's World include the **Open** panel. With it you can opt to create a random world instead of an empty one when a new world is created by selecting the **Open with random world** checkbox. You can view an animation as the world is opened or created by selecting the **Open with flythrough** checkbox, and you can opt to always open worlds in 2-D by selecting **Open in 2D**. The **Window Layout** panel determines whether the world and sentence sections are side by side (landscape or horizontal, the default) or world section above sentence section (portrait or vertical).

You can control the speed of animations, or switch all animations off using the **Animation** panel. The speed of animations is controlled by a slider. When the slider is set to the **Fast** end of the scale, the animations will have fewer frames, resulting in a more jerky animation which takes less time. The **Smooth** end of the scale will result in smoother, but longer, animations. You might like to play with this setting to get the effect that is just right for your computer. If nothing seems right, then you can switch all animation off.

You can choose a different effect for how new blocks are created with the **Block Creation Effect** panel, varying from dropping from the sky, materializing or growing in place. We think that the effects are pretty nifty. You might like to try them out.

You can determine the size and shape of the block that is created when the **New Block** button is pressed with the **Initial Block Choice** panel. You have the option of allowing Tarski's World to choose a size and shape for you, having a dialog, or always creating the same kind of block.

Chapter 4

Using Fitch

Fitch is an application that makes it easy to construct formal proofs in first-order logic. We begin with instructions on how to start and stop Fitch, and explain the basic layout of the screen.

4.1 Getting started

The Fitch application is contained inside the folder called Fitch Folder. Also in this folder is a folder called Fitch Exercise Files, in which you will find the Fitch exercise files referred to in the book.

When Fitch is running, you will see (from top to bottom) the ubiquitous *menu bar*, a narrow *proof toolbar*, a wider, gray FOL *toolbar*, and a large, mostly blank window, called the *proof window*, with a *goal strip* at the bottom. Here are the basic facts to remember about each of these.

4.1.1 The menus

Fitch has the following menus:

- **File:** This menu lets you start new proof files, open existing proofs, save proofs, and export proofs for printing.
- **Edit:** This is the customary edit menu allowing one to cut, copy and paste items in proofs. In addition one can toggle author mode (see section [4.6](#))

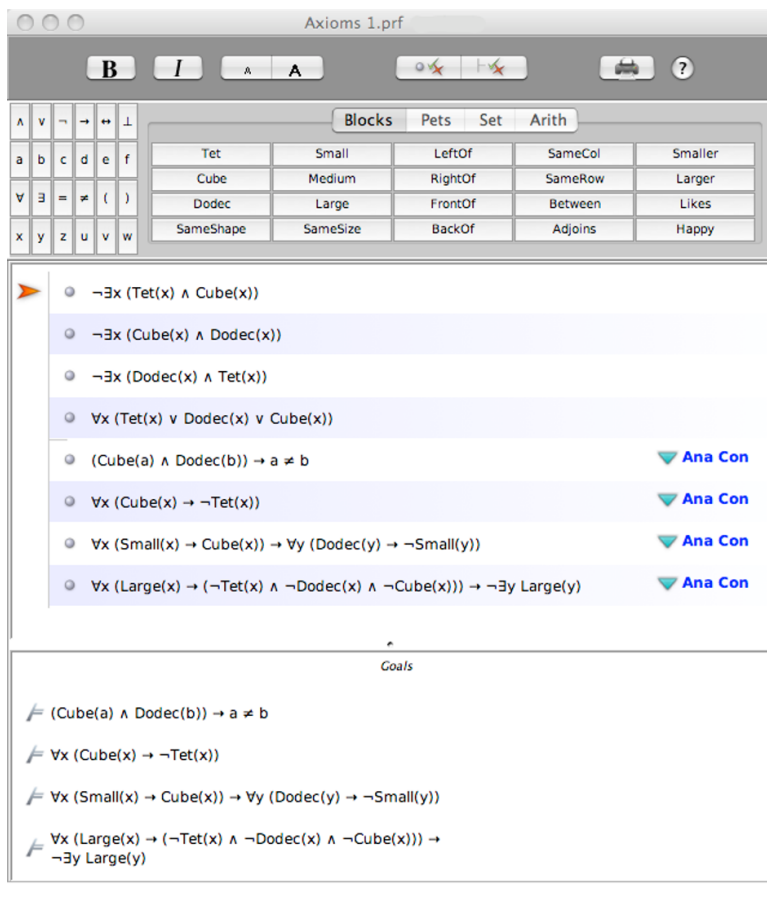


Figure 4.1: Main Fitch window.

- **Proof:** The items on this menu allow you to add steps to proofs, to begin and end subproofs, to check the correctness of proofs, and to display step numbers.
- **Goal:** The items on this menu allow your instructor to set goals in problems and allow you to see whether any special constraints apply to the goals.

- **Window:** This menu gives you access to various Fitch files you have open, and allows you to change the font and appearance of your open windows.
- **Help:** This menu allows you to get help on using the application, and also to check whether updates are available.

4.1.2 The proof toolbar

The narrow toolbar at the top of the window is called the *proof toolbar*. This is like the window toolbar in Tarski's World, and contains buttons which allow you to control the display of the proof. These tools include buttons for changing the font to bold or italic, and to control the size of text in the proof. There are also buttons for checking the current step, and the whole proof. Finally, there are buttons to print (actually export to the default browser) the proof, and to open the help system for the application. One can toggle the visibility of this toolbar under the **Window** menu.

4.1.3 The FOL toolbar

The strip containing logical symbols and predicates located at the top of the proof window is called the FOL *toolbar*. It is like the FOL toolbar in Tarski's World, and is used for writing and editing in the proof window. Clicking on a button enters the symbol or predicate into the proof. One can toggle the visibility of this toolbar under the **Window** menu.

4.1.4 The proof window

The proof window is itself divided into two areas. The larger top part or *proof panel* is where you construct proofs. When you add steps or subproofs to proofs, for example, this is where they will show up. When you add a step to a proof, the word **Rule?** appears on the right. This is a popup menu that you click on to choose the rule you want to use at the step.

The bottom panel of the proof window is where the problem's goals appear, that is, the sentences to be proven. If the *goal strip* is not visible, choose **Show Goal Strip** from the **Goal** menu. To hide it, and give yourself more room in the proof panel, choose **Hide Goal Strip** from the **Goal** menu.

At the extreme bottom of the proof window, under the goal strip, is a *status line* that displays comments and error messages. It is usually blank but is used to present useful information, especially when proof or goal steps don't check out. In some cases (e.g., Taut Con errors) a **Show It** button appears there, click on that to get more information. You can also check the step of the proof in focus by simply clicking on the status line.

4.2 Creating and editing proofs

The body of a proof appears in the large *proof panel* in the middle of the window, sandwiched between the FOL toolbar and the goal strip. In this section, we explain how to create, modify, and navigate around a proof. Before doing this, we present a list of some of the graphical elements you will encounter in the proof panel:



Proof line and Fitch Bar. Proofs and subproofs are demarcated by a vertical gray line. Attached to the line is a horizontal bar called the Fitch bar. The Fitch bar separates the assumptions of the proof from the steps that follow from those assumptions.



Focus Slider. The focus slider appears just to the left of the proof and points to the currently focused step. If there are goals in the goal strip, the focus slider can also point at one of the goals. There is only one focus slider in the window at any time.



Step Bullet. This circular icon indicates the presence of a step in the proof. You can either add a new sentence at the step, if none is present, or you can edit an existing sentence. If step numbers are displayed (**Show Step Numbers** from the **Proof** menu), the step bullet is replaced by the step number.



Goal Bullet. This “turnstile” icon precedes each of the goal sentences in the goal strip (unless one is in focus, in which case the turnstile is replaced by the focus slider).



Constant Box. The constant box appears at the top of subproofs in which a new constant or constants have been introduced. The constant box depicted here indicates that **a** is a newly introduced constant. The downarrow to its right indicates a menu where constants can be added or removed from this box.



Rule? **Rule menu icon.** This indicates a popup menu where you can choose a rule to justify a step of the proof.

4.2.1 Step numbers

Normally, Fitch does not display step numbers, but simply indicates the steps with bullets. Choosing **Show Step Numbers** from the **Proof** menu will replace the bullets with numbers. When you display step numbers, the support steps are indicated by number next to the rule name, exactly as they are shown in the text.

4.2.2 The current focus

As you work on a proof, there is always one step that is “in focus.” This step is indicated by a red triangle on the far left of the proof, called the **Focus Slider**. The focused step is the step affected when you perform any of the editing functions. It is also the step that is checked if you click on the **Check Step** button or on the status line at the very bottom of the proof window.

4.2.3 Moving the focus

There are three ways to change the focus from one step to another: You can drag the focus slider up and down, you can click in the focus slider area next to the step you want to focus on, or you can use the arrow keys on the keyboard to move the focus up and down.

Clicking on a step other than the currently focused step won’t move the focus, unless you click in the focus slider area to the left. This is because clicking on steps is the way we enter supports for the currently focused step (see page 37).

Shift-clicking in the focus slider area beside a step other than the currently focused step selects a range of steps which can then be copied or deleted.

4.2.4 Adding new steps

To add a new step to a proof, choose **Add Step After** or **Add Step Before** from the **Proof** menu. These commands will give you a new proof step immediately after or before the step you were focused on, unless you were focused on a premise, in which case the new step will be the first step following the premises. To add a step at the end of the proof, focus on the last step of the proof and choose **Add Step After**. If you are working backwards in a proof, you will often want to add a step immediately before the step in focus, using **Add Step Before**. Steps can be inserted in the middle of a proof in the obvious way, by first moving the focus and then choosing the appropriate add step command.

New steps added within a subproof will appear in the same subproof. Starting and ending subproofs require different commands. These are described on page 38. Adding premises to a proof is discussed on page 42. Normally, though, you will not be adding premises to your proofs, since the exercise files contain the premises already.

4.2.5 Entering sentences into steps

When you add a new step to a proof, the step icon (a small circle) will appear. At this point you can enter a sentence. To enter, either use the FOL toolbar or type directly from the keyboard. To type the logical symbols from the keyboard, refer to the list of keyboard equivalents found in Table 3.1 on page 22. These are the same keyboard equivalents used in Tarski's World and Boole.

In general, entering sentences is faster using the toolbar. However some sentences must be entered using the keyboard, since the predicates, names, or sentence letters may not appear on the toolbar. For example, you will have to type parts of the sentence $P \rightarrow (Q \vee R)$, since P, Q, and R don't appear on the toolbar.

4.2.6 Deleting steps

To delete a step, focus on the step and choose **Delete Step** from the **Proof** menu. If you delete the assumption step of a subproof (the step just above the Fitch bar), the entire subproof containing that step will be deleted. Be careful in deleting assumption steps, since you could lose a lot of work. If you simply want to change the assumption, just edit the sentence.

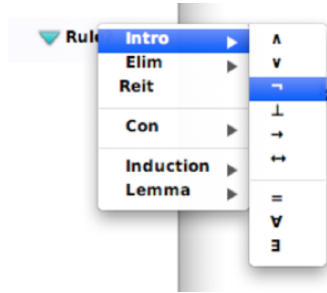


Figure 4.2: Using hierarchical menus to specify rules.

4.2.7 Specifying a step's rule

When you add a new step, the word “**Rule?**” appears to the right of the step. To specify a rule for the step, focus on the step, click down on the word “**Rule?**” A popup menu will appear. This menu has five submenus plus the rule of **Reit**.

The submenus are attached to **Intro**, **Elim**, **Con**, **Lemma**, and **Induction**. Moving the cursor over these submenus will cause a second menu to appear with a list of further options. To specify the rule of, say, Negation Introduction, first move the cursor over the item **Intro**. Then, when the second menu appears, move the cursor to the item \neg (or “not”) and release the mouse button. This operation is illustrated in Figure 4.2. Similarly, to specify the rule of **Taut Con**, first move the cursor over the item **Con** and then choose **Taut** from the second menu.

You can also specify a rule from the keyboard, by typing the appropriate keystroke equivalent. A list of these appears in Table 4.1. Note that in using these keyboard equivalents, you should not hold down the shift key unless it is explicitly mentioned in the table. For example, to specify the rule \forall **Elim**, you will actually type Option-2 (Macintosh) or Alt-2 (Windows); holding down the shift key will change the rule to \forall **Intro** instead. In most cases, we have listed the shifted (“uppercase”) characters because they are easier to remember. For example, it is easier to remember that \wedge **Elim** is Option-& than to think of it as Option-7.

Table 4.1: Keystroke equivalents for specifying rules.

Rule	Equiv	Rule	Equiv
\wedge Elim	Opt(Alt)-&	\wedge Intro	Shift-Opt(Alt)-&
\vee Elim	Opt(Alt)-	\vee Intro	Shift-Opt(Alt)-
\neg Elim	Opt(Alt)-~	\neg Intro	Shift-Opt(Alt)-~
\perp Elim	Opt(Alt)-^	\perp Intro	Shift-Opt(Alt)-^
\rightarrow Elim	Opt(Alt)-\$	\rightarrow Intro	Shift-Opt(Alt)-\$
\leftrightarrow Elim	Opt(Alt)-%	\leftrightarrow Intro	Shift-Opt(Alt)-%
$=$ Elim	Opt(Alt)-=	$=$ Intro	Shift-Opt(Alt)-=
\forall Elim	Opt(Alt)-@	\forall Intro	Shift-Opt(Alt)-@
\exists Elim	Opt(Alt)-/	\exists Intro	Shift-Opt(Alt)-/
Reit	Control-Opt(Alt)-R	FO Con	Control-Opt(Alt)-F
Taut Con	Control-Opt(Alt)-T	Ana Con	Control-Opt(Alt)-A

4.2.8 The Lemma Rule

Lemma allows you to reference another proof file and use it in your proof. Once a lemma is referenced, its name appears as the rule and two extra buttons will be to the right of it. The left one (**display lemma**, a rectangle with lines) will open a new window with the lemma in it (do not edit this displayed lemma but you can do a 'save as' and then edit). The right one (**expand lemma**, an up arrow) will expand the lemma within the proof and make any necessary substitutions (be aware that this is not reversible so save first).

The Lemma rule will check out when the formula in the step matches the only goal in the lemma file, the same number of steps are cited as there are premises in the lemma file, and these cited steps all match the premises in the lemma (in any order). The premises and the cited steps have to *match* but do not have to be identical. If the lemma contains the premises P and Q, for example, citing the formula $P \wedge Q$ won't work. However some substitution does work. If for instance the lemma proves $P \vee \neg P$ you can use it on a step that states $\text{Tet}(a) \vee \neg \text{Tet}(a)$.

4.2.9 Changing a step's rule

To change the rule of an existing step, you must first move the focus to that step. Then specify the new rule using either the popup menu or the keystroke equivalents.

4.2.10 Specifying a step's supports

Most rules require that you cite other steps as justification or “support.” To specify the supports for a step, focus on that step and click on the steps to be cited. The steps you click on will be highlighted. If your support is a subproof, clicking anywhere in the subproof will cite the whole subproof. If you click on a step or subproof that has already been cited, it will be uncited.

To see a step's supports, just focus on the step in question. The supporting steps will then become highlighted. To change a step's supports, focus on the step and click on the steps you wish to add or delete from the step's supports.

If you are displaying step numbers in a proof, the support steps are highlighted and the step numbers appear to the right of the rule name. Thus with step numbers displayed, your proofs will look like the proofs in the text.

4.2.11 Checking steps and verifying proofs

To check whether a step is correct, focus on the step and either press the **Check Step** button on the proof toolbar or click on the status line at the bottom of the window. (On the Macintosh, you can also check a step by hitting the Enter key on the numeric keypad.)

You can check all of the steps in your proof, plus the goals, by clicking **Verify Proof** on the proof toolbar or by choosing **Verify Proof** on the **Proof** menu.

After you check a step, one of four symbols will appear to the left of the rule name.



Check mark. A check mark means that the step is logically correct.



X. An X means that the step is logically incorrect.



Asterisk. An asterisk means that the sentence at that step is not syntactically well-formed.



Question mark. A question mark will appear if a **Con** rule is unable to determine the validity of your step.

If you don't get a check mark for one of your steps, focus on that step and look at the message in the status line. It will provide you with some (hopefully) helpful information about why your step did not check out.

4.2.12 Rule Defaults

Many of rules have defaults that can save you considerable time when constructing a proof. For example, if you choose the rule \rightarrow **Elim**, cite two sentences of the form $P \rightarrow Q$ and P , and then check the step, Fitch will automatically fill in the step with the sentence Q . To get Fitch to provide a default for a step, the sentence must be blank, that is, there must not be any text already in that step. If the sentence is blank when the step is checked, Fitch will try to provide a default sentence for that step. The defaults for the rules are described in detail in the textbook.

Not all inference rules have defaults.

4.2.13 Add Support Steps

Many of the rules allow you to use the **Add Support Steps** command to automatically insert the appropriate support steps needed to derive a particular formula using that rule. To use this feature, focus on a step and insert the formula to be derived at this step. Then select the inference rule to use. Finally select the **Add Support Steps** command from **Proof** menu. If this is disabled, then the rule does not support this option (or you didn't choose a rule or enter a formula.) Otherwise using this command will insert the necessary steps into the proof.

4.2.14 Starting and ending subproofs

A subproof is started by choosing **New Subproof** from the **Proof** menu. When you start a new subproof, you can enter a sentence (or boxed constant) in the first step. Once you are within a subproof, any new steps you add will be part of that subproof. To add a step *after* a subproof, you need to know how to end the subproof. To end the subproof, focus on any step in the subproof and choose **End Subproof** from the **Proof** menu. This will end the subproof and give you a new step following that subproof.

If, when you end a subproof, the last step of the subproof is empty, then that step will just be moved out of the subproof. This means you can end two embedded subproofs by choosing **End Subproof** twice. The first time, you will end the innermost subproof and get a new step in the outer subproof. The second time, the new step will be shifted out of that subproof as well.

4.2.15 Boxed constants in subproofs

When you start a subproof, a downward-pointing triangle appears next to the step bullet. This triangle indicates the presence of a popup menu. If you click down on the triangle, the menu will appear. In this case, the menu presents you with a list of all the names available in Fitch. Choosing one of these names adds the name as a boxed constant—unless it is already boxed, in which case it is removed from the box. Boxed constants are used in the rules \forall **Intro**, and \exists **Elim**.

4.2.16 Collapsing subproofs

When you complete a subproof, you might want to “collapse” it so that you don’t have to think about it any more. This is particularly true if the subproof is long, and you want to avoid scrolling back and forth over it while working on a different part of the proof. You can do this using the **Collapse Subproof** command from the **Proof** menu, when you are focused on any step within the subproof. The subproof will be replaced by a single step containing an icon to represent the collapsed proof. To open the proof again, chose **Expand Subproof** from the **Proof** menu when you are focused on the step containing the collapsed proof.

4.2.17 Deleting subproofs

To delete a subproof, focus on the assumption step that begins the subproof and choose **Delete Step** from the **Proof** menu. This will delete the entire subproof, so make sure you really want to do that. If you simply want to change the assumption step, edit the sentence, don’t delete the step.

4.3 Goals

The goals for a problem are represented by sentences that appear in the goal strip at the bottom of the proof window. These are the sentences that are to be proven in your proof. If the goal strip is not visible and you would like it to be, choose **Show Goal Strip** from the **Goal** menu. If the goal strip is visible but you would like more room for the proof, choose **Hide Goal Strip** from the **Goal** menu.

When you are working on a problem and think that you have satisfied one or more of the goals, choose **Verify Proof** from the **Proof** menu. Either a check or an X will appear to the right of each goal. If an X appears, focus on the goal by clicking on it, and read the error message that will appear in the status line.

4.4 Copying and pasting

Fitch allows you to cut, copy, and paste various parts of a proof. Mastering these operations will make the construction of proofs much easier.

When you cut or copy something from a proof, it is placed on the “clipboard.” The clipboard is a part of the computer’s memory that you can’t see, but which stores whatever you have cut or copied so you can later paste it somewhere else in the proof. The difference between cut and copy is that the former deletes the item in question from its current place in the proof, while the latter leaves the proof itself untouched, and stores a copy of the item on the clipboard.

Once something is on the clipboard, you can paste copies of it into the proof as many times as you want. It will remain on the clipboard until something else is cut or copied, at which point the new item replaces what used to be on the clipboard.

4.4.1 Copying and pasting sentences

To cut or copy a sentence, or part of a sentence, you must first be focused on the step that contains the sentence. Select the portion you want by clicking down at one end and dragging to the other, holding the mouse button down as you drag. Once it is selected, choose **Copy** or **Cut** from the **Edit** menu. Both of these

commands place a copy of the selected text on the clipboard; the second simultaneously deletes it from the step.

If you want to copy an entire sentence from a step, you simply focus on the step and choose **Copy** from the **Edit** menu. There is no need to select the sentence. This places a copy of the whole sentence from that step on the clipboard, ready to paste elsewhere in the proof (or in another proof). This shortcut is particularly useful if you want to copy one of the premises when the authoring mode is off, for in that case the premise will be locked and you will not be able to select it.

Once the sentence is on the clipboard, you can paste it into another step by moving to that step and choosing **Paste** from the **Edit** menu. The text will appear wherever text typed from the keyboard would, so if you want to paste it into the middle of some existing text, make sure the blinking insertion point is located where you want the text to appear.

4.4.2 Copying and pasting goal sentences

You can copy a goal sentence by focusing on it and choosing **Copy** from the **Edit** menu. This is an easy way to grab the desired sentence and paste it into your proof.

4.4.3 Copying and pasting ranges of steps

Fitch allows you to cut or copy a range of steps and paste them at another location in the same proof or in another proof. This is especially useful if your proof requires several similar subproofs, each containing a similar sequence of steps.

To cut or copy a range of steps, you must first select the steps. Shift-clicking on a step will select the range of steps between the focus step and that step. You cannot select a range containing just one step, so shift-clicking on the focus step will do nothing.

A gray box will appear, showing which steps are selected. If the rectangle doesn't contain the steps you want, click somewhere else in the proof and the rectangle will disappear. You can then try selecting the steps again. Note, however, that Fitch will not allow the selection rectangle to cut a subproof in half: you must either select steps entirely from within a subproof, or else select the subproof as a whole. When the rectangle contains exactly the steps you want, choose **Cut** or **Copy** from the **Edit** menu. Both

of these commands place a copy of the steps on the clipboard; **Cut** also deletes the selected steps from the proof.

Once a sequence of steps is on the clipboard, choosing **Paste** will insert the steps at the point of focus. If you are currently focused on an empty step, the pasted steps will replace the empty step. If you are currently focused on a step that is not empty, the pasted steps will be inserted *after* the focused step.

If you want to paste steps into your proof immediately following a subproof, but not as part of the subproof, you will have to end the subproof before pasting. This will give you an empty step outside the subproof and **Paste** will replace this empty step with the steps on the clipboard.

When you paste steps into a proof, Fitch will try to keep track of the appropriate supports for those steps. Sometimes, though, the supports for the pasted steps will no longer be “legal” in the new location, for example if you paste a step into the proof at a point earlier than one of its support steps. In such cases, Fitch will remove the illegal support from the step’s list of supports.

Note that this method also gives you a handy way to delete a large number of steps. Rather than repeatedly choosing **Delete Step** from the **Proof** menu, simply select all of the steps you wish to delete, and choose **Clear** from the **Edit** menu. Simply hitting the Delete key will also delete the selected range of steps.

4.5 Printing proofs

To print a proof, choose **Export HTML** from the **File** menu. When you do this, your default browser will open a new tab with the proof displayed in it and you can print using your browser’s standard print command. In the displayed proof Fitch adds numbers to all of the steps and uses these numbers to indicate each step’s supports so that the printed proofs look like the proofs in the textbook. Note that you should print in the browser with background colors and images turned on.

4.6 Setting up exercises

Fitch has two modes of operation, user mode and author mode. Exercise files are always opened in user mode. This mode allows you to construct proofs, but not to change the premises or goals of

the proof, which of course is not permitted in solving the exercises. (The Grade Grinder always checks to make sure that no changes have been made to the premises or goals of a proof.)

Author mode is used for creating new exercises, and so new files are always opened in author mode. This mode allows you to enter premises into the proof, add goals to the proof, and specify any constraints that apply to the goals.

You can tell which mode you are in by looking at **Author Mode** on the **Edit** menu. If there is a check in front of **Author Mode**, Fitch is in author mode; otherwise, it is in user mode. Choosing **Author Mode** will toggle between these two modes.

The current mode is saved with the file, so if you create a new problem, you should turn off author mode before saving it. Alternatively, you can use **Save As Problem...** from the **File** menu. This command is available only in Author Mode, and will save only the premises and goals in the new file. The created file is saved in user mode.

4.6.1 Adding and deleting premises

To add a premise, you must be in Author Mode. Choose **Add Premise** from the **Proof** menu. If you are currently focused on a premise step, the new premise will appear immediately after the focused step. If you are focused on a step in the body of the proof, the new premise will appear at the end of the list of premises.

To delete a premise, focus on its step and choose **Delete Step** from the **Proof** menu.

4.6.2 Adding and deleting goals

To add a goal to a problem, choose **New Goal** from the **Goal** menu and enter the goal sentence you want. As noted above, this can only be done in Author Mode. To delete a goal from a problem, click on the goal in the goal strip and choose **Delete Goal** from the **Goal** menu.

You can check use the **Check Goal Forms** from the **Goal menu** to check that the goal formulas are well formed. Goals that are not well formed will be marked with a star.

If you want to modify the constraints on a goal, choose **Edit Goal Constraints** from the **Goal** menu (see figure 4.3). When you initially add a goal, Fitch assumes that you want the goal to

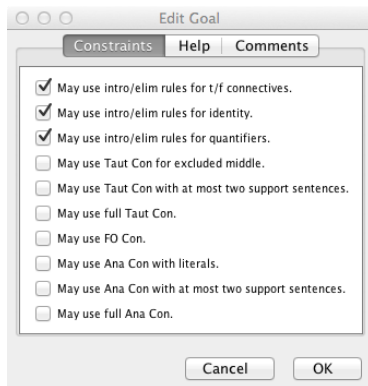


Figure 4.3: Fitch Edit Goal Constraints window

be proven using just the introduction and elimination rules of \mathcal{F} . If you want to allow the use of the **Con** procedures, or if you want to disallow the use of any standard rules, you will have to modify the constraints associated with the goal. Constraints are associated with individual goals, so a problem can have different constraints for different goals.

4.6.3 Saving new problems

To save a newly created file as an exercise to be solved, choose **Save As Problem...** from the **File** menu. This saves the file, but also turns on User Mode in the file, so that users will not accidentally change the premises or goals of the exercise.

4.7 Preferences

Some aspects of the behavior of Fitch can be controlled using the preferences dialog. This can be accessed by choosing the **Preferences...** command from the application menu (**Edit** Menu on Windows). The preferences dialog is shown in figure 4.4.

The preferences allow you to control various aspects of the applications. The **FOL Editor Preferences** are shared by Fitch, Tarski's World, and Boole and control the style of the text you type such as bold/italic and whether parentheses give hints on how they are paired. **Use Native File Chooser** (only visible

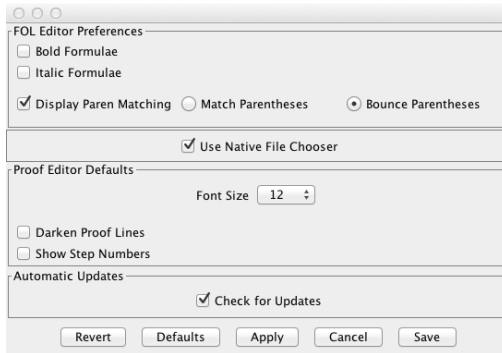


Figure 4.4: Fitch Preferences Dialog

in MacOS) is also shared; in almost all cases you should leave it checked. **Automatic Updates** is also a global preference which controls whether the applications check for updates when they are launched. If this box is checked, the application will determine if an update is available, and ask if you want to download and install it.

Proof Editor Defaults are specific to Fitch and allow you to darken proof lines, always show step numbers, and set the font size for the proof.

Chapter 5

Using Boole

Boole is an application that makes it easy to construct truth tables. We begin with instructions on how to start and stop Boole, and explain the basic layout of the screen.

5.1 Getting started

The Boole application is contained inside the folder called **Boole Folder**. When Boole is running, you will see (from top to bottom) the *menu bar* (on the mac this is not part of the window), the *table toolbar*, the *FOL toolbar*, a narrow *assessment panel*, and finally a large, mostly blank area for constructing truth tables, called the *table panel*. At the very bottom is the *status line* which is used to display useful information about your work, including a description of errors that you may have made. Here are the basic facts to remember about each of these.

5.1.1 The menus

Boole has the following menus:

- **File:** This menu lets you start new truth tables, open existing tables, save tables, and print (actually export to the default browser) tables.
- **Edit:** This is the customary edit menu allowing you to cut, copy and paste items in truth tables. See section [5.2.3](#) for the *By Row* feature.

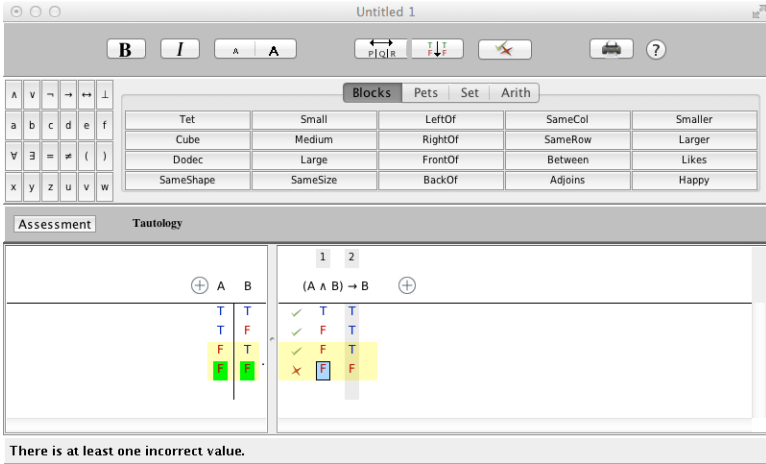


Figure 5.1: Main Boole window.

- **Table:** This menu has commands for adding new columns, for generating reference columns automatically, for verifying your table, and for showing or hiding the column order (see section 5.2.4).
- **Window:** This menu gives you access to various Boole files you have open and allows you to adjust font size and style.
- **Help:** This menu allows you to get help on using the application, and also to check whether updates are available.

5.1.2 The toolbars

At the very top of the window is the *table toolbar* which is very like those in the corresponding places in Fitch and Tarski's World. There are buttons here for changing the size and format of the sentences to be displayed in the truth tables, as well as three buttons for checking and making edits to the table.

- **Build Reference Columns:** This shows a collection of columns and a bidirectional arrow. This button can be used after you have entered one or more target sentences on the right side of your table. If you choose this button, Boole will automatically create all of the reference columns needed

for your table. You should use this feature only if the exercise says that you can, since otherwise the Grade Grinder will complain that you did not create your own reference columns. Learning how to create reference columns is part of what you must master when learning to build truth tables.

- **Fill Reference Columns:** This button fills in the truth values under the reference columns for you. It shows a collection of truth values and a downward pointing arrow. Like the preceding button, you should only use this if the exercise says that you can, since filling in the reference columns is another crucial skill to master in building truth tables.
- The remaining button verifies the correctness of your table by first checking whether the rows of the table are correctly filled in, then that all of the required rows are present (and there are no extra rows), and finally checks to see whether your assessment is correct. You can also check the table using the **Verify** menu item from the **Table** menu.

Immediately below this toolbar is the FOL *toolbar* containing logical symbols and predicates. It is just like the FOL toolbar in Tarski's World and Fitch. Clicking on a button enter the corresponding symbol or predicate, if the insertion point is located in one of the sentence fields at the top of your table.

5.1.3 The assessment panel

The assessment panel is similar to the goal panel in Fitch. This is where you look to see whether your truth table is correctly constructed. It also contains a button that allows you to assess the target sentence or sentences in your table. When you click the **Assessment** button, you are presented with a list of possible assessments. For example, if you are asked to determine whether a sentence is a tautology, you can specify whether or not it is. If you are asked to determine whether a sentence is a tautological consequence of other sentences, you can specify this here as well.

5.1.4 The table panel

The large, mostly white area is where you construct truth tables. A thin horizontal line divides the headings of the columns from

the truth value columns. A thick vertical line divides the reference columns from the body of the table. When the insertion point is in the heading area, you can enter target sentences on the right or reference sentences on the left.

New sentence columns are added using the “plus” buttons in the heading area. These buttons add a new column at the end of the existing columns, nearest to the button. If you hold the Alt key down when you click, the new column is added at the end furthest from the button. You can also use the commands **Add Column After** or **Add Column Before** from the **Table** menu. These add new columns to the left (right, respectively) of the column that is currently in focus. Once sentences are entered, truth values can be entered in the appropriate places under those sentences.

5.1.5 Adjusting the table panel

The table panel is divided into two parts: the *reference columns* on the left and the *target columns* on the right. These are separated by a vertical divider. Often when you open or construct a table, you will not be able to see all of the columns on one or the other side. You can adjust the overall size of your window in the usual way, and you can also change the position of the divider by grabbing it and dragging to the left or right.

5.2 Writing and editing tables

There are three steps in creating a truth table: specifying the target sentence (or sentences), building the reference columns, and filling in the truth values. Once a table is complete, it can be used to assess the logical properties of the target sentences.

5.2.1 Entering target sentences

To enter a target sentence, the insertion point must be blinking in the upper right section of the table. If it is not, click in this area to place the insertion point where you want it. Then enter the desired sentence using the toolbar or keyboard. To type the logical symbols from the keyboard, refer to the table of keyboard equivalents on page 22. Notice that as you enter the sentence, the characters you type change color. If the expression is ill-formed, some of the characters will be displayed in red.

If you are constructing a truth table for two or more sentences (a *joint* truth table), you will need to use the “plus” button, or choose **Add Column After** or **Add Column Before** from the **Table** menu to add a new target sentence to your table.

Another way to enter sentences is by copying them from Fitch or Tarski’s World and pasting them into the appropriate place in Boole.

5.2.2 Creating reference columns

There are two ways to create reference columns: you can do it by hand or you can have Boole do the work for you. You should always do the work yourself unless the exercise or your instructor gives you permission to let Boole build the reference columns for you.

To enter a reference sentence, click at the top of the first column to the left of the thick dividing line. Then enter the desired atomic sentence. (Boole will allow you to enter any formula in a reference column. As in the case of the target columns the characters change color to indicate ill-formed expressions.) To add additional reference sentences, choose **Add Column After** or **Add Column Before** from the **Table** menu or the “plus” button, and enter the sentence.

If you want Boole to build the reference columns for you, click on the button **Build Ref Cols** on the table toolbar. This will generate the necessary reference columns for the target sentences currently appearing on the right. If Boole generates the reference columns the column header will be displayed with a grey background to remind you that Boole built them. Reference columns built by Boole cannot be edited, but clicking anywhere in the header area will enable you to remove all of the reference sentences.

5.2.3 Filling in truth values

To fill in truth values in your table, click in the desired column and type T or F. (You can also type “1” for T and either “0” or “2” for F, if you find this more convenient.) After the letter is entered, the insertion point will move down one row in the same column. If you are working in a column on the right side of the table, Boole will highlight the values elsewhere in the table that the current value depends on. In other words, Boole implements the “two finger” method described in the textbook.

Boole will also fill in truth values in the reference columns of a table automatically, should you so desire. Only do this if the exercise or your instructor says that you can. (The Grade Grinder will complain if you do this without permission.) To fill them in automatically, select the **Fill Reference Columns** item from the **Table** menu. If Boole fills the truth values, the table is shown with a grey background. You may not edit the truth values, but clicking on the panel will enable you to have Boole remove all of the values, so that you can start over and fill them yourself.

In its default mode, Boole will fill in values column by column. That is, when you type a T or F, the insertion point will move down one row so that you can fill in the next value in the column. If you prefer filling in your table row by row, choose **By Row** in the **Edit** menu. In Row mode, Boole will move the insertion point to the next column after you type a truth value.

5.2.4 Using the Column Ordering Panel

The column ordering panel appears above the sentences. It is closed initially, and can be opened using the **Show Column Ordering** command from the **Table** menu. When it is open the panel contains a single-line table with columns for each of the connectives in each target sentence. Unlike the main truth table, these columns should contain numbers.

To fill the column ordering panel correctly, you should place numbers in the columns, starting at 1 and increasing until you reach the number of connectives in the sentence. The numbers should reflect the order in which you can fill the columns of the truth panel. The column containing the number 1 indicates that you think that you can fill this column of the truth table first. To be able to fill it first, the values that the column contains must depend only on values in the reference columns. The column numbered 2 could depend column number 1 and on reference columns, and so on. Each numbered column can depend only on reference columns and columns with smaller numbers. Since only single digit numbers can be used, the column ordering panel can only be used for sentences containing less than 10 connectives.

Filling the column ordering panel is optional. If the column ordering panel contains at least one number, then the ordering that is given is checked as part of the assessment of the table. If all of the columns are empty, then the column ordering panel is

Your Assessment

☐ is
☐ is not
 a tautology

☐ is
☐ is not
 TT-possible.

☐ is
☐ is not
 TT-Contradictory

☐ are
☐ are not
 tautologically equivalent.

☐ is
☐ is not
 tautological consequence of others.

☐ is
☐ is not
 tautological consequence of others.

☐ are
☐ are not
 Tautologically Inconsistent

Clear Cancel OK

Figure 5.2: Boole Assessment window

ignored when checking the table.

5.3 Specifying your assessment

Typically you will be asked to use truth tables to determine whether a sentence is a tautology, whether two sentences are tautologically equivalent, or whether a sentence is a tautological consequence of others. Thus after you have constructed your truth table, you will need to specify the relevant assessment of the sentence or sentences in your table. To do this, click on the **Assessment** button in Boole's assessment panel. This will open a window (see figure 5.2) allowing you to specify your assessment.

5.4 Verifying your table

There is a single command for verifying your table. This is available both from the **Table** menu and on the table toolbar. Verifying the table involves the following steps.

1. If the column ordering panel contains any numbers, then these are checked first. If the numbers in the panel are correct, then a checkmark will appear to the left of the row, one

for each sentence on the right hand side of the table. If you are going to give a column ordering for any of the sentences, then you must give one for *all* of them, or the missing ones will be marked incorrect.

2. Each of the rows is checked next. If all of the references columns are present and the values filled in under the target sentence are correct, a checkmark will appear to the left of the row on the target side of the table.
3. Next, Boole checks whether the table has exactly the required rows that it needs. If a row is missing, or if there are extra rows, then this check will fail.
4. Finally Boole checks to see whether your assessment is correct. Before it can do this, you must specify an assessment, as described above. The checkmark for this will appear in the assessment area.

5.5 Saving or printing your table

To save your table, choose **Save** or **Save As...** from the **File** menu. If you are submitting a table to the Grade Grinder, you should name it **Table n.m**, where n.m is the number of the exercise.

To print a table, choose **Export HTML** from the **File** menu. When you do this, your default browser will open a new tab with the table displayed in it and you can print using your browser's standard print command. Note that you should print with background colors and images turned on.

5.6 Preferences

Some aspects of the behavior of Boole can be controlled using the preferences dialog. This can be accessed by choosing the **Preferences...** command from the application menu (**Edit** Menu on Windows). The preferences dialog is shown in figure 5.3.

The preferences allow you to control various aspects of the applications. The **FOL Editor Preferences** are shared by Fitch, Tarski's World, and Boole and control the style of the text you type such as bold/italic, font size, and whether parentheses give hints on how they are paired. Use **Native File Chooser** (only visible

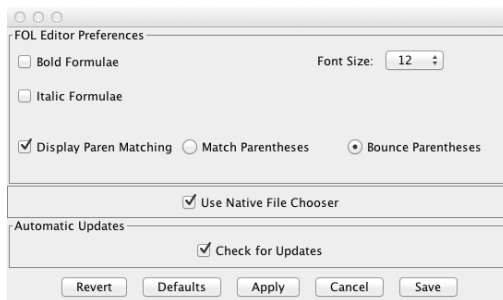


Figure 5.3: Boole Preferences Dialog

in MacOS) is also shared; in almost all cases you should leave it checked. **Automatic Updates** is also a global preference which controls whether the applications check for updates when they are launched. If this box is checked, the application will determine if an update is available, and ask if you want to download and install it.

Chapter 6

Languages

Fitch, Tarski's World, and Boole all share a common FOL toolbar and one part of that are shortcuts for various languages used in the three programs though the Blocks language is the most used and is shown by default.

6.1 Blocks

Predicates

Tet(*a*) *a* is a tetrahedron

Cube(*a*) *a* is a cube

Dodec(*a*) *a* is a dodecahedron

SameShape(*a*, *b*) *a* is the same shape as *b*

Small(*a*) *a* is small

Medium(*a*) *a* is medium

Large(*a*) *a* is large

SameSize(*a*, *b*) *a* is the same size as *b*

LeftOf(*a*, *b*) *a* is located nearer to the left edge of the grid than *b*

RightOf(*a*, *b*) *a* is located nearer to the right edge of the grid than *b*

FrontOf(*a*, *b*) *a* is located nearer to the front of the grid than *b*

BackOf(*a*, *b*) *a* is located nearer to the back of the grid than *b*

SameCol(a, b) a is in the same column as b

SameRow(a, b) a is in the same row as b

Between(a, b, c) a, b and c are in the same row, column, or diagonal, and a is between b and c

Adjoins(a, b) a and b are located on adjacent (but not diagonally) squares

Larger(a, b) a is larger than b

Smaller(a, b) a is smaller than b

Happy(a) a is happy

Likes(a, b) a likes b

6.2 Pets

Predicates

Pet(x) x is a pet

Person(x) x is a person

Student(x) x is a student

Home(x) x is at home

Happy(x) x is happy

Angry(x, t) x was angry at time t

Hungry(x, t) x was hungry at time t

Fed(x, y, t) x fed y at time t

Owned(x, y, t) x owned y at time t

Gave(x, y, z, t) x gave y to z at t

$t < t'$ t is earlier than t' (for times)

Names

max, claire *Max, Claire* (people)

pris, scruffy *Pris, Scruffy* (cats)

folly, carl *Folly, Carl* (dogs)

2:00, 2:05, 3:00 *2:00pm, 2:05pm, 3:00pm, Jan 2, 2011* (times)

6.3 Sets

Predicates

$a \in b$ a is an element of b

$a \subset b$ a is a subset of b

6.4 Arith

Functions

$s(a)$ the successor of a

$a \times b$ a multiplied by b

$a + b$ a added to b

Predicates

$a < b$ a is less than b

The only name is 0 (zero).

Index

- add
 - Boole
 - Column After, [49](#)
 - Column Before, [49](#)
 - Fitch
 - goal, [43](#)
 - premise, [43](#)
 - Step After, [34](#)
 - Step Before, [34](#)
 - Submit
 - files to list, [8](#)
 - Tarski
 - blocks, [17](#)
 - Sentence After, [21](#)
 - Sentence Before, [21](#)
- animation, Tarski, [28](#)
- arrow keys
 - Fitch, [33](#)
 - Tarski, [21](#)
- assessment, Boole
 - button, [48](#)
 - panel, [48](#), [52](#)
- asterisk, Fitch, [37](#)
- author mode, Fitch, [42](#)
- blocks, Tarski
 - 2-D view, [19](#), [20](#)
 - add, [17](#)
 - copying, [19](#)
 - cutting, [19](#)
 - delete, [19](#)
 - hiding labels, [19](#)
 - in Game, [25](#), [26](#)
 - moving, [18](#)
 - naming, [18](#)
 - pasting, [19](#)
 - preferences, [28](#)
 - selecting, [17](#)
 - sizing or shaping, [18](#)
- Book ID, Submit, [1](#), [6](#), [7](#), [10](#)
- By Row, Boole, [51](#)
- check mark, Fitch, [37](#), [38](#)
- Check Step, Fitch, [37](#)
- Choose Files, Submit, [7–8](#)
- Closing tabs, Tarski, [16](#)
- column order, Boole, [47](#), [51–52](#)
- columns, Boole, *see* reference columns
 - and target columns
- Con menu, Fitch, [35](#)
- constant box, Fitch, [33](#)
- copy
 - Boole, sentences, [46](#)
 - Fitch
 - goals, [41](#)
 - sentences, [40–41](#)
 - steps, [41–42](#)
 - Tarski
 - blocks, [19](#)
 - sentences, [23](#)
- current focus, Fitch, [33](#)
- cut

- Fitch
 - sentences, [40](#)
 - steps, [41](#)
- Tarski
 - blocks, [19](#)
 - sentences, [23](#)
- delete
 - Fitch
 - goal, [43](#)
 - premise, [43](#)
 - step, [34](#), [39](#)
 - Tarski
 - blocks, [19](#)
 - sentence, [22](#)
- display lemma, Fitch, [36](#)
- Done
 - Submit, [8](#)
- Edit
 - Tarski, [19](#), [23](#)
- Elim menu, Fitch, [35](#)
- End Game, Tarski, [25](#)
- exercise files
 - Fitch, [29](#)
 - creating, [42–44](#)
 - Tarski's World, [15](#)
- expand lemma, Fitch, [36](#)
- Export HTML
 - Boole, [53](#)
 - Fitch, [29](#), [31](#), [42](#)
 - Tarski
 - Sentences, [17](#)
 - World, [17](#)
- Fitch bar, Fitch, [32](#)
- focus, Fitch
 - current, [33](#)
 - moving, [33](#)
 - slider, [32](#)
- Game, Tarski
 - backing up, [25](#)
 - ending, [25](#)
 - giving up, [25](#)
 - playing, [24](#), [27](#)
- GG Status, Submit, [10](#)
- goal, Fitch, [30](#)
 - adding, [43](#)
 - bullet, [32](#)
 - deleting, [43](#)
 - strip, [31](#)
- Grade Grinder, [1](#), [5](#), [6](#), [8–10](#)
- hide
 - Fitch
 - FOL toolbar, [31](#)
 - Goal Strip, [31](#), [40](#)
 - proof toolbar, [31](#)
 - Tarski
 - Labels, [19](#)
- home page, LPL, [4](#)
- Induction menu, Fitch, [35](#)
- installing the applications, [2](#), [3](#)
- Instructor Too, Submit, [8](#)
- Intro menu, Fitch, [35](#)
- Just Me, Submit, [8](#)
- launching
 - Boole, [46](#)
 - Fitch, [29](#)
 - Submit, [7](#)
 - Tarski, [13](#)
- Lemma menu, Fitch, [35–36](#)
- LPL home page, [4](#)
- menus
 - application, [17](#)
 - Boole, [46](#)
 - Edit, [46](#), [51](#), [53](#)

- File, [46](#), [53](#)
- Help, [47](#)
- Table, [47–52](#)
- Window, [47](#)
- Fitch, [29–31](#)
 - Constant, [33](#), [39](#)
 - Edit, [29](#), [40–44](#)
 - File, [29](#), [42–44](#)
 - Goal, [30](#), [31](#), [40](#), [43](#)
 - Help, [31](#)
 - menu bar, [29](#)
 - Proof, [30](#), [32–34](#), [37–40](#), [42](#), [43](#)
 - Window, [31](#)
- Fitch rule, [31](#), [33](#), [35](#), [37](#)
 - Con menu, [35](#)
 - Elim menu, [35](#)
 - Induction menu, [35](#)
 - Intro menu, [35](#)
 - Lemma menu, [35–36](#)
- Submit
 - File, [8](#)
- Tarski
 - Edit, [19](#), [23](#), [27](#)
 - File, [14–17](#), [20](#)
 - New Random World, [15](#)
 - New Sentences, [15](#)
 - New World, [15](#)
 - revert file, [16](#)
 - Save File, [15](#)
 - save file, [16](#)
 - Sentence, [21–23](#)
 - tab popup, [16](#), [17](#)
 - World, [19](#), [20](#)
- mode, Fitch
 - author, [42–43](#)
 - user, [42–43](#)
- moving blocks, Tarski, [18](#)
- naming blocks, Tarski, [18](#)
- open
 - Fitch, [29](#)
 - Submit, [8](#)
 - Tarski, [14](#), [15](#)
- panel
 - Boole
 - assessment panel, [48](#), [52](#)
 - table panel, [48–49](#)
 - Fitch
 - goal strip, [31](#), [32](#), [40](#), [43](#)
 - proof panel, [31](#), [32](#)
 - Tarski, [15](#), [16](#)
 - sentence, [13](#), [20–23](#)
 - world, [17–20](#)
- paste
 - Boole, [46](#)
 - Fitch
 - goals, [41](#)
 - sentences, [40–41](#)
 - steps, [41–42](#)
 - Tarski
 - blocks, [19](#)
 - sentences, [23](#)
- premise, Fitch
 - adding, [43](#)
 - deleting, [43](#)
- printing, *see* Export HTML
- Proceed, Submit, [8](#)
- proof, Fitch
 - creating, [32–40](#)
 - editing, [32–40](#)
 - line, [32](#)
 - verifying, [37](#)
 - window, [31–32](#)
- question mark, Fitch, [38](#)
- quitting, Tarski application, [17](#)
- reference columns, Boole, [47](#), [49–51](#), [53](#)

- add, 49
 - build, 47
 - fill, 48
- Reit, Fitch, 35
- reverting a file
 - Tarski, 16
- Rotating Worlds, Tarski, 20
- rule, Fitch
 - Add Support Steps, 38
 - changing, 37
 - defaults, 38
 - keyboard equivalents, 36
 - popup menu, 31, 33
 - specifying, 35
- save
 - Boole, 53
 - Fitch, 29
 - as problem, 43, 44
 - Tarski, 15
- selecting blocks, Tarski, 17
- sentences, *see* writing wffs
- shaping blocks, Tarski, 18
- Shift-Return
 - Tarski, 21
- show
 - Fitch
 - Goal Strip, 31, 40
 - Step Numbers, 33
 - Tarski
 - Labels, 19
- sizing blocks, Tarski, 18
- solution files, Submit, 5
- starting
 - Boole, 46
 - Fitch, 29
 - Submit, 7
 - Tarski, 13
- Status Line
 - Boole, 46
 - Fitch, 32, 33, 37, 38, 40
- step, Fitch
 - bullet, 32
 - checking, 37
 - numbers, 33
- submitting files, Submit, 7–9
- subproof, Fitch
 - as support, 37
 - boxed constants, 33, 39
 - collapsing, 39
 - copying, 41
 - cutting, 41
 - deleting, 39
 - deleting assumption, 34
 - ending, 30, 38–39
 - starting, 30, 38–39
- support, Fitch
 - specifying, 37
- syntax
 - Tarski, 23–24
- table panel, Boole, 46, 48–49
 - adjusting, 49
- target columns, Boole, 49, 50, 53
 - add, 49
 - order, 51–52
- target sentences, Boole, 47–50, 53
- toolbar
 - FOL (First-Order Logic), 55
 - Boole
 - FOL (First-Order Logic), 46
 - sentence, 48–50
 - table, 47–48, 50, 52
 - Fitch
 - FOL (First-Order Logic), 29, 31–32, 34
 - proof, 29, 31, 37

- Tarski's World
 - FOL (First-Order Logic),
 - [14](#), [20](#), [21](#)
 - Tarski, [13](#), [23](#), [24](#)
 - world, [13](#), [17](#), [18](#), [20](#)
- truth
 - Tarski, [23–24](#)
- user data, Submit, [10](#)
- user mode, Fitch, [42](#)
- verify
 - Boole, [48](#)
 - assessment, [53](#)
 - column ordering, [52](#)
 - rows, [53](#)
 - table, [53](#)
 - Fitch
 - proof, [37](#), [40](#)
 - Tarski
 - all, [23](#)
 - all sentences, [23](#)
- vertical divider, Boole, [49](#)
- view
 - Tarski
 - 2-D, [19](#)
 - 3-D, [20](#)
- web page, [4](#)
- Window menu
 - Fitch, [31](#)
- windows layout, Tarski's World,
 - [28](#)
- world panel
 - Tarski, [13](#)
- writing wffs
 - Boole
 - using keyboard, [49](#)
 - using toolbar, [49](#)
 - Fitch
 - using keyboard, [34](#)