6.871 Assignment 1: Minesweeper

Due: February 23, 2006, 9:35AM

The Problem

This is a knowledge engineering task, i.e., a task involving writing down the knowledge needed to be good at something. In this case the something is the Minesweeper game from Windows. Your job is to create the best set of rules to play minesweeper that you can. We will run all the rule sets against a challenging set of examples and see if we can crown one player the class champion. In doing this assignment you should be sensitive to what it means to specify the knowledge needed to be good at the game. That is,

- be aware of how writing rules differs from writing a traditional program, what we referred to in the lectures as "telling it what to know, not what to do."
- be aware of what goes through your mind as you figure out the rules to begin with, i.e., what the thought processes are in figuring out what you need to know and in stating it explicitly. (Simply put, try to tune in to the process of knowledge engineering.)

We have created SmartSweeper, an MSPDE – MineSweeper Player Developer Environment. It will enable you to write, test, and debug a set of rules for the game. You can see how well you do, and can improve your rules until you're happy with them.

Assignment Part One

Play several games of minesweeper. Even if you've played it before, play it again, but try to be very aware of the patterns you are reacting to and the reasoning you are doing. If you've never played it before, you may actually have an advantage, as it's easier to be conscious of how you're thinking about the game when it's still unfamiliar.

Assignment Part Two

Use SmartSweeper to write rules capable of beating the 5 boards provided. You will turn in your "rules.txt" file as evidence (just e-mail this file to the TA). You will need java version 1.5 to run SmartSweeper. This is available at java.sun.com (it should also be available on Athena using "add java"). You can get SmartSweeper from the class website (http://courses.csail.mit.edu/6.871/SmartSweeper/). After you unpack it, you can run SmartSweeper with "run.bat" (for Windows) or "run.sh" (for Linux and hopefully Mac OS X).

When you run SmartSweeper, you will see a tempting green button with an arrow on it. Pressing this button applies the rules on the left to the board on the right. The slider

beneath the board then allows you to see the sequence of rule applications. Incidentally, if you see a green rectangle on the board, then a rule has matched at that location, and the panel on the right will show more information about this match.

About rules.txt

The text area on the left of SmartSweeper loads “rules.txt” when the program starts (from the same directory as SmartSweeper). Whenever you close the program or press the green button, it saves it. It also saves periodically (every 10 seconds). Note that SmartSweeper will *not* reload “rules.txt” if it is modified outside of SmartSweeper while SmartSweeper is running (sorry).

If you want to write the rules in a separate text editor, you can either edit “rules.txt” while SmartSweeper is closed, or else you can cut and paste rules from your text editor into SmartSweeper.

Rule Specification

A rule has a name (for your convenience), a left hand side (the condition), and a right hand side (the action to take if the left hand side is satisfied).

```
Rule: name
LHS
=>
RHS
```

These rules can appear one after the other in “rules.txt”.

The Left-Hand Side

The left hand side of each rule consists of a Pattern, optionally follows by a scheme expression.

```
LHS ::= Pattern [scheme expression]
```

A Pattern is a matrix (of any width and height) of characters describing a subsection of the board. Legal characters are:

`M`	an explored mine
`0-8`	a square with this number in it
`-`	a square known not to be a mine
`?`	an unexplored square
`*`	any square
`a-z`	a square with a number in it, and the number is remembered
`|`	(the “pipe” character) a square off the board

NOTE: You do not need to create separate rules for symmetric patterns; the system will automatically try each rule in every possible rotation and reflection (all 8 of them).

If a scheme expression is provided, then the left hand side of the rule will only be satisfied if the scheme expression evaluates to true. Here are some predefined variables available in the scheme expression:

`star-mine-count`	If the pattern has any `*` characters in it, then this says how many of the `*`'s matched known mines.
`star-unknown-count`	If the pattern has any `*` characters in it, then this says how many of the `*`'s matched unexplored squares.
`a-z`	Let's say the pattern has `x` in it somewhere, then `x` in the scheme expression says what number `x` matched in the pattern.

The Right-Hand Side

On the right-hand-side (RHS), you specify what action to take if the LHS matches. This can be expressed in the form of another pattern, and/or a predefined action:

RHS ::= [Pattern] [mark-stars] [click-stars]

Note that you must provide *something* for the RHS.

If you provide a pattern, it must have the same dimensions as the pattern on the LHS. All the system looks for in the pattern are '`M`'s and '`-`'s. If it sees an '`M`' in the pattern where there is an unexplored square on the board, then it marks this square as a mine. If it sees a '`-`' in the pattern where there is an unexplored square, then it "clicks" this square in order to explore it.

If you provide the text "mark-stars", then any stars in the *LHS* pattern corresponding to unexplored squares in the board will be marked as mines. Similarly, if you provide the text "click-stars", then any stars in the LHS pattern corresponding to unexplored squares will be "clicked" in order to explore them.

About the Boards

It is worth noting that the 5 boards provided are solvable by logic alone, and do not require guessing. In fact, you will never need a pattern larger than 5x5 to solve any board (probably none larger than 4x4 is required, but this is not guaranteed).

The Competition

You are fine as far as your grade goes if your rules solve the 5 boards provided with SmartSweeper, but we will run your rule sets against a different set of boards as part of a competition. The rule set which explores the most squares on these boards without making any mistakes will be crowned as champion, with appropriate fanfare and a prize for the author. If two rule sets correctly explore the same number of squares, then the rule set with fewer rules will win.