

CS 523 – Expert Systems

GO Game Expert

Term Paper

John Leung

Winter '04

1. Introduction

GO is an ancient board game invented in China four thousand years ago, making it the oldest game exists today. Despite that, it is the most popular game in the world, played in many Asian communities around the world. It is simple to learn, yet difficult to master. The game is played on a 9x9, 13x13, or 19x19 boards. Because pieces can be placed anywhere on the board, computer game programs that rely on brute force search can only play at an amateur level at best. This GO game expert is not an attempt to be a strong GO player; it is only an expert in terms of knowing the rules of the game, and to give reasons of some of the rules during the game.

2. Problem Domain Description and Justification

Expert Systems have been shown to be extremely useful in some applications, but useless in others. In particular, after a doing a bit more research in the types of expert systems exist today, the ones that prove to be the most useful are ones where a specific computer program is a better form of expert than just looking up printed material, notes, or even web pages (human experts are not counted since they will usually be the most useful **if** they are available whenever you need them). This is the starting point for my motivation to build an expert system – to build one that is really useful, does not exists already, and is not better served by any other means.

In narrowing my choice of the expert system to build, my second criterion was to find expertise from printed material or web pages. The reason is first, I could not think of any person who is really an expert in something. The other reason is that from reading

the book and watching the video, I learned that knowledge gathering from human experts are very time intensive and I do not have that much of free time. The third criterion was to build something that I am interested in and is really going to use it myself. This leaves me with only two choices – a PDA (Personal Digital Assistant) selector or a GO game expert.

I was tending towards the PDA selector since I follow PDA news almost religiously. I know of every good PDA site and I can say that I am a decent PDA expert myself. The biggest motivation was that I know it is going to be easy, much easier than the GO game expert (actually now that I have finished the GO game expert and realized how difficult it is, “much easier” is really an understatement). However, I can not get over one of my first criterion – to build an expert that does not exist already. This is because there are a few good PDA selector experts on the web already; the *Active Sales Assistant* at activebuyersguide.com is one, and pdabuyersguide.com is another. These two sites do such a good job both in terms of user interface and updating (new PDA comes out about once a week) that I really do not believe I can make a better one. Thus I went with the GO game expert.

The idea of needing to keep it updated also gives me more motivation to build the GO game expert since it will truly be a useful expert system for a long time because the rules in GO have not changed much in four thousand years. This GO expert system should be of real help to beginners trying to get started in GO.

3. Project Scope

A true computer GO expert, one that can play the game well, is still years away. The best computer GO program available today can only play at an amateur's level. Thus, instead of trying to build an "expert" program that plays the game rather poorly, I have focused on building an expert that **knows** all the rules. Even being just the rules expert, it still serves my main goal of building an expert system that is useful and does not exist already. Since rules such as capturing and illegal moves are built-in into the system, when such occasions arises, the user (a beginning player) does not have to look up why it's happening in books or internet; the GO expert will tell the user why automatically. This is really helpful for beginners to learn the game faster and better – I know since I wished I had this to help me when I started to learn the game a few years back.

The second limitation is that since three different sets of facts must be implemented for the three different board sizes, this can be a very time consuming task. Thus, in this first implementation of the system, I will only allow a 9x9 board. This is sufficient as it is since the system is intended for beginners and they are recommended by GO experts to begin on 9x9 boards to feel and learn the rules of the game. The addition of 13x13 and 19x19 boards will be added after other more important future additions, which will be discussed in the future developments section.

The third and last limitation is that scoring turns out to be more difficult than previously thought. Thus, the scoring system is not fully implemented yet and will be part of future development.

4. Knowledge Acquisition

Even though I know most, if not all of the rules in GO, I still need some form of written material in order to freshen up my memory. The main source of this is *The Way To Go* by Karl Baker, published by The American Go Association. Actually, their web site is very helpful as well. I found many GO software programs which can be helpful and complements my GO expert very well. More on these programs will follow later. *The Way To Go* is a guide on how to play the game and in doing so, they have laid out all of the rules. It is very useful in helping me sort out the rules' priorities.

5. Knowledge Representation and Verification

The main problem in this GO expert is how to formalize the rules of the game into rules and facts in the expert system. In doing so, I have categorized them into the following “engines” of the GO expert system (will be described in details later):

- Turn Keeping
- Move Legality Check
- Groups and Liberties Tracking
- Capturing System
- End Game and Scoring

Before describing these in details, I will first go over my approach to the problem and also some basics about the game.

5.1 Approach

Before I got started in this, I knew that it has to be implemented in CLIPS. CLIPS' template definitions and pattern matching is needed. Trying to implement this in

Level 5 will be a nightmare. Also this will be a forward chaining or facts driven system. On first thought, it may seem better fitted as goal driven since this is a game expert system. However, as mentioned before, this is not an expert in terms of trying to win the game (thus goal driven). Instead, it is an expert in terms of providing the rules and helping the user to play the game. Thus, every event is driven by the input or the positions of the play.

5.2 Basic of GO

The basics of the game are as follows:

- Black and white pieces (stones) and are played on points, which are the intersections of the lines on the board.
- Play begins with black placing one stone, then white, then alternate back to black, and so on.
- Once stones are played, it is fixed in that location unless it is being captured by the opponent.
- The main idea in GO is to capture as much territory and stones as possible and at the same time, trying to prevent the opponent from doing so.
- To capture the opponent's stone, you must cut off all liberties of the opponent.
- Liberties are immediate points of groups on board not taken by stones of other color.

Further descriptions of liberties and groups are needed. In the following diagram, point C is the only liberty of the white stone.

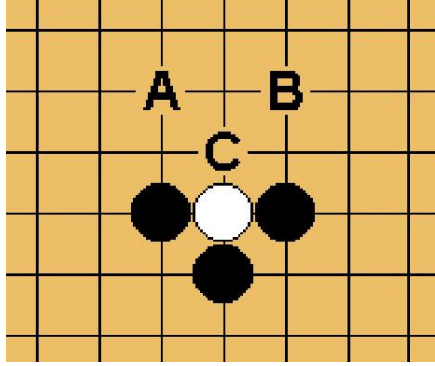


fig.1 Liberty example

Groups are formed when one or more of the same colored stones meet on the board. In fig.1, there are four groups, one white group of a single stone and three black groups of also a single stone. The notion of liberties also applies to groups as well. For example, in the following diagram, point A is the only liberty for the white group.

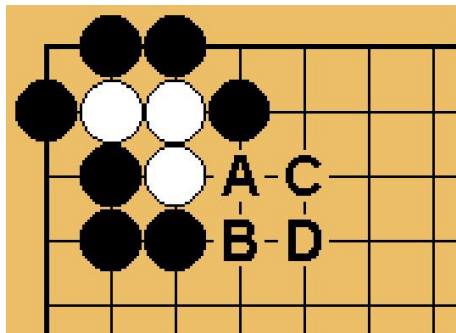


fig. 2 Group example

Now we are ready to proceed to the “engines” of the GO expert system.

5.3 Turn Keeping

Of course the first and most fundamental key to a good GO expert is to keep whose turn it is (black or white’s turn). There are two solutions to this. First, the user can enter the players’ color followed by the position, that is: <color> <position>, e.g.

black a2. However, in a full board of 19x19, the user will typically enter more than a hundred such moves. Even if I have shortened the <color> field to be just 'b' or 'w', it will still be a hassle. Thus, I have implemented automatic turn keeping instead. Since black always starts first, black is asserted at the beginning. All the user needs to enter is the position. After the user enters the position, it will check to see if it is legal (described later). If so, the new position will be entered into the knowledge base and the current color will be retract and the other color will be assert. This facilitates automatic turn keeping since the system will request next move from the other color once it sees the assertion of a new color. If move is illegal however, it will request move for current color again by reasserting the current color.

5.4 Move Legality Check

This is divided into two parts. The first part is a simple check to see if a stone exists on the board already or if the move is an invalid label of the board space. If so, it will ask the user to reenter move using the turn keeping method described previously. The other possible invalid move is a move where the stone will be captured immediately. Thus, the second part to the legality check is done within the capturing system. The capturing system requires the notion of grouping of stones, which brings us to our next section, Groups and Liberties Tracking.

5.5 Groups and Liberties Tracking

Liberties must be tracked since it is the only means to ensure valid capturing of stones. Group tracking is need for this since liberties apply to a group of stones, and not

only to a single stone. The method I used for this is if a new stone is place, check to see if there are same colored stones in the four neighbors. If so, add it to that group. If the new stone is a connecting stone for two or more groups, retract all groups and create a new group for all stones connected. The liberties of the group will be slots within the group and will be done by pattern matching to the facts of the stones played already. The groups and liberties must be retracted if the groups are captured.

5.6 Capturing System

The capturing system engine starts after a valid position is played. First it checks all 4 neighboring opponent's stone groups. If after placement of the new stone causes the opponent's group(s) to have no liberties left, remove (retract) the group(s) and add the captured stones to the player's inventory.

5.7 End game and Scoring

If both players pass, the game ends. At end of game, count enclosed points surrounded by the stones. But first, if there are stones inside a territory, then it is removed and is added to the player's captured stone inventory. After this is done, add the captured stones to the total enclosed points. This will be the score for the player. The player with the most points is the winner. However, as mentioned in project scope, this scoring feature is much harder to do than these simple words. Thus, the scoring system is not fully implemented yet.

5.8 Knowledge Verification

I have done much testing on this to both during development and afterwards to verify that it is working according to the rules of the game. After I am satisfied with all the extra features I may add later, I will give this out to my friends whom I have introduced the game to. They will be glad to “test” it out for me since they are real beginners to the game.

6. User Instruction

To run the GO expert, just enter (batch "go.bat") within the CLIPS environment. It will provide the user with additional help information. The same information will also be provided in GoExpert.txt. In addition, if the user does not have a GO board and partner to play with, the SmartGo board and other GO programs will be of help. The SmartGo is a program that just features the GO board and does not provide you with a computer player. This is really helpful to beginners who is just trying to understand the rules and is not ready to play with the computer player yet. When the user is ready to move on, there are other recommended GO programs that feature a computer player. Just be sure to select a 9x9 board from the menu. There links are provided in the reference section.

7. Conclusion/Suggestions for Future Developments

The first needed development is to finish the scoring system. The second addition to the system will be to incorporate hints to the user. If this is done, not only will this expert serve as a rules and scoring expert, but also as a GO teacher that teaches step by step dynamically. This will be truly something new since I have found none on the

internet. However, this will only be a beginner's teacher since the engine of the system is just rules. Specifically, it can keep track and warn groups that have only 1 or 2 liberties left. It can store the most popular start game moves and also some general tips. The third addition will be to add facts about the 13x13 and 19x19 boards and let the user select among them. One last addition is to provide a graphical interface. SmartGo is a good one but it seems like only interface for C++ is supported. There may be ways to get around this, I have to dig further. With these additions, maybe my GO expert may even be good enough to be distributed on the web.

8. References

Way To Go, Karl Baker, The American Go Association, <http://www.usgo.org>

SmartGo, <http://www.smartgo.com>

Other computer Go programs, <http://www.usgo.org/resources/computer.asp>