Strategies Explained

Ian Frank
Complex Games Lab
Electrotechnical Laboratory
Umezono 1-1-4, Tsukuba
Ibaraki, JAPAN 305
ianf@etl.go.jp

David Basin
Institut für Informatik
Universität Freiburg
Am Flughafen 17
Freiburg, Germany
basin@informatik.uni-freiburg.de

Abstract

For many problem-solving tasks, it is important not only to produce solutions, but also to be able to explain them. In this paper, we describe how we are addressing this question within the framework of our computer game-playing research.

We build on the computer Bridge system FINESSE, which finds optimal lines of play for single-suit Bridge problems. To explain FINESSE's strategies, we developed an approach based on three distinct steps. First, we identify from a strategy the possible sequences of MAX plays that needs to be explained. Second, we remove from consideration move sequences or game situations that would be considered too simple to explain to human players. Third, we produce natural English text with the aid of both game-general and game-specific patterns and idioms that can explain each MAX and MIN move. We explain each of these three steps, and demonstrate the effectiveness of the overall approach by comparing automatically generated explanations against those found in an expert Bridge text.

1 Introduction

"Tell me more..." is the familiar refrain of the ELIZA program. But with most computer programs the role is typically reversed: it is the user that wants more information from the computer. In the case of computer games, this problem can be overlooked.

How can we add the automatic generation of explanations to games? In this paper, we discuss our preliminary experiences with this question in the game of Bridge. Specifically, we describe our progress in automating the English explanation of computer-generated Bridge strategies. We detail our solution to this problem, and show that the quality of the resulting explanations is well on the way to being comparable to that found in expert texts.

The strategies we attempt to explain are generated by the Bridge system, FINESSE [Frank, 1996]. This system uses specialised search techniques [Frank and Basin, 1998a] to identify optimal strategies for single-suit Bridge problems. FINESSE's strategies are in the form of (possibly large) game trees that contain the feasible moves when a given single-suit problem is played to completion. FINESSE automatically identifies the optimal moves for one player (MAX) at each node of these trees. Such a specification of optimal moves is effectively equivalent to a strategy.

To succinctly explain a given strategy, we developed an approach with three distinct steps. The first of these steps is to identify from a strategy the possible sequences of MAX plays that needs to be explained. The second is to remove from consideration move sequences or game situations that would be considered too simple to explain to human players. Then, the third is to order the explanation and to produce natural English text with the aid of (game-general and game-specific) patterns and idioms.

The effectiveness of our explanations is to some extent a result of the high-level formalisation of 'tactics' employed by the FINESSE system. These tactics reduce the number of possible types of play in a single suit to seven distinct manoeuvres such as playing winners and finessing missing cards. Since FINESSE's game trees are constructed of tactics rather than simple card plays, part of the work of explanation is already done.

We should note that our original motivation for this tactic-based approach came from automated theorem proving. As in Bridge, high-level tactics are useful in theorem-proving because they prune the search space and also enable users to communicate with a planner in terms that are more meaningful. In theorem-proving, Alan Robinson has coined the following slogan to capture the requirement that a convincing proof should be understandable:

$$\textit{Proof} = \textit{Guarantee} + \textit{Explanation}.$$

When FINESSE's explanation capability is completed, it will be capable of producing 'proofs' of optimal strategies in this general sense. It will also satisfy the criteria for the solving of (part of) a game [Allis *et al.*, 1991]. That is, in addition to producing optimal play, it will also be capable of explaining its actions.

2 FINESSE's Game Trees

For a detailed description of Bridge, readers are referred to one of the excellent books on the subject (e.g, [Goren, 1986]). Here, we can give some idea of the type of game trees produced by FINESSE with the aid of an example. Consider the following situation:

♠ A Q

N
W E
S

♠ 2

When presenting single-suit problems such as this, it is conventionally assumed that South is the *declarer*. This means that South's partner, North, is the *dummy* and must place his cards on the table for all to see. The division of the remaining cards in the suit between East and West is unknown, and the task is to specify the optimal way for South to play the cards from both his own and hand and from North's. Typically, the goal is to win a certain number of *tricks*, or rounds of play.

For the simple example above, declarer can win one certain trick (assuming there is no *trump* suit) by playing the Ace. However, if this card is played immediately, the only chance of making two tricks is if one of the defenders holds the singleton King, so that it falls when the Ace is played. So, this play of *cashing* the Ace succeeds in winning two tricks in only two of the possible 2^{10} distributions of the outstanding cards.

A better return is offered by playing the two from the South hand. By covering whatever card West plays, declarer can expect to win two tricks whenever West holds the King — a 50/50 chance. This line of play is based on the elementary principle of card play that the best results can be obtained by forcing an opponent to play ahead of you. It is a typical example of a very standard manoeuvre called a *finesse*.

When planning a hand, human players will make use of their knowledge of commonly occurring patterns like the finesse to avoid having to consider all the possible combinations of plays of single cards. FINESSE attempts to replicate this capability by restricting declarer's options at each stage of the play to a pre-determined set of such manoeuvres, or *tactics*.

The current version of FINESSE has seven tactics[1], which it represents using the following Prolog predicates:

1 - 4. `finesse(Type, Player, Card, Suit)` — `Type` represents the type of finesse being used (four different types of finesse were identified when designing FINESSE; in the example above, the finesse of the Queen was a Type 1 finesse); `Player` is the defender being finessed; `Card` and `Suit` specify the finesse card.

5. `cash(Card, Suit)` — represents a trick on which declarer plays the card specified by `Card` and `Suit` from one hand, and plays a low card (or throws away a card from another suit) in the other.
6. `duck(Suit)` — represents a trick on which declarer plays low from both his hands.
7. `sequence(Card, Suit)` — represents a trick on which declarer plays the card specified by `Card` and `Suit` (which must come from a sequence of length 2 or more) from one hand, and plays low (or throws away a card from another suit) in the other.

FINESSE's tactics do not specify complete lines of play to be followed for any particular card combination, but only continuations for the next trick. Lines of play are built up by a search algorithm that constructs a tree of tactics resembling a minimax tree. To build this tree, FINESSE must be able to determine the tactics that are applicable in any state. This is achieved by specifying the minimal applicability preconditions for each tactic with a set of Prolog clauses of the form:

```
applicable(State, Tactic) :- PreConds.
```

To form a game tree, an `applicable(+State, -Tactic)` goal is used to find a `Tactic` applicable to the current `State`. The possible responses by the defence are then generated, and the post-conditions are determined. This process is continued recursively for the resulting states until branches for each applicable tactic have been generated. In general, there will be more than one tactic applicable to a particular state, and the defenders may also make a number of different responses, leading to trees of the structure illustrated in Figure 1. In this tree, the North and South cards are shown at

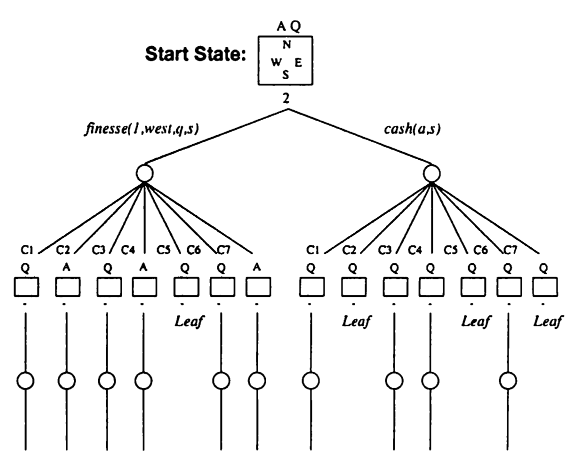

Figure 1: An example game tree

[1] To produce these tactics, we generalised from examples of card play we found in Bridge books. What is perhaps surprising is that just seven tactics were all that was needed. Although the tactics represent knowledge that human players usually learn through experience, we are not aware of any Bridge text that explicitly presents the required knowledge in such a compact and manageable way.

each non-terminal MAX node, and the MIN nodes are represented by circles. The labels C1 to C7 denote MIN's possible responses to MAX's first tactic as shown in Figure 2 ('discard' refers to playing a card from a different suit — allowed when the defender holds no cards in the suit that starts the trick). After the first trick, MAX has only one remaining card. If this can be cashed a further branch is generated.

Label	West's card	East's card
C1	King	discard
C2	low	discard
C3	discard	King
C4	low	King
C5	discard	low
C6	King	low
C7	low	low

Figure 2: Key for Figure 1

To choose between multiple branches at MAX nodes, FINESSE employs search techniques that model the dependencies between MAX nodes in the tree [Frank and Basin, 1998a]. Using these algorithms, FINESSE can automatically identify the best moves for MAX at each node of a tree with respect to a goal that can be a) winning the maximum possible number of tricks, b) winning a specified number of tricks, or c) producing the maximum expected return. In the example of Figure 1, all of these metrics result in the same choice of strategy: select the finesse branch at the root of the tree.

FINESSE also calculates further information about a strategy, such as its chance of success and the actual distributions for which it succeeds. For the purposes of this paper, however, we assume that the input to our explanation generation mechanism is in the form of a tree of tactics with the best choices at each MAX node marked as *selected*.

3 Collapsing Game Trees

The first step in explaining a game tree is to extract the branches that form part of the optimal strategy. We achieve this by *collapsing* the input game tree into a tree whose paths are a subset of the original.

The aim of this collapsing stage is to reduce the size of the tree that has to be explained. The obvious paths to exclude are those in which any of the MAX branches are not selected. However, we can make one further refinement. We noted above that the criterion for selecting the best branch at each MAX node can be a) winning the maximum possible number of tricks, b) winning a specified number of tricks, or c) producing the maximum expected return. Consider the first of these, and note that some paths in a tree — despite containing only selected MAX branches — may fail to lead to the maximum possible number of tricks. For example, when finessing in the example of Figure 1, the MIN branch C4 (corresponding to the Queen losing to East's King) leads to a continuation where declarer only wins one trick (with the Ace). Thus, when collapsing a tree in order to explain how to take the maximum possible number of tricks, we restrict the collapsed tree to contain only paths through the original for which :

- all the MAX branches are selected branches,
- the payoff at the leaf node is the maximum possible.

Similarly, when collapsing a tree in order to explain how to best guarantee winning a specified number of tricks, we include only paths that achieve or exceed this payoff at the leaf node. And, when collapsing a tree in order to explain how to produce the maximum expected return, we include only paths for which the payoff at the leaf nodes is higher than the *minimum* possible payoff for the problem.

Having narrowed the set of paths to be considered in this way, we can now describe our collapsing algorithm. This algorithm builds a new tree, $\mathcal{T}'$, from an input game tree $\mathcal{T}$ as follows. First a path p (meeting the requirements described above) is extracted[2] from $\mathcal{T}$. This path is added to $\mathcal{T}'$ by comparing branches, starting at the root. If the root of $\mathcal{T}'$ and p are MAX nodes, a 'match' requires the root of $\mathcal{T}'$ to have a branch labelled with the same tactic as the root node branch of p. If such a branch is found, a recursive call is made to combine the subtree of $\mathcal{T}'$ rooted on this branch with the remainder of p. For MIN nodes, a 'match' requires that there is a match for the daughter of the root node of p along some branch of the root of $\mathcal{T}'$; if such a branch exists its labels are combined with the label on the root branch of p, and a recursive call made to combine the subtree of $\mathcal{T}'$ rooted on this branch and the remainder of p. When the recursion reaches a point at which no match can be found, any remaining nodes in p are inserted into $\mathcal{T}'$.

Figure 3 gives an example of a collapsed tree produced by this algorithm. In fact, this is the tree produced by FINESSE for the example of Figure 1. The collapsed tree contains just two branches, which represent four paths through the original tree. The left-hand branch of the collapsed tree represents the two paths in the original where West plays the King, so that declarer wins with the Ace and cashes the Queen on the second round (MIN branches C1 and C6). The right-hand branch represents the two paths (C2 and C7) where West plays low and East either discards, or also plays a low card. It should already be clear that this collapsed tree is more amenable to explanation than the original of Figure 1.

[2]Often, a tree contains MAX nodes at which there is no single branch that offers a better return than all the others. Thus, there may be nodes where multiple branches are marked as selected. We currently treat such nodes by deterministically picking the tactic that is most commonly selected at the node's siblings. A future improvement would be to allow such multiple possibilities to be represented disjunctively.

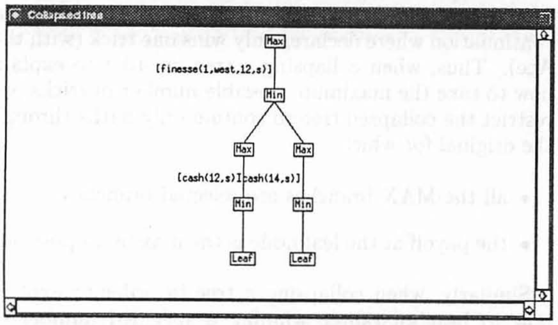

Figure 3: Screen capture of collapsed tree generated by FINESSE from the game tree for the AQ-2 problem

4 Pruning Game Trees

Some parts of a strategy may be so straightforward that no human player above the level of a beginner would expect an explanation. The pruning step of our approach identifies and removes such branches from collapsed trees. To do this, it uses the following two principles.

1. It is not necessary to explain how to proceed against *all* the possible moves that the opponent might make. In particular, some MIN branches may correspond to particularly bad moves. Such branches are pruned.
2. It is not necessary to explain how to proceed in *all* of the game situations that can arise when following the strategy. In particular, some MIN branches may leave the game in situations that are particularly easy to play. Such branches are pruned.

Concepts like "particularly bad moves" and "particularly easy to play" are clearly game-specific. For example, in chess a particularly bad move might be to place a Queen on an unguarded square, allowing it to be captured at no cost by the opponent. Or, a situation that is particularly easy to play might be an endgame position such as that shown in Figure 4. Here, White's pawns can easily promote to Queens, irrespective of Black's moves. (In fact, Black's best strategy also requires no explanation: resign).

However, we attempted to avoid game-specific pruning techniques, instead relying on the two general pruning strategies described below. What is left after these two pruning steps is a tree containing the "interesting" parts of the original strategy.

4.1 Identifying Bad Moves

One way of identifying bad moves is by comparing them to other possible moves in the same situation. In general game trees, this could be done by examining the heuristic evaluations of sibling nodes and discarding any that are clearly inferior. However, our game trees (at least as described in §2) do not have evaluations. Rather, each

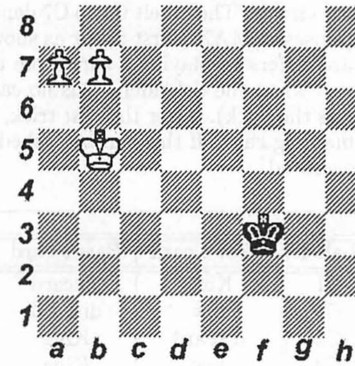

Figure 4: A chess position requiring no explanation of how to win

MIN branch can only be followed under certain distributions of the outstanding cards. We therefore find bad moves by looking for branches that should *never* be followed by the defenders. Such branches can be identified by checking that for every possible distribution, there is a different path through the tree that offers a better return for the defence.

To give an example of what we mean here, let us return to consider the example tree of Figure 1. We did not draw attention to this before, but in this tree there are actually not four paths that lead to two tricks, but five. The extra path is one where MAX finesses, and West discards from another suit with East playing the King (branch C3). If the first defender discards in this way when a finesse is attempted, the tactic is doomed to fail (since all the remaining cards in the suit are held by the other defender). FINESSE's rule-base therefore tries to win the trick if possible in such circumstances. In this example, this means aborting the finesse attempt after seeing West's play and playing the Ace from the North hand on the first trick. For East to play the King under North's Ace is clearly giving away a free trick to the Queen, so branch C3 should never be followed by the defence. In practice, FINESSE identifies such branches via the *payoff-reduction* step of the payoff-reduction minimaxing algorithm [Frank *et al.*, 1998].

4.2 Identifying Easy Situations

One way of characterising an "easy situation" is as a game state where there is a reasonable and straightforward way of achieving the desired goal (a win, say). We use one game-general and one Bridge-specific heuristic to identify such situations. The game-general heuristic we use is:

- If each branch of a MIN node leads to a linear tree, the MIN node is pruned.

The justification for this pruning is that a linear tree implies that MAX's best moves are not affected by the responses made by MIN: whatever moves MIN makes, MAX's best strategy is the same. In Bridge, such situations are among the easiest possibilities to play. For example, the top-most MIN node in the example of Figure 3 would be pruned by this rule.

We expect that this game-general heuristic will be useful in other games as well as Bridge. To aid in our specific attempts at producing Bridge explanations, however, we also use a similar heuristic tailored to consider a specific Bridge tactic:

- If any subtree of a MIN node contains only cash tactics, the subtree is pruned.

This heuristic is justified by noting that cashing a top card is the simplest possible tactic. Without this pruning step, FINESSE would generate explanations with long strings such as "then cash the ten and the nine and the eight" at the end of many sentences. The reader of FINESSE's explanations is expected to bear in mind that if no explicit direction is given, the default action is to cash a top card.

5 Pattern Matching

Pattern matching is the final step in our approach to automatically explaining strategies. It is this step that is responsible for mapping the branches of a collapsed and pruned tree into English text. Most of the basic operations carried out in the pattern matching phase are game-general:

- For a tree such as that of Figure 5, the natural explanation is not "Do tactic T1 then whatever MIN does, do tactic T2,...". Much better is "Do tactics T1 and T2,...". We incorporate a sequentialisation step to compile down such linear sequences in trees.

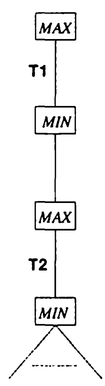

Figure 5: A linear sequence in a tree

- MIN branches (as discussed in the introduction of the collapsing algorithm) can have multiple labels. At MIN nodes with more than one branch, the branches are ordered so that the branch with fewest labels comes first. This allows the final (longest) branch label to be explained as "otherwise...".
- As a special case of sequentialisation, a tree may contain a linear sequence where T1 and T2 are the same tactic. This is rendered as "Do tactic T1 and repeat if necessary".

We implemented a simple recursive function for collapsed and pruned game trees that incorporates the above operations. The basic action is very simple: if the leaf node is a MAX node, explain the tactic, and if the leaf node is a MIN node, explain each branch. To produce formatting, newlines are added after each MAX tactic, and the text describing the nth MIN level is indented n units.

Limited Bridge knowledge is required to explain the tactics, since they already correspond directly to Bridge plays that humans understand. It is also relatively simple to describe the MIN plays in terms of the cards they represent. Only one further piece of Bridge-specific knowledge is required. If collapsing a game tree produces a linear tree, the tree can only contain cash tactics. We explain such trees as simply "cash top honours".

6 Performance and Commentary

In this section, we present and comment on a number of examples of system output. We take as our problem source the established and authoritative Official Encyclopedia of Bridge [ACBL, 1994], which contains a section detailing the solutions to over 650 single-suit problems. Each of the boxes below gives a problem from the Encyclopedia, along with its number and the Encyclopedia's and FINESSE's explanations (in each case, the strategy FINESSE explains is equivalent to the Encyclopedia's strategy). FINESSE's explanations are reproduced in typewriter font. We concentrate on just the explanations of the line of play producing the maximum number of tricks (similar results hold for lines of play guaranteeing fewer tricks, or the maximum expected number of tricks). Note that an 'x' signifies an arbitrary low card.

#	Cards (N-S)	Max Tricks
1	AKQJ9-x	5

Cash top honors in the hope of dropping the ten
`cash top honours`

The Encyclopedia solutions often add to the description of a strategy by describing the important card distributions under which the strategy brings a reward. FINESSE is also capable of generating textual explanations of the conditions under which strategies succeed (see [Frank, 1996]). Incorporating such information will be a useful improvement to future versions of FINESSE's explanation generation mechanism.

#	Cards (N-S)	Max Tricks
2	AKQJ9x-x	6

Cash top honors
`cash top honours`

#	Cards (N-S)	Max Tricks
3	AKQJ9-xx	5

Cash top honors in the hope of dropping the ten

```
cash the Ace
 if East shows out finesse the nine
```

#	Cards (N-S)	Max Tricks
4	AKQ9x-Jx	5

Cash top honors in the hope of dropping the ten

```
cash the Jack
 if East shows out finesse the nine
```

In the previous two problems, FINESSE concentrates on what to do when East discards from another suit on the first round, rather than the larger picture of dropping the ten. However, this is acceptable, since the default action to take when FINESSE's explanations give no explicit direction is to cash a top card.

#	Cards (N-S)	Max Tricks
5	AKQJ8-xx	5

Cash top honors. (But against defenders who would not falsecard from 109x or 109xx, cash the jack and finesse the eight if the nine or ten appears from East)

```
cash top honours
```

The Encyclopedia implicitly uses a model of *best defence* [Frank and Basin, 1998b] to assess Bridge strategies. However, sometimes it also gives alternative strategies — such as the one in parentheses above — that are designed to exploit weak play by East and West. Usually (as in this case), if the assumption of weak defence is incorrect, the chance of success of the alternative strategy will be lower than that of the 'best defence' strategy. FINESSE is currently not capable of relaxing the best defence assumptions to produce strategies that can take advantage sub-optimal defence; this is an open area for future research.

#	Cards (N-S)	Max Tricks
6	AKQ8x-Jx	5

See (5) above

```
cash top honours
```

#	Cards (N-S)	Max Tricks
7	AKQ8x-Jxx	5

Cash the jack first in case East is void

```
cash the Jack
 if East shows out finesse the eight
  and repeat if necessary
```

Above, FINESSE's explanation gives explicit detail on what to do when East is void. This example also illustrates the recognition of a repeated tactic.

#	Cards (N-S)	Max Tricks
8	AKQT-x	4

Finesse the ten

```
finesse the ten
```

#	Cards (N-S)	Max Tricks
9	AKQT9-x	5

Play off the top honors. This is fractionally better than the immediate finesse

```
cash top honours
```

In problem #9, the Encyclopedia contrasts the chance of success of the best line of play against the chance of success of the nearest competitor. Although currently not utilised, FINESSE also has all the information required to produce such contrasts. In fact, FINESSE's use of tactics simplifies the task of looking for sub-optimal strategies. In this case, for instance, "the immediate finesse" is simply the second MAX branch at the root of FINESSE's game tree.

#	Cards (N-S)	Max Tricks
10	AKQTx-x	5

Finesse the ten

```
finesse the ten
```

#	Cards (N-S)	Max Tricks
11	AKQT-xx	4

Cash the queen, and then finesse the ten

```
cash the Ace
 if both play low or East shows out finesse the 10
```

#	Cards (N-S)	Max Tricks
12	AKQ9-xx	4

Finesse the nine: hope that West has both the jack and ten

```
finesse the nine
 and repeat if necessary
```

#	Cards (N-S)	Max Tricks
13	AKTx-Qx	4

Cash the queen, and then finesse the ten

```
cash the Queen
 if both play low or East shows out finesse the 10
```

The above three examples all show FINESSE explaining the same strategies as the Encyclopedia in slightly different terms. The differences are simply a matter of style; any Bridge player will understand either explanation equally well.

#	Cards (N-S)	Max Tricks
14	AK9x-Qx	4

Play off the queen, king, and ace, hoping that the jack and ten fall in three rounds. (But against defenders who would not falsecard from J10x, cash the queen and finesse the nine if East drops an honor)

```
cash top honours
```

#	Cards (N-S)	Max Tricks
15	AKQT9x-x	6

Cash the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
16	AKQTxx-x	6

Cash the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
17	AKQT9-xx	5

Play off the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
18	AKQTx-xx	5

Play off the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
19	AKQ9x-xx	5

Play off the top honors, hoping that the jack and ten drop in three rounds

```
cash top honours
```

#	Cards (N-S)	Max Tricks
20	AKT9x-Qx	5

Play off the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
21	AKTxx-Qx	5

Play off the top honors

```
cash top honours
```

Problems #14 to #21 illustrate an important feature of explaining multiple similar problems in sequence: variation. In each of these problems, the basic line is to cash the top cards. Rather than simply stating this, however, the Encyclopedia sometimes adds detail (problem #19), sometimes gives lines of play that take advantage of weak defence (problem #14), and sometimes varies the language used ("Play off the top honors" and "cash the top honors"). If FINESSE were ever to be used to generate a database of explanations, such variations would be worthwhile considering.

#	Cards (N-S)	Max Tricks
22	AK9xx-QT	5

Cash the queen, king and ace

```
cash the Queen
```

In the above example, it is important to begin with the cashing of the Queen, since this allows 5 tricks to be won even when the Jack is singleton. Both the Encyclopedia and FINESSE explicitly state that the Queen should be cashed first, but neither state why.

#	Cards (N-S)	Max Tricks
23	AKxxx-QT	5

Finesse the ten

```
finesse the ten
```

#	Cards (N-S)	Max Tricks
24	AK9xx-Qx	5

Play off the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
25	AKQT-xxx	4

Cash the king and queen; if both follow, play the ace. This is 2% better than a third-round finesse

```
cash the Ace and King
 if East shows out finesse the ten
```

Problem #25 shows an example of sequentialisation. Note that FINESSE also concentrates on giving explanations of when to finesse, rather than when to cash, since cash branches are specifically removed by the pruning heuristics.

#	Cards (N-S)	Max Tricks
26	AKQ9-xxx	4

Cash the queen and king; if an honor drops from East, finesse the nine next. This is 6% better than cashing the three top honours regardless

```
cash the Ace
 if East shows out finesse the nine
  and repeat if necessary
 otherwise, cash the King
   if East shows out or plays the Jack or ten
finesse the nine
```

#	Cards (N-S)	Max Tricks
27	AKTx-Qxx	4

See (25) above

```
cash the Ace
 if both follow low cash the Queen
  if East shows out finesse the ten
 if East shows out finesse the ten
```

By referring to a previous example, problem #27 introduces another way of varying the presentation style: identifying examples with similar solutions. In this case, the optimal strategies are not trivially recognised as being equal, since in one problem the King and Queen are in the same hand, and in the other they are separated.

#	Cards (N-S)	Max Tricks
28	AK9x-Qxx	4

See (26) above

```
cash the Queen
 if East shows out finesse the nine
  and repeat if necessary
 otherwise cash the Ace
   if East shows out or plays the Jack or ten
finesse the nine
```

#	Cards (N-S)	Max Tricks
29	AKxx-QTx	4

Cash the ace, queen, and king. This is 4% better than a second-round finesse.

```
cash the Ace
 if West shows out finesse the ten
```

#	Cards (N-S)	Max Tricks
30	ATxx-KQx	4

See (25) above

```
cash the King
 if both follow low cash the Queen
  if East shows out finesse the ten
 if East shows out finesse the ten
```

#	Cards (N-S)	Max Tricks
31	A9xx-KQx	4

See (26) above

```
cash the King and Queen
 if both follow low or East shows out or East
plays the Jack or ten, finesse the nine
```

Although the Encyclopedia relates Problem #31 to #26, it is now not possible to win 4 tricks when East is void. This difference changes FINESSE's explanation.

#	Cards (N-S)	Max Tricks
32	Axxx-KQT	4

Cash the king, queen, and ace. This is 4% better than a second-round finesse

```
cash the King
 if West shows out finesse the ten
```

#	Cards (N-S)	Max Tricks
33	AKQTxxx-x	7

Play off the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
34	AKQ9xx-xx	6

Play off the top honors. (But against defender who would not falsecard from J10x, cash the ace and finesse the nine if an honor appears from East)

```
cash top honours
```

#	Cards (N-S)	Max Tricks
35	AKQ8xx-Tx	6

Play off the top honors

```
cash the Ace
```

#	Cards (N-S)	Max Tricks
36	KQ9xxx-Ax	6

Play off the top honors

```
cash top honours
```

#	Cards (N-S)	Max Tricks
37	AK9xx-Q8x	5

Cash the ace and queen (or queen and ace) hoping for a 3-2 break or a singleton honor with East. (But against defenders who would not falsecard with J10x, cash the ace, and finesse the eight if West drops an honor)

```
cash the Ace and Queen
 if East shows out finesse the nine
```

#	Cards (N-S)	Max Tricks
39	AK9x-Q8xx	4

Cash the ace. If an honor appears, cash the next top honor from the hand on the left of the Jack or ten.

```
cash the Ace
 if East plays the Jack or ten, cash the Queen
  if East shows out finesse the nine
 otherwise, cash the King
  if West shows out finesse the eight
```

For problem #39, FINESSE is not capable of replicating the complexity of the Encyclopedia's explanation, yet produces an equivalent description that is easy to understand.

#	Cards (N-S)	Max Tricks
62	AK98x-Jx	5

Run the jack *or* lead small to the nine

```
run the jack and finesse the eight
```

We finish with problem #62, since it illustrates one of the shortcomings of our current implementation. Whereas the Encyclopedia gives a choice of two lines of play with an equal chance of success, FINESSE just deterministically commits to explaining one line of play.

7 Conclusions and Further Work

We have described our progress in automating the English explanation of computer-generated Bridge strategies. We detailed the three steps of collapsing, pruning, and pattern matching a game tree, and gave examples of explanations generated by these techniques. For future work, we identified the following possible directions:

- Include descriptions of card distributions under which strategies succeed.
- Vary the presentation and terminology of explanations when giving multiple explanations of similar problems.
- If there is a slightly inferior way to play a given problem, include a comparison for contrast.
- Allow for disjunction, so that multiple equivalent strategies can be explained.
- Identify and give pointers to problems with similar solutions to the one currently under consideration.
- Investigate how to generate strategies that can take advantage of weak defence.

Let us end by emphasising that although our explanation techniques take advantage of the tactic-based representation found in FINESSE, this is not a pre-requisite. The trees of tactics produced by FINESSE are simply a compact representation of a subset of the space of possible moves. Any game tree can be converted into such a tactic tree if the specification of a set of relevant tactics for the game can be produced.

References

[ACBL, 1994] ACBL. *The Official Encyclopedia of Bridge.* American Contract Bridge League, Inc., 2990 Airways Boulevard, Memphis, Tennessee 38116-3875, fifth edition, 1994.

[Allis *et al.*, 1991] L.V. Allis, H.J. van den Herik, and I.S. Herschberg. Which games will survive? In D.N.L. Levy and D.F. Beal, editors, *Heuristic Programming in Artificial Intelligence 2 - The Second Computer Olympiad*, pages 232–243. Ellis Horwood, 1991.

[Frank and Basin, 1998a] I. Frank and D. Basin. Optimal play against best defence: Complexity and heuristics. In H.J. van den Herik and H. Iida, editors, *Proceedings of CG98*, pages 50–73, Tsukuba, Japan, 1998. Springer LNCS 1558, ISBN 3-540-65766-5. Longer version to appear in *Theoretical Computer Science.*

[Frank and Basin, 1998b] I. Frank and D. Basin. Search in games with incomplete information: A case study using bridge card play. *Artificial Intelligence*, 100(1–2):87–123, 1998.

[Frank *et al.*, 1998] I. Frank, D. Basin, and H. Matsubara. Finding optimal strategies for imperfect information games. In *Proceedings of AAAI-98*, pages 500–507, 1998.

[Frank, 1996] I. Frank. *Search and Planning under Incomplete Information: A Study using Bridge Card Play.* PhD thesis, Department of Artificial Intelligence, Edinburgh, 1996. Also published by Springer-Verlag in the Distinguished Dissertations Series, ISBN 3-540-76257-4, 1998.

[Goren, 1986] Charles H. Goren. *Goren's New Bridge Complete.* Century Hutchinson Limited, 1986.