

0. まえがき	2
a) 動作環境	2
b) 仕様上の制限	2
c) 特殊な機能	2
1. 取り扱うルール	2
2. 使用方法	3
(1) コマンドプロンプトからの使用	3
(2) Kifu for Windows からの起動	3
(3) プログラムの実行と出力	6
3. 詰将棋情報の設定内容	8
4. 特殊な局面指定	11
5. 動作の概要	12
6. 出力ファイル	15
(1) 検討結果の出力	15
(2) 証明木の出力	16
(3) 現探索情報の出力	18
7. 中断・再開・途中経過表示	20
(1) 中断	20
(2) 再開	21
(3) 途中経過表示	22
(4) 自動待避	23
8. 本プログラムで扱うルールに関する補足説明	23
(1) 普通の詰将棋と「最善詰」の違い	23
(2) 千日手の扱い	24
(3) 左右対称の扱い	25
(4) 自動で付加される要素	25
(5) 「長躯」条件について	26
(6) 「打歩」条件について	26
(7) 同時指定の制限	27
9. 各種検討支援機能	27
(1) 手順指定	27
(2) 余詰検討	33
(3) 作意強制 (※仮実装)	34
(4) 不詰出力 (※仮実装)	37
10. トレース、モニタ機能と本プログラムの動作原理	40
(1) トレース、モニタ機能について	40
(2) モニタの表示・非表示の切り替えと深度の変更	43
11. 性能の限界	44
12. 連絡先	44
99. 更新履歴	44

0. まえがき

SnifferDogは対抗系（協力系でない）ルールの変則詰将棋を検討するために作成したプログラムです。最悪詰を検討するため2007年より開発を続けてきたWorst1.exeは、最善詰や自玉詰等にも適用範囲を広げ、名称と機能が乖離していました。そこで今回、名称を改めたのが本プログラムです。使い方や機能はWorst1.exeと基本的に変わりません。SnifferDogの名称は「探知犬」の意味で、作品の問題点（余詰・早詰・変同等）を嗅ぎつける働きに期待しての命名です。開発における最重点項目は「全検」。時間が掛かってもすべての変化・紛れを検討できるよう、中断・再開の機能を備えています。

本ソフトウェアはフリーソフトとして配布します。無保証であることをご承知の上でお使いください。再配布も自由に行って構いませんが、必ず本説明書を添付してください。

【注意】 本プログラムは検討開始時に無条件に約30MBのメモリを消費します。

また、使用時間の増加と共に追加のメモリも必要とします。

使用時にはメモリ残量にご注意ください。

a) 動作環境

本プログラムはWindows 10以降で使われることを想定しています。

64bit版と32bit版があり、64bit版は32bit版より多くのメモリを使うことができます。

b) 仕様上の制限

本プログラムでは1手から2999手が仕様上取り扱える範囲です。

また、「連続王手の千日手の禁止」というルールは採用していません。

従って、それを利用した作は解けません。

c) 特殊な機能

本プログラムは検討や創作に資するため、以下の機能を提供しています。

- ・「非標準駒数」をサポートしています。
- ・「不透過・不可侵」のマスを定義できます。
- ・「受先形式」（受方から開始する作）も検討できます。

1. 取り扱うルール

本プログラムは「最善詰」「最悪詰」を検討することができます。

また、以下のような項目を指定することで、様々なルールに変化させることができます。

- ・「自玉」指定を加えて、「詰」の対象を攻方とする
- ・達成すべき目標を「詰」ではなく、「スタイルメイト」や「千日手」（一般的な千日手の意味ではなく、初形に戻すという意味）等に変える
- ・「強欲」「禁欲」等、選べる着手を制限したり、拡張・変更したりする条件を加える

現時点で用意されているバリエーションを以下にまとめます。

論理	最善、最悪
対象	攻方、受方
目標	詰、スタイルメイト、勝（詰またはスタイルメイト）、千日手
条件	禁欲、強欲、マキシ、ミニ、長駆、打歩、非王手可
手番	双方、受方、攻方、受先

「手番」は特定の「条件」や「目標」と組み合わせて使用します。

詳しくは「3. 詰将棋情報の設定内容」の「分類」をご覧ください。

2. 使用方法

(1) コマンドプロンプトからの使用

本プログラムを適当なフォルダに置き、入力ファイルを kif 形式で作成し、コマンドプロンプトから以下の形式で入力ファイルを指定します。

C:¥>本プログラムへのパス 入力ファイルへのパス

例えば入力ファイル名を test1.kif とし、本プログラムと共に C:¥work というフォルダに置いた場合は、コマンドプロンプトを起動してから以下のように操作します。

```
Microsoft Windows [Version 10.0.19044.2364]
```

```
(c) Microsoft Corporation. All rights reserved.
```

```
C:¥Users¥七郎>cd ¥work
```

```
C:¥work>SnifferDog test1.kif
```

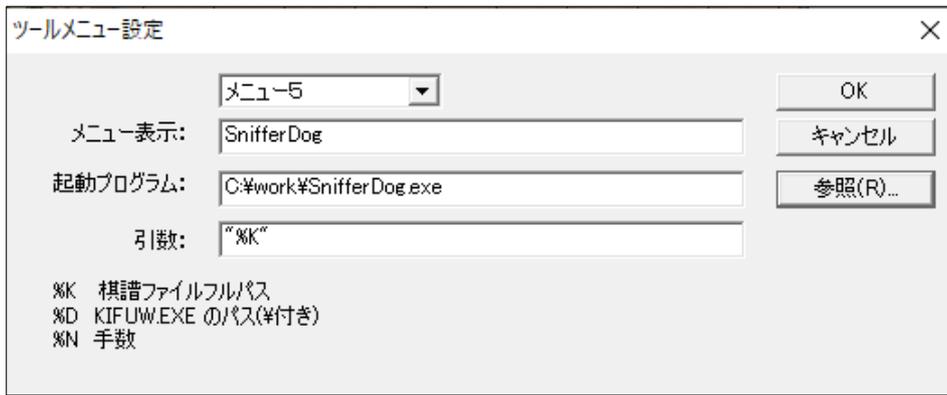
コマンドプロンプトを使うのが面倒な場合は、デスクトップ等にショートカットを作成し、入力ファイルを SnifferDog またはそのショートカットにドラッグ&ドロップしても良いでしょう。

(2) Kifu for Windows からの起動

もっと便利なのは柿木義一氏の作成した Kifuw ([柿木の将棋ソフトウェア](#)から入手してください) から本プログラムを呼び出せるようにすることです。



Kifuw のメニューから、[ツール]→[ツールメニュー設定]を選び、空いた場所を適当に選んで SnifferDog を使うよう設定します。以下は「メニュー5」に登録した例です。



「引数」は二重引用符 (") で囲ってください。

これで間にスペースが入るようなフォルダに棋譜ファイルが置いてあっても解析することができます。設定が終わったら実際に何か入力して試してみましょう。

ここでは「最悪自玉詰」でのサンプルを用いて説明します。

まずは、通常の詰将棋の編集と同様の手順で盤面・持駒を編集します。



次にメニューの[編集]→[詰将棋情報]を選んで、**手数**と**ルール**を設定します。

ルールは「分類」の項目に入れてください。

詰将棋情報

作品番号：

作品名：

作者：

発表誌：

発表年月：

出典：

手数：

完全性：

分類：

受賞：

備考：

OK

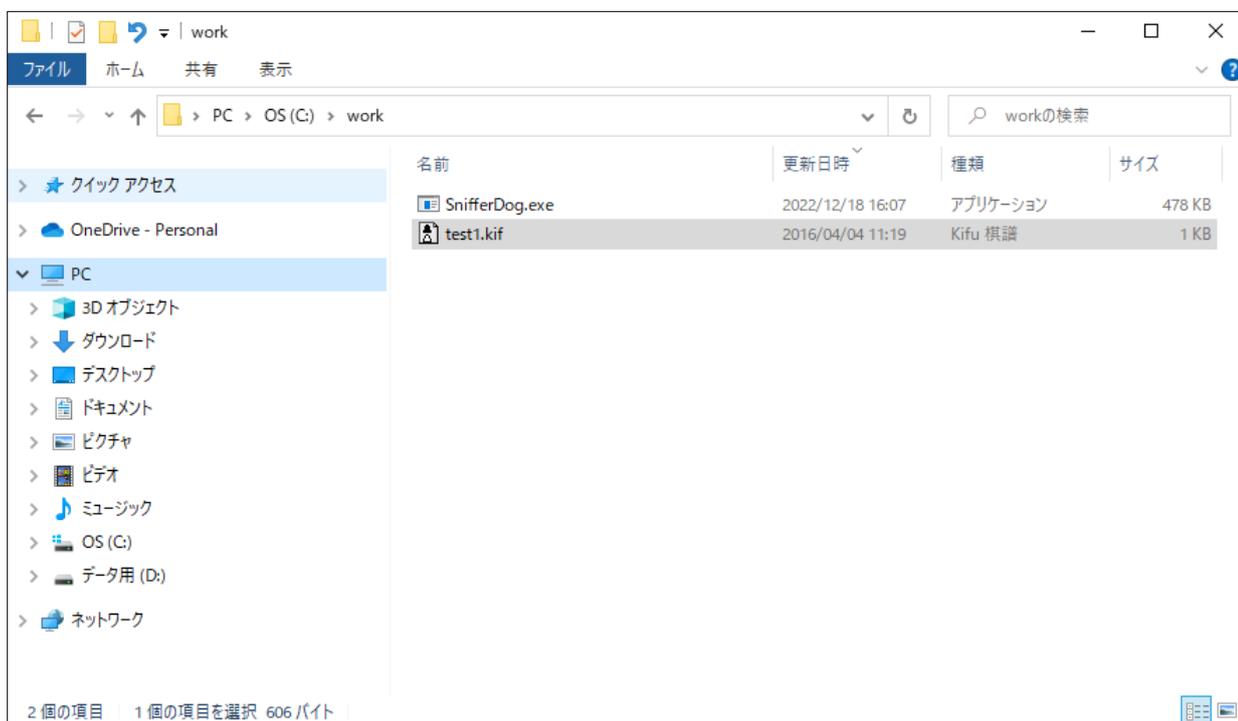
キャンセル

「手数」「分類」以外の項目の入力は任意です。

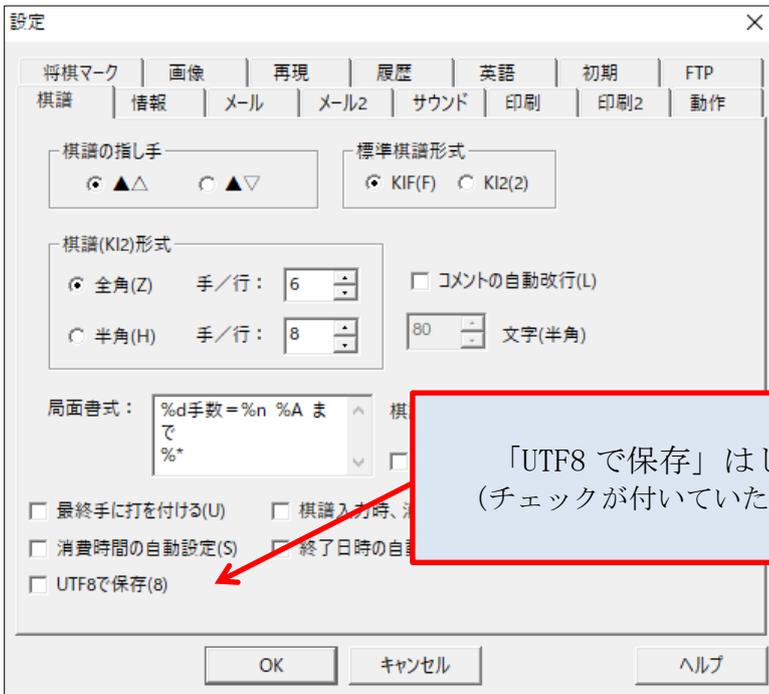
また、「作品名」「作者」「発表誌」「発表年月」の項目は出力時に1行にまとめて出力されます。

設定が終わったら、適当なファイル名で保存してください。

このときファイル形式は **kif** か **ki2** にしてください。 (※**bod** 形式だと「詰将棋情報」が保存されません)



【注意】 Kifuw V7系をお使いの方は、[設定]→[棋譜]の「UTF8で保存」のチェックを外してください。
SnifferDogは今のところUTF8に対応していません。

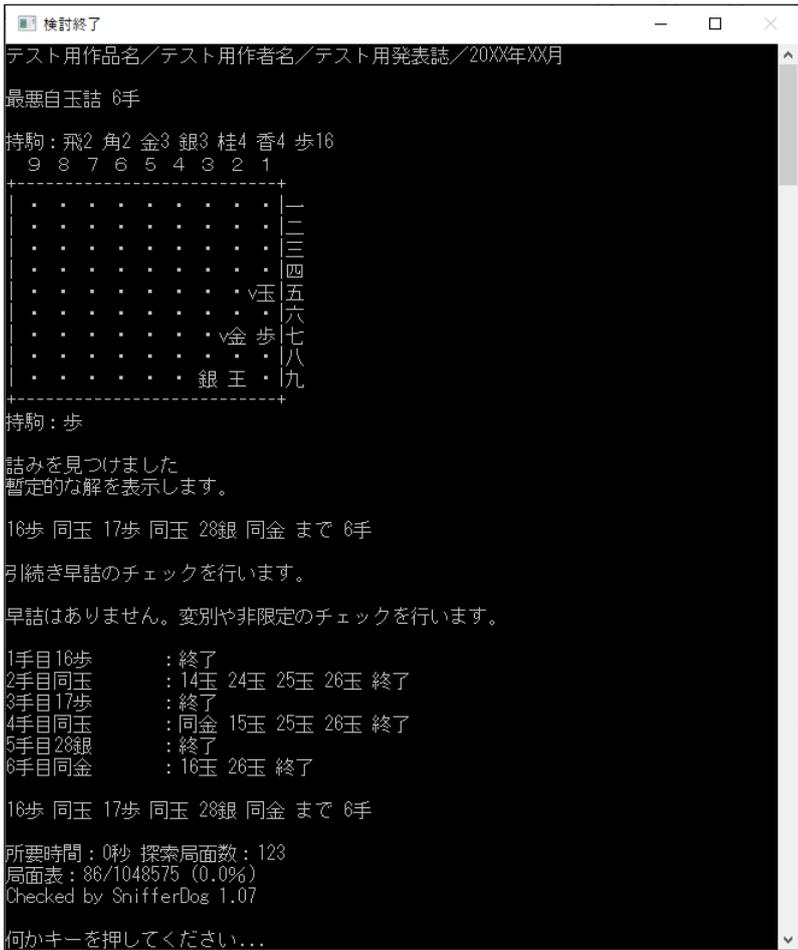


「UTF8で保存」はしない
(チェックが付いたら外す)

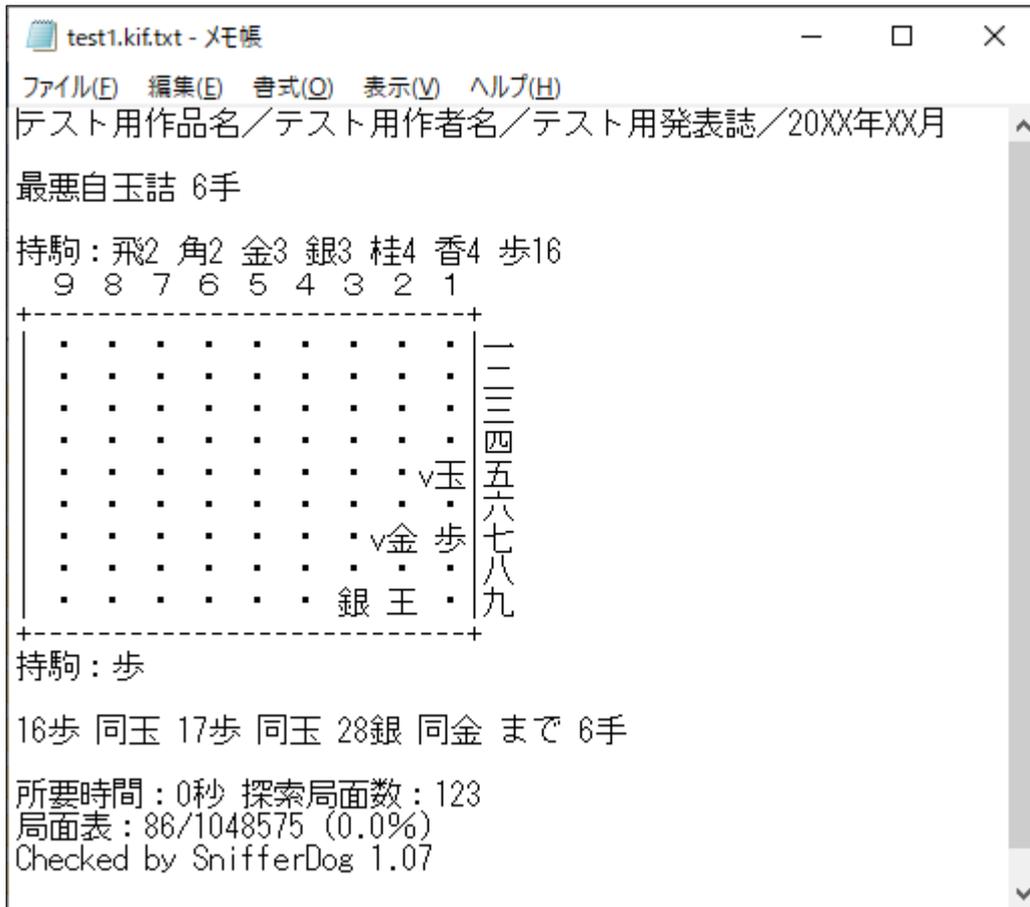
保存が終わったらメニューの[ツール]からツールメニューを選んで実行してください。

(3) プログラムの実行と出力

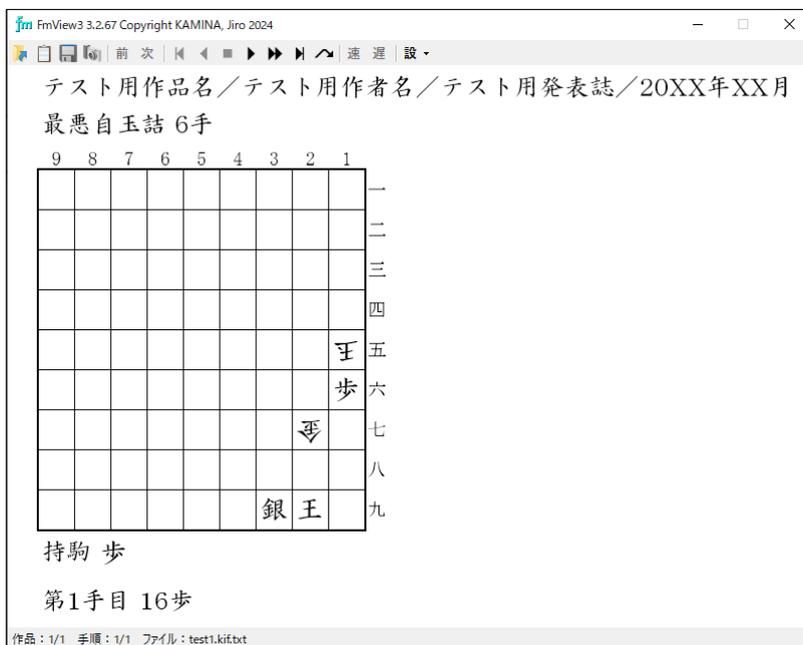
先程の例では「ツールメニュー5」に SnifferDog を登録したので、これを選びます。



解析が行われ、作意手順が検出されました。(上の例では見易くするため普通の画面より縦長にしています。) 「何かキーを押してください...」と表示されるので、出力内容を確認したら、Enter キーを押してください。Enter キーを押すとプログラムは終了し、ウィンドウが閉じます。結果は元のファイルがあったのと同じ場所に、元の名前に.txt という拡張子を付けて保管されています。既にファイルがある状態で再度解析をすると追記されます。出力内容は以下になります。



このファイルは fmview や FmView3 で閲覧可能です。



入力に使うソフトと閲覧に使うソフトが異なるのは変則的ですが、特殊なマスや非標準の駒数を使う場合に備えてこのような仕様になっています。

【注意】 余詰などを検出して配置を修正した場合は、必ずファイルの保存を行ってください。
保存せずに Kifuw 上の配置だけ変えても、検討対象に反映されません。

3. 詰将棋情報の設定内容

kif ファイルの「詰将棋情報」には次の情報を設定します。

項目	説明
手数	<p>手数の上限を指定します。1 から 2999 手が指定可能です。</p> <p>最悪詰、最善詰では偶数手のとき受先形式、(最悪・最善) 自玉詰系ルールでは奇数手のとき受先形式となります。</p>
分類	<p>ルールを指定します。ルールは〔論理〕〔対象〕〔目標〕〔条件〕〔手番〕の5つの項目を組み合わせて指定します。以下に各項目の概要と指定可能な要素について説明します。</p> <p>〔論理〕 攻方と受方が着手を選択する基準を指定します。本プログラムでは「最善」「最悪」が選べます。不詰（目標達成不可）は手数無限として扱います。</p> <p>【最善】 攻方はどう応じられても目標を達成できるよう王手を選び、受方はなるべくそれを妨げるよう応じる。攻方最短・受方最長手数の手順のみが正解。</p> <p>【最悪】 受方はどう王手されても目標を達成できるよう着手を選び、攻方はなるべくそれを妨げるよう王手する。攻方最長・受方最短手数の手順のみが正解。</p> <p>〔対象〕 〔目標〕を適用する対象を指定します。通常（無指定時）は受方が対象です。</p> <p>【自玉】 攻方を対象とする</p> <p>〔目標〕 達成すべき目標を指定します。通常は「詰」ですが、他の目標も指定できます。</p> <p>【詰】 相手番側に王手が掛かっており、合法的な応手がない状態</p> <p>【ステイルメイト】 相手番側に王手が掛かっておらず、合法的な応手がない状態</p> <p>【勝】 相手番側に合法的な応手がない状態（詰またはステイルメイト）</p> <p>【千日手】 初形と同一局面に戻った状態（手順中に同一局面が現れる一般的な「千日手」とは異なる）</p> <p>〔条件〕 条件を指定することで、選べる着手の制限・拡張・変更ができます。</p> <p>【禁欲】 駒を取らない手を優先して着手を選ぶ</p> <p>【強欲】 駒を取る手を優先して着手を選ぶ</p> <p>【マキシ】 受方は最長距離の応手を選ぶ。距離は将棋盤を9×9の正方格子とみなした時のユークリッド距離とする。駒を打つ手は距離1と定義する。</p> <p>【長軀】 受方は最長距離の応手を選ぶ。距離は将棋盤を9×9の正方格子とみなした時のチェビシェフ距離とする。駒を打つ手は距離1と定義する。</p> <p>【ミニ】 受方は最短距離の応手を選ぶ。距離は将棋盤を9×9の正方格子とみなした時のユークリッド距離とする。駒を打つ手は距離1と定義する。</p> <p>【打歩】 打歩詰以外での目標達成を失敗とする（単純打歩）。自玉詰系は指定不可。</p> <p>【非王手可】 攻方王手義務がない（王手をしていても良い）。「非連続王手」でも同じ。</p> <p>※注1 「禁欲」「強欲」「マキシ」「長軀」「ミニ」は互いに排他で、同時に指定できません。</p> <p>※注2 「非王手可」で目標が「詰」でないとき、玉の存在チェックは行われません。</p> <p>〔手番〕 条件を適用する手番や、初形をどちらの手番から開始するかを指定します。</p> <p>【双方】 条件を双方の手番に適用します。「禁欲」「強欲」のデフォルト解釈です。</p> <p>【受方】 条件を受方の手番のみに適用します。「マキシ」「長軀」「ミニ」のデフォルト解釈です。「打歩」には指定できません。</p> <p>【攻方】 条件を攻方の手番のみに適用します。</p> <p>【受先】 受方手番から開始します。目標が「千日手」の場合のみ有効。他は手数により自動的に判定されます。</p>

備考	<p>各種のオプションを指定します。複数指定するときはスペースで区切ってください。</p> <p>〔動作に関するオプション〕</p> <p>暫定的な解を求めた後の動作を指定するオプションです。現在用意されているのは「解図のみ」「略式検討」「余詰検討」「作意強制」「不詰出力」の5つです。</p> <p>解図のみ：</p> <p>「解図のみ」と記入すると、早詰や非限定・変別のチェックを省略します。変別解が出力される危険があるので、詰みの有無のみを調べるときに使います。</p> <p>略式検討：</p> <p>「略式検討」と記入すると早詰チェックを省略します。早詰がないことが確実な場合はこれで検討時間を短縮できます。</p> <p>余詰検討：</p> <p>「余詰検討」は最善系のルールで利用可能なオプションで、攻方の手番で早詰や非限定だけでなく、長手数之余詰も検出する場合に指定します。単に「余詰検討」とすると、ルールに応じ 2999 手または 2998 手以内の余詰検討が行われます。検討する手数を指定する場合は「余詰検討 99」のように、手数を表す数値を指定します。</p> <p>作意強制：(※仮実装)</p> <p>「手順指定」の一種で、変化手順をチェックせず、合法手である限り指定された手順に沿って検討を進めます。特定の変化・紛れを検討するときを使うことを想定しています。</p> <p>不詰出力：(※仮実装)</p> <p>不詰の場合に参考手順を出力する機能です。出力されるのはあくまで不詰解の一例です。</p> <p>〔求解に関するオプション〕</p> <p>問題によっては、使用者が条件を外挿した方が速く効率的に解ける場合があります。この種のオプションで現在用意しているのは「持駒節約」「対称性検出」の2つです。</p> <p>持駒節約：</p> <p>このオプションは盤面配置が同一で、手数が同じか大きく、持駒が真に少ない局面が攻方手番で現れた場合、読みを打ち切ります。最善系ルールで指定可能です。誤って指定すると詰むのに不詰となる場合があるので、慎重に使ってください。なお、条件の付かない「最善詰」は自動的にこれが適用されます。</p> <p>対称性検出：</p> <p>左右対称形からの左右対称解を無視するオプションです。途中左右対称もチェックの対象となります。※手順指定時、このオプションは無効化されます</p> <p>〔出力内容に関するオプション〕</p> <p>備考欄に「トレース」と記入すると、探索中の候補手を画面とファイルに出力します。デフォルトのトレース深度は 2 となっています。深度を変える場合は「トレース 4」のように、深度を表す数値を添えてください。「モニタ」は「トレース」と同様の機能ですが、出力先は画面のみです。「モニタ」は解析時に表示／非表示を切り替えることができるので、通常はこちらをお使いください。「トレース」はデバッグ用の機能であり、出力ファイルが巨大になるため、使用をお勧めしません。</p> <p>〔メモリに関するオプション〕</p> <p>局面表のサイズを「メモリ 100MB」「メモリ 1GB」のように指定します。1GB=1024MB です。数値は整数値のみが使えます。GB の単位を使う場合は、メモリ残量にくれぐれも注意してください。無指定時は 24MB と解釈されます。32bit 版では約 2000MB が OS から割り当てられるメモリの上限ですが、64bit 版ではより大きな値を指定することができます。解析中は局面表以外にもメモリを使います。マシンのメモリ残量に注意して、十分に余裕を持った値を指定してください。</p>
----	--

〔追加出力に関するオプション〕

追加の出力を行うオプションとして、現在「証明木作成」及び「最小証明木作成」の2つを用意しています。

証明木作成：

証明木の作成を指示するオプションです。

「証明木」は「詰」を証明するため、すべての変化の詰め方を網羅したものです。詳細は「6. 出力ファイル」「(2) 証明木の出力」をご覧ください。

最小証明木作成：

最小証明木の作成を指示するオプションです。

「最小証明木」は「証明木」の各変化手順の最短性を保証したものです。詳細は「6. 出力ファイル」「(2) 証明木の出力」をご覧ください。

自動待避：

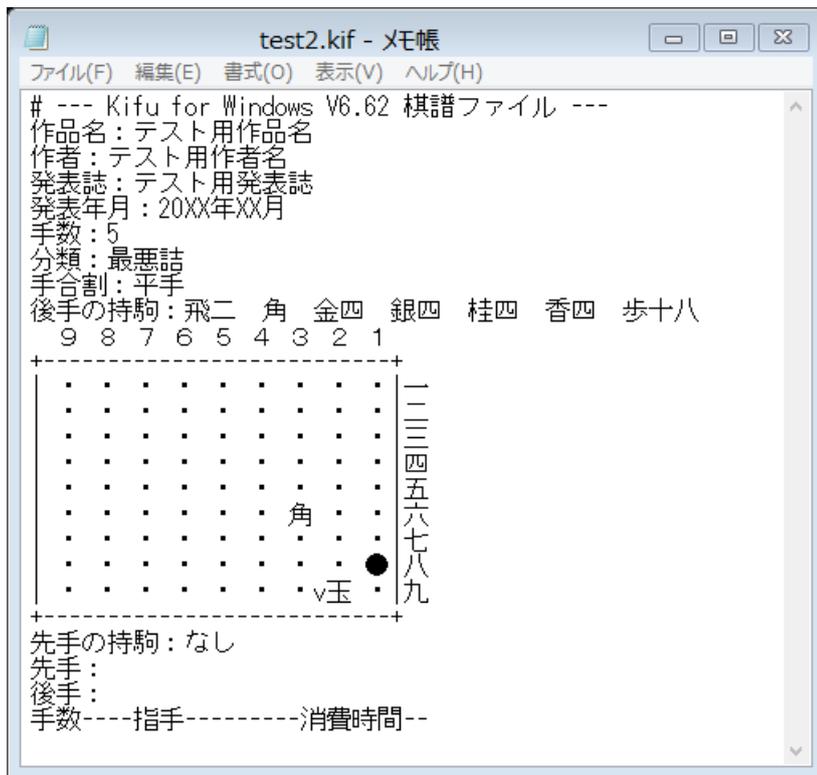
待避ファイル作成を指定された時間間隔で自動的に行います。単位は「時間」です。

例えば4時間ごとに自動待避を行う場合「自動待避4」と指定します。

4. 特殊な局面指定

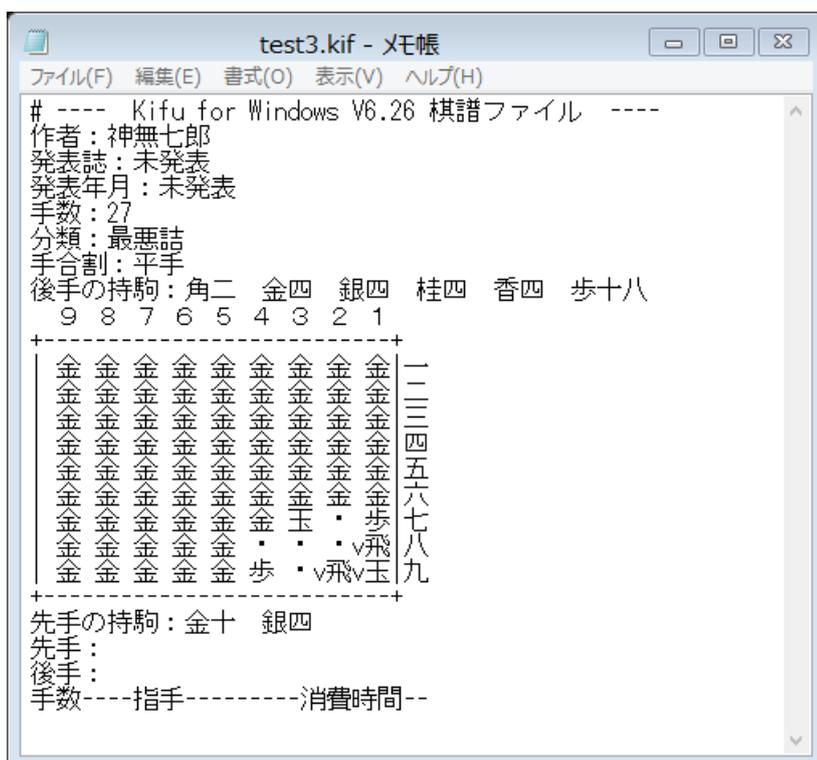
本プログラムでは f m の「石」と同様、不透過・不可侵のマスを指定することができます。(表記は●)
これは Kifuw では扱えませんので直接テキストファイルで指定してください。

例) 石 (●で表記) を使った例



また、本プログラムは非標準駒数にも対応しています。通常より駒数を多く使う問題が指定可能です。持駒に関しては 99 枚までという制限があります。枚数は漢数字で指定し、各持駒の右には全角空白を入れてください (この指定は kif 形式の書式に準じています)。盤上の駒数には特段の制限はありません。

例) 非標準駒数の例



5. 動作の概要

このプログラムは主に3段階の動作をします。

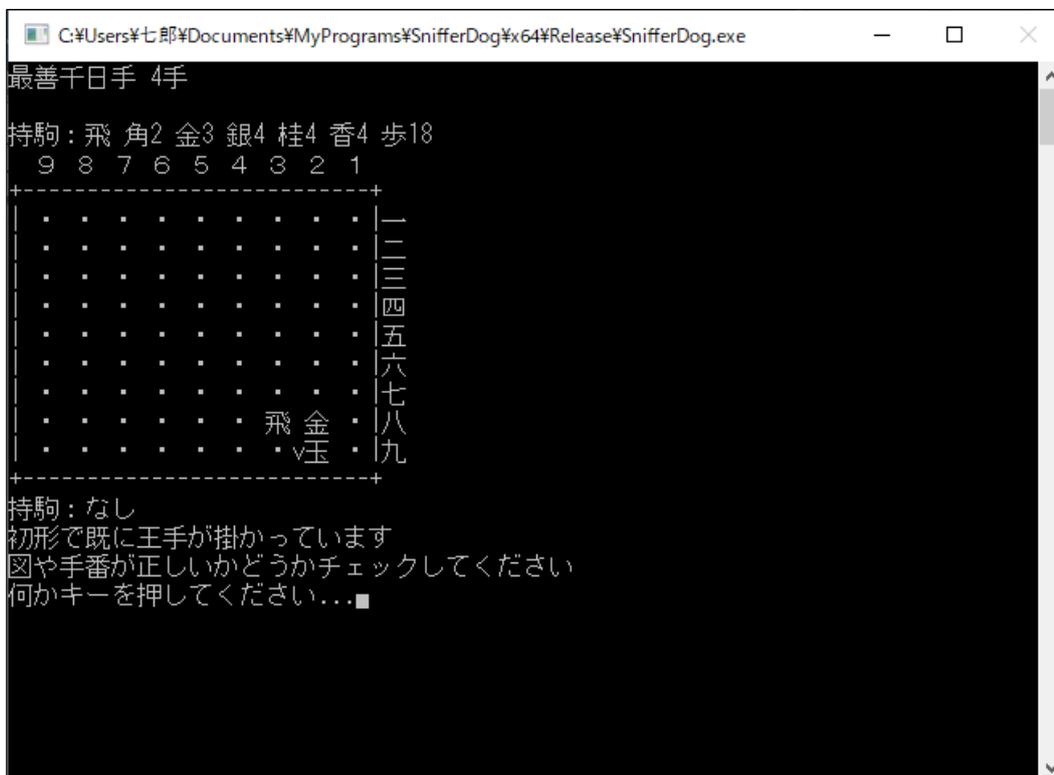
- a. 詰みを見つける
- b. 早詰がないことを確かめる
- c. 非限定・変別を検出する

本格的な動作の前に、簡単なチェック処理が行われます。

起動時には実機に搭載されている物理メモリの残量をチェックします。もし不足していた場合、次のようなメッセージが出るので、何かキーを押して一旦プログラムを終了し、メモリ残量を確認してください。

指定のメモリサイズが実際に使える値より大きくなっています
指定：6144MB, 使用可能：4838MB
何かキーを押してください...

メモリが確保できたら、入力データに対して最低限のチェックを行います。
以下は「受先最善千日手」を「最善千日手」で入力してしまった場合の例です。



入力データのチェックが無事に終わると、「詰」(与えられた目標が達成できるかどうか)の探索(a.の動作)が開始されます。動かし始めたらとにかく解が出るまで待つしかありません。途中で強制終了する場合は、Escキーの押下で終了してください。

「詰」を発見したときに次のようなメッセージが出てきます。

詰みを見つけました。
暫定的な解を表示します。

このメッセージの後に“暫定的な”解が表示されますが、これは変別を含む可能性があるため、鵜呑みにしてはいけません。ただし、詰みの有無のみを問題にしている場合は、ここで打ち切っても良いでしょう。

もし、早詰があれば次のようなメッセージと、検出された早詰解が出ます。

少なくとも 2 手短い早詰があります。
図や手数指定に誤りがないか確認してください。
暫定的な解を表示します。

この段階で出力された早詰解が最短である保証はないので、プログラムは最短解を求める作業に移ります。
最短解を求めず打ち切る場合は、**Esc** を入力して中断してください。

なお、指定手数で詰まない場合は、次のメッセージが出ます。

詰みませんでした、最大検討深度：XX 手

この場合、図や手数を再確認してやり直してください。

「最大検討深度」が指定手数未満の場合は、指定手数不足ではなく、変化が不詰の可能性がります。

(1.07 版より指定手数に達した局面が更に展開可能な場合は、展開した局面の深度も含めることにしたので、最大検討深度は通常「指定手数+1」になります。)

指定通りの手数の解が検出されれば、次のメッセージが出ます。

引続き早詰のチェックを行います。

これで早詰チェック (b.の動作) の段階に入り、最短解を求める作業に入ります。
早詰チェックが無事終了すると、次のようなメッセージが出てきます。

早詰はありません。変別や非限定のチェックを行います。

このメッセージの後は自動で c.の非限定・変別検出の段階に移行します。

早詰チェックに入る前の段階で既に早詰が見つかった場合や、早詰チェックで新たに早詰が見つかった場合は、次のようなメッセージが出ます。(XX は最短解の手数。)

最短解は XX 手でした。

早詰順 (暫定解) をもとに、変別や非限定の検討を行いますか? (Y/N) :

引続き変別や非限定の検討を行う場合は、「Y」または「y」を入力してください。ここで「N」または「n」を入力すると検討は終了し、暫定解を出力します。

後で変別や非限定の検討を行う場合は、待避ファイルを作成してください。待避ファイルの作成と、待避ファイルからの再開については「7. 中断・再開・途中経過表示」をご覧ください。

変別や非限定のチェック中は、暫定解の各手に対して検討を行っている状況が表示されます。
これは、画面のみでファイルには出力されません。

1 手目 16 歩 : 終了
2 手目 同玉 : 14 玉 24 玉 25 玉 26 玉 終了
:

まず何手目を検討しているかが表示されます。

変化に対してはまず変別のチェック (暫定解で選んだ手に対し早詰がないことの確認) が行われ、次に暫定解で選択されなかった手が早く詰むこと (同手数変化にならないこと) を確認します。

紛れに対しては暫定解以外の手で非限定 (余詰) がないかチェックします。その局面でのすべての選択肢に対する検討が終わると「終了」と表示されます。

変化の検討で変別を検出した際には、次のメッセージが出ます。

変別検出！ 代替手を探しています...

この後代替の手が見つかり、「手順修正」のメッセージが表示され、修正した手順を元に検討が続行されます。そのまま最後まで続行すれば、変別のない手順が出力されます。

非限定（余詰）が検出された場合は、画面とファイルの双方に出力されます。詳しくは「6. 出力ファイル」をご覧ください。

解析が終わった後は次のようなメッセージが出るので **Enter** キーを押下して終了してください。

所要時間：0 秒 探索局面数：869

局面表：526/1048575 (0.1%)

Checked by SnifferDog 1.07

何かキーを押してください...

解が出るまで待ち切れない時は、頭の手順をカットした図を解かせてみて、徐々に手数を増やすのも良いかもしれません。だんだん解析に時間が掛かるようになるので、最終的な所要時間がある程度予測できます。

また、詰めることはできても、早詰のチェック等に時間が掛かる場合は、備考欄に「解図のみ」と入れて、早詰のチェック以降の工程を省略する方法もあります。この場合、出力される解には変別が混じっている危険もありますが、とりあえず詰むかどうか、どんな手順で詰むかを確認したい場合に有効です。

動作中にメモリが足りなくなった場合は次のようなメッセージが出ることがあります。

メモリが不足気味です（局面表利用率 80.0%）

30.0%の非保護データ（探索レベル 3 以下）を破棄して再利用します

本プログラムでは検討中のデータを局面表に保存し、再利用することによって検討効率を上げていますが、問題が難しいと局面表が満杯に近くなることがあります。そのときに、一部のデータを破棄して、空いた領域を再利用します。

このメッセージが出た時は、できれば多めのメモリを割り当てて再度やり直してください。

このままでも解析は続けられますが、検討の効率は低下します。破棄対象データの探索レベルが一桁くらいなら余裕ですが、二桁を越えるとかなり効率が低下しており、状況が逼迫していると考えられます。

※探索レベルについての補足

「探索レベル」は対象局面の探索木の大きさを、2 を底とした対数の整数部で表したものです。例えば探索レベルが 3 の場合、2 の 3 乗、つまり 8 局面分位の探索を行ったことを示しています。つまり「探索レベル」が 1 上がるごとに倍のコストを掛けて得られた貴重なデータということになります。従って、なるべく小さな探索レベルのデータから再利用に回す方式になっています。

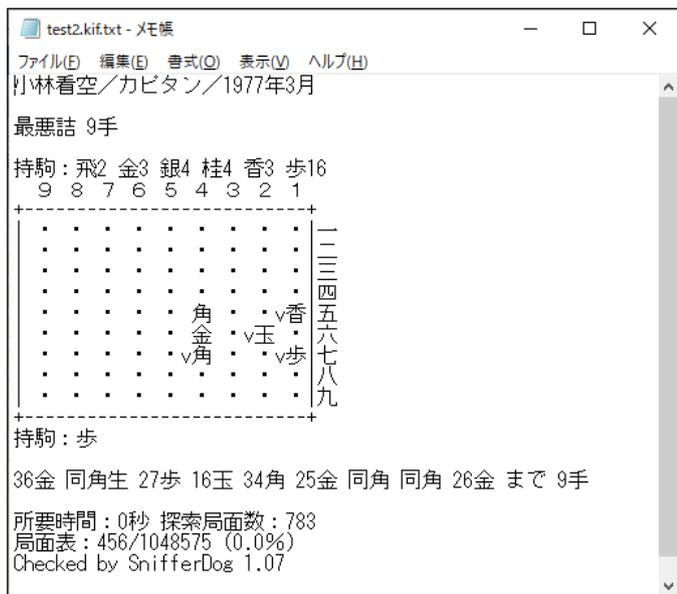
再利用には「詰データは極力保護する」とか「直近のデータはなるべく保護する」などの基準もありますが、これらはあくまで内部的な処理であり、表には出てきません。

6. 出力ファイル

(1) 検討結果の出力

解析結果は入力ファイルのある場所に「入力ファイル名」+「.txt」の名前で出力されます。従って二重括弧子のファイルになります。また、同じ入力ファイルで連続して解かせると、結果が追記されます。ファイルサイズの増加に注意してください。出力の末尾にはプログラムの版数が表示されます。バグなどの報告のときは、版数も併せて報告してください。出力ファイルは `fmview` や `FmView3` で鑑賞可能な形式になっています。

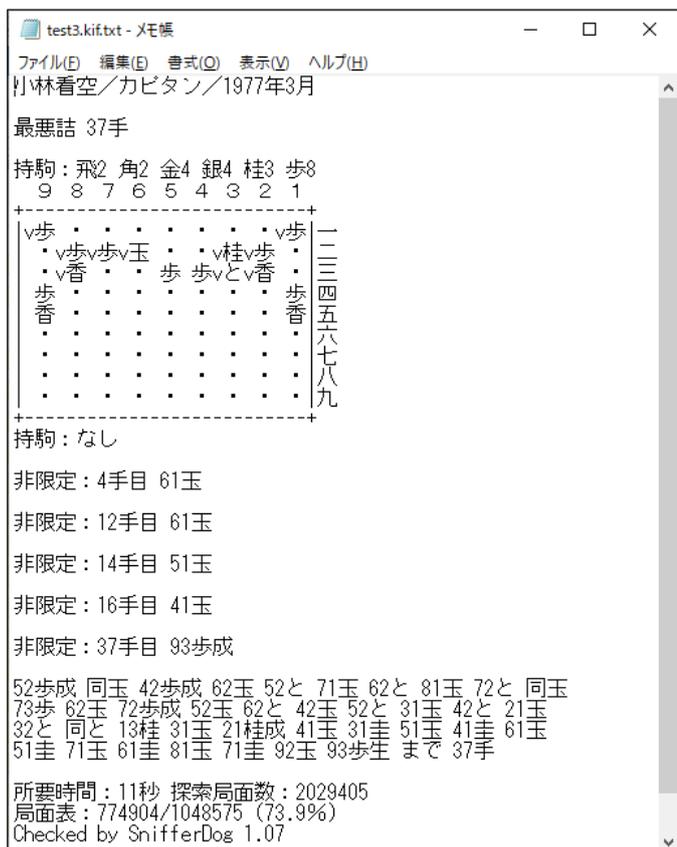
出力例) 小林看空／カピタン／1977年3月



非限定がある場合は、最初に見つかった解とそれに対する非限定が出力されます。

非限定はその存在のみが報告され、手順は示されません。非限定の中に更に非限定があっても無視されます。

非限定の出力例) 小林看空／カピタン／1977年3月

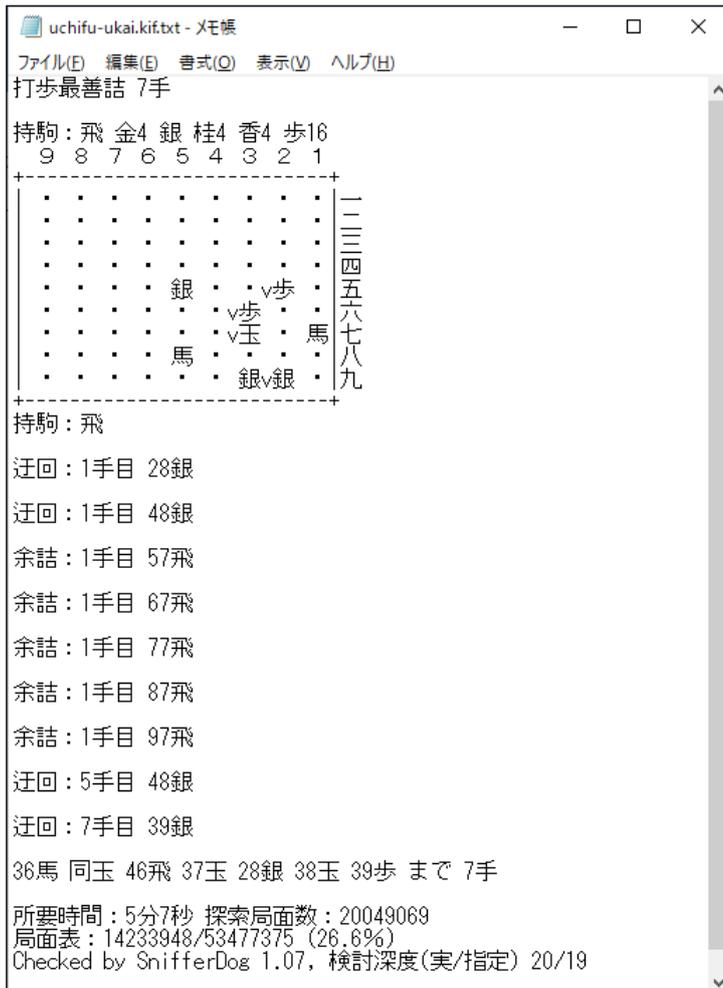


受方も攻方も作意以外の解は「非限定」と表示されます。

左右対称形からの左右対称解も非限定扱いされますが、それを無視するかどうかは使用者の方で判断してください。無視する場合は、あらかじめ「備考」に「対称性検出」と指定すると、左右対称形からの左右対称解を除外します。ただし「手順指定」の場合は「対称性検出」は無視されます。

また、最善系ルールで「余詰検討」のオプションを使った場合、詰を証明する側の非限定は「余詰」と表示されます。既出の局面に戻る余詰、または手順の先で作意に合流する余詰は「迂回」と表示されます。「迂回」と表示されても、その過程に余詰が潜んでいないことの保証にはなりません。あくまで、チェックの目安程度に考えてください。

出力例) 最善系ルールで「余詰検討」を指定した場合の出力例



```
uchifu-ukai.kif.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
打歩最善詰 7手
持駒: 飛 金4 銀 桂4 香4 歩16
      9 8 7 6 5 4 3 2 1
+-----+
| . . . . . . . . | 一
| . . . . . . . . | 二
| . . . . . . . . | 三
| . . . . . . . . | 四
| . . . . 銀 . . . . | 五
| . . . . . v歩 . . . | 六
| . . . . . v歩 . . . | 七
| . . . . . v玉 . 馬 . | 八
| . . . . . . . . . . | 九
+-----+
持駒: 飛
迂回: 1手目 28銀
迂回: 1手目 48銀
余詰: 1手目 57飛
余詰: 1手目 67飛
余詰: 1手目 77飛
余詰: 1手目 87飛
余詰: 1手目 97飛
迂回: 5手目 48銀
迂回: 7手目 39銀
36馬 同玉 46飛 37玉 28銀 38玉 39歩 まで 7手
所要時間: 5分7秒 探索局面数: 20049069
局面表: 14233948/53477375 (26.6%)
Checked by SnifferDog 1.07, 検討深度(実/指定) 20/19
```

(2) 証明木の出力

証明木とは与えられた局面が「詰」であることを証明するため、必要な変化を木構造で表したものです。本プログラムで出力する証明木のファイル形式は、FmTreeView (本来は fmza のトレース出力を木構造データとして表示するためのツール) で表示可能な形式になっています。

証明木のファイル名は、「入力ファイル名」+「-pt.txt」で、入力ファイルのある場所に出力されます。従って二重拡張子のファイルになります。同じ入力ファイルで連続して解かせると、上書きされます。ファイルサイズが非常に大きくなったり、証明木作成の時間が長くなったりすることがあるので、ご注意ください。

証明木を出力するときは、オプションに「証明木作成」または「最小証明木作成」と記入します。

「証明木作成」は正解手順より2手短い手順で詰むかどうかまでしかチェックしません。そのため作成時間は短くなりますが、ファイルサイズが大きくなったり、変化手順が最短でない場合があります。証明木は一時的なメモリを必要とするので、メモリを使い果たすと作成を強制的に中止する場合があります。

「最小証明木作成」は変化手順も最短の手順を求めます。ファイルサイズが小さくなり、手順も綺麗になりますが、作成時間は長くなります。変化の中に変化があってもその最短を求めると、処理時間が長くなるこ

とを前提にお使いください。

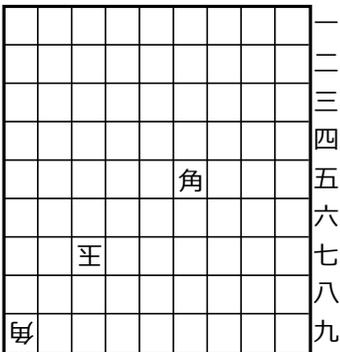
以下に本プログラムで出力した証明木を FmTreeView で表示した例をご覧ください。この例であれば「最小証明木作成」も数秒で終わります。

出力例)

神無七郎

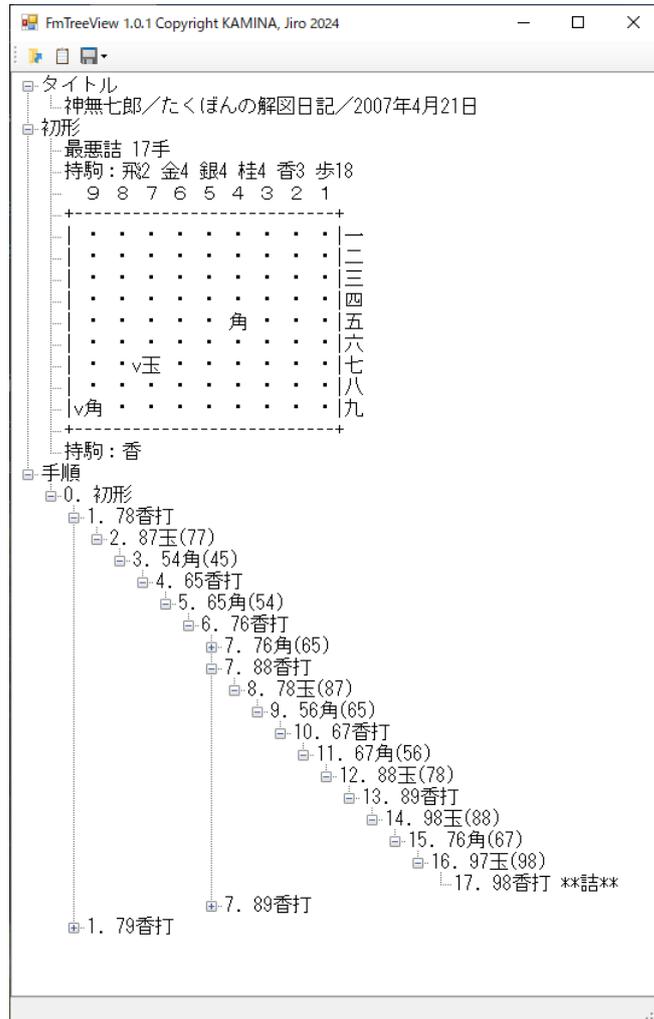
最悪詰 17手

9 8 7 6 5 4 3 2 1



持駒 香

(たくぼんの解図日記,2007年4月21日)



最悪詰は「攻方最長・受方最短」の手順を求めるものですが、その解は既に展開された形で表示されます。もし変化同手数順があれば、「詰」で終わる展開された枝が2つ以上表示されるので、問題点をすぐ把握することができます。

変化手順の「初手 79 香」「7 手目 76 角」「7 手目 89 香」は早く詰むので、折りたたまれた状態 (⊖) で表示です。これをクリックすると⊕になり、その着手以下の枝が展開されます。

実際に「初手 79 香」以下の枝を展開すると以下ようになります。



これを見ると「初手 79 香」の変化は「79 香 78 香 同香 88 玉 89 香 98 玉 54 角 76 香 同角 97 玉 98 香」で詰むことが分かります。

(3) 現探索情報の出力

その時点で探索中の手順、及び各局面及びその子局面に関する情報をファイルに出力します。出力内容は以下の3つです。

- a. 証明数・反証数の閾値
- b. 前回の証明数・反証数
- c. 現時点の証明数・反証数

証明数・反証数や閾値の意味については、「10. トレース、モニタ機能と本プログラムの動作原理」の「モニタ、トレースについての解説」をご覧ください。

探索中に「v」キーを押下すると、以下のようなメッセージが表示されます。

現在探索中の手順情報を出力しますか？ (Y/N) :

実行する場合は「Y」を入力します。それ以外の文字を入力すると何もせず解析を続行します。実行中は「作成中...」が表示され、完了すると「完了」のメッセージが付加されます。

作成中... 完了

出力ファイル名は、「入力ファイル名」+「-st.txt」で、入力ファイルのある場所に出力されます。従って二重括弧子のファイルになります。同じ入力ファイルで連続して解かせると、上書きされます。

出力ファイルの形式は FmTreeView での閲覧を前提としていますが、目視でも閲覧は可能です。

出力例) メモ帳で開いた例

```
七郎最悪45.kif-st.txt
ファイル 編集 表示
早詰チェック中：29手目44金(35)まで：神無七郎／妖精都市／2009年7月
最悪詰 43手
持駒：飛2 金2 銀4 桂3 香3 歩14
  9 8 7 6 5 4 3 2 1
+-----+
| . . . . . | 一
| . . . . . | 二
| . . . . . | 三
| . . . . . | 四
| . . 歩 . . 歩 金 . . . | 五
| . . . . . 歩v歩 . . . | 六
| . . . 角 . . . . . | 七
| . . . 金 角 . 桂 . . . | 八
| . . . v玉 . 香 . . . | 九
+-----+
持駒：なし
0. 初形 [8388607, 8388607] (1, 1)→(1, 1)
1. 69金(68) [8388607, 8388607] (1, 1)→(1, 1)
2. 48玉(59) [8388607, 8388607] (1, 1)→(1, 1)
3. 59金(69) [8388607, 8388607] (1, 1)→(53, 48302)
4. 59玉(48) [8388607, 8388607] (1, 1)→(2147483647, 0)
4. 37玉(48) [66, 8388607] (53, 44877)→(53, 44877)
5. 46桂(38) [8388607, 8388607] (1, 1)→(0, 2147483647)
5. 26桂(38) [8388607, 8388607] (1, 1)→(0, 2147483647)
5. 48金(59) [66, 8388607] (53, 44877)→(53, 44877)
6. 48玉(37) [8388607, 8388607] (1, 1)→(2147483647, 0)
6. 27玉(37) [66, 8388607] (53, 44877)→(53, 44877)
7. 37金(48) [66, 8388607] (53, 44877)→(56, 45945)
8. 37玉(27) [71, 8388607] (56, 26877)→(56, 26877)
9. 46桂(38) [71, 8388607] (56, 26877)→(65, 30520)
10. 46玉(37) [8388607, 8388607] (1, 1)→(2147483647, 0)
10. 27玉(37) [73, 8388607] (60, 1990)→(73, 2136)
10. 28玉(37) [8388607, 8388607] (1, 1)→(2147483647, 0)
10. 46玉(37) [8388607, 8388607] (1, 1)→(2147483647, 0)
10. 38飛打 [74, 8388607] (65, 21111)→(65, 21111)
行1,列1 6,143文字 100% Windows (CRLF) ANSI
```

[]内の数字が証明数・反証数それぞれの閾値、矢印の前の()内の数字が前回の証明数・反証数、矢印の後の()内の数字が現在の証明数・反証数です。

これを FmTreeView で表示した例をご覧ください。

出力例) FmTreeView での表示例

タイトル
早詰チェック中: 29手目44金(35)まで: 神無七郎/妖精都市/2009年7月

初形
最悪詰 43手
持駒: 飛2 金2 銀4 桂3 香3 歩14
9 8 7 6 5 4 3 2 1

・	・	・	・	・	・	・	・	・	一
・	・	・	・	・	・	・	・	・	二
・	・	・	・	・	・	・	・	・	三
・	・	・	・	・	・	・	・	・	四
・	・	歩	・	・	歩	金	・	・	五
・	・	・	・	・	歩	歩	・	・	六
・	・	・	角	・	・	・	・	・	七
・	・	・	金	角	桂	・	・	・	八
・	・	・	・	玉	香	・	・	・	九

持駒: なし

手順

- 0. 初形 [8388607, 8388607] (1, 1)→(1, 1)
- 1. 69金(68) [8388607, 8388607] (1, 1)→(1, 1)
- 2. 48玉(59) [8388607, 8388607] (1, 1)→(1, 1)
- 3. 59金(69) [8388607, 8388607] (1, 1)→(53, 48302)
- 4. 59玉(48) [8388607, 8388607] (1, 1)→(2147483647, 0)
- 4. 37玉(48) [66, 8388607] (53, 44877)→(53, 44877)
- 5. 46桂(38) [8388607, 8388607] (1, 1)→(0, 2147483647)
- 5. 26桂(38) [8388607, 8388607] (1, 1)→(0, 2147483647)
- 5. 48金(59) [66, 8388607] (53, 44877)→(53, 44877)
- 6. 48玉(37) [8388607, 8388607] (1, 1)→(2147483647, 0)
- 6. 27玉(37) [66, 8388607] (53, 44877)→(53, 44877)
- 7. 37金(48) [66, 8388607] (53, 44877)→(56, 45945)
- 8. 37玉(27) [71, 8388607] (56, 26877)→(56, 26877)
- 8. 16玉(27) [63, 8388607] (54, 4)→(63, 7)
- 8. 17玉(27) [8388607, 8388607] (1, 1)→(2147483647, 0)
- 8. 18玉(27) [8388607, 8388607] (1, 1)→(2147483647, 0)
- 8. 28玉(27) [61, 8388607] (53, 17996)→(65, 19061)
- 6. 28玉(37) [8388607, 8388607] (1, 1)→(2147483647, 0)
- 4. 57玉(48) [59, 8388607] (45, 1741)→(59, 3423)
- 4. 39玉(48) [49, 8388607] (39, 8)→(92, 2)

これは作意 (45 手) の検出が終わり、早詰 (43 手以下) があるかどうかチェックしている段階で出力した情報です。

これを見ると初手から「69 金 48 玉 59 金 37 玉 48 金 27 玉 37 玉 …」の手順を読んでおり、例えば 4 手目 59 玉の紛れが完全に詰まないこと、逆に、4 手目 37 玉・39 玉・57 玉の紛れについてはまだ検討中であり、結論が出ていないことがわかります。

上の図では 8 手目までしか展開していませんが、更に先の情報を知りたいければ、折りたたまれた箇所 (⊕で表示) を開いてください。

7. 中断・再開・途中経過表示

本プログラムは長時間の検討に資するため、検討を途中で中断する機能と、中断した状態から再開する機能、途中経過を表示する機能を備えています。

(1) 中断

本プログラムは探索中に **Esc** キーの押下で、プログラムを一旦停止させることができます。プログラムが **Esc** キーの押下を検出すると、現在探索中の手順、局面、検討深度及び局面表使用率が表示され、中断するかどうか確認するメッセージが出ます。

```
検討中： C:\work\umanoko65.kif
現在の読み筋：開始0手目、検討深度(実/指定)：46/65
32馬 11玉 33馬 21玉 43馬 11玉 44馬 21玉 54馬 11玉
55馬 21玉 65馬 11玉 66馬 21玉 76馬 11玉 77馬 21玉
87馬 11玉 88馬 21玉 98馬 11玉 88馬 22歩 12歩 21玉
87馬 65香 32銀成 12玉 22全 同銀 13歩

37手目：13歩
持駒：飛2 角4 金4 銀2 桂4 香3 歩16
  9 8 7 6 5 4 3 2 1
+-----+
|   .   .   .   .   .   .   .   .   | 一
|   .   .   .   銀   .   .   v銀v玉 | 二
|   .   .   .   .   .   .   .   .   | 三
|   .   .   .   v香   .   .   .   .   | 四
|   .   .   .   .   .   .   .   .   | 五
|   .   馬   .   .   .   .   .   .   | 六
|   .   .   .   .   .   .   .   .   | 七
|   .   .   .   .   .   .   .   .   | 八
|   .   .   .   .   .   .   .   .   | 九
+-----+
持駒：歩
局面表使用率：8.9%
中断しますか？ (Y/N) :
```

上の例では「検討深度(実/指定)」が「46/65」となっており、指定手数 65 手に対し、検討した最大深度が 46 手だった（つまりまだ余り深く読めていない）ことを示しています。

ここで「Y」または「y」を入力すると解図を打ち切り、「中断します。」というメッセージを表示します。逆に「Y」や「y」以外の文字を入力すると解図は続行されます。（※途中経過を観察する目的でこの機能を利用することもできますが、その場合は「途中経過表示」の使用を推奨します。）

中断が選択されると待避ファイルを作成するかどうか問い合わせるメッセージが出ます。

```
中断します。
現在の状態を保存しますか？ (Y/N) :
```

このメッセージに対して「Y」または「y」を入力すると待避ファイルの作成を開始し、以下のメッセージが出力されます。

```
待避ファイル作成中... しばらくお待ちください
```

待避ファイルには現在の問題・探索中の手順・局面表のデータ等が保存されます。このデータは元のファイルがあったのと同じ場所に、元の名前に「.sds」という拡張子を付けて保管されます。

【注意】 待避ファイルのサイズは使用メモリのサイズとほぼ同じ大きさになります。
使用メモリに大きな値を割り当てていた場合は、ディスクの残り容量にも気を付けてください。

なお、**Esc** キー押下が、暫定解を求めた後だった場合、待避ファイルを作るかどうかの前に、暫定解をファイルに残しておくかどうか問い合わせるメッセージが表示されます。

暫定的な解を出力しますか？ (Y/N) :

ここで「Y」または「y」を入力すると出力ファイルに暫定解が出力されます。
暫定解を残しておきたい場合は、「Y」または「y」を入力してください。
最終的な解のみを残す場合はここで「Y」や「y」以外の文字を入力してください。

待避ファイルの作成が完了すると、以下のようなメッセージが表示されます。

所要時間 : 5 分 22 秒 探索局面数 : 442899
局面表 : 225823/1048575 (21.5%)
Checked by SnifferDog 1.07
(中断 : 求解中)

何かキーを押してください...

解析終了時とほぼ同じメッセージですが、どの工程で中断されたかの情報が付加されます。**Enter** キー等を押下して終了してください。

(2) 再開

再開時に特別な操作は不要です。最初から解析を開始するのと同じ操作をしてください。

このとき、元のファイルの名前を元に待避ファイルを探しに行きます。

ですから、**ファイル名の変更や移動はしないでください。**

また、待避ファイルを作成したときと問題の設定（ルールや手数など）や、オプション（使用メモリの設定など）が異なっていた場合は、再開ができません。

待避ファイルが見つかったら以下のメッセージが出ます。

待避ファイルがあります。これを使って再開しますか？ (Y/N)

ここで「Y」や「y」を入力すると、プログラムは「待避ファイルから再開します。」というメッセージを表示し、待避ファイルに保存されたデータを元に解析を再開します。逆に「N」や「n」を入力すると、待避ファイルは無視して最初から解析を行います。

待避ファイルからのデータの読み込みが終わると、問題図・中断時の手順・中断時の局面が表示されます。ただし待避ファイルを作成したのが早詰チェックの段階の場合は、再開時に手数が元の手数より2手少なく表示され、手数の横に「早詰チェック」の文字が付加されます。

待避ファイルを作成したのが変別・非限定チェックの段階の場合は、再開時に何手目のどの手から再開するか示されます。検討が済んだ手順の分については「検討済」と表示され、検討をスキップします。

解析が終了したとき待避ファイルを削除するかどうか問い合わせるメッセージが出ます。

待避ファイルを削除しますか？ (Y/N) :

ここで「Y」や「y」を入力すると、待避ファイルを削除します。

「Y」や「y」以外の文字を入力した場合、待避ファイルは自動で削除されません。

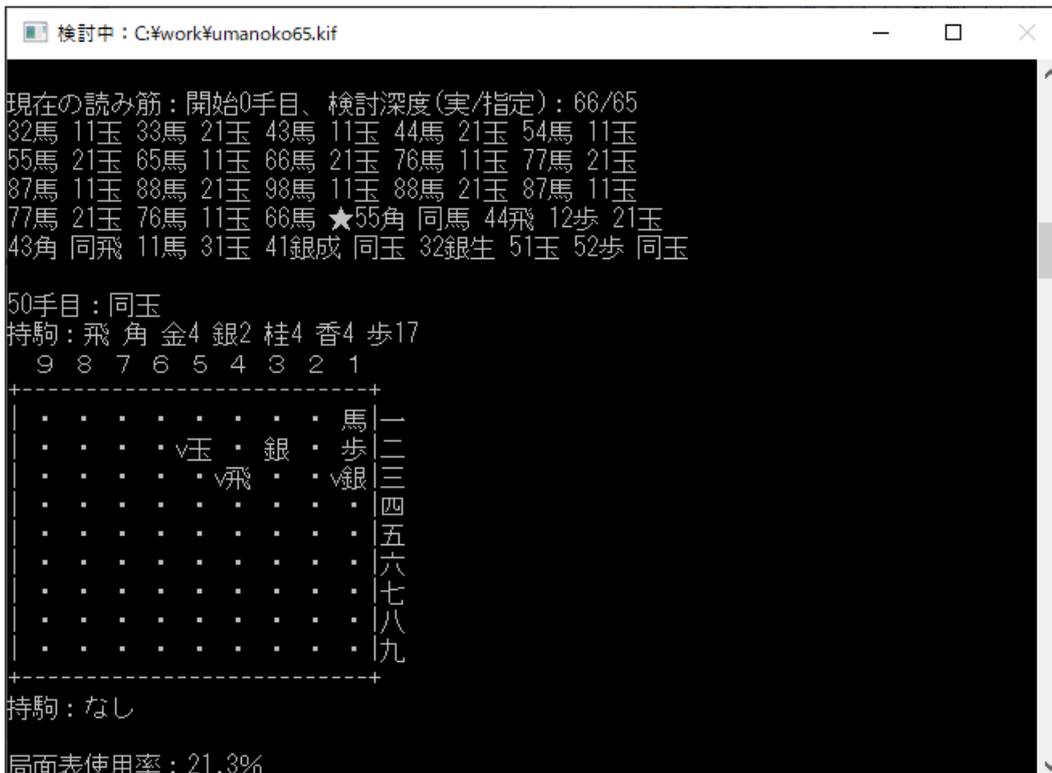
待避ファイルを後で削除する場合は、エクスプローラなどから手動で削除してください。

(3) 途中経過表示

解析中に「D」または「d」のキーを押下すると、その時点で読んでいる手順と図面が表示されます。



「手順指定」での検討の場合、指定手順と違っている手に「★」マークが付加されます。



このマークが変化に付いている場合は、順調に作意通り進んでいることを意味し、作意では詰まないはずの紛れに付いている場合は、どこかで詰まない変化があったために別の手順を読んでいることを意味します。

作意に問題がない場合、「★」マークは詰上りから初形まで遡るように移動するはずですが、早詰チェック以降の工程にすぐ進まない場合、難しい変化を読んでいるのか、作意自体が誤っているのか確認するため、途中経過表示を利用して進捗を確かめてください。また、早詰チェックの段階では、どの手でも詰まないはずなので、必ず紛れの手に「★」マークが付きます。この時も最終手から初手に遡るように「★」マークが移動するので、進捗の確認に使うことができます。

(4) 自動待避

「備考」に「自動待避」と指定すると、一定の時間間隔で自動的に待避ファイルの作成を行います。作成する時間間隔は「自動待避 4」のようにオプション名の後の数で指定します。指定できる時間間隔の範囲は「1」から「24」です。長時間の検討を実行中するとき、PCの思わぬ再起動や停電等でせっかくの検討時間が無駄にならないよう、適度な時間を指定すると良いでしょう。

8. 本プログラムで扱うルールに関する補足説明

(1) 普通の詰将棋と「最善詰」の違い

本プログラムでは「最善詰」が機能に含まれ、普通の詰将棋もある程度解くことができます。ただし、「最善詰」では「攻方最短・受方最長」以外の規則はすべて除外しています。具体的に違いが生じるのは以下のような事項です。

- ・無駄合概念はない
- ・駒が余るかどうかで手順に優劣を付けない
- ・長手数之余詰は無視する

このようにしているのは開発の手間を省くためではなく、本プログラムがフェアリー詰将棋用のプログラムだからです。

フェアリーではルール単体を最適化するために細則を設けるより、ルールの応用を重視してルール群全体の最適化を考えます。その場合、コアとなる「どのような手を選ぶか」というロジックは可能な限りシンプルでなければなりません。普通の詰将棋の規則をその附則と一緒にフェアリーに導入するのは望ましくないのです。

そのため、本プログラムでも「攻方最短・受方最長」以外の要素を排除した「最善詰」を扱っています。(目標を「詰」以外にしたルールでも、同様の扱いです。)

以下に「最善詰」と普通の詰将棋でどのような違いが現れるか、具体的に見てみましょう。

例)

9	8	7	6	5	4	3	2	1	
					馬		王		一
				銀					二
							銀	歩	三
									四
				糸					五
									六
									七
									八
									九

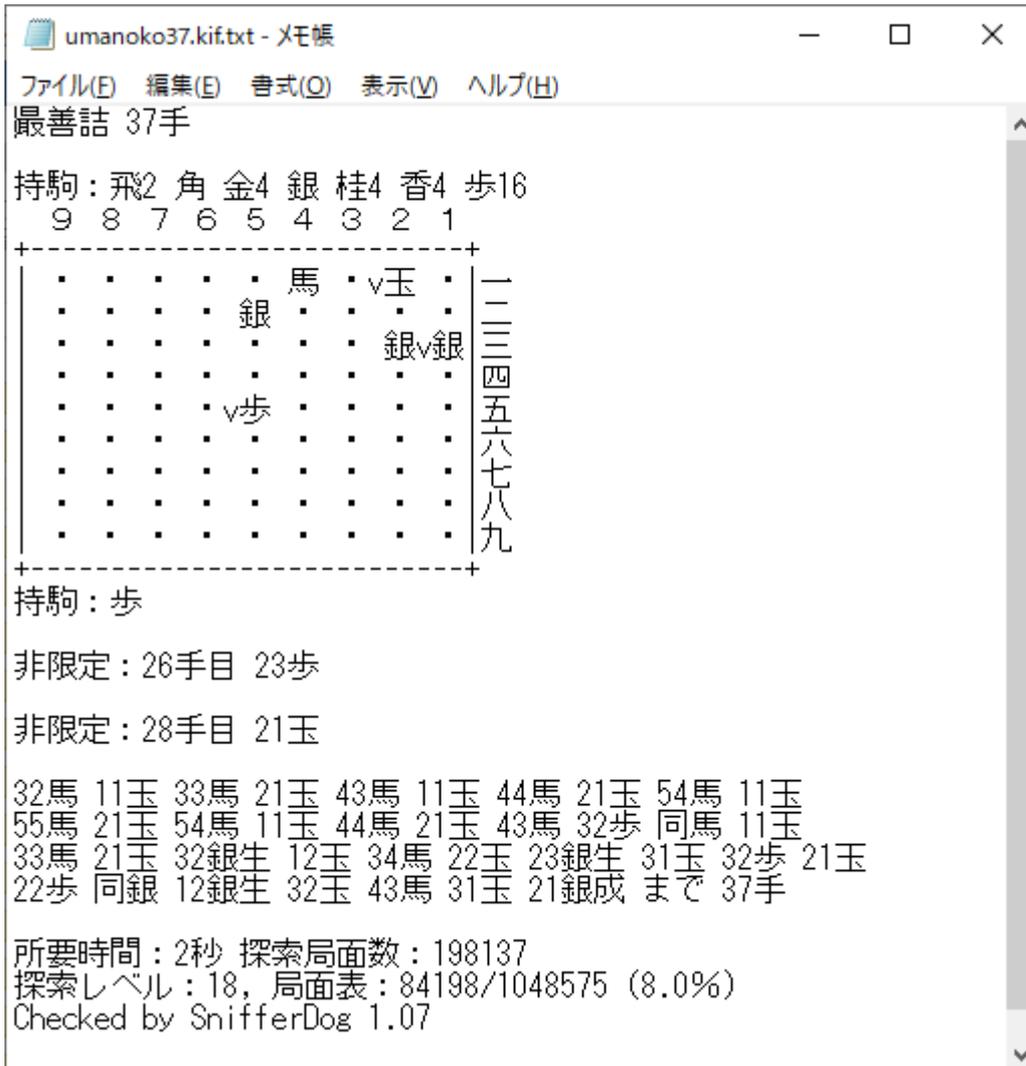
持駒 歩

これを普通の詰将棋として解くと、以下の 35 手詰が正解となります。

32 馬 11 玉 33 馬 21 玉 43 馬 11 玉 44 馬 21 玉 54 馬 11 玉
55 馬 21 玉 54 馬 11 玉 44 馬 21 玉 43 馬 11 玉 33 馬 21 玉

32 銀生 12 玉 34 馬 22 玉 23 銀生 31 玉 32 歩 21 玉 22 歩 同銀
 12 銀生 32 玉 43 馬 31 玉 21 銀成 まで 35 手

しかし、これを「最善詰」として本プログラムで解くと、以下のようになります。



17 手目「32 歩」は普通の詰将棋では「無駄合」とされる手ですが、「最善詰」ではこれを省略しません。普通の詰将棋に慣れていると、この種の合駒を除外することに慣れてるので、しばしば手数不足の「不詰」を引き起こします。特に無駄合が変化に隠れている場合に要注意です。

また、26 手目「23 歩」は「同手数駒余り」であり、普通の詰将棋では手数の短い変化と同様に扱われますが、「最善詰」では手数の同じ「22 玉」の変化とまったく同格の扱いです。上の例では、選択された解が 26 手目「22 玉」だったため 28 手目「31 玉」の非限定も検出されています。たまたま 26 手目「22 玉」が選択された場合は、この非限定は出力されません。「非限定」の中の「非限定」は無視されるのです。

本プログラムで出力される「非限定」が普通の詰将棋でも非限定かどうかは、使用者自身が判断する必要があります。

(2) 千日手の扱い

本プログラムは「連続王手の千日手」を禁手と扱いません。そのため、このルールを適用する作品の解図はできません。

また本プログラムにはルールとしての「千日手」があります。これは初形に戻すことを達成すべき目標とするルールであり、手順の途中で同一局面を繰り返しても目標達成とはなりません。「千日手」は協力系ルールで出題されることが多いですが、以下のように対抗系ルールでも実現は可能です。

例) 最善千日手 12手

	3	2	1	
龍	飛			一
			王	二
	香			三
		金		四
				五

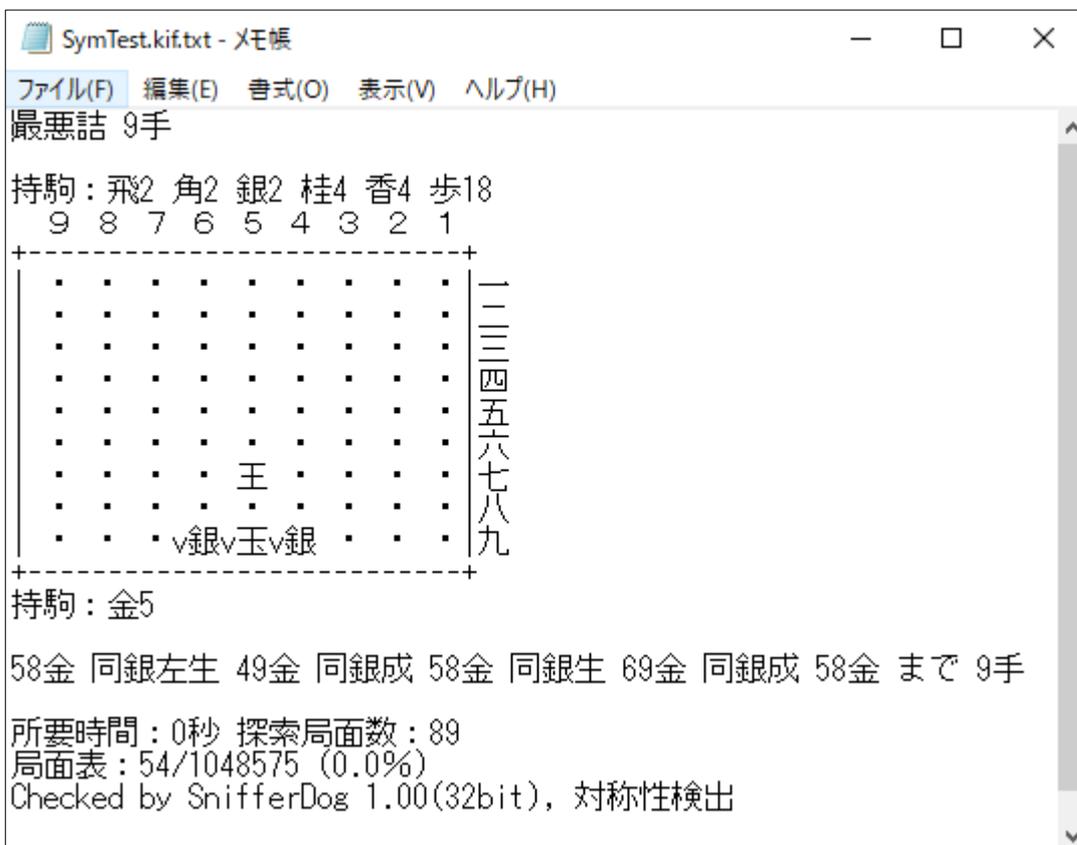
攻方持駒 桂
受方持駒 なし

24 桂 同金 32 龍 22 桂 同飛生 11 玉 23 桂 同金 31 龍 21 桂 同飛生 12 玉 まで 12 手

(3) 左右対称の扱い

本プログラムのデフォルトは、左右対称形で手順が左右対称であっても「非限定」として扱います。それが本当に非限定かどうかは使用者が判断してください。

「備考：」欄に「対称性検出」のオプションを指定すると、左右対称形からの左右対称解を無視します。



このオプションは初形だけでなく、途中左右対称形からの左右対称解にも適用されます。なお、手順指定時はこのオプションは無視されます。

(4) 自動で付加される要素

本プログラムではルールの特性に応じて自動的に付加される条件やオプションがいくつかあります。

・最善詰

「禁欲」や「打歩」の条件がないとき「持駒節約」のオプションが付加される。

・スタイルメイト

「最善スタイルメイト」には「非王手可」が付加される。(最終手のみ非王手可ではなく、最初から最後まで攻方王手義務は適用されない。)
「最善自玉スタイルメイト」に「非王手可」は付加されない。

・勝

「勝=詰+スタイルメイト」なので、「スタイルメイト」と同様の方式で「非王手可」が付加される。

・禁欲、強欲

この条件は双方の手番に適用されるので、「双方」が自動的に付加される（ルール名には反映されない）。片方の手番に適用する場合は「受方」「攻方」のどちらかを指定する。

・マキシ、ミニ、長駆

この条件は受方の手番に適用されるので、「受方」が自動的に付加される（ルール名には反映されない）。適用する手番を変更する場合は「双方」「攻方」のどちらかを指定する。

(5) 「長駆」条件について

「長駆」は「マキシ」の距離の測り方をユークリッド距離からチェビシェフ距離に変更したルールです。

チェビシェフ距離という堅苦しい用語に聞こえますが、要は「玉なら何手で行けるか」を基準にした距離の測り方と思ってください。つまり「玉距離」です。

距離をこのように定義すると、通常のような着手が可能なルールと、「マキシ」の中間的な挙動が現れます。

将来的には「マキシ」や「ミニ」に適用する距離の選択肢を増やす（「臣距離-縦横が距離1・斜めが距離2」、「金距離-斜め下が距離2・それ以外の方向が距離1」、「銀距離-横が距離2・下が距離3・それ以外の方向が距離1」等）構想もありますが、現時点では実装していません。

(6) 「打歩」条件について

本プログラムで扱う「打歩」は単純打歩です。打歩以外の詰を禁手とし、双方の手番に再帰的に適用する完全打歩とは異なり、打歩以外の詰（目標達成）は失敗と見なされます。

また、自玉系ルールでは「打歩」の指定はできません。例えばルール名に「最善自玉スタイルメイト」と指定すると、エラーメッセージが表示されます。

今のところ「完全打歩」や自玉系ルール、「千日手」での「打歩」指定の提供は予定していません。

双玉で自玉の打歩詰を利用する「逆打歩詰」は本プログラムで扱う範囲に含まれます。受方は逆王手の打歩詰となる応手を指すことはできません。

例) 逆打歩詰の変化がある例 (受方は2手目22歩を指せない)



(7) 同時指定の制限

通常の合法手から指せる着手を絞り込むタイプの条件、「禁欲」「強欲」「マキシ」「ミニ」「長駆」は同時指定できません。「禁欲」と「強欲」、「マキシ」と「ミニ」のように反対の性質を持っていたり、「禁欲」と「マキシ」のように、適用する順序の違いにより、指せる着手が異なったりする場合があるためです。

優先度等の概念を導入して同時指定を可能にすることも論理的には可能ですが、需要は少ないと思うので、今後も導入予定はありません。

また「打歩」と「千日手」も同時指定はできません。ただし歩を打って初形に戻る作品は検討できます。「千日手」において突歩で初形に戻る手と打歩で初形に戻る手を（今のところは）区別していないだけです。

9. 各種検討支援機能

(1) 手順指定

「手順指定」は、あらかじめ作意を入力しておく、その作意を元に検討する機能です。

ただし作意を盲目的に信頼するのではなく、変化が詰むことを確認してから、以降の検討作業を行います。作意は途中まででも構いません。作意が入力されているところは、変化が詰むことだけを確認し、作意が途切れたところから通常の解図と検討の作業を行います。

変化が詰まない時、作意以外の手順で詰む可能性があるので、以降は通常の解図と検討を行います。

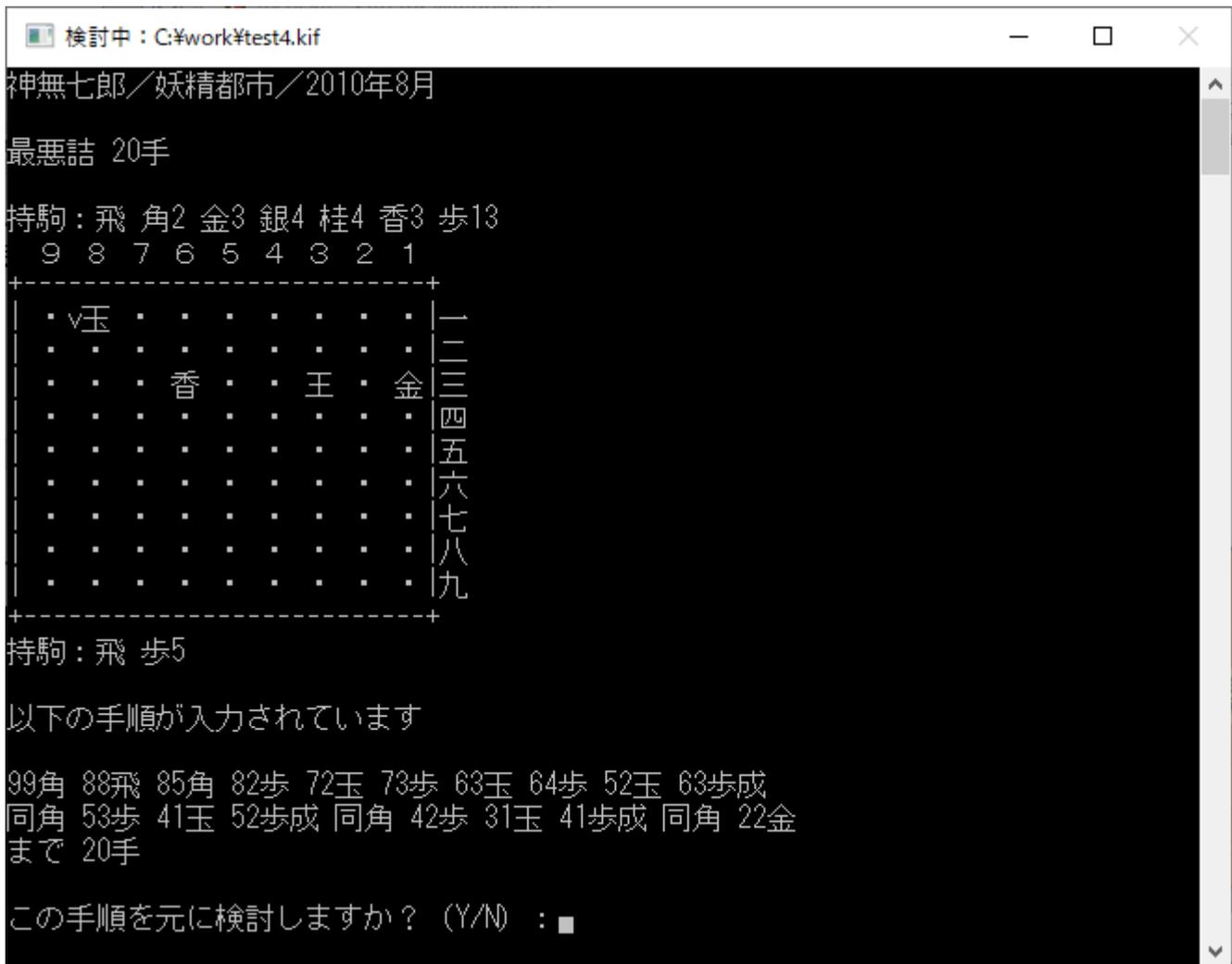
また、作意不詰で作品自体も不詰の場合、作意の末端に近い不詰箇所を表示します。

手順の指定は kif 形式で行います。(ki2 形式や他の棋譜形式には対応していません)

以下のように手順を入力した kif ファイルを用意し、本プログラムを起動します。



すると、解図の前に入力手順を利用するかどうかの確認が行われます。



ここで「Y」または「y」を入力してください。
以降は変化の確認と検討が行われます。

「手順指定」の主な用途としては以下を想定しています。

- ・ 検討時間の短縮（作品にある程度の自信のある場合）
- ・ 変化の読み抜けがないことの確認
- ・ 手順の清書用
- ・ 非限定がある場合に特定の手順を検討させたい 等々

【補足1】作意不詰の場合

入力された手順に不詰の変化がある場合、本プログラムは指定以外の手順で詰むかどうか調べます。もし詰めば通常通り解を出力しますが、不詰の場合は指定手順の一番末端に近い不詰箇所を表示します。

〔不詰作の例1〕 森茂作、最悪詰 29手（詰将棋パラダイス 2001年9月）

森茂作

最悪詰 29手

9 8 7 6 5 4 3 2 1

									一
									二
						角	角		三
									四
					馬	香			五
					飛				六
				金					七
					金	金			八
					玉	王			九

持駒 なし

(詰将棋パラダイス,2001年9月,不詰)

上図の作意は以下の通りですが、19手目 28銀のところ、46角成で不詰です。

- 18金 同玉 28金 19玉 18金 同玉 38飛 19玉 18飛 同玉
- 45角成 36銀 同馬 19玉 18馬 同玉 29銀 19玉 28銀 18玉
- 19銀 同玉 46角生 28香 同角 18玉 19香 27玉 37金 まで 29手

実際に作意手順を入力した.kifファイルを用意し、本プログラムで解かせてみましょう。



すると、以下のように表示されます。

【補足2】普通詰将棋で不詰を確認する場合の注意点

手順指定による不詰箇所を検出は最悪詰以外のルールでも有効です。

ただし、これを普通詰将棋の不詰チェックに使う場合、少し注意が必要です。

SnifferDog で取り扱うのはあくまで「最善詰」なので、無駄合によって伸びる手数 of 考慮をしないと、手数不足による不詰を検出してしまいます。

また、不詰の変化の意味が単なる「手数稼ぎ」の場合は、検出された不詰箇所以外に「真の不詰」が隠れていることがあります。

具体的な例で説明しましょう。

〔不詰作の例2〕宮越宗太郎作「刻」、215手（将棋世界 1961年1月）

宮越宗太郎 作「刻」

詰将棋 215手

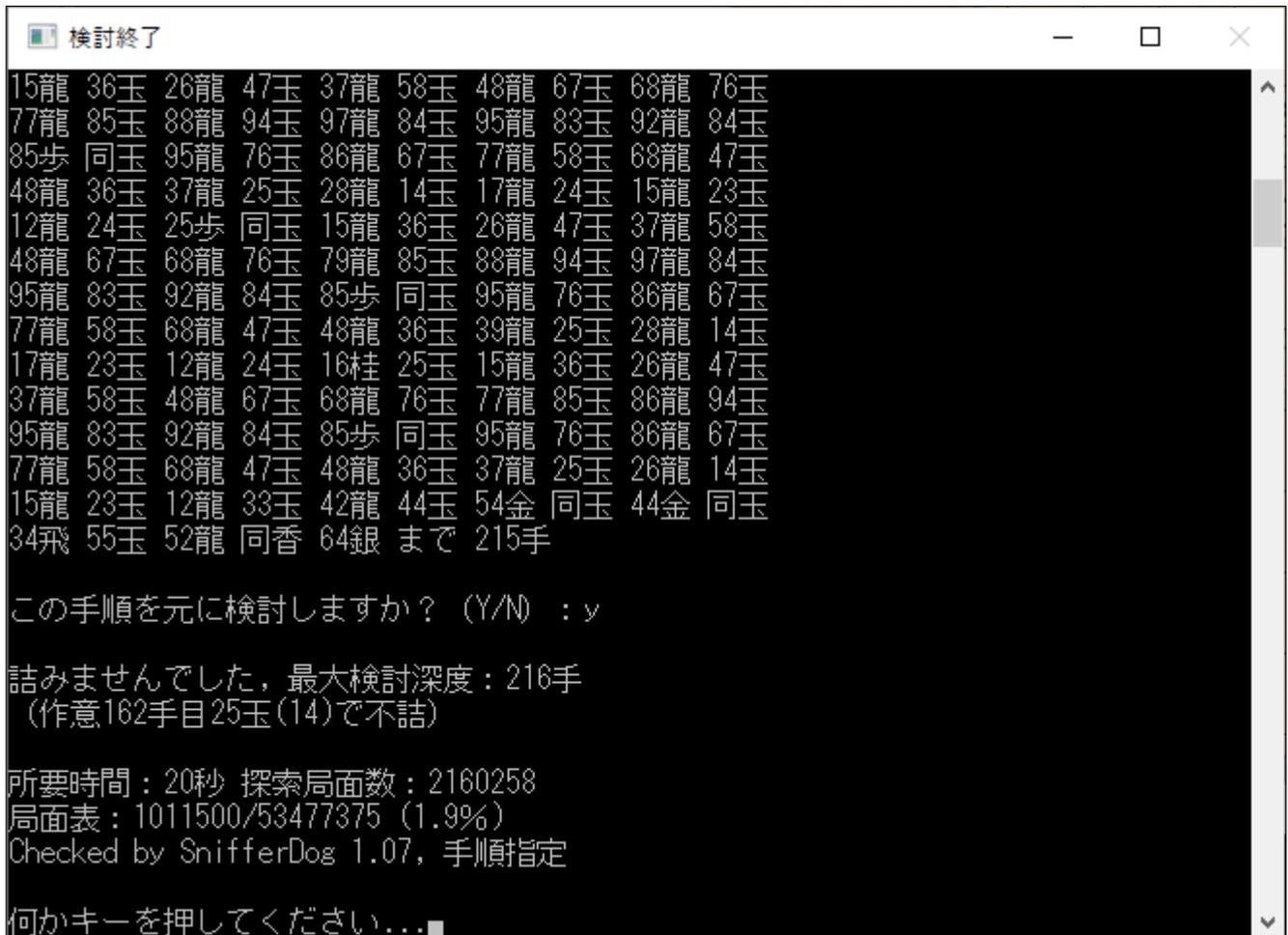
9 8 7 6 5 4 3 2 1

と				皇				と	一
			香	馬	香				二
			香	金	香				三
				馬					四
歩		歩	桂	歩	歩		歩		五
			歩						六
ス		ス	角		龍		ス		七
	ス		王				ス		八
歩		ス	角		手		歩		九

持駒 金

(将棋世界,1961年1月,不詰)

これを本プログラムで解いてみます。



「作意 162 手目 25 玉(14)で不詰」と出ていますね。
でも、これではなぜこの手で不詰なのか分かりません。ちょっとその局面を見てみましょう。

	9	8	7	6	5	4	3	2	1	
と				馬					と	
			馬	龍	馬					
			香	金	香		王			
				龍						
		歩	歩	桂	歩	歩				
				歩						
				角				龍		
歩				角					歩	

持駒 桂 歩
(162手目23玉の局面)

作意は 162 手目 23 玉です。そして、ここから「12 龍 24 玉 16 桂…」と進みます。

162 手目 25 玉としても「26 龍 14 玉 15 龍 23 玉」以下作意に合流できるので、これでは「手数稼ぎ」があることしか分かりません。

こんな場合に、真の「不詰」を検出する方法があります。

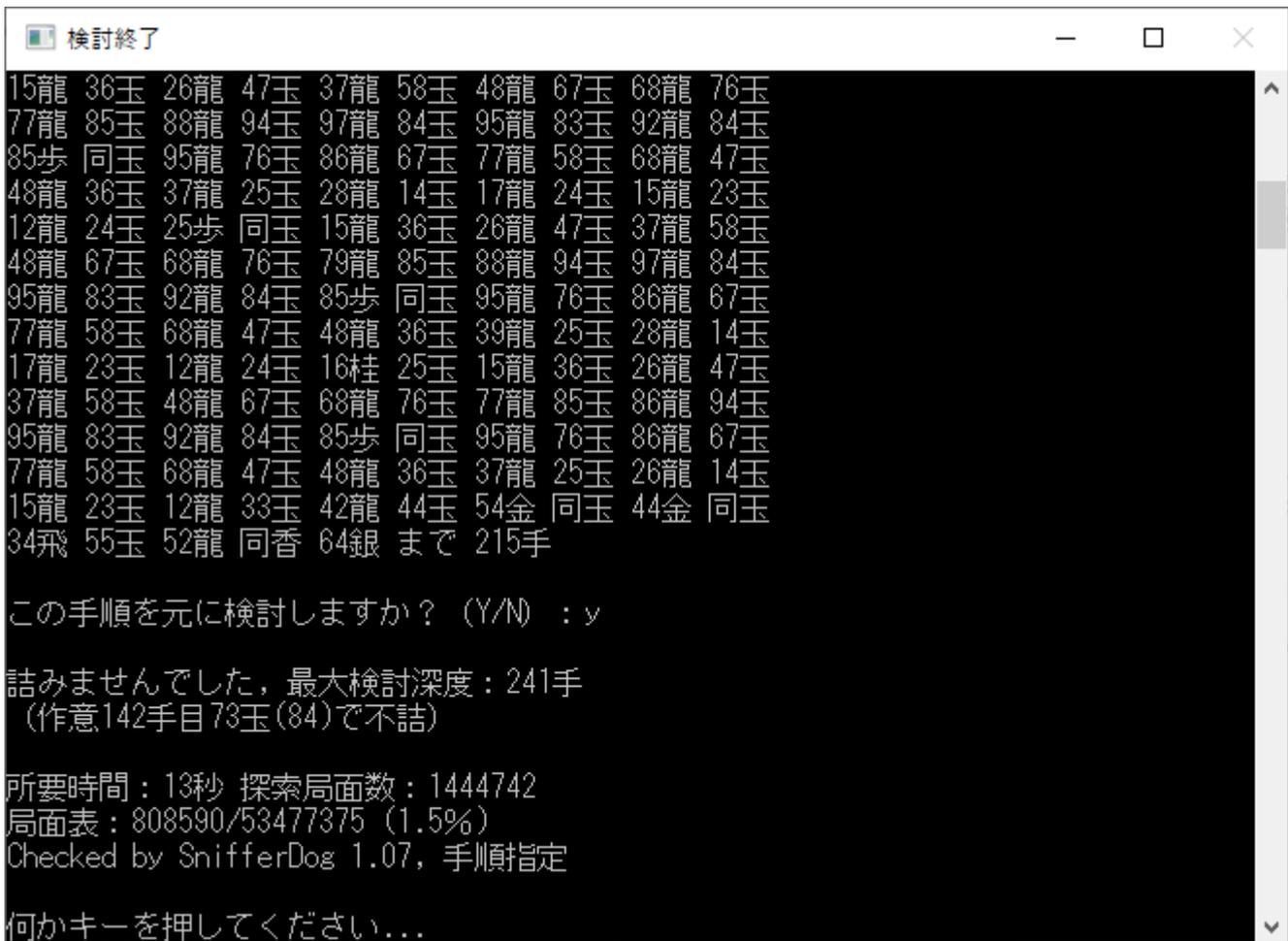
手数を作意ちょうどにするのではなく、余裕を持って指定するのです。

「詰将棋情報」の「手数」欄に書く手数を「215」から「999」に変えて、棋譜を保存します。

詰将棋情報 ✕

作品番号：	<input type="text"/>	OK
作品名：	刻	キャンセル
作者：	宮越宗太郎	
発表誌：	将棋世界	
発表年月：	1961年1月	
出典：	<input type="text"/>	
手数：	999	
完全性：	不詰	
分類：	詰将棋	
受賞：	<input type="text"/>	
備考：	メモリ1GB	

本プログラムで解析すると、今度は「作意 142 手目 73 玉(84)で不詰」と出ました。



実際に作意の 142 手目と比べてみましょう。

	9	8	7	6	5	4	3	2	1	
と				皇					と	一
			香	金	香					二
	王		香	金	香					三
				銀						四
龍		歩	歩	桂	歩	歩				五
				歩						六
				角						七
										八
歩				角		手		歩		九

持駒 歩2
(142手目83玉の局面)

作意 142 手目は 83 玉でした。確かにここで 73 玉とされたら詰みませんね。

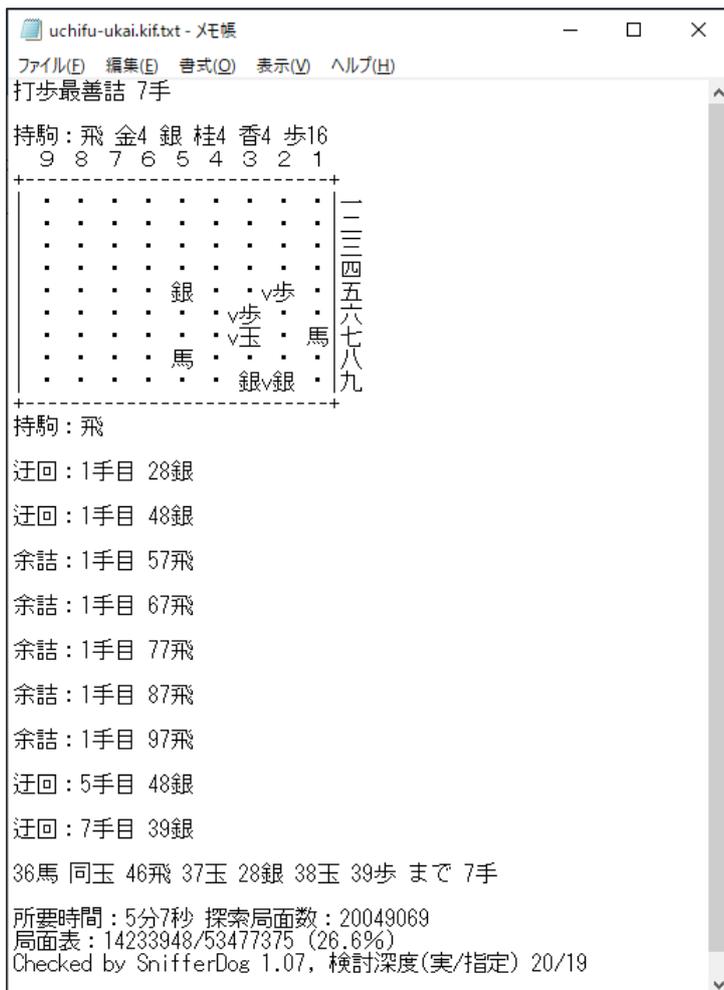
以上のように、本プログラムで普通詰将棋を検討し、手順指定で不詰箇所を特定したい場合は、「手数」を多めに指定することをお勧めします。

(2) 余詰検討

「最善詰」は攻方最短・受方最長の手順を求めるものですが、普通詰将棋では長手数余詰でも作品は不完全作となります。フェアリーでも「自玉詰」等はその規定に準じているので、長手数余詰の有無を検出する「余詰検討」のオプションを用意しました。これは最善系のルールでのみ有効です。

単に「余詰検討」とすると、ルールと指定手数に応じ 2999 手または 2998 手以内の余詰検討が行われます。検討する手数上限を変更する場合は「余詰検討 99」のように、手数を表す数値を指定します。

以下は「余詰検討 20」を指定して検討を行った例です。



指定した検討深度は「20」ですが、手数は20を越えない最大の奇数である「19」に調整されています。長手数余詰の中でも、既出の局面に戻るものや、余分な手数を掛けて作意に合流するものは「迂回」と表示されます。ただし「迂回」の途中で余詰が潜んでいても検出されないことがあるので、鵜呑みにしないでください。

出力の末尾に「検討深度(実/指定) 20/19」とあるのは、指定された検討深度と実際に検討を行った最大の深度を示しています（深度＝手数と考えて結構です）。実深度の方が指定深度より1上回った場合、更に深度を深めると新たな余詰が見つかる可能性もあります。**余詰が検出されなかった場合**、指定した手数が充分だったかどうかの目安としてください。

このオプションを使うと、手順前後、成不成非限定、以遠打のような比較的軽微とされる非限定についても「余詰」と表示されます。それを本当に無視して良いものかどうかは使用者が判断してください。

(3) 作意強制（※仮実装）

「作意強制」は「手順指定」の一種で、変化が詰むかどうかの確認を行わず、入力された手順が合法である限り、作意に従って進め、それ以降の検討作業を行う機能です。作意を途中まで入力し、特定の変化・紛れを確認する使い方を想定しています。

例えば「(2) 余詰検討」で挙げた例図では、初手に57飛を打つ余詰が検出されています。その余詰が具体的にどんな手順か知りたいとしましょう。

この場合、作意として初手のみ kif ファイルに入力します。



「詰将棋情報」の「備考」欄には「作意強制」のオプションを指定します。
 既に得られている検討結果から余詰の手数が19手以内であることは確定しているため、手数は「19」を指定します。

The dialog box titled "詰将棋情報" (Jigsaw Puzzle Information) contains the following fields and controls:

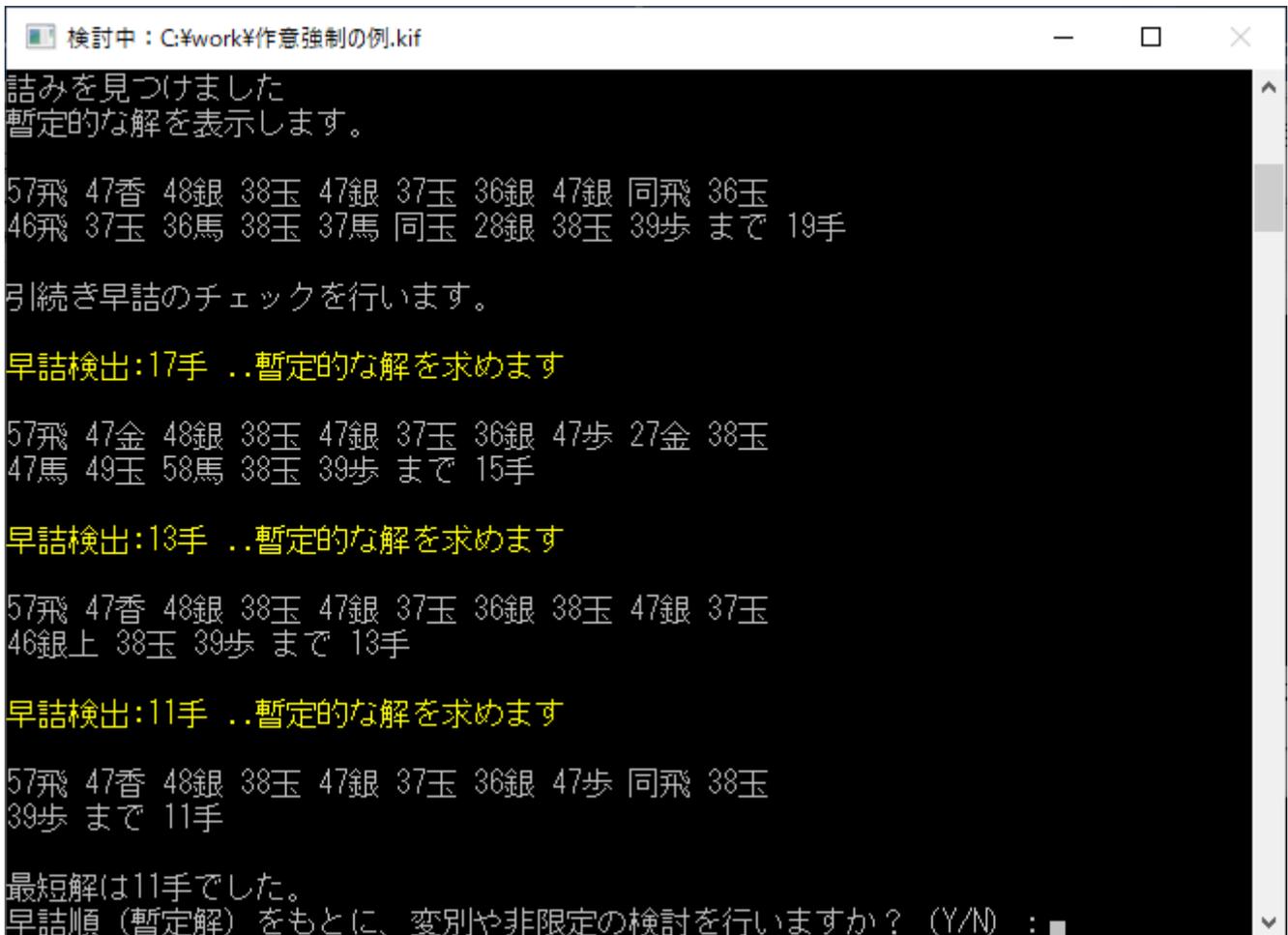
- 作品番号: []
- 作品名: []
- 作者: []
- 発表誌: []
- 発表年月: []
- 出典: []
- 手数: 19
- 完全性: []
- 分類: 打歩最善詰
- 受賞: []
- 備考: 作意強制

Buttons: OK, キャンセル

後は「(1) 手順指定」と同じです。
 「この手順を元に検討しますか? (Y/N):」の問い合わせに対し、「Y」または「y」を入力して検討を行ってください。

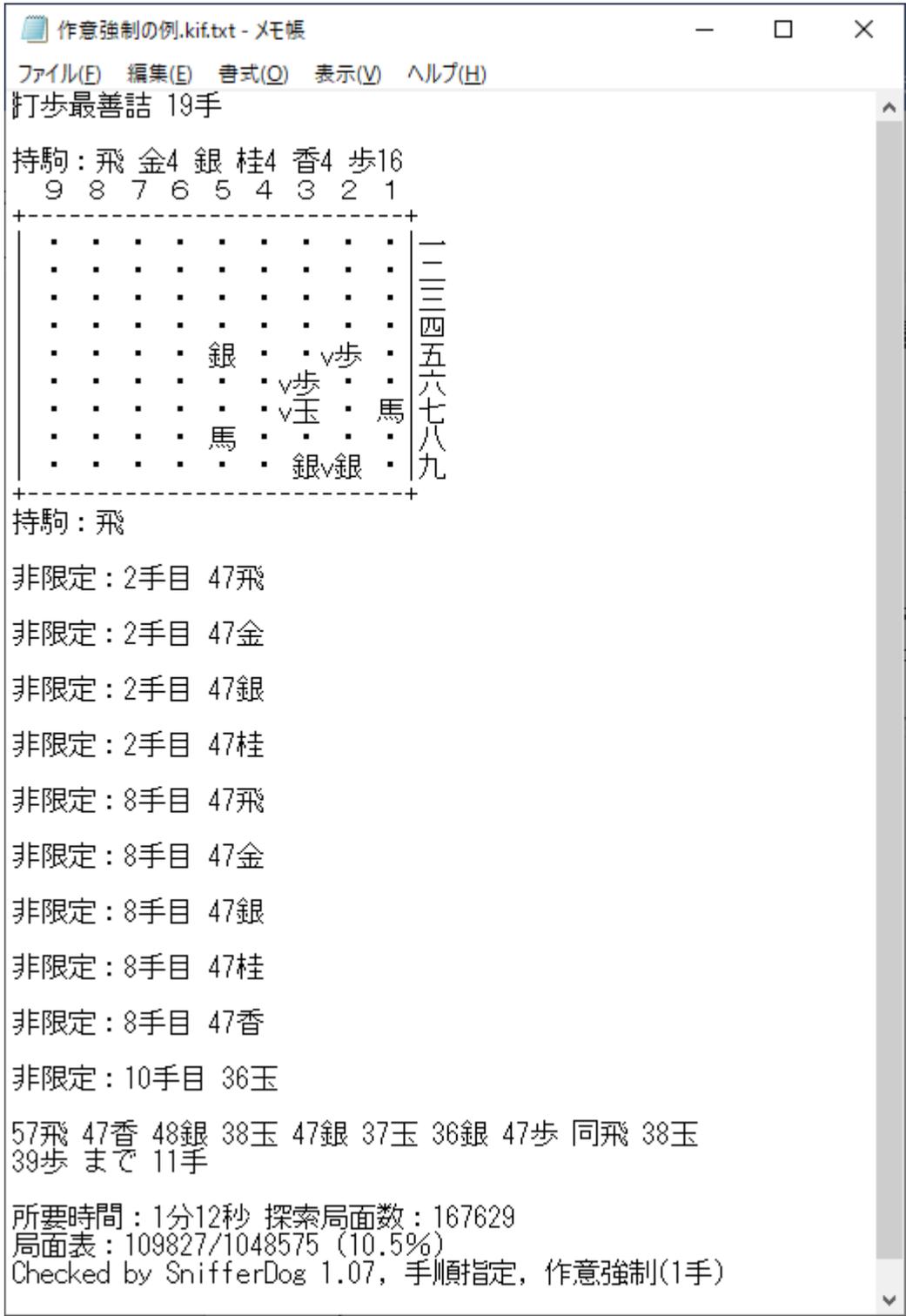


指定手数 19 手に対し、実際は 11 手で詰むので、検討継続の問い合わせが表示されます。



暫定解は変別を含む可能性があるため、時間に余裕があれば「Y」または「y」を入力して検討を継続してください。暫定解で十分な場合はここで「N」や「n」を入力して検討を打ち切っても構いません。

以下は検討を継続した場合の出力です。



(4) 不詰出力 (※仮実装)

「不詰出力」は与えられた問題が詰まなかった場合に参考手順を出力するものです。読んだ範囲内での最適不詰解 (最善系なら攻方最長・受方最短、最悪系なら攻方最短・受方最長の手順) を求めるものであり、すべての紛れを読んだ上での最適不詰解を求めるものではありません。また、長手数作品では不詰手順も長くなるため、求解に長時間掛かる場合もあります。

以下の例で「不詰出力」を指定してみましょう。



「詰将棋情報」で「備考」に「不詰出力」と記入し、手数も多めに指定しています。(手数不足だと手順の途中で打ち切ったような解が出力されます。)

詰将棋情報		×
作品番号 :	<input type="text"/>	OK
作品名 :	<input type="text"/>	
作者 :	<input type="text"/>	キャンセル
発表誌 :	<input type="text"/>	
発表年月 :	<input type="text"/>	
出典 :	<input type="text"/>	
手数 :	99	
完全性 :	<input type="text"/>	
分類 :	最善詰	
受賞 :	<input type="text"/>	
備考 :	不詰出力	

すると、以下のような結果が表示されます。



この例では「最大検討深度 40」となっているので、指定上限の 99 手より短い手数で不詰の結論が得られました。従って手数不足の不詰ではなく、真の不詰だということが分かります。例示された手順中、攻方は同一局面に戻る手を避けていることにご注目ください。本プログラムでは「詰」を目指す側は手順中に同一局面が出現しないようにしています。

作品が不詰のとき、ここで出力される手順が参考になるかどうかは分かりません。出力される不詰解が作者の見落とし箇所を含むとは限らないからです。不詰の原因を特定したい場合は、「不詰出力」より「手順指定」を推奨します。

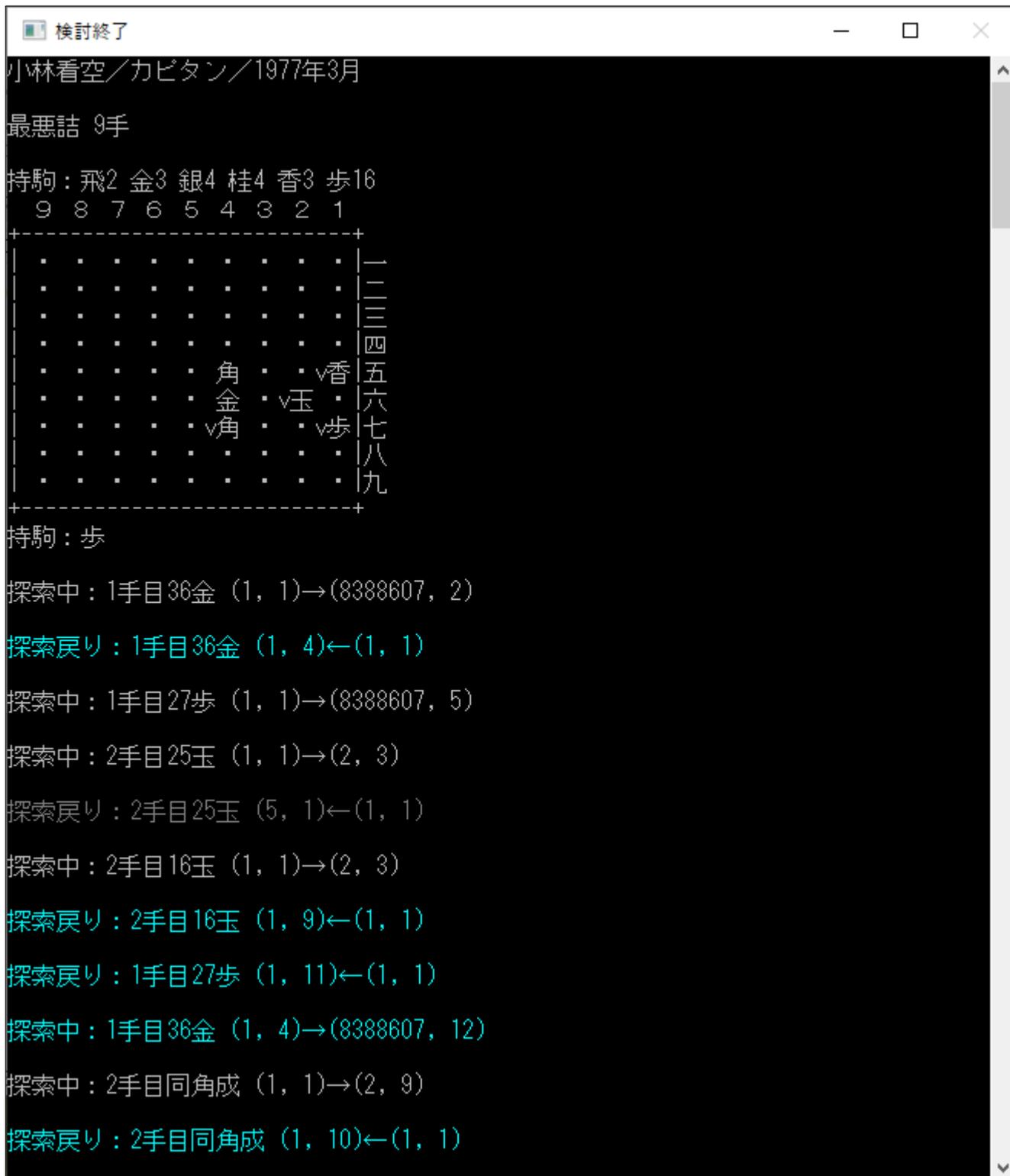
また、ここで示される手順はあくまで「読んだ範囲内の」不詰解です。それが如実に現れているのが、36 手目「21 玉」です。ここは「同玉」の方が短い手数で不詰となりますが、たまたま「21 玉」の不詰を先に発見したため、本プログラムはそれで満足し、より短い不詰順があるかを読まなかったというわけです。

詰む作品の詰手順を求めるとき「変別解」を除去するのと同様、不詰作品の不詰順を求めるとき「紛別解」を除去すれば、もっときれいな不詰順が得られますが、不詰順に時間を掛けるのは勿体ないので、今のところ紛別解除の処理を入れる予定はありません。

10. トレース、モニタ機能と本プログラムの動作原理

(1) トレース、モニタ機能について

本プログラムには、プログラムの挙動を知るために「トレース」「モニタ」という2つの機能を備えています。これは出力先が「ファイルと画面の両方」か「画面のみ」かという違いだけで、ほとんど同じ機能です。「トレース」はしばしば出力ファイルのサイズを膨大にしてしまうので、普段は「モニタ」を使うことを推奨します。以下では「モニタ」の使用例とその見方について説明します。



初形図の下に「探索中」や「探索戻り」で表示されているのが「モニタ」の内容です。デフォルトではモニタの深度は「2」なので、2手目まで、どのような手を読んでいるかここに示されています。

横に数字の組が表示されていますが、それが何を意味しているか気にする必要はありません。興味のある方

は後述の解説をお読みください。

一般のユーザーにとって、注目すべきは作意と合致する手の出現頻度です。プログラムが最初のうちは迷走しているように見えても、だんだん作意順に読みが絞られてきたら、正解に近づいていることが推測できます。

1.07 版からは、検討の進み具合の目安として、「詰み易そうな手」を青色で、「詰みにくそうな手」を暗い色で表示するようにしました。ただし、これは固定ではなく、読みの進展により変化していきます。上図の例では初手「36 金」と「27 歩」が共に青色で表示されていますが、「36 金」は詰、「27 歩」は不詰なので、この段階では半分しか当たっていません。また、初手「27 歩」に対する 2 手目「25 玉」が暗色で表示されており、これは当たりです。検討が進むに従って、詰むはずの手が青色で、不詰のはずの手が暗色で表示されるようになったら、順調に検討が進んでいるものと考えてください。早詰チェックの工程では、作意に沿ったすべての手が不詰となるはずなので、青色は逆に危険信号となります。

【モニタ、トレースについての解説】（※興味のある方のみお読みください）

上の図を例として説明します。

まず 1 行目「探索中：1 手目 36 金」は、初手 36 金を読んでいることを表しています。その横に(1, 1)という数字の組がありますが、これは 36 金を指した後の局面に与えられている「証明数」と「反証数」の組を表します。現時点では 36 金から先はまったく読んでいないので、証明数と反証数は共に「不明」を表す「1」が与えられています。

今、説明なしに「証明数」「反証数」という言葉を使いましたが、証明数は「詰み」（あるいは目的達成）を証明するのに読む必要のある局面数、反証数は「不詰」（あるいは目的達成不可能）を証明するのに読む必要のある局面数としてみてください。

次に「→」の後に (8388607, 2) という数字の組が出てきました。これは証明数の閾値、及び反証数の閾値の組です。閾値というのは、この数に達したら読みを打ち切ろうという基準値を表します。つまり証明数の閾値が 8388607 というのは「36 金とした局面は証明数が 8388607 になるまで頑張っ調べてよう」ということを表し、反証数の閾値が 2 というのは「36 金とした局面は反証数が 2 になったら早々に出直そう」ということを表しています。最悪詰では攻方は「不詰」を証明する立場ですから、なるべく少ない読みで「不詰」を証明できると都合が良いので、その期待に基づいて、なるべく小さい閾値を反証数に割り当てているのです。ただし、この閾値は固定ではなく、読みの進展に合わせて変化していきます。

次の行は早々に探索から戻ってきたことを示しています。ここで証明数・反証数の組は (1, 4) となっています。実際 36 金に対しては「同角成」「同角生」「16 玉」「27 玉」の 4 つの受けがあり、攻方はそのどれもが詰まないことを示さなくてははいけません。仮にすべての受けが反証数 1 の範囲で詰まないとしても、反証数は 4 より小さくはなりません。そして「4」は反証数の閾値として与えた「2」よりも大きかったため、探索を切り上げて戻ってきたのです。

なお、「←」の後に (1, 1) という数字の組が出ていますが、これは探索を行う前の証明数・反証数の組の値を表しています。一般には証明数が大きくなり、反証数が小さくなったら「不詰」の可能性が強くなり、逆に証明数が小さく、反証数が大きくなったら「詰」の可能性が強くなります。ただし、その先の局面を読み進めると突然その傾向が逆転することがあり、あくまで現時点での詰みやすさ詰みにくさを評価した目安に過ぎません。

三行目は攻方の別の初手が出てきます。今度は 27 歩を選択し、閾値に (8388607, 5) を与えていますね。先程読んだ 36 金の反証数が 4 だったので、それより一つ大きい 5 を反証数の閾値に設定したわけです。

四行目では 27 歩に対し、25 玉を読んでいきます。これも最初は「不明」の局面なので(1, 1)の組なのですが、閾値は (2, 3) が与えられています。まずは証明数の閾値がたったの 2 であることに着目してください。最悪詰では受方は「詰み」を証明する側なので、なるべく証明数を小さく抑えたいわけですね。これは最初の 36 金で攻方が反証数の閾値を 2 にしたのと同様の理屈です。

一方、25 玉の反証数の閾値が 3 である理由は少し考えないと分からないでしょう。まず、初手 27 歩とした局面では反証数の閾値に既に 5 が与えられています。一方、この局面には今回選択する「25 玉」の他に「16 玉」と「37 玉」という 2 つの応手があります。それぞれの反証数が 1 だとしても、合計は 2。従って「25 玉」の反証数が 3 に達したら、上限の 5 に達してしまいます。というわけで、25 玉に対する反証数の閾値は 3 より大きくできません。その結果、閾値の組は (2, 3) となっています。

更に次の行は「探索戻り」となっています。これは先ほどの 25 玉を指した局面が、期待に反して「証明数 2」「反証数 3」の範囲内で解けなかったため、これを指す前の局面に戻ってきたことを示しています。そして、戻ってきたときの数字の組は(5, 1)に更新されています。証明数と反証数のどちらかが閾値に達したら戻ってくるわけですね。

実際 25 玉に対しては「26 歩」「34 角」「36 角」「35 金」「36 金」の 5 つの王手があります。だから、この局面の詰・不詰について結論を出すには、5 つの王手すべてについて「詰」を証明するか、どれか 1 つについて「不詰」を証明しないとはいけません。証明数と反証数の組が (5, 1) になったのはこのためです。

少し飛ばして 8 行目。「25 玉」も「16 玉」も打ち切りになって、この時点で「37 玉」を読むまでもなく、「27 歩」の読みを打ち切っています。これは「16 玉」の反証数が 9 で、「27 歩」に与えられた反証数の閾値である 5 を上回ってしまったからです。

そして 9 行目、再び初手 36 金を読んでいます。ただし、今度は反証数の閾値を 12 に設定し直しています。このように局面を前へ後ろへと移動しながら、証明数や反証数を徐々に変化させ探索範囲を広げていきます。

証明数・反証数や各閾値の更新方法には「df-pn」*¹と、そのアレンジ版「WPNS」*²を併用しています。合流やループの多い詰将棋は基本的に WPNS の方が向いていますが、問題の性質により df-pn の方が適している場合があるためです。

また、難問を解く場合、細かく閾値を上げていくのではなく、最初からある程度大きめの閾値を与えるという小細工 (1+ε Trick) *³なども使用しています。

さて探索が始まる方は見たので、今度は探索が終わるところを見てみましょう。

```
検討終了
探索戻り：1手目36金 (1, 19)←(1, 13)
探索中：1手目27歩 (1, 16)→(8388607, 22)
探索中：2手目16玉 (1, 14)→(2, 20)
探索戻り：2手目16玉 (0, 2147483647)←(1, 14)
探索戻り：1手目27歩 (0, 2147483647)←(1, 16)
探索中：1手目36金 (1, 19)→(8388607, 8388607)
探索中：2手目同角成 (1, 16)→(2, 8388607)
探索戻り：2手目同角成 (2, 21)←(1, 16)
探索中：2手目同角生 (1, 1)→(2, 8388607)
探索戻り：2手目同角生 (0, 2147483647)←(1, 1)
探索戻り：1手目36金 (0, 2147483647)←(1, 19)
詰みを見つけました
暫定的な解を表示します。
36金 同角生 27歩 16玉 34角 25金 同角 同角 26金 まで 9手
```

上の 4 行目に注目してください。「探索戻り：2 手目 16 玉 (0, 2147483647)」となっています。

証明数が 0 —— これは実際に証明ができたことを示します。(このとき反証数は理論上 ∞ (無限大) を割り当てるのですが、普通のプログラミング言語にはそんな数はないので、代わりにできるだけ大きい数を使っています。)それによって必然的に、初手 27 歩も詰みとなり、5 行目「探索戻り：1 手目 27 歩 (0, 2147483647)」が表示されています。実際初手 27 歩は以下 16 玉 34 角 25 金 同角 同角 26 金 まで 7 手で詰むので、確かにこの結果は正しいのです。

更に 11 行目、今度は初手 36 金も(0, 2147483647)となりました。初手 36 金も初手 27 歩も詰みだと分かったので、初形が詰みだということも判明しました。解図終了です。

不詰の場合は数字の組が(2147483647, 0)と逆になるので、**作意手順に(0, 2147483647)が現れ、詰まないはずの手順に(2147483647, 0)が現れば、順調に解図が進んでいる**と推測することができます。逆に、このような極端な数字が出ないうちは、プログラムが苦戦しているのだと思ってください。(8388607, 0)という数字の

組が現れることもありますが、これは「千日手」や「手数不足」が絡む「条件付の不詰」を示します。

また、証明数や反証数の閾値の最大値は 8388607 に設定しています。従って(8388607, 8388607)という数字の組が出始めたら、そこで読んでいる手以外は詰か不詰の結論が出たと思って良いでしょう。これも解図進行の一つの目安になります。早詰チェック以降の工程では、作意以外の手や、作意通りの手でも与えられた手数が短いと詰まないはずなので、ある程度調べた段階で、いきなり閾値を(8388607, 8388607)に引き上げることがあります。

[参考文献]

- ※1 長井 歩、今井 浩「df-pn アルゴリズムの詰将棋を解くプログラムへの応用」、VoL43, No6, 情報処理学会論文誌, June, 2002 (<http://ci.nii.ac.jp/naid/110002726401>)
- ※2 橋本 剛、上田 徹、橋本 隼一「オセロ求解へ向けた取り組み」、組合せゲーム・パズル ミニプロジェクト、第3回ミニ研究集会 2008年3月7日 (<http://www.alg.cei.uec.ac.jp/itohiro/Games/Game080307.html#anchor3>)
- ※3 Jakub Pawlewicz, Łukasz Lew「Improving Depth-first PN-Search:1+εTrick」、5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. (<http://www.mimuw.edu.pl/~pan/papers/lm-pns.pdf>)

(2) モニタの表示・非表示の切り替えと深度の変更

モニタを表示するか否かは「M」または「m」キーの押下で切り替えられます。モニタをしていない状態で「M」キーを押下すると以下の表示が出て、モニタが開始されます。

モニタ開始：2手

もう一度「M」キーを押下すると、以下の表示が出て、モニタが停止します。

モニタ停止

モニタ深度の初期値は「2」なので、プログラムが深い探索をしている場合は、何も表示されません。そこで、モニタ深度の変更を行う機能を使います。「+」キーの押下で深度を増やし、「-」キーの押下で深度を減らします。

モニタ開始：2手	(ここで + キーを押下)
モニタ深度：3手	(ここで + キーを押下)
モニタ深度：4手	(ここで + キーを押下)
モニタ深度：5手	(ここで - キーを押下)
モニタ深度：4手	(ここで - キーを押下)
モニタ深度：3手	

なお、この「深度」は開始手数からの相対的な手数を表します。

従って変別や非限定を検討している段階では、検討中の手数を足した手数の状況が表示されます。

例えば 5 手目の非限定の検討中にモニタ深度を 2 とすれば、5 手目から 7 手目の探索の状況が表示されます。1.15 版から「深度」の直接指定が可能になりました。「1」～「9」のキーを押下すると、「深度」がその数に応じて調整されます。

プログラムが現在どの付近を探索しているかは「D」キーの押下で途中経過表示することにより分かるので、それを参考に、モニタ深度を調整してください。

また、早詰チェック終了時、非限定・変別の各着手のチェックの終了時、画面表示が流れないようにモニタ表示は自動的にオフになります。モニタ深度は維持されます。

11. 性能の限界

本プログラムの使用は忍耐力との戦いです。

詰みを見つけるのはまあまあ早いのですが、早詰チェック以降は大抵多大な時間が必要です。

ルール固有の高速化ロジック（例えば「最善詰」における局面間の優越関係の利用）は入れていません。

特定のルールで有効なロジックが別のルールではまったく役に立たなかったり、むしろ有害な影響を及ぼしたりすることがあるからです。今後も高速化は検討していきますが、それはルールに依存しない範囲で行うか、オプションとして提供したいと思います。

どの程度の手数が扱えるかは作品の性質によるので一概には言えませんが、本プログラムのメインターゲットである「最悪詰」や「自玉詰」では、だいたい 50 手が実用上の限界とってください。短手数でも多大な時間が掛かるもの（裸玉や合駒物）もあれば、超長手数でも実用的な時間に収まるものもあります。

解図・検討に掛かる時間を前もって予測することはできませんが、モニタ機能を活用し、プログラムの挙動から解けそうかどうか判断してください。本プログラムには中断・再開機能があるので、性能の不足は大量の時間を投入することで補ってください。

12. 連絡先

本プログラムについて不具合の指摘、要望などがありましたら神無七郎 (k7ro.ts@gmail.com) までご連絡ください。掲示板 (<http://k7ro.sakura.ne.jp/cgi/upload/fairytalks.cgi>) でお知らせいただいても結構です。

99. 更新履歴

2024.9.24	1.15	<ul style="list-style-type: none">➤ 探索中の情報の出力機能追加➤ 自動待避オプション追加➤ モニタ深度の直接指定機能追加➤ 解析開始時コンソール画面の「簡易編集モード」を OFF に➤ 非王手可のとき攻方が持駒を譲渡する手を読まなかった不具合を修正
2024.5.24	1.10	<ul style="list-style-type: none">➤ 証明木作成機能追加
2024.4.2	1.07	<ul style="list-style-type: none">➤ 「非王手可」で達成目標が「詰」以外のとき、受方玉なしを許容➤ 証明数・反証数の増減によりモニタされる手の表示色を変更➤ 手順指定時「対称性検出」を無効化➤ 長手数余詰検出時、同手数の成・生非限定を「余詰」でなく「非限定」とした➤ 各種不具合修正及び内部処理の変更
2023.1.24	1.00	<ul style="list-style-type: none">➤ 初版公開

以上