Formal Analysis of Correctness of a Strategy for the KRK Chess Endgame

Fifth Workshop on Formal and Automated Theorem Proving and Applications.
Belgrade, February 3-4, 2012.

Marko Maliković
Faculty of Humanities and Social Sciences in Rijeka,
University of Rijeka, Croatia

Mirko Čubrilo
Faculty of Organization and Informatics in Varaždin,
University of Zagreb, Croatia

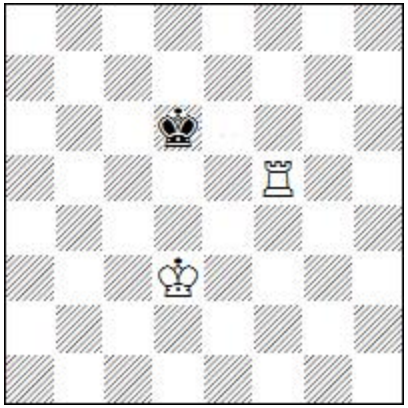
Predrag Janičić
Faculty of Mathematics in Belgrade,
University of Belgrade, Serbia

Motivation

- To our knowledge, there are no formalizations (within a proof assistant) of strategies for chess endgames or of correctness of these strategies (existing proofs of correctness are informal)
- We want to show that the game of chess can be suitable described within a relatively simple theory such as Presburger arithmetic (linear arithmetic over natural numbers) which is decidable (in contrast to the whole of arithmetic)

Problem

- ♔ King and Rook vs. King

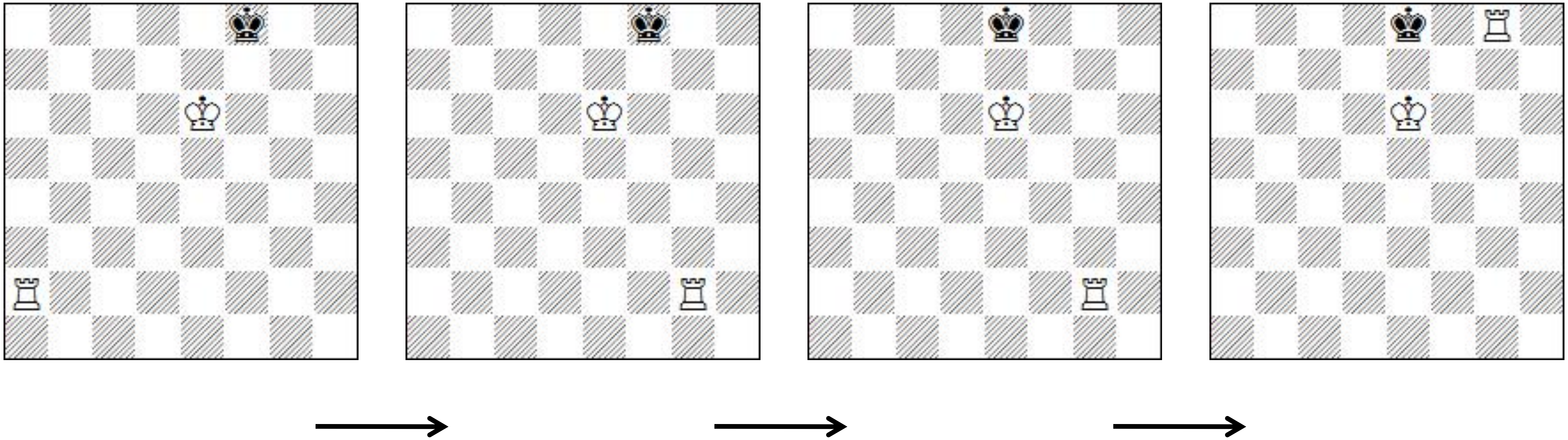


- ♔ There exists a strategy for white which always leads to checkmate (with regard to 50-rule)*

* Bratko, Ivan. Prolog programming for artificial intelligence. Harlow, England; New York: Addison Wesley, 2001.

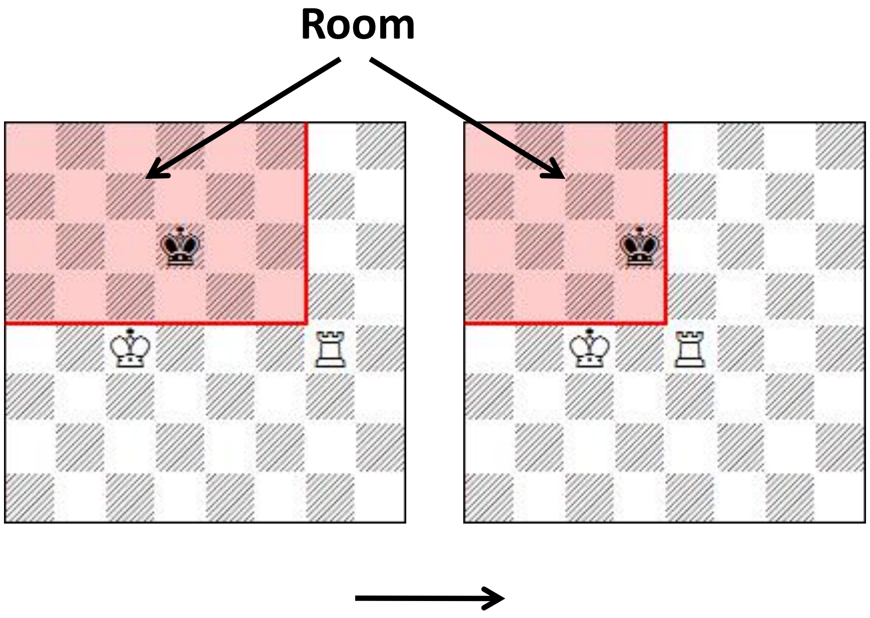
Strategy for KRK endgame by Bratko

1) **Mate in 2**: Look for a way to mate the opponent's king in two moves



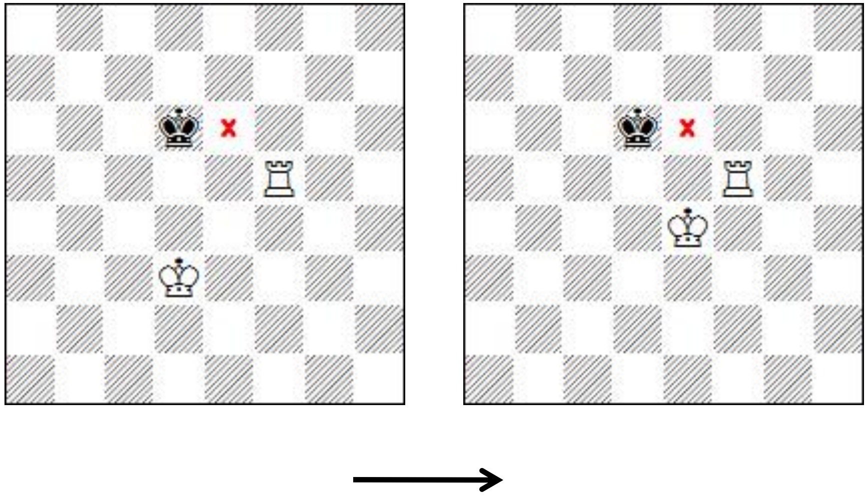
Strategy for KRK endgame by Bratko

2) **Squeeze**: If mate in 2 is not possible, then look for a way to constrain further the area on the chessboard to which the opponent's king is confined by our rook



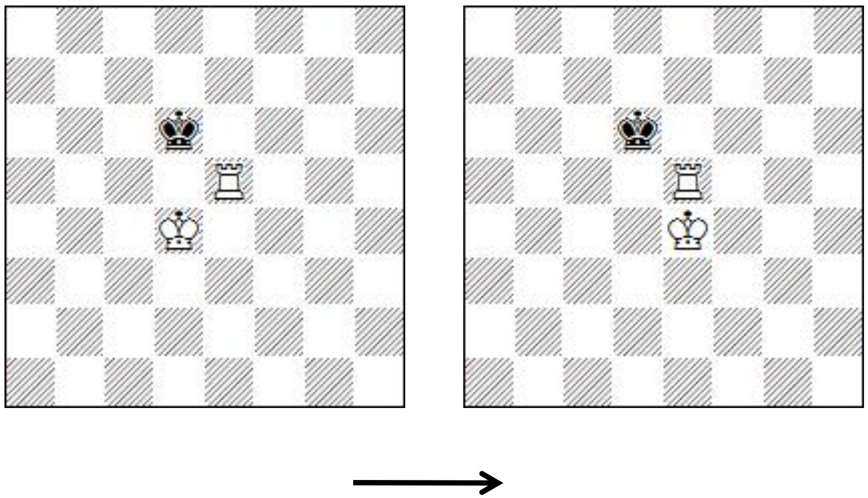
Strategy for KRK endgame by Bratko

3) **Approach**: If the above is not possible, then look for a way to move our king closer to the opponent's king



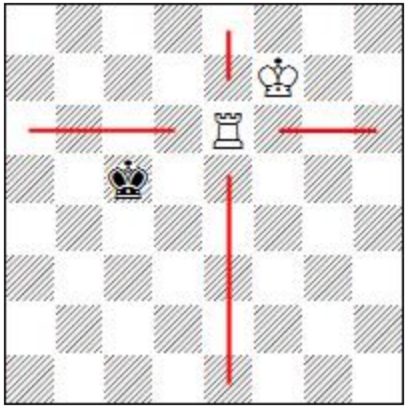
Strategy for KRK endgame by Bratko

4) **Keep Room**: If none of the above piece-of-advice 1, 2 or 3 works, then look for a way of maintaining the present achievements in the sense of 2 and 3 (that is, make a waiting move)



Strategy for KRK endgame by Bratko

5) **Divide**: If none of 1, 2, 3 or 4 is attainable, then look for a way of obtaining a position in which our rook divides the two kings either vertically or horizontally



More concrete descriptions of strategy

- Each step of the strategy is described in more detail by using more concrete predicates
- For example:
  - Squeeze:
    - Better goal: newroomsmaller /\
      ~ rookexposed /\
      rookdivides /\
      ~ stalemate
    - Holding goal: ~ rooklost

Outline of the informal proof of correctness by Bratko**

- Theorem: Starting from any legal KRK position, Bratko's strategy always leads to mate
- Proof of the above theorem is based on a number of lemmas
- The idea is:
  - to use the termination measure in position P that is equal to room in P plus distance of kings in P
  - to prove that such measure monotonically decreases if white uses Bratko's strategy

- **Bratko, Ivan. Proving correctness of strategies in the AL1 assertional language. Information Processing Letters, 7(5):223-230, 1978.

Outline of the informal proof of correctness by Bratko

- Bratko provides a completely informal proof
- For example, Bratko used the chess diagrams in which some claims are "obvious"
- Bratko does not even specify the theory in which he works

Our aims

1. Formalization of KRK endgame
2. Formalization of Bratko's strategy for white
3. Formal proof of the main theorem
4. It is necessary to prove many auxiliary lemmas
5. Some automation thanks to Omega - the available decision procedure for quantifier-free fragment of Presburger arithmetics

Formalization of the chess game in Coq
- Main sections -

- General definitions and declarations
- Metric on the chessboard (Chebyshev and Manhattan distance of the squares)
- Legality of the chess positions
- Specific positions (check positions)
- Moves of pieces and their legality
- Final positions (checkmate and stalemate)

Examples from formalization

♚ Legality of the chess positions:

- The coordinates of the chess pieces:
  - We use zero-based representation of rows and columns (all coordinates should be less than or equal to 7- more suitable then between 1 and 8)

- Kings can not be placed next to each other:

  Definition NotKingNextKing (P : Position) := WKx P > BKx P + 1 \/ BKx P > WKx P + 1 \/ WKy P > BKy P + 1 \/ BKy P > WKy P + 1.

- One and only one player is on turn in some position:

  Definition OnePlayerOnTurn (P : Position) := (OnTurn P = W \/ OnTurn P = B) /\ ~ (OnTurn P = W /\ OnTurn P = B).

Examples from formalization

- Specific positions:

Definition BlackKingAttackedByWhiteRook (P : Position)
:= WRx P = BKx P /\ (WKx P <> WRx P \/ WKx P = WRx P /\ (WKy P - BKy P = 0 /\ WKy P - WRy P = 0 \/ BKy P - WKy P = 0 /\ WRy P - WKy P = 0)) \/ WRy P = BKy P /\ (WKy P <> WRy P \/ WKy P = WRy P /\ (WKx P - BKx P = 0 /\ WKx P - WRx P = 0 \/ BKx P - WKx P = 0 /\ WRx P - WKx P = 0))

- Moves of pieces:

Definition MoveWhiteKing (P : Position) :=
Chebyshev (WKx P) (WKy P) (WKx (next P)) (WKy (next P)) = 1.

Examples from formalization

- Final positions:

  - Checkmate: Definition Mate (P : Position) :=
    BlackChecked P /\ ~ LegalMoveBlack P.

  - Stalemate: Definition Stalemate (P : Position) :=
    ~ BlackChecked P /\ ~ LegalMoveBlack P.

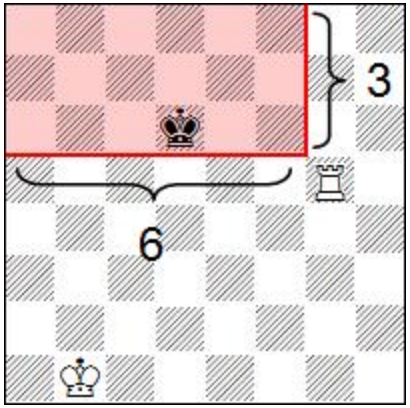
- All definitions given in terms of Presburger arithmetic

Formalization of mate in two moves

- Mate in two moves can be defined in terms of (minimax) search
- Search is not suitable for our needs (as it involves alternation of quantifiers)
- We solved this problem by explicit description of these positions (with the help of analogies and symmetries)

Formalization of steps of strategy - Example: NewRoomSmaller -

- Bratko used the multiplication to compute the Room
- Room = 3 x 6 = 18
- That approach is more natural



- We use the sum because it is:
  - sufficient
  - there is no multiplication in Presburger arithmetic
- Room = 3 + 6 = 9
- Thanks to:

$$\forall\ x_1, x_2, y_1, y_2 > 0,\ x_1 = x_2 \lor y_1 = y_2 \rightarrow (x_1 + y_1 < x_2 + y_2 \leftrightarrow x_1 * y_1 < x_2 * y_2)$$

Domain of Omega

- Solver of quantifier-free (actually universally quantified) problems in Presburger arithmetic
- Omega applied only to goals built from:
    - connectives: /\, \/, ~, ->
    - predicates: =, <, <=, >, >=
    - operators: +, -, *, pred, S, O
    - multiplication only if at least one the two multiplicands is a constant

Decision procedure of Omega

- Our problem can serve as a test case for exploring limits of automation for linear arithmetic in Coq
- Decision procedures of Omega is quite simple
- Alternatives: - using external SMT solvers and certificates
  - using bitvector arithmetic
  - reduction to SAT
- But, our intention was to try the simplest technique available within Coq
- It turned out that even simple decision procedures can be very useful, with additional techniques, in reasoning about non-trivial problems

How do we solve some problems

- Problem of subtraction:
  - For every expression of the form "minus x y" Omega generates two subgoals:
    - One for x < y where minus x y = 0
    - Second for x >= y where minus x y = x - y
- Can result in a memory overflow due to many generated subgoals
- Our solution:
  - We deal mainly with absolute difference
    - We defined function: AbsDiff x y = (x - y) + (y - x)
    - and we proved (and used only) the lemma about properties of AbsDiff:

      x <= y -> (AbsDiff x y) + x = y

      x > y -> (AbsDiff x y) + y = x

How do we solve some problems

- Problem of complex propositional structure of the (sub)goals
- e.g. omega can't prove this goal (memory overflow):

```
P : Position
H : AbsDiff (BKy P) (BKy (next P)) = 1
H4 : ~ (WRx P = BKx (next P) /\ (WKx P <> WRx P \/ WKx P = WRx P /\ (WKy P <= BKy (next P) /\ WKy P <= WRy P
        \/ BKy (next P) <= WKy P /\ WRy P <= WKy P)) \/ WRy P = BKy (next P) /\ (WKy P <> WRy P \/ WKy P = WRy
        P /\ (WKx P <= BKx (next P) /\ WKx P <= WRx P \/ BKx (next P) <= WKx P /\ WRx P <= WKx P)))
H6 : WKx P > BKx (next P) + 1 \/ BKx (next P) > WKx P + 1 \/ WKy P > BKy (next P) + 1 \/ BKy (next P) > WKy P + 1
H1 : WKx P > WRx P
H2 : WRx P > BKx P
n : nat
H0 : AbsDiff (BKy P) (BKy (next P)) = AbsDiff (BKx P) (BKx (next P)) + S n
H3 : BKy P <= BKy (next P) -> AbsDiff (BKy P) (BKy (next P)) + BKy P = BKy (next P)
H10 : BKy P > BKy (next P) -> AbsDiff (BKy P) (BKy (next P)) + BKy (next P) = BKy P
H5 : BKx P <= BKx (next P) -> AbsDiff (BKx P) (BKx (next P)) + BKx P = BKx (next P)
H9 : BKx P > BKx (next P) -> AbsDiff (BKx P) (BKx (next P)) + BKx (next P) = BKx P
============================
 WRx P > BKx (next P)
```

How do we solve some problems

- Our solution: Ltac functions for simplification of propositional structure of the goals which recursively:

• Eliminate implications in hypotheses:	A -> B converts into ~ A \/ B
• Eliminate negations:	~ (A /\ B) converts into ~ A \/ ~ B ~ (A \/ B) converts into ~ A /\ ~ B
• Eliminate conjunctions in hypotheses:	A /\ B destructs into A, B
• Eliminate disjunctions in hypotheses:	goal A \/ B |= C split into two subgoals: A |- C, B |- C
• Avoid split goal into subgoals:	in goals with hypotheses of form: A \/ B, A we clear hypothesis A \/ B

How do we solve some problems

- Irrelevant hypotheses in the goals

```
H : x1 < 7
H0 : x1 > 5
============================
x1 = 6

Time omega.
Proof completed.
Finished transaction in 0. secs (0.015u,0.s)
```

```
H : x4 < 5
H0 : x2 < 7
H1 : x3 - x2 - 3 + x4 < 7
H2 : x2 <= 8 \/ x2 = 4
H3 : x3 > 5
H4 : x3 - 2 = x2 -> x2 + x4 = 5
H5 : x4 < 5
H6 : x1 < 7
H7 : x3 - x2 - 3 + x4 < 7
H8 : x2 <= 8 \/ x2 = 4
H9 : x1 > 5
H10 : x3 - 2 = x2 -> x2 + x4 = 5
============================
x1 = 6

Time omega.
Proof completed.
Finished transaction in 26. secs (26.391u,0.s)
```

- Our solutions:
  - Cleaning of the goals of irrelevant hypotheses
  - Matching relevant hypotheses

Correctness proof of strategy - Main lemmas -

- For example:

  - Lemma 5. Starting to play from any position P in which white is on turn, satisfying (rookdivides P /\ ~rookexposed P \/ lpatt‘ P) /\ mdist (OK,CS,P) <= 2 /\ room P > 2 the KRK strategy forces Squeeze-ing in at most 3 moves

- To prove such lemmas we have to break them down into several sub-lemmas and to prove many auxiliary lemmas

Current Status

- ♔ What have we done so far:
  - We finished with the formalization:
    - of the core of the system (axiomatization)
    - of the strategy
    - of mate in two moves
  - We have solved most of the efficiency problems that arise due to the complexity of the system
  - We have formally proved (roughly) around half needed conjectures
- ♔ What still needs to be done:
  - To prove the remaining lemmas
  - To prove the main theorem
  - To generalize ways to solve efficiency problems

Future work

- We plan to consider:

  - Other types of endgames

    - For example King and Rook vs. King and Knight where exists strategy for black to draw

  - Related chess problems (e.g., retrograde analysis), within the same setting
  - Other techniques for using decision procedures for linear arithmetic within Coq