Solving program Jacobi equipped with constraints - a new tool to check proof games

Michel Caillaud, Nicolas Dupont, François Labelle

November 5, 2018

1 Introduction

The main proof game solving programs are Natch and Euclide for orthodox, and Popeye and Jacobi for fairy chess. The performance of such programs is largely determined by the amount and the quality of the deductions the program can make when analyzing a problem. No program has a complete set of deductions, and this situation probably won't change for the foreseeable future, given the sheer variety of possible maneuvers in proof games.

To mitigate this, Natch offers the “watch” option, and Euclide offers user-defined constraints, which can speed up solving in some cases, as shown in the programs' respective documentations. Recently, Jacobi 0.5.3 was equipped with its own system of user-defined constraints, which is more expressive than the constraints offered in Euclide 1.10. We briefly describe Jacobi's constraints below, and then we give six examples demonstrating their usefulness by verifying some proof games that stump solving programs otherwise.

Jacobi constraints let the user specify some or all of a piece's moves. One design goal was to be able to represent the “strategy” notation displayed by Natch and Euclide when solving a proof game, so the notation is somewhat similar. This common notation includes the “-” and “x” symbols, and a minimum number n of moves played by a piece is marked “(n)”.

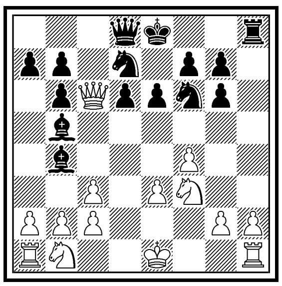
For extra flexibility, Jacobi offers more symbols. The arrow-like “>” is used to represent exactly one move, and “~” and “*” are counterparts to “-” and “x” but allowing captures along the way. This overview should be sufficient for this article, more precise definitions are given in the program's documentation.

We recommend the notation “HC+” (human-computer checked) for problems that are verified using human-derived constraints, no matter how “obvious” these constraints might be. This prevents a slippery slope in the meaning of “C+”, but we encourage the composers to use only constraints which are easily deductible from the diagram position, so that an “HC+” problem can almost be considered “C+”.

2 Six human-computer checked proof games

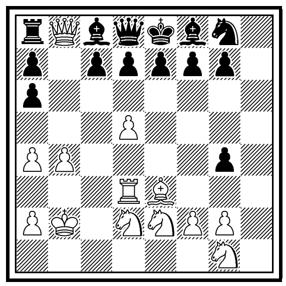
We first list the diagrams and stipulations (three orthodox and three fairies), then provide Jacobi's solutions (which are obviously the same as the published ones), and finally describe, for each problem, the constraints and the underlying reasoning leading to them. None of those problems is C+, mainly because they contain spare moves from both sides.

1
Joachim Iglesias
France-Echecs 2004



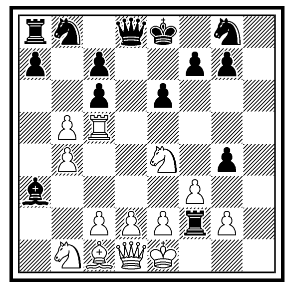
PG in 16.0 moves HC+ (14+15)

2
Unto Heinonen
7942v Die Schwalbe 1992 138



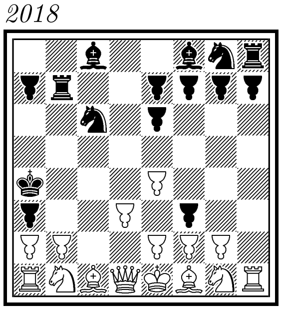
PG in 24.5 moves HC+ (13+14)

3
Satoshi Hashimoto
R164 Problem Paradise 2009 52



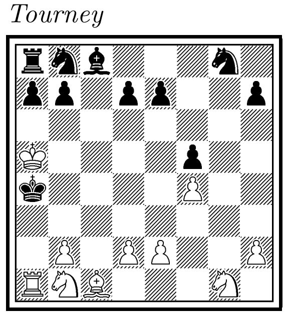
PG in 18.5 moves HC+ (13+14)

4
Nicolas Dupont
Murfatlar Tourney 2018



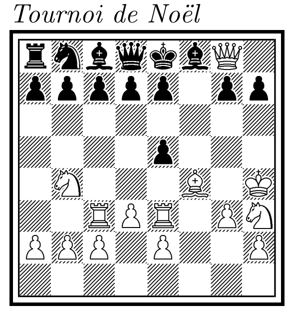
PG in 18.5 moves HC+ (15+15)
Lortap Berolina

5
Timothy Luffingham
Retros mailing List 1997
Quick Composing Tourney



PG in 23.5 moves HC+ (10+11)
Monochromatic

6
Nicolas Dupont
France-Echecs 2011
Tournoi de Noël



PG in 21.0 moves HC+ (14+14)
Isardam

Problem 1: 1.e3 e6 2.Bd3 Bb4 3.Bg6 hxg6 4.f4 Rh5 5.Qf3 Rc5 6.Kd1 Rc3 7.dxc3 d6 8.Bd2 Bd7 9.Be1 Bb5 10.Bh4 Sd7 11.Bf6 Rc8 12.Bd4 Sgf6 13.Bb6 cxb6 14.Ke1 Rc5 15.Qc6 Rh5 16.Sf3 Rh8

Problem 2: 1.d4 Sc6 2.d5 Sd4 3.h4 Sxe2 4.Rh3 Sc3 5.Ba6 h5 6.Rd3 Rh6 7.Be3

Rb6 8.Sd2 Rb3 9.cxb3 Sa4 10.bxa4 bxa6 11.Qb3 Bb7 12.O-O-O Bc6 13.Qb8 Bb5 14.b4 Bc4 15.Kb2 Bb3 16.Rc1 Bd1 17.Rc4 Bc2 18.Rg4 hxg4 19.h5 Bb3 20.h6 Bc4 21.h7 Bb5 22.h8=S Bc6 23.Sg6 Bb7 24.Sf4 Bc8 25.Sfe2

Problem 3: 1.a4 b5 2.axb5 Sc6 3.Ra6 Sd4 4.Rc6 dxc6 5.h3 Bg4 6.hxg4 e6 7.Rh5 Ba3 8.Rc5 h5 9.b4 hxg4 10.Bb2 Rh1 11.Qc1 Sf3+ 12.Kd1 Se1 13.Sh3 Rxf1 14.f3 Rf2 15.Sg5 Sd3 16.Se4 Se5 17.Ke1 Sd7 18.Qd1 Sb8 19.Bc1

Problem 4: 1.BPhf4 BPbd5 2.BPe5 BPe4 3.BPd6 BPf3 4.BPe4 BPa5 5.Qa4 BPb4 6.Kd1 BPa3 7.Kc2 Qc7+ 8.Kb3 Qxc1 9.Sc3 Qb1 10.BPc7 Qd3 11.BPxd3 BPde6 12.Kc2 Kd7 13.Kd2 Kc6 14.Qd1 Kb5 15.Sb1 Sc6 16.BPb8=B Ka4 17.Bf4 Rb8 18.Ke1 Rb7 19.Bc1

Problem 5: 1.f4 f5 2.Kf2 Kf7 3.Kg3 Kg6 4.Kh4 Kh5 5.g4+ Kxg4 6.Bh3+ Kxh3 7.a4 g5+ 8.Kxg5 Bh6+ 9.Kxh6 Kg2 10.Kg7 Kxh1 11.Kxh8 Kg2 12.Kg7 Kh3 13.Kh6 Kg4 14.Kg5 Kh5 15.Kh4 Kg6 16.Kg3 Kf7 17.Kf2 Ke6 18.Ke3 Kd5 19.c4+ Kxc4 20.Qb3+ Kxb3 21.Kd4 c5+ 22.Kxc5 Kxa4 23.Kb4 Qa5+ 24.Kxa5

Problem 6: 1.Sc3 Sh6 2.Sd5 Sg4 3.Sb4 Sxf2 4.Kxf2 Rg8 5.Kg3 Rh8 6.Kh4 Rg8 7.g3 Rh8 8.Bh3 Rg8 9.Be6 fxe6 10.Sh3 Kf7 11.Rf1+ Kg6 12.Rf3 Rh8 13.Rc3 Kf6 14.d3 Kf7 15.Qf1+ Ke8 16.Qf7 Rg8 17.Bf4 Rh8 18.Rf1 Rg8 19.Rf3 Rh8 20.Re3 Rg8 21.Qxg8 e5

In problem 1, a black Rook was captured on square c3, hence before Bc1 moved, and hence after the capture of Bf1 on square g6 (to open a door to a black Rook). It implies that Bc1 moved after e2-e3 was played. As this Bishop was captured on square b6, it needed a 6-long journey via square e1, implying in turn at least 2 moves played by Ke1. The related constraints deductible from the position (but not obvious for a solving program), and enough for Jacobi to prove the uniqueness of the solution are therefore:

Ke1(2) Bc1-d2-e1-b6(6)

In problem 2, the counting of white moves allows 5 Rook moves with castling. Sb1 moved before this castling, and hence it costs 3 moves for Rd1 to go to square d3. On the other hand, it is impossible for Rh1 to have been captured on the g-file in 2 moves, otherwise this Rook stood behind the h-file white Pawn, and can't therefore have been captured by the h-file black Pawn.

This implies that Rh1 went to square d3 via square h3 in 2 moves, while Rd1 (after castling) went to the g-file in 3 moves. The only such possibility is via the c-file (Be3 was played before 0-0-0) and, as Qd1 moved to square b3 also before 0-0-0, Rh8 was already captured and hence the h-file black Pawn stood on square h5 before capturing. The related constraints deductible from the position (but not obvious for a solving program), and enough for Jacobi to prove the uniqueness of the solution are therefore:

Rh1>h3>d3 Ra1>d1>c1>c4>g4

In problem 3, the Rh8 stood at one moment on square f1, and hence the white King must have escaped from check. The white move counting implies that this maneuver was at most 6-moves long. The only possibility was to play Ke1-d1 and to shield on square e1. As Ke1-d1 implies 2 moves from each of Bc1 Qd1 and Ke1, there was not enough time to shield with a white piece, e.g. Sg1.

The black side must therefore have put a piece on square e1, to allow Rf2 without non-parried check. This piece can't have been the black-squared Bishop, as it would have been in-cage after Rf2. The black move counting implies that the only remaining candidate is Sb8. The related constraints deductible from the position (but not obvious for a solving program), and enough for Jacobi to prove the uniqueness of the solution are therefore:

Ke1(2) Qd1(2) Bc1(2) Sb8-e1-b8

In problem 4, the capture BPe3xe4 was impossible as square e3 is observed by the immovable BPe2. Hence the capture BPd2xd3 was mandatory, which implies that Sb1 Qd1 and Ke1 each left their observation of square d2 (and then went back home in respectively at least 2, 2 and 6 moves), and that Bc1 must have been home-captured before BPd2xd3, and then replaced by a Pronkin promotion.

The captured black piece was Qd8, and the counting of black moves asserts that it is also this piece which captured Bc1. Finally the Pronkin Bc1 was clearly issued from the promotion of the BPh2. The related constraints deductible from the position (but not obvious for a solving program), and enough for Jacobi to prove the uniqueness of the solution are therefore:

Sb1(2) Qd1(2) Ke1(6) BPd2xd3 BPh2-c1=B Qd8xc1-d3

In problem 5, the missing white-squared Rh1 has been home-captured by the black King, as there is no other white-squared candidate. As square f3 is observed, the black King reached its destination via square h3, leading to a minimum of 16 moves. The same conclusion holds for the white King, but it is not enough for Jacobi to check the problem (in a reasonable amount of time).

Fortunately, some captured pieces moved during the solution: Pa2 Pc2 Pg2 Bf1 and Qd1 for white (because the black King, the only available capturing piece, would have been in self-check otherwise), Bf8 and Qd8 for black (for the same reason, and the fact that those pieces can't have been home-captured by Ra1). The related constraints deductible from the position (but not obvious for a solving program), and enough for Jacobi to prove the uniqueness of the solution are therefore:

Ke1*h8˜a5(16) Ke8*h1˜a4(16)

Pa2(1) Pc2(1) Pg2(1) Bf1(1) Qd1(1) Bf8(1) Qd8(1)

In problem 6, the Isardam condition prevents the possibility for Pf2 to have been captured by Pf7 (after having captured the missing black Knight on the e-file). It implies that Bf1 was captured on square e6, and the counting of white moves implies that the trajectory of each white piece is fixed, except for the Queen. The white moves which are not fully obvious are inputted as constraints:

Qd1xg8(3) Ra1>f1>f3>e3 Rh1>f1>f3>c3 Bf1>h3>e6

But it is not enough for Jacobi to prove the uniqueness of the solution (in a reasonable amount of time), mainly because nothing can easily be deduced concerning the black side. We thus use another possibility offered by Jacobi (already available in a previous release), the “addpieces” trick. It allows to more or less fix the white move order, which is easily deductible from the diagram position. For example the first step is to claim that Sg1 can’t have moved before the nineteenth ply. Indeed Sb1-c3-d5-b4 were played before the home-capture of Pf2, and then Ke1-f2-g3-h4, g2-g3, Bf1-h3-e6 were played before Sg1-h3 (see below the full “addpieces” trick description).

For the convenience of the readers who would like to test themselves the above examples, the respective Jacobi inputs are:

stipulation dia 16.0
forsyth 3qk2r/pp1s1pp1/1pQppsp1/1b6/1b3P2/2P1PS2/PPP3PP/RS2K2R
constraints Ke1(2) Bc1-d2-e1-b6(6)

stipulation dia 24.5
forsyth rQbqkbs1/p1ppppp1/p7/3P4/PP4p1/3RB3/PK1SSPP1/6S1
constraints Rh1>h3>d3 Ra1>d1>c1>c4>g4

stipulation dia 18.5
forsyth rs1qk1s1/p1p2pp1/2p1p3/1PR5/1P2S1p1/b4P2/2PPPrP1/1SBQK3
constraints Ke1(2) Qd1(2) Bc1(2) Sb8-e1-b8

stipulation dia 18.5
pieces
white Ke1 Qd1 Ra1h1 Bc1f1 Sb1g1 BPa2b2d3e2e4f2g2
black Ka4 Rb7h8 Bc8f8 Sc6g8 BPa7a3e6e7f3f7g7h7
condition lortap berolina
constraints Sb1(2) Qd1(2) Ke1(6) BPd2xd3 BPh2-c1=B Qd8xc1-d3

stipulation dia 23.5
forsyth rsb3s1/pp1pp2p/8/K4p2/k4P2/8/1P1PP2P/RSB3S1
condition monochromatic
constraints Ke1*h8~a5(16) Ke8*h1~a4(16) Pa2(1) Pc2(1) Pg2(1) Bf1(1) Qd1(1) Bf8(1) Qd8(1)

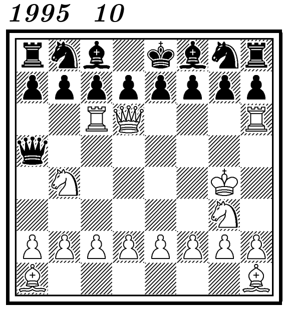
```
stipulation dia 6.0
pieces white Sg1 Pg2 addpieces
stipulation dia 3.0
pieces white Sg1 addpieces
stipulation dia 4.0
pieces white Pd2 addpieces
stipulation dia 8.0
forsyth rsbqkbQ1/ppppp1pp/8/4p3/1S3B1K/2RPR1PS/PPP1P2P/8
condition isardam
constraints Qd1xg8(3) Ra1>f1>f3>e3 Rh1>f1>f3>c3 Bf1>h3>e6
```

3 Conclusion

Proof game solving programs are great tools, but are not currently able to check problems with many invisible moves from both sides. The examples above show that inputting human deductions may sometimes help to overcome this obstacle. We hope that Jacobi will offer new such possibilities in a near future.

Finally note that, even when Jacobi is unable to claim "HC+", it might nevertheless help the composer in various manners. For example, inputting almost all the moves and getting uniqueness of the related solution, shows at least that the move order is correct. This trick is useful for a partial checking of very complicated problems:

7
Michel Caillaud
2856 Probleemblad
1995 10



PG in 22.0 moves C? (16+16)
Circe

With the input:

```
stipulation dia 22
forsyth rsb1kbsr/pppppppp/2RQ3R/q7/1S4K1/6S1/PPPPPPPP/B6B
condition circe
```

constraints Qd1xd6 Ra1-a6xc6 Pa2xb3~ Pb2xa3~ Pd2xe3~ Pe2xd3~ Pg2xh3~ Ph2xg3~ Qd8(15) Bc8(5) Pc7-c6~ Pd7-d6~

Jacobi is able to show that the solution fulfilling those constraints is unique:

1.Sc3 c6 2.Sd5 Qc7 3.Sb4 Qg3 4.hxg3 [+bQd8] d6 5.Rh6 Bh3 6.gxh3 [+bBc8] Be6 7.Bg2 Bb3 8.axb3 [+bBc8] Qb6 9.Ra6 Qe3 10.dxe3 [+bQd8] Qa5 11.Bh1 Qa3 12.bxa3 [+bQd8] Qd7 13.Bb2 Qxh3 [+wPh2] 14.Ba1 Bf5 15.Qxd6 [+bPd7] Bd3 16.exd3 [+bBc8] Qxg3 [+wPg2] 17.Ke2 Qg6 18.Kf3 Qxd3 [+wPd2] 19.Se2 Qxb3 [+wPb2] 20.Sg3 Qxe3 [+wPe2]+ 21.Kg4 Qxa3 [+wPa2] 22.Rxc6 [+bPc7] Qa5