Basic modeling in ASP and more via the n-Queens puzzle

Torsten Schaub[*]
torsten@cs.uni-potsdam.de
http://potassco.sourceforge.net/videos.html

May 18, 2014

Contents

[*]based on encodings by Roland Kaminski (BIG thanks!)

1 The problem

- Problem statement:

  Place n queens on an n x n chess board such that no two queens attack one another

- `http://en.wikipedia.org/wiki/Eight_queens_puzzle` (here n=8)

- Example: n-Queens puzzle, Section 3.2 and 8.1 in [1]

2 The systems

- Commands

  gringo4 –version clasp3 –version

3 Basic encoding

- Example: n-Queens puzzle, Section 3.2 in [1]

  – **Note** [1] uses language of gringo 3, while we use the language of gringo 4 (cf [2]).

- Generate and Test Methodology okular –presentation methodology.pdf

3.1 Playground

- Files

  view-file queensB.lp4 [3]

[1]M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub: Answer Set Solving in Practice. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool December 2012, 238 pages, `10.2200/S00457ED1V01Y201211AIM019`

[2]F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, F. Ricca, and T. Schaub: ASP-Core-2: Input language format. 2012. Available at `https://www.mat.unical.it/aspcomp2013/files/ASP-CORE-2.0.pdf`.

[3]We use extension lp4 to indicate encodings for gringo 4 (along the ASP-Core-2 standard [2])

  - queen(X,Y) indicates a queen on position (X,Y)

- Commands

  gringo4 queensB.lp4 -c n=4 | clasp3 gringo4 queensB.lp4 -c n=4 | clasp3 0

  gringo4 queensB.lp4 -c n=4 –text gringo4 queensB.lp4 -c n=4 –text | wc

  gringo4 queensB.lp4 -c n=13 | clasp3 gringo4 queensB.lp4 -c n=14 | clasp3

3.2 NB: Keep in mind that programs are normalized

- Files

  view-file queensB.lp4 view-file queensBB.lp4

- Commands

  gringo4 queensB.lp4 -c n=4 | clasp3 –stats gringo4 queensBB.lp4 -c n=4 | clasp3 –stats

  - Note: Both encodings result in the same constraints!

3.3 NB: You may as well want to try gringo 3 for more (grounding) options:

- Files

  view-file queensB.lp

- Commands

  gringo3 queensB.lp -c n=4 –text gringo4 queensB.lp -c n=4 –text

4 Advanced encoding

- Example: n-Queens puzzle, Section 8.1 in [1]

  - Note: [1] uses language of gringo 3, while we use the language of gringo 4 (cf [2]).

- Files

  view-file queensB.lp4 [3] view-file queensA.lp4 [3]

- Commands

```
gringo4 queensB.lp4 -c n=2 –text gringo4 queensA.lp4 -c n=2 –text
```

```
gringo4 queensB.lp4 -c n=14 | clasp3 gringo4 queensA.lp4 -c n=14 |
clasp3
```

```
gringo4 queensB.lp4 -c n=14 | wc -l gringo4 queensA.lp4 -c n=14 | wc
-l
```

```
gringo4 queensB.lp4 -c n=14 | clasp3 –stats gringo4 queensA.lp4 -c
n=14 | clasp3 –stats
```

4.1 NB: An even more succinct but identical advanced encoding

- Files

```
view-file queensA.lp4 view-file queensAA.lp4
```

- Commands

```
gringo4 queensAA.lp4 -c n=14 | clasp3 –stats gringo4 queensA.lp4 -c
n=14 | clasp3 –stats
```

- Note: Both encodings result in the same constraints!

4.2 NB: Keep in mind that clasp partly unfolds cardinality constraints

- Commands

```
gringo4 queensA.lp4 -c n=14 | wc -l gringo4 queensA.lp4 -c n=14 |
clasp3 –stats | grep Rules
```

```
gringo4 queensA.lp4 -c n=14 | clasp3 –stats –trans-ext=dynamic
gringo4 queensA.lp4 -c n=14 | clasp3 –stats –trans-ext=no gringo4
queensA.lp4 -c n=14 | clasp3 –stats –trans-ext=all
```

4.3 NB: You may as well want to try gringo 3 for more (grounding) options:

- Files

```
view-file queensA.lp
```

- Commands

```
gringo3 queensA.lp -c n=4 –text gringo4 queensA.lp -c n=4 –text
```

5 Corrupted encoding

- Files

  view-file queensX.lp4 [3]

- Commands

  gringo4 queensB.lp4 -c n=14 | wc -l gringo4 queensA.lp4 -c n=14 | wc -l gringo4 queensX.lp4 -c n=14 | wc -l

6 Declarativity versus Scalability

- **Declarativity**

  - ASP does separate a problem's representation from the algorithms used for solving it

- **Scalability**

  - Modeling ASP does not separate a problem's representation from its induced combinatorics
  - Solving Boolean constraint technology is rather sensitive to search parameters

- **Challenge** Source code optimization! [4]

7 More advanced encoding

- Example: n-Queens puzzle, Section 8.1 in [1]

  - Note: [1] uses language of gringo 3, while we use the language of gringo 4 (cf [2]).

- Files

  view-file queensA.lp4 [3] view-file queensApre.lp4 [3]

[4]M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub: Challenges in Answer Set Solving. Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday: 74–90. Springer, 2011 Available at `http://www.cs.uni-potsdam.de/wv/pdfformat/gekakasc11a.pdf`

- Commands

  gringo4 queensA.lp4 -c n=300 | clasp3 –quiet gringo4 queensApre.lp4 -c n=300 | clasp3 –quiet

  gringo4 queensA.lp4 -c n=300 | clasp3 –quiet –stats gringo4 queensApre.lp4 -c n=300 | clasp3 –quiet –stats

- Note: Both encodings result in the same constraints!

8 Constraint-based encoding

- **NOTE** This is an experimental feature!
- Background
  - feature
    - express finite linear constraint satisfaction problems within ASP's modeling language and
    - solve them with off-the-shelf ASP solvers
  - language
    - All constraint relations and constraint variables are (currently) preceded by a dollar symbol
    - #disjoint{ term : value }
      - Idea: Sets of values labeled with the same term(s) must be disjoint :)
  - order encoding [5]
    - Idea: Introduce a Boolean variable for each statement 'X <= k' where X is a variable over integers and k is an integer bound
- Files

  view-file queensC.clp view-file queensB.lp4 [3] view-file queensApre.lp4 [3]

[5]N. Tamura, A. Taga, S. Kitagawa, M. Banbara: Compiling finite linear CSP into SAT. Constraints: 14(2):254-272, 2009. Available at `http://springer.r.delivery.net/r/r?2.1.Ee.2Tp.1gRdFJ.BxsAdG..N.HAQa.38pS.CLWEcC00` (see also `http://bach.istc.kobe-u.ac.jp/sugar`)

- Commands

```
gringo4 queensC.clp -c n=10 | clasp3
```

```
gringo4 queensApre.lp4 -c n=300 | clasp3 –quiet gringo4 queensC.clp
-c n=300 | clasp3 –quiet
```

```
gringo4 queensApre.lp4 -c n=300 | clasp3 –stats –quiet gringo4 queensC.clp
-c n=300 | clasp3 –stats –quiet
```

```
gringo4 queensC.clp -c n=300 | clasp3 –stats –quiet | grep Constraints
gringo4 queensC.clp -c n=300 | clasp3 –stats –quiet –sat-prepro | grep
Constraints
```

```
gringo4 queensC.clp -c n=300 | clasp3 –quiet gringo4 queensC.clp -c
n=300 | clasp3 –quiet –sat-prepro
```

9 M

- Commands

```
gringo4 queensC.clp -c n=1000 | clasp3 –stats –quiet gringo4 queen-
sApre.lp4 -c n=1000 | clasp3 –stats –quiet gringo4 queensApre.lp4 -c
n=1000 | clasp3 –stats –quiet –configuration=jumpy
```