Final Project - Evaluation for Clingo ASP Solver

Course: EECS 4401 – Section M – Term W

Team Member #1 (Student #): Muhammad Adeel Zafar (216047144)

Team Member #2 (Student #): Amirhossein Razavi (216715963)

Team Member #3 (Student #): Arya Mazloomi (217418609)

Team Member #4 (Student #): Akaf Khurshid (217181041)

Introduction (Common)

Clingo is an ASP system that offers users the ability to ground and solve logic-based problems. ASP here stands for Answer Set Programming and acts as a modelling language that allows users to solve combinatorial problems. Problems that arise in a number of different facets of computer science, such as finding models of propositional formulae and finding the shortest or cheapest paths in certain instances. Overall, combinatorial problems mostly involve the grouping and ordering of various objects in a discrete set of objects, under explicitly defined conditions.

In this project, the team went through and evaluated the utility this system offers by making use of it to solve a number of such problems to assess its limitations and capabilities. The types of problems used involve varying degrees of complexity and practicality, as will be discussed in the next section.

Methodology (by Amirhossein Razavi)

Given that we have tested and gone over the capabilities of a given tool, we can justifiably say that our methodology of analysis is Observational. Although our nature of testing involved us providing a variety of problems for Clingo to solve, we collected data regarding the types of solutions the system presented us with and checked the validity of these provided solutions alongside the resources required by the software in solving them.

Software

We used the online version of Clingo made available on the Potassco website for our tests. [1] The software itself offers several features which we used to conduct our testing, ranging from the ability to use pre-built problems of variable difficulty to allowing custom problems to be defined for the software to solve, all the way to offering different configurations for solving a given problem.

Procedure

The team's procedure for conducting the tests was simple in design. A selection of problems was chosen that were intended to be used to test Clingo and its various features. The problems chosen were of varying complexity to assess the limitations of the software properly, as well as having two versions of some of the problems but of escalating complexity to determine the relationship between complexity and results. Also, each of the chosen problems were ran on the four different running modes (default, brave, cautious, and enumerate all) of Clingo and the results were used to collect data on the following criteria to satisfy the analysis of the software:

- Time taken to solve the problem.
- Accuracy of resulting solutions.
- Data provided regarding a given problem.
- Resources used by the software in solving the problem (in terms of memory allocation and/or CPU usage).

The team also examined the usability of Clingo in terms of syntax (how hard it would be to learn the syntaxes), ease of use (how easy would it be to encode a problem), and understanding the results it gives using more general modeling languages as a contrasting point. Since we are aware that Clingo offers a wide selection of features that aren't necessarily present in many other modeling languages, the authors assessed the effects of these extra features on the software's performance and usability as well.

The team also contrasted these results with another ASP system called SModels, which helped us see where Clingo stands among similar modeling languages. The team chose specific problems from the set of problems we used for Clingo based on a variety of criteria and then had the other software attempt to solve them as well. This allowed a measure of comparison between the different datasets and solutions, allowing us to distinguish between the strengths and weaknesses of Clingo.

Problems

For this project, the team used eight problems to test the capabilities of Clingo and used some of them in the SModels to compare their results with each other. The eight problems are as follows:

1. Guilty or Innocent
2. Guilty or Innocent (Modified)
3. Flying Birds
4. N-Queens
5. Four-Color Theorem
6. Logic Puzzle Problem #1
7. Logic Puzzle Problem #2
8. Restaurant

Most of the problems chosen are well-known examples for modeling languages, resulting in the possibility of expanding this report to other logic programming languages. All the problems in logic programming present a set of facts and rules about some problem domain to the system and ask the system to provide the answers it finds, but why did we use eight problems to test Clingo?

These eight problems had specific reasons to be used in order to test Clingo. The original "Guilty or Innocent" and "Flying Birds" problems were included in our test set to see how Clingo performs when the given problem has only one possible answer set and no other options. The "Guilty or Innocent (Modified)" problem is to see how the performance metrics such as time and memory/CPU usage change regarding the original version of the problem where the problem is very simple. The "N-Queens" and "Four-Color Theorem" problems help us see how the Clingo gets affected by having many answer sets. The "Logic Puzzle #2" and "Restaurant" problems are in our examples to test for problems that the Clingo might encounter on a more general level, when presented with scenarios that may be found in games or books. "Logic Puzzle #1" does the same, but it also tests to see if there are any effects on Clingo's performance when the problem incorporates a vast amount of code.

<u>Syntax and Ease of Use (by Akaf Khurshid)</u>

ASP (Answer Set Programming), is a technique based on logic programming that allows us to define a generative space in abstract terms, then iteratively place constraints on this generative space to change the output to what we want. This method is oriented towards combinatorial search problems, where the goal is to find a solution among a large, but finite, number of possibilities. An example of a simple program in Clingo is:

```
{a}.
b :- a.
{c} :- not a.
d :- not b, not c.
:- c, not a.
```

Running Clingo to find answers by using this command:

```
! clingo example.lp 0
```

Which example.lp contains the code and 0 is the number of answer sets. By default, Clingo computes 1 answer set. If 0 is replaced by another number greater than zero (e.g. N), Clingo will compute N answer sets.

After executing, The output will be:

```
Reading from example.lp
Solving...
Answer: 1
d
Answer: 2
a b
SATISFIABLE

Models      : 2
Calls       : 1
Time        : 0.002s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time    : 0.001s
```

It should be noted that there is no way to tell Clingo in which order to print the answer sets.

	True	False	If	And	Or	Iff	Negation	Strong Negation
Clingo			:-	,	;		not	-
Logic	⊤	⊥	→	∧	∨	↔	¬	~

Table 1 (Conventions)

As seen with the example above it can be seen that Clingo can have multiple answers where each answer is a stable model for the constraints that the program gives. To define a problem in Clingo it is crucial to know the syntax as it allows users to code the problem in Clingo without potential misrepresentation. To define the knowledge base for the program such as ‘penguins can swim’ or that ‘q’ exist within the knowledge base in Clingo write:

1. swim(penguins).
2. q.

Line 1 swim(penguins).’ states that there is a function ‘swim’ and that function contains ‘penguins’ through this the program has defined that ‘penguins can swim’ in Clingo. Line 2 states that there is ‘q’ therefore ‘q’ is a fact for the program. Note that functions and constants cannot start or have uppercase letters within them, as that defines a variable in Clingo that will be explained later on. Also note that every statement in Clingo needs to end with ‘.’, this allows Clingo to know when the start and the end of a statement. Just defining the knowledge base doesn’t represent a problem as there are no rules defined. Rules use the knowledge base to find more facts/atoms for a given program, such as:

```
1. q.
2. p :- q.
```

Line 1 states that ‘q’ is in the knowledge base, line 2 defines that there is ‘p’ if there is ‘q’, ‘:-’ is the if condition on Clingo. ‘p’ is the head for the rule. This is the same as ‘q implies p’ from logic, with line 2 Clingo will derive the atom ‘p’, making the answer for this program:

```
q p
```

There can be multiple conditions for a rule that are separated by a ‘,’ this means ‘and’ within the condition. Such as:

```
1. swim(penguins).
2. bird(penguins).
3. -fly(X) :- swim(X), bird(X).
```

Line 1 and 2 states ‘penguins’ is in ‘swim’ and ‘bird’. In line 3 states ‘swim AND bird implies -fly’ to write this in Clingo we need to use a variable, the variable used is ‘X’ for this program. ‘X’ is a variable that is in ‘swim’ and ‘bird’ then ‘X’ is in ‘-fly’ as well.

Another type of rule that can be defined in Clingo is *Choice Rule*, which allows the program to have disjunction in the head of the rule, to write choice rule in Clingo such as ‘the knowledge base may or may not contain ‘p’, if the knowledge base contains ‘p’ then ‘q’ may or may not be an atom’, choice rule are written with curly braces around the head:

```
1. {p}.
2. {q} :- p.
```

Line 1 states that ‘p’ is a choice rule, hence there is an answer set that does not contain ‘p’ and another answer set that does contain ‘p’, if line 1 was the whole program then Clingo will give the following as the answer to the program:

```
Answer: 1

Answer: 2
p
```

This is the same as ‘the knowledge base may or may not contain ‘p’’, with line 2 ‘{q} :- p.’ it is defined that ‘p implies {q}’ the choice rule ‘{q}’, therefore there are two answer sets produced from this rule, one without ‘q’ and one with ‘q’. together with the last answers there is:

```
Answer: 1

Answer: 2
```

```
p
Answer: 3
p q
```

There are only 3 answer sets as the second choice rule branches off the first one.

With multiple choice rules the answer sets increase proportionally to the number of choice rules, however not all the answer sets produced are valid for the problem that is being coded. *Integrity Constraints* solve this problem by putting constraints on the answer set produced by Clingo, this is done by writing the constraint/s after ':-'. Such as:

```
1. {p}.
2. {q}.
3. :- p, q.
```

In this program there are two choice rules, without line 3 the answer sets are:

```
Answer: 1

Answer: 2
p
Answer: 3
q
Answer: 4
q p
```

However with line 3, it is defined that 'p AND q' cannot be in the answer set, thus only having the first 3 answers the program.

There can be multiple variables in a function, this helps define more complex problems such as graphs in Clingo.

```
1. city(a).
2. city(b).
3. road(X,Y) :- city(X), city(Y), Y!=X.
```

This program states that is city a, city b and if there are two cities that are different then there is a road between them. Another way to declare city a and city b is 'city(a;b).' with a and b separated by a ';'. Line 3 defines that 'X' from city, 'Y' from city and 'Y' is not that same as 'X' then there is a road from 'X' to 'Y'. Thus the answer set for this program is:

```
Answer: 1
city(a) city(b) road(b,a) road(a,b)
```

Note that the answer set has all the atoms for the program, however this can be hard to read when looking for something, for instance if there are 100 cities then it will be hard to read the answer set. To remedy the program can states that only show the roads in the answer set, to do this the the program will have '#show road/2.'. This will show all roads with two predicates stated by the 'road/2':

```
Answer: 1
road(b,a) road(a,b)
```

Results of Tested Problems (by Muhammad Adeel Zafar)

Over the course of the team's testing, several key points were made incredibly apparent. Both in terms of the effectiveness of Clingo, but also the speed it employs and the ease of its use in contrast to other similar software.

In this section, we'll be discussing the effectiveness of Clingo, as opposed to its ease of use which was discussed in the previous section. In order to test the effectiveness of Clingo, the team utilised a wide variety of different test problems. These problems ranged from the example problems provided by Clingo itself, modified versions of those problems and then more complex problems that we found and implemented that were not present in the provided example set.

As seen in the appendix, one of the most distinct points to consider when it comes to deciding on how effective Clingo is falls to how quickly it is able to solve given problems. As such, our first value of note in the results for each problem and each mode of Clingo was the time taken for Clingo to return the answer set. The team tested the time needed by Clingo through problems of varying complexity, beginning with something simple like the *Guilty or Innocent Problem*, then increasing the complexity of said problem, all the way to replicating a logic puzzle for Clingo to work through in *Logic Puzzle Problem #1*. In these three example cases, the complexity of the problems being solved undoubtedly increased, given that the number of rules and atoms grows exponentially. We go from a meagre 4 rules and atoms in the original *Guilty or Innocent Problem*, to 64 rules and 51 atoms in the modified version of the same problem, all the way to 5024 rules and 1266 atoms in *Logic Puzzle Problem #1*.

In order to test whether different types of changes in complexity caused any substantial differences in efficiency, we started simple with basic problems like the *Guilty or Innocent Problem* or the *Flying Birds Problem*. Upon finding that Clingo had no issues with solving these types of simple problems in a miniscule amount of time (less than 1 second), we moved on to using choices as a means of increasing complexity. By providing Clingo with choices in our problems, as seen in the modified version of the *Guilty or Innocent Problem*, we began to escalate the model-level and the number of rules and atoms. However, despite even massive leaps in these given values, Clingo continued to perform with no real issues, offering results in under a second in most cases. In fact, the only time we pushed Clingo anywhere close to a tenth of a second was in the case of *Logic Puzzle Problem #1*, where Clingo took 0.096s to find a solution. This was also the only case where the software took long enough to warrant a distinct output of 0.01s on both its solving time and its 1st model time, whereas in all other problems tested the results had shown to be zero due to the time taken being too minor for Clingo to consider noteworthy. Due to this, it was near impossible to judge how much of the time being used was allocated to which parts of finding a solution.

With complexity not being an issue for Clingo to handle, the team moved to a different avenue of evaluation. Next, we tested the different modes Clingo had to offer so that we could assess if there were any possible cases where these modes would be ineffective, or perhaps simply be too tedious for users to make use of. This is where we first ran into a potential weakness Clingo displayed which was not as apparent in its default running mode. When it comes to problems with particularly large answer sets, such as the *N-Queens Problem*, Clingo's 'Brave' or 'Enumerate All' modes end up listing all possible answers in an instant. In such a case, it becomes incredibly difficult for users who may be searching for a specific answer case to sort through all the possible cases in order to find a single distinct solution.

Additionally, there are also cases where certain modes of Clingo may crash due to the software reaching some internal quota of resources being used. As is the case in the ‘Enumerate All’ mode for the *Four-Color Theorem Problem* where the browser simply crashes, very likely due to the sheer amount of possibilities that Clingo has to process. This is further confirmed by the fact that Clingo has no issue presenting users with solutions to the problem itself in its other modes. It is only in one specific mode that the software seems to crash, highlighting a weakness in that specific type of scenario, regardless of how niche it may be.

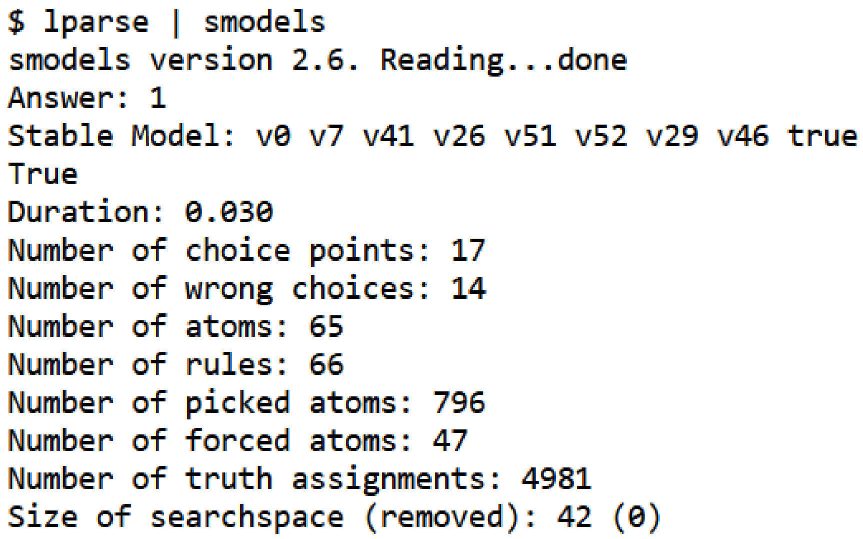
As far as data presented to users through results is concerned, however, Clingo is extremely detailed in the statistical results it provides, if the user so requests these results. The software provides ample information ranging from the more expected details such as number of atoms, rules and constraints to the more nuanced facets of data such as the amount of loops or backjumps. Overall, the statistical data provided by Clingo is incredibly powerful and detailed, allowing users access to knowledge that would in many cases be unavailable.

Comparing Clingo to other Software (by Arya Mazloomi)

The program SModels is an implementation of the stable model semantics for logic programs. SModels can be used either as a C++ library that can be called from user programs or as a stand-alone program together with a suitable front-end.

The main front-end is Lparse. Lparse is a front-end that adds variables to the accepted language and generates a variable-free simple logic program that can be given to SModels. Lparse also implements several other semantics (classical negation, partial stable models) by translating them into normal logic programs.

Assume that the file program contains a logic program. A stable model would typically be computed as follows:



```
$ lparse | smodels
smodels version 2.6. Reading...done
Answer: 1
Stable Model: v0 v7 v41 v26 v51 v52 v29 v46 true
True
Duration: 0.030
Number of choice points: 17
Number of wrong choices: 14
Number of atoms: 65
Number of rules: 66
Number of picked atoms: 796
Number of forced atoms: 47
Number of truth assignments: 4981
Size of searchspace (removed): 42 (0)
```

Logic programs in SModels format consist of:

- Normal Rules
- Choice Rules
- Cardinality Rules
- Weight Rules
- Optimization Statements

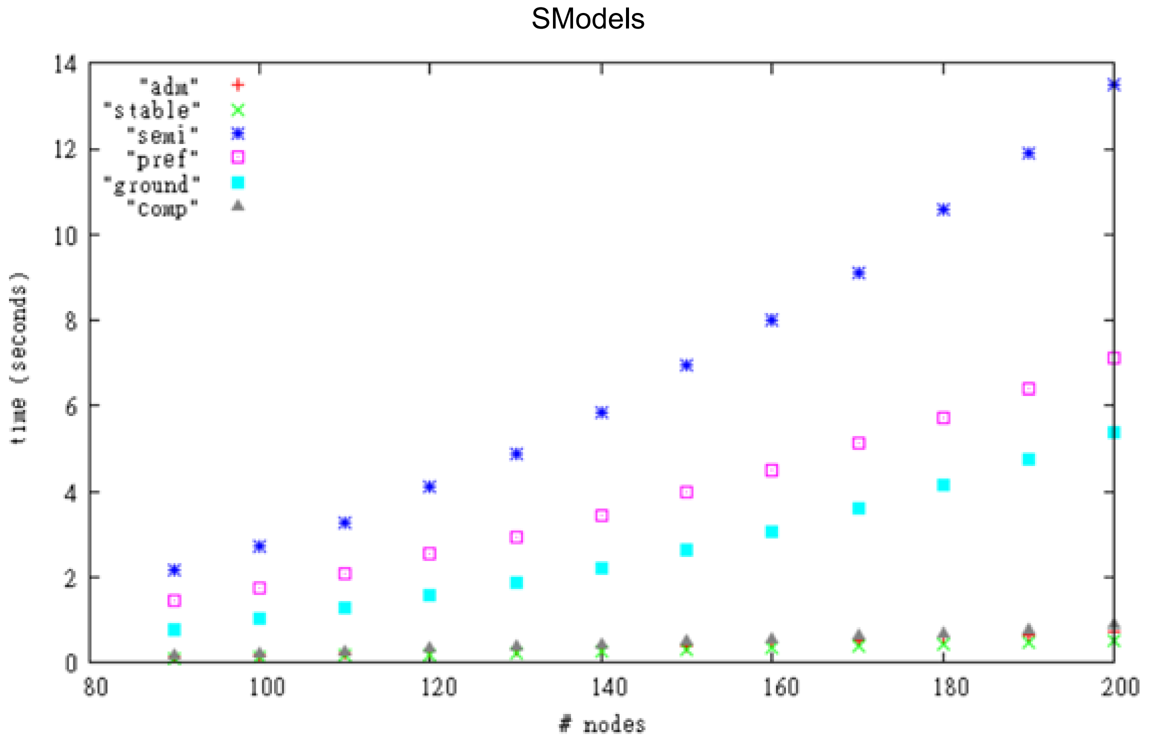
Such a format is obtained by grounder Lparse.

Below is a table which shows the running time of SModels + Lparse vs Clingo, in seconds, for the N-Queens Problem.

N - Queen Problem	Clingo	SModels
15-queens	0.049 s	0.62 s
16-queens	0.046 s	0.102 s
17-queens	0.050 s	0.352 s
18-queens	0.054 s	0.588 s
100-queens	1.98 s	-

As we can see, SModels is far slower than Clingo by comparison. As we increase the number of queens, SModels struggles with handling the data. Particularly, in the case of 100 queens, SModels crashed whereas Clingo was able to manage the same problem in approximately 2 seconds..

Another experiment comparing SModels + Lparse performance over different types of semantics is shown below. SModels has the worst performance in handling semi-semantics.



The team also compared the outputs of Clingo vs Smodels in the Appendix for the following problems:

- Guilty or Innocent Problem
- Flying Birds Problem
- N-Queens Problem

In all problems, SModels returns different numbers of atoms for the same problems which implies that the way in which Lparse and SModels derives atoms from a given problem is fundamentally different when compared to Clingo. But the results are the same.
On the other hand, The outputs of Flying Birds Problem are different. Clingo solves the problem in a short time. However, unlike Clingo, SModel cares about the weakly restricted variables.
Since the syntax of SModels and Lparse is not the same as Clingo, The code for the SModels is different. The Clingo solved the problem with a better running time. SModels is not as powerful as Clingo since Clingo has more modes and better running time than SModels and Lparse.

In summary, Clingo has a better performance overall, and is more reliable for solving ASP problems. And whilst SModels can perform relatively well on simpler cases, it is quite fairly outperformed by Clingo the more complex our problems become.

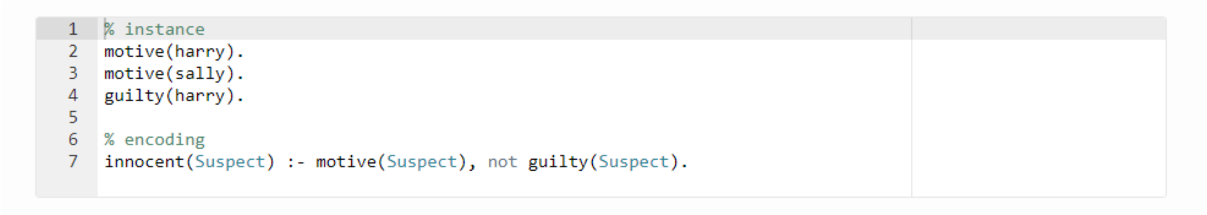
Conclusion (Common)

Overall, the purpose of this study was to push the limits of Clingo and ascertain its value as a tool used to solve logical problems. From their findings, the authors have come to the verdict that whilst there are certain cases where Clingo may cause users frustration when dealing with particularly large answer sets, the software largely has no issues in dealing with increasingly complex logical problems both quickly and efficiently. In addition, Clingo also offers users access to statistical data that may prove incredibly useful for a variety of different purposes, whether they be algorithm analysis or simply optimizing previously found solutions to a given problem.

Appendix (Common)

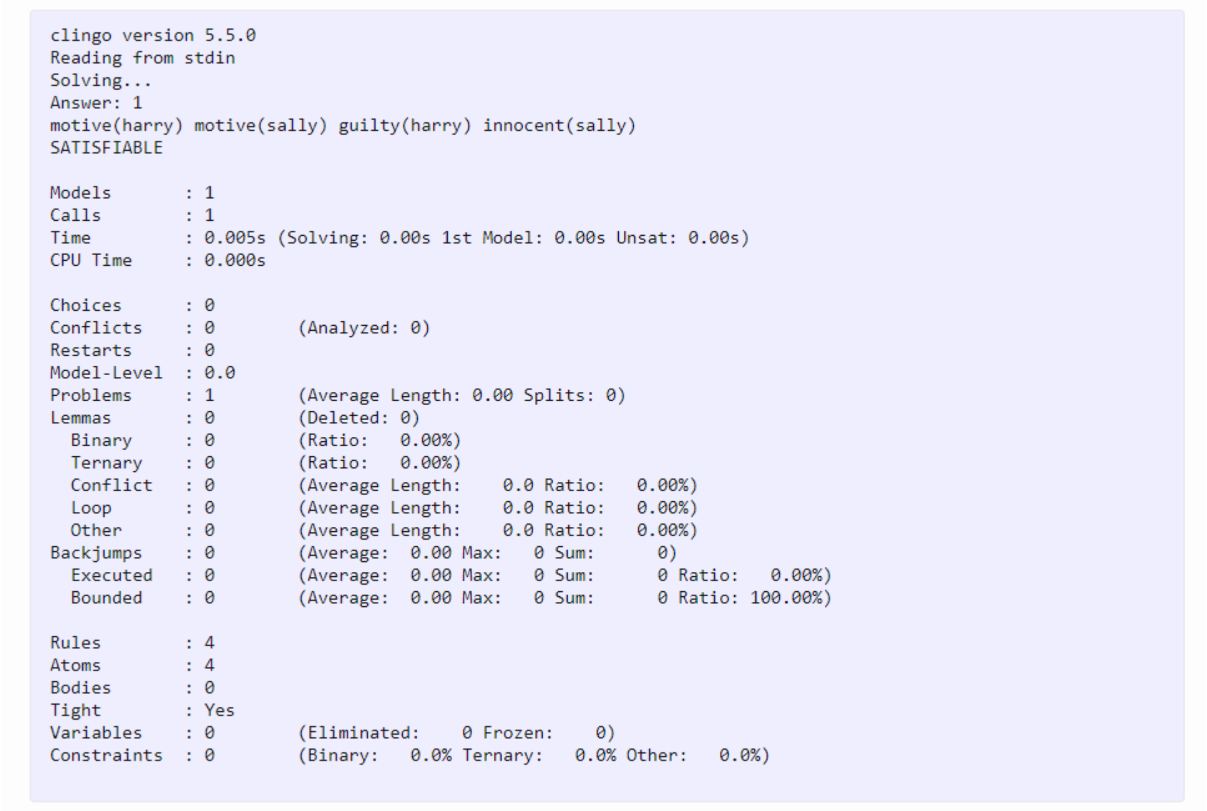
Guilty or Innocent Problem

Problem



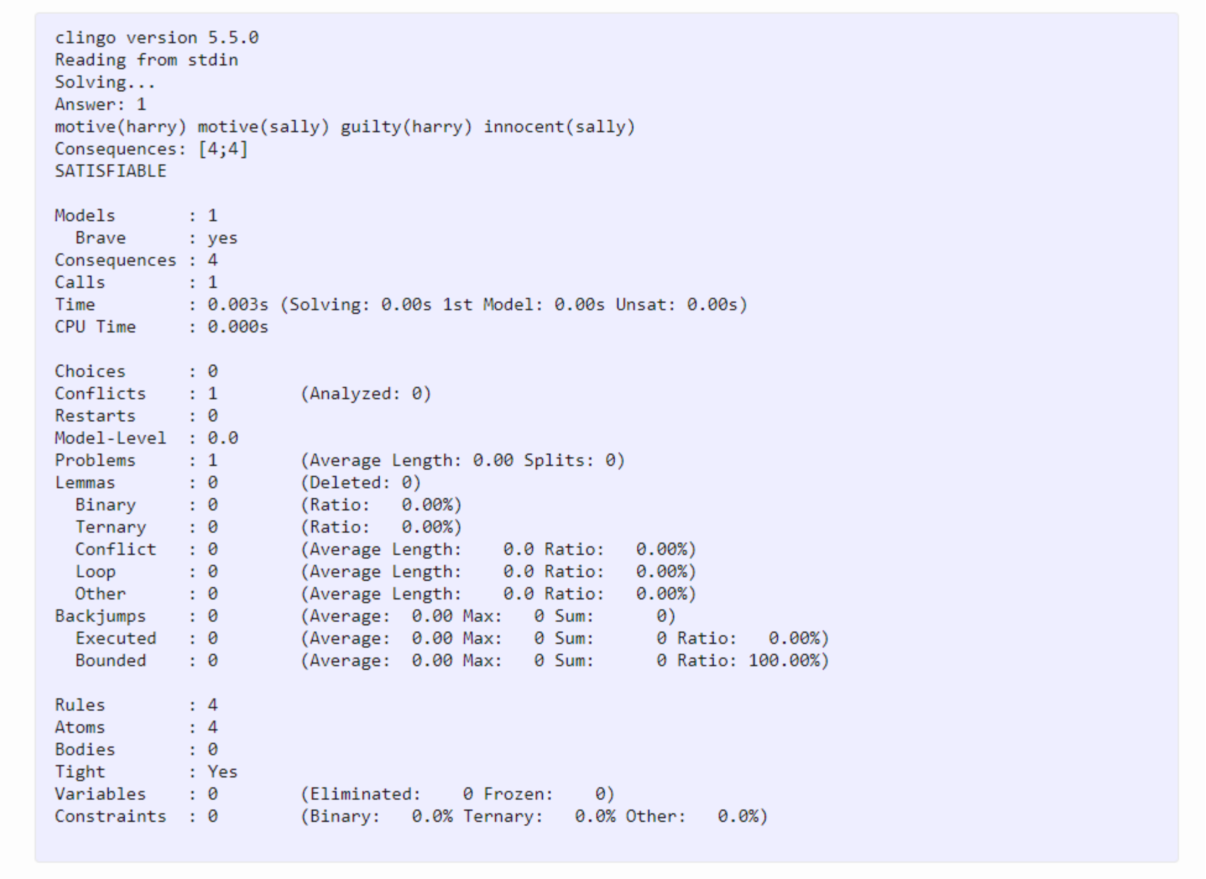
```
% instance
motive(harry).
motive(sally).
guilty(harry).

% encoding
innocent(Suspect) :- motive(Suspect), not guilty(Suspect).
```

Default



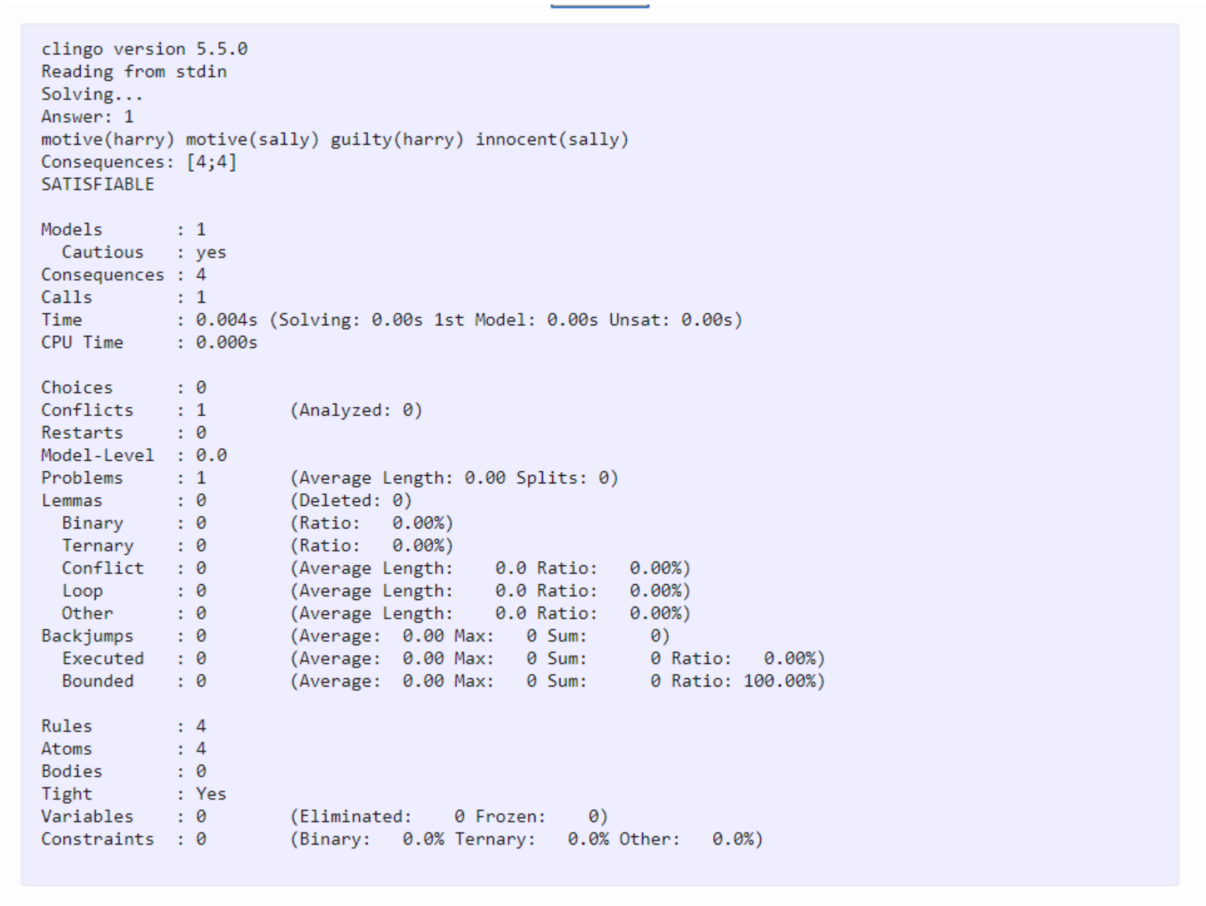
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
motive(harry) motive(sally) guilty(harry) innocent(sally)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.005s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 4
Atoms        : 4
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

Brave



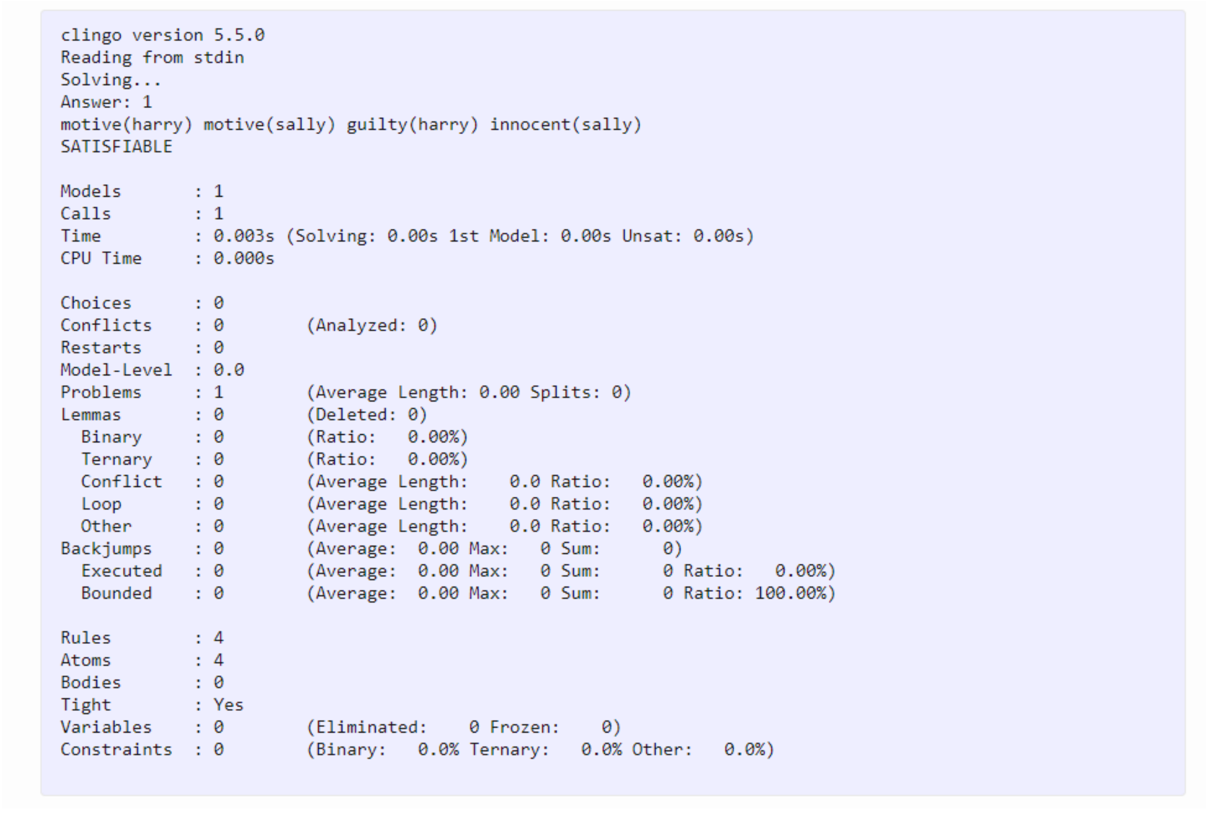
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
motive(harry) motive(sally) guilty(harry) innocent(sally)
Consequences: [4;4]
SATISFIABLE

Models       : 1
  Brave      : yes
Consequences : 4
Calls        : 1
Time         : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 1        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 4
Atoms        : 4
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

Cautious



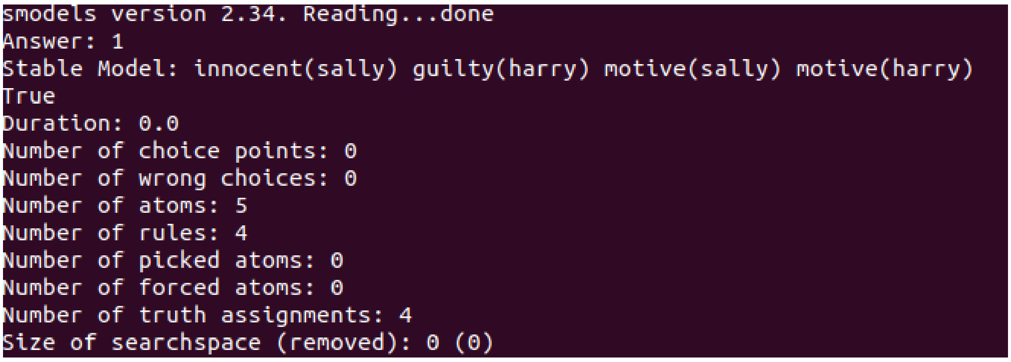
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
motive(harry) motive(sally) guilty(harry) innocent(sally)
Consequences: [4;4]
SATISFIABLE

Models       : 1
  Cautious   : yes
Consequences : 4
Calls        : 1
Time         : 0.004s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 1        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 4
Atoms        : 4
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

Enumerate All



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
motive(harry) motive(sally) guilty(harry) innocent(sally)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 4
Atoms        : 4
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

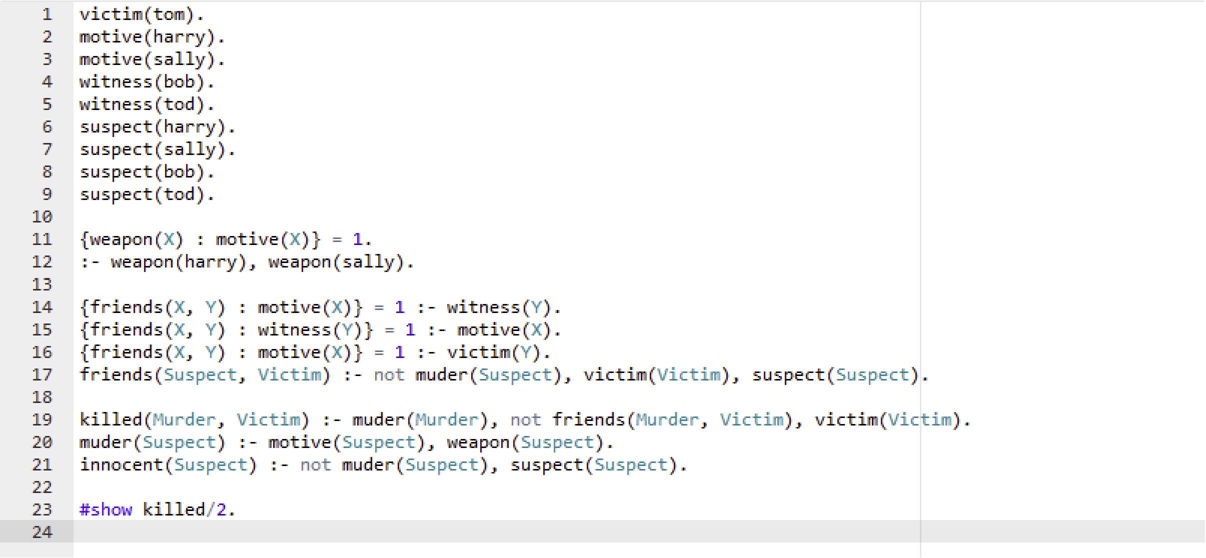
SModels Output



```
smodels version 2.34. Reading...done
Answer: 1
Stable Model: innocent(sally) guilty(harry) motive(sally) motive(harry)
True
Duration: 0.0
Number of choice points: 0
Number of wrong choices: 0
Number of atoms: 5
Number of rules: 4
Number of picked atoms: 0
Number of forced atoms: 0
Number of truth assignments: 4
Size of searchspace (removed): 0 (0)
```

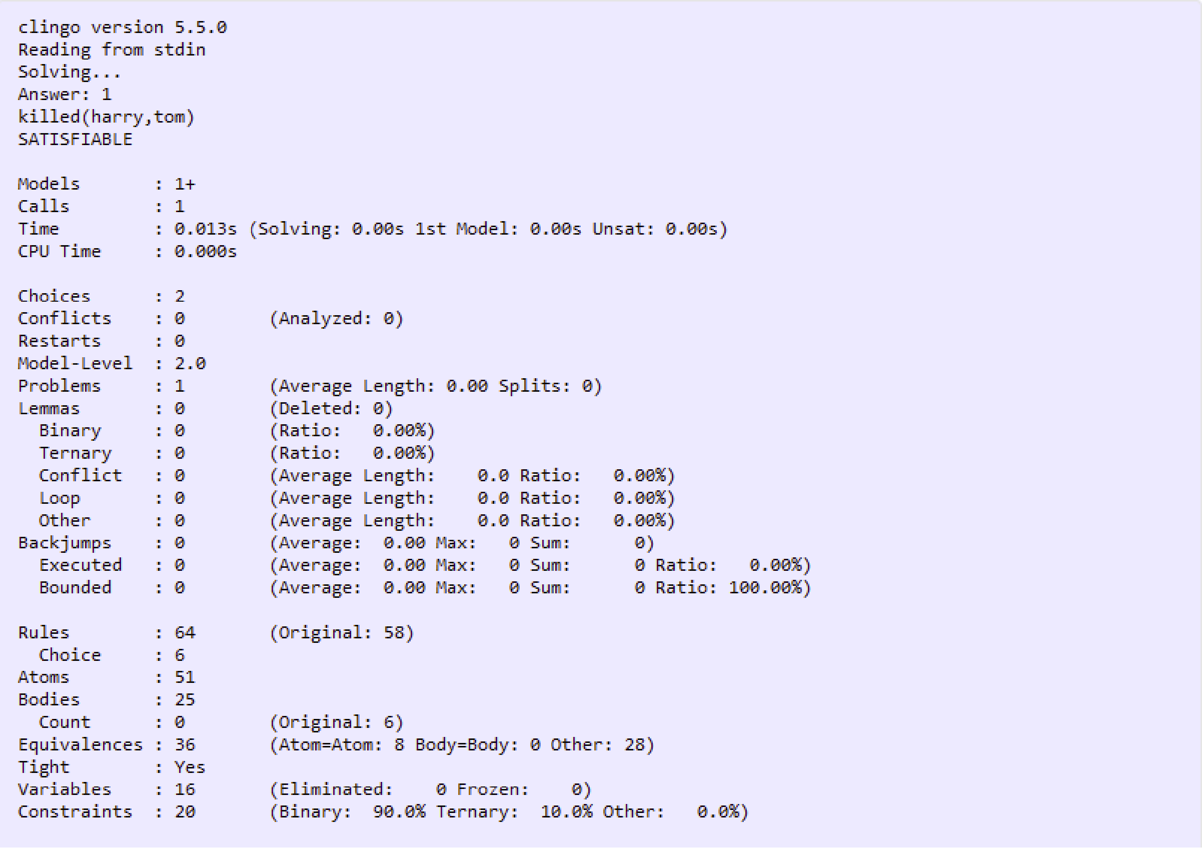
Guilty or Innocent Problem (Modified)

Problem



```
victim(tom).
motive(harry).
motive(sally).
witness(bob).
witness(tod).
suspect(harry).
suspect(sally).
suspect(bob).
suspect(tod).

{weapon(X) : motive(X)} = 1.
:- weapon(harry), weapon(sally).

{friends(X, Y) : motive(X)} = 1 :- witness(Y).
{friends(X, Y) : witness(Y)} = 1 :- motive(X).
{friends(X, Y) : motive(X)} = 1 :- victim(Y).
friends(Suspect, Victim) :- not muder(Suspect), victim(Victim), suspect(Suspect).

killed(Murder, Victim) :- muder(Murder), not friends(Murder, Victim), victim(Victim).
muder(Suspect) :- motive(Suspect), weapon(Suspect).
innocent(Suspect) :- not muder(Suspect), suspect(Suspect).

#show killed/2.

```

Default



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
killed(harry,tom)
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 2
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 2.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 64       (Original: 58)
  Choice     : 6
Atoms        : 51
Bodies       : 25
  Count      : 0        (Original: 6)
Equivalences : 36       (Atom=Atom: 8 Body=Body: 0 Other: 28)
Tight        : Yes
Variables    : 16       (Eliminated:    0 Frozen:    0)
Constraints  : 20       (Binary:  90.0% Ternary:  10.0% Other:   0.0%)
```

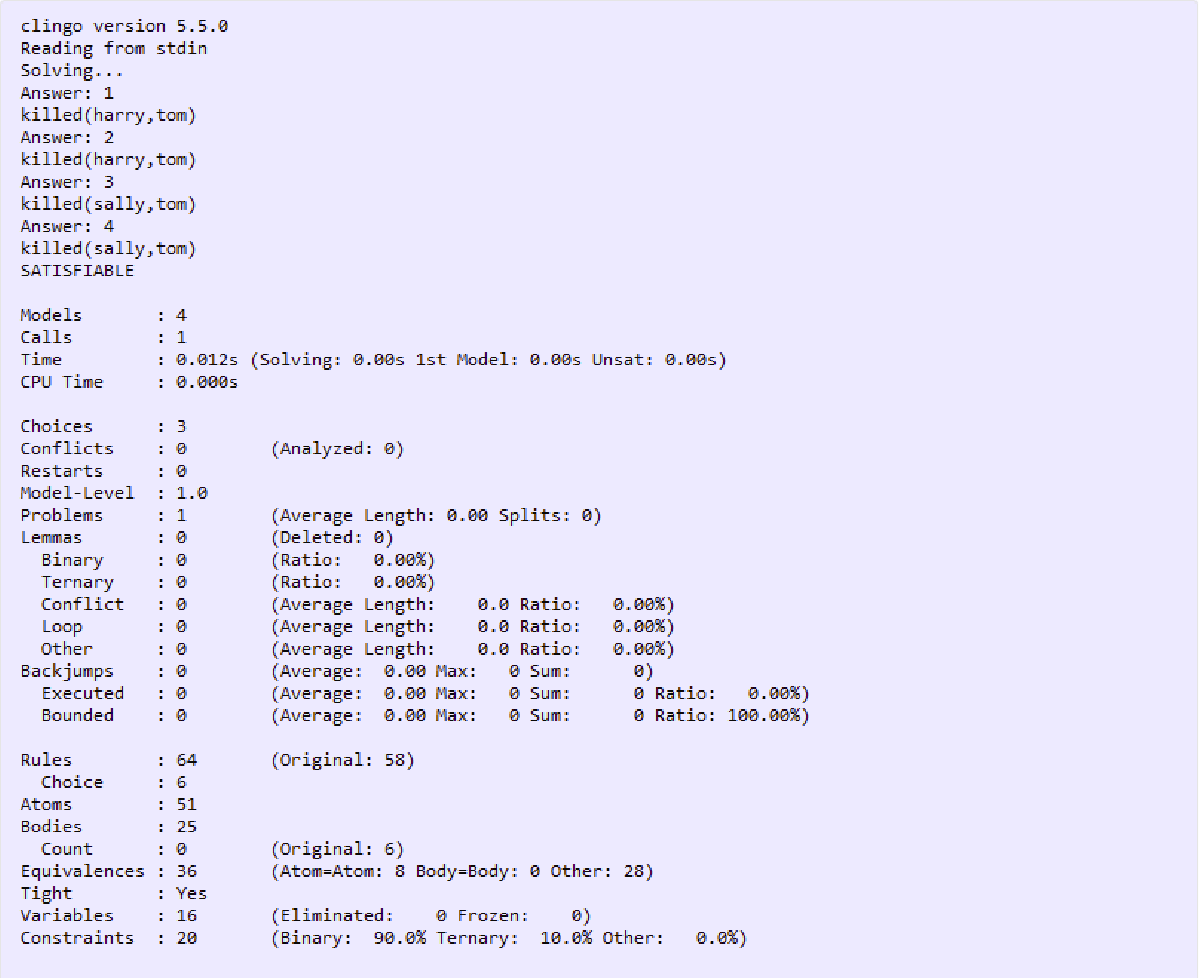
Brave

```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
killed(harry,tom)
Consequences: [1;2]
Answer: 2
killed(harry,tom) killed(sally,tom)
Consequences: [2;2]
SATISFIABLE

Models       : 2
  Brave      : yes
Consequences : 2
Calls        : 1
Time         : 0.015s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 3
Conflicts    : 1        (Analyzed: 0)
Restarts     : 0
Model-Level  : 1.5
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 2        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 2        (Average Length:    0.5 Ratio: 100.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 64       (Original: 58)
  Choice     : 6
Atoms        : 51
Bodies       : 25
  Count      : 0        (Original: 6)
Equivalences : 36       (Atom=Atom: 8 Body=Body: 0 Other: 28)
Tight        : Yes
Variables    : 16       (Eliminated:    0 Frozen:    2)
Constraints  : 20       (Binary:  90.0% Ternary:  10.0% Other:   0.0%)
```

Cautious

```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
killed(harry,tom)
Consequences: [0;1]
Answer: 2

Consequences: [0;0]
SATISFIABLE

Models       : 2
  Cautious   : yes
Consequences : 0
Calls        : 1
Time         : 0.010s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 5
Conflicts    : 2        (Analyzed: 1)
Restarts     : 0
Model-Level  : 1.5
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 3        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 1        (Average Length:    1.0 Ratio:  33.33%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 2        (Average Length:    0.5 Ratio:  66.67%)
Backjumps    : 1        (Average:  2.00 Max:   2 Sum:      2)
  Executed   : 1        (Average:  2.00 Max:   2 Sum:      2 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 64       (Original: 58)
  Choice     : 6
Atoms        : 51
Bodies       : 25
  Count      : 0        (Original: 6)
Equivalences : 36       (Atom=Atom: 8 Body=Body: 0 Other: 28)
Tight        : Yes
Variables    : 16       (Eliminated:    0 Frozen:    2)
Constraints  : 20       (Binary:  90.0% Ternary:  10.0% Other:   0.0%)
```

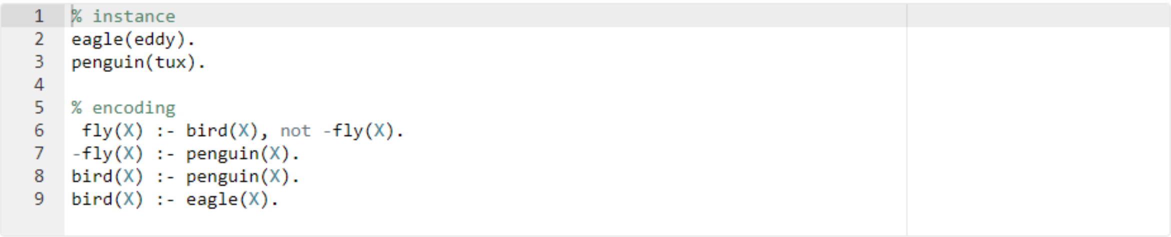
Enumerate All



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
killed(harry,tom)
Answer: 2
killed(harry,tom)
Answer: 3
killed(sally,tom)
Answer: 4
killed(sally,tom)
SATISFIABLE

Models       : 4
Calls        : 1
Time         : 0.012s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 3
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 1.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 64       (Original: 58)
  Choice     : 6
Atoms        : 51
Bodies       : 25
  Count      : 0        (Original: 6)
Equivalences : 36       (Atom=Atom: 8 Body=Body: 0 Other: 28)
Tight        : Yes
Variables    : 16       (Eliminated:    0 Frozen:    0)
Constraints  : 20       (Binary:  90.0% Ternary:  10.0% Other:   0.0%)
```

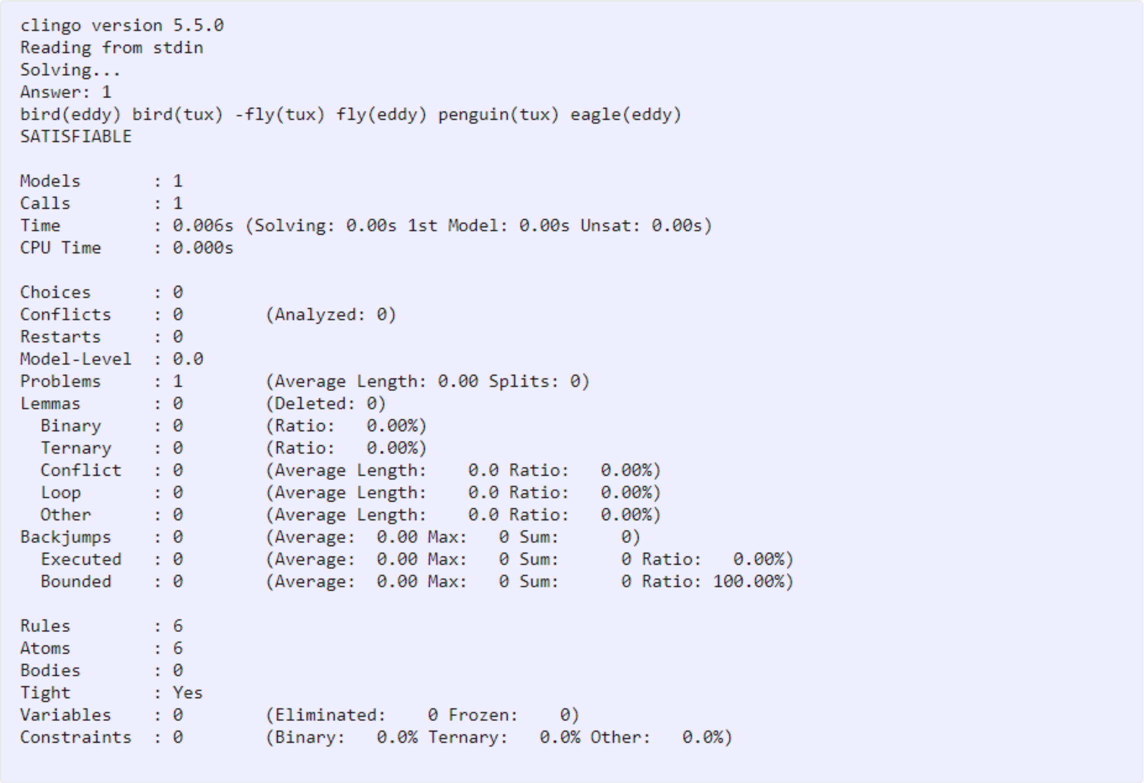
Flying Birds Problem

Problem



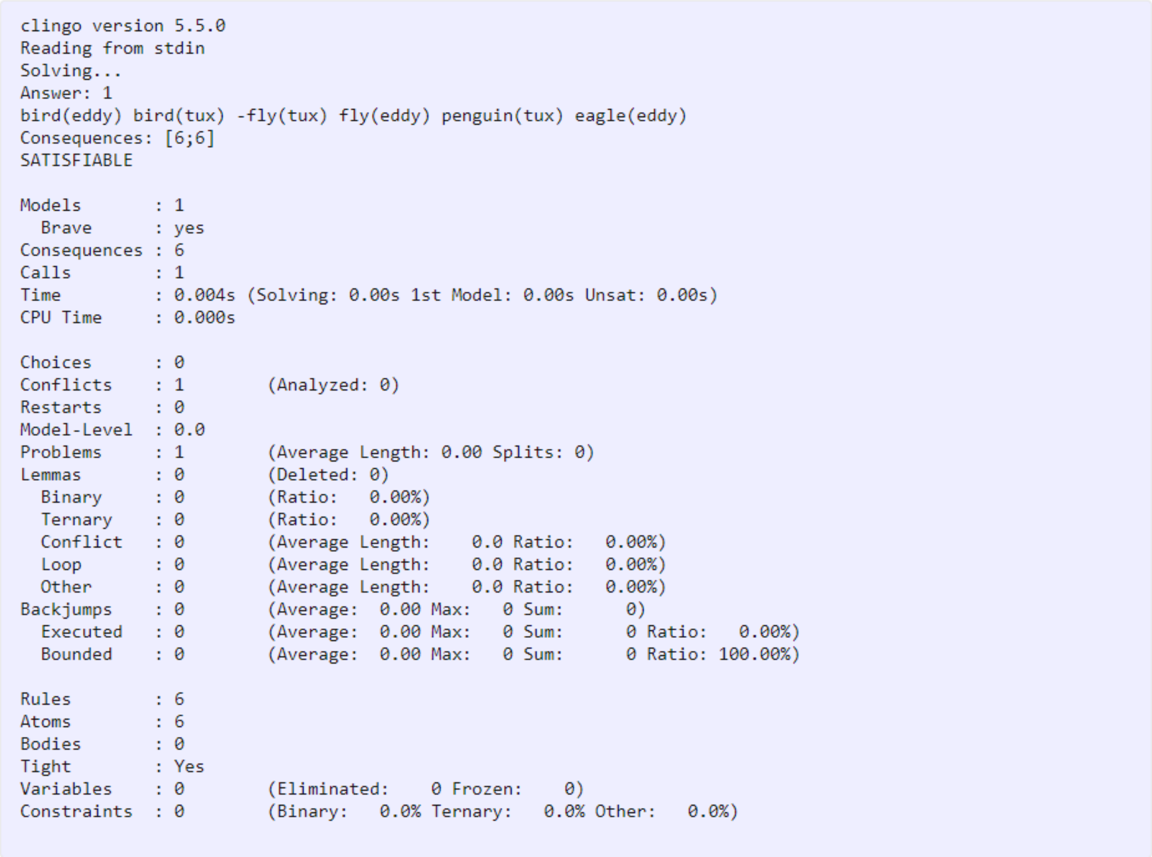
```
% instance
eagle(eddy).
penguin(tux).

% encoding
 fly(X) :- bird(X), not -fly(X).
-fly(X) :- penguin(X).
bird(X) :- penguin(X).
bird(X) :- eagle(X).
```

Default



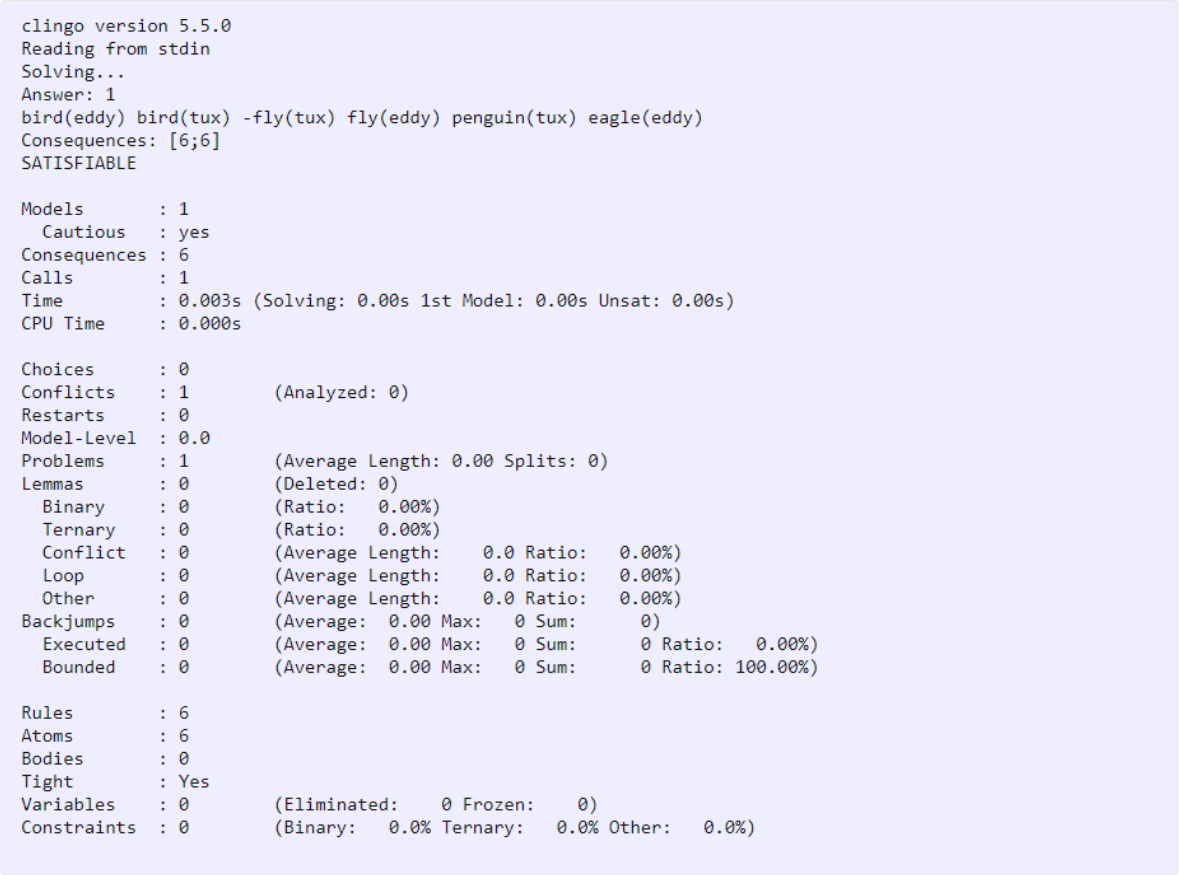
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
bird(eddy) bird(tux) -fly(tux) fly(eddy) penguin(tux) eagle(eddy)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.006s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 6
Atoms        : 6
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

Brave



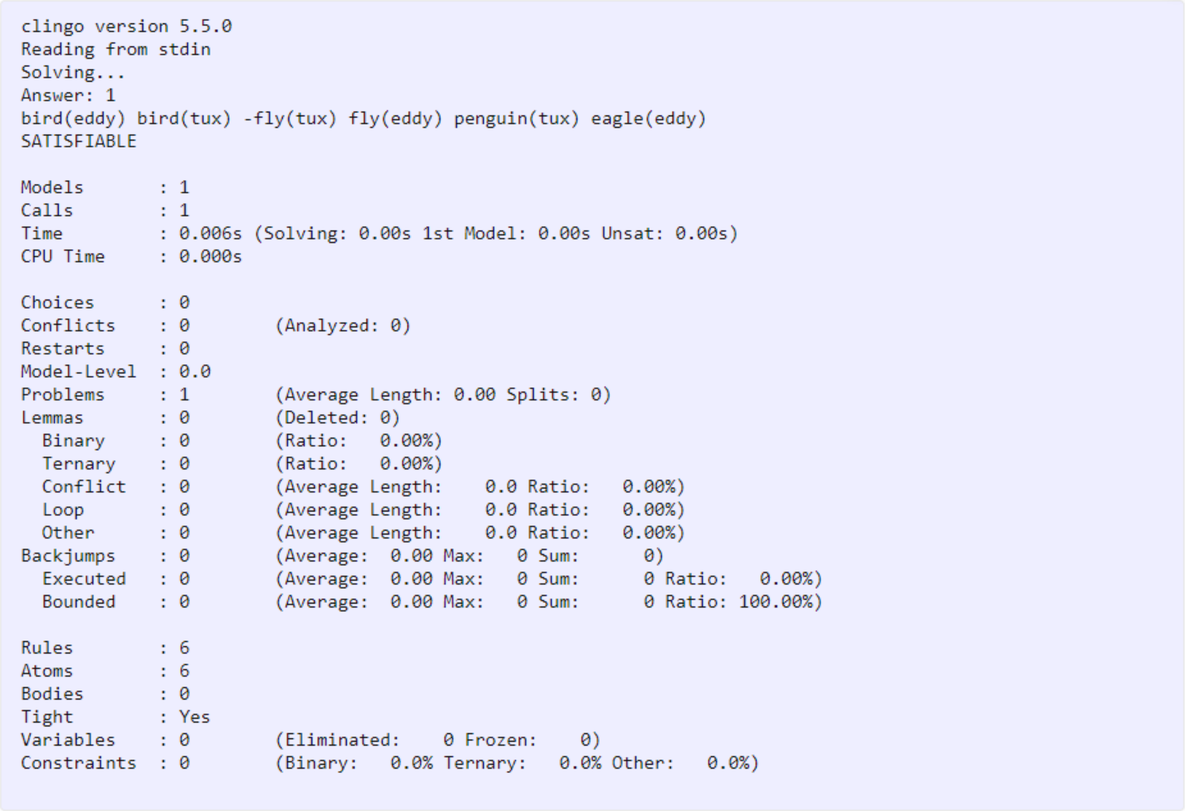
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
bird(eddy) bird(tux) -fly(tux) fly(eddy) penguin(tux) eagle(eddy)
Consequences: [6;6]
SATISFIABLE

Models       : 1
  Brave      : yes
Consequences : 6
Calls        : 1
Time         : 0.004s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 1        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 6
Atoms        : 6
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

Cautious



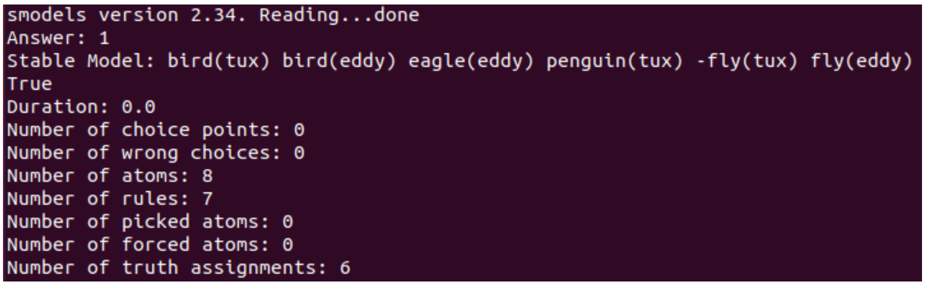
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
bird(eddy) bird(tux) -fly(tux) fly(eddy) penguin(tux) eagle(eddy)
Consequences: [6;6]
SATISFIABLE

Models       : 1
  Cautious   : yes
Consequences : 6
Calls        : 1
Time         : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 1        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 6
Atoms        : 6
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

Enumerate All



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
bird(eddy) bird(tux) -fly(tux) fly(eddy) penguin(tux) eagle(eddy)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.006s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 0
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 6
Atoms        : 6
Bodies       : 0
Tight        : Yes
Variables    : 0        (Eliminated:    0 Frozen:    0)
Constraints  : 0        (Binary:   0.0% Ternary:   0.0% Other:   0.0%)
```

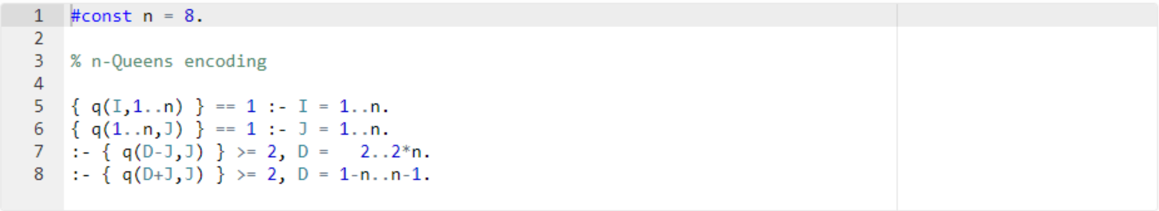
SModels Output



```
smodels version 2.34. Reading...done
Answer: 1
Stable Model: bird(tux) bird(eddy) eagle(eddy) penguin(tux) -fly(tux) fly(eddy)
True
Duration: 0.0
Number of choice points: 0
Number of wrong choices: 0
Number of atoms: 8
Number of rules: 7
Number of picked atoms: 0
Number of forced atoms: 0
Number of truth assignments: 6
```

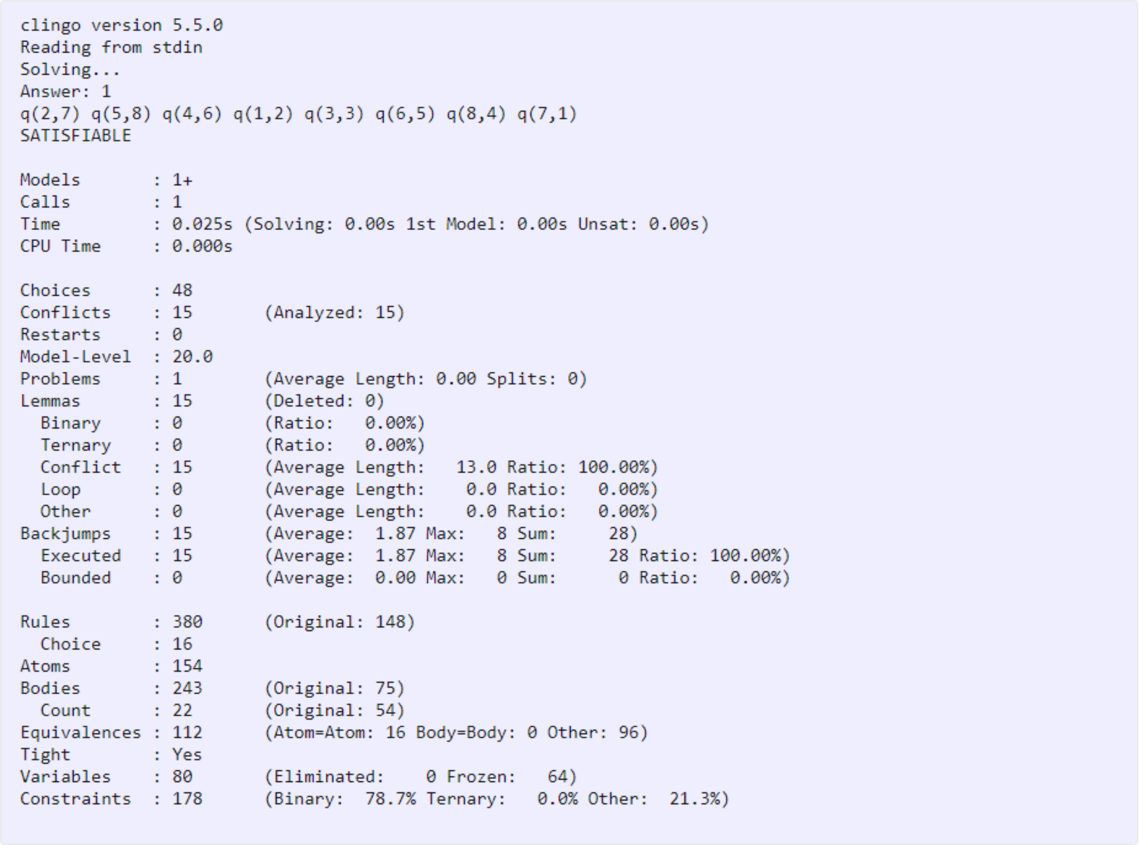
N-Queens Problem

Problem



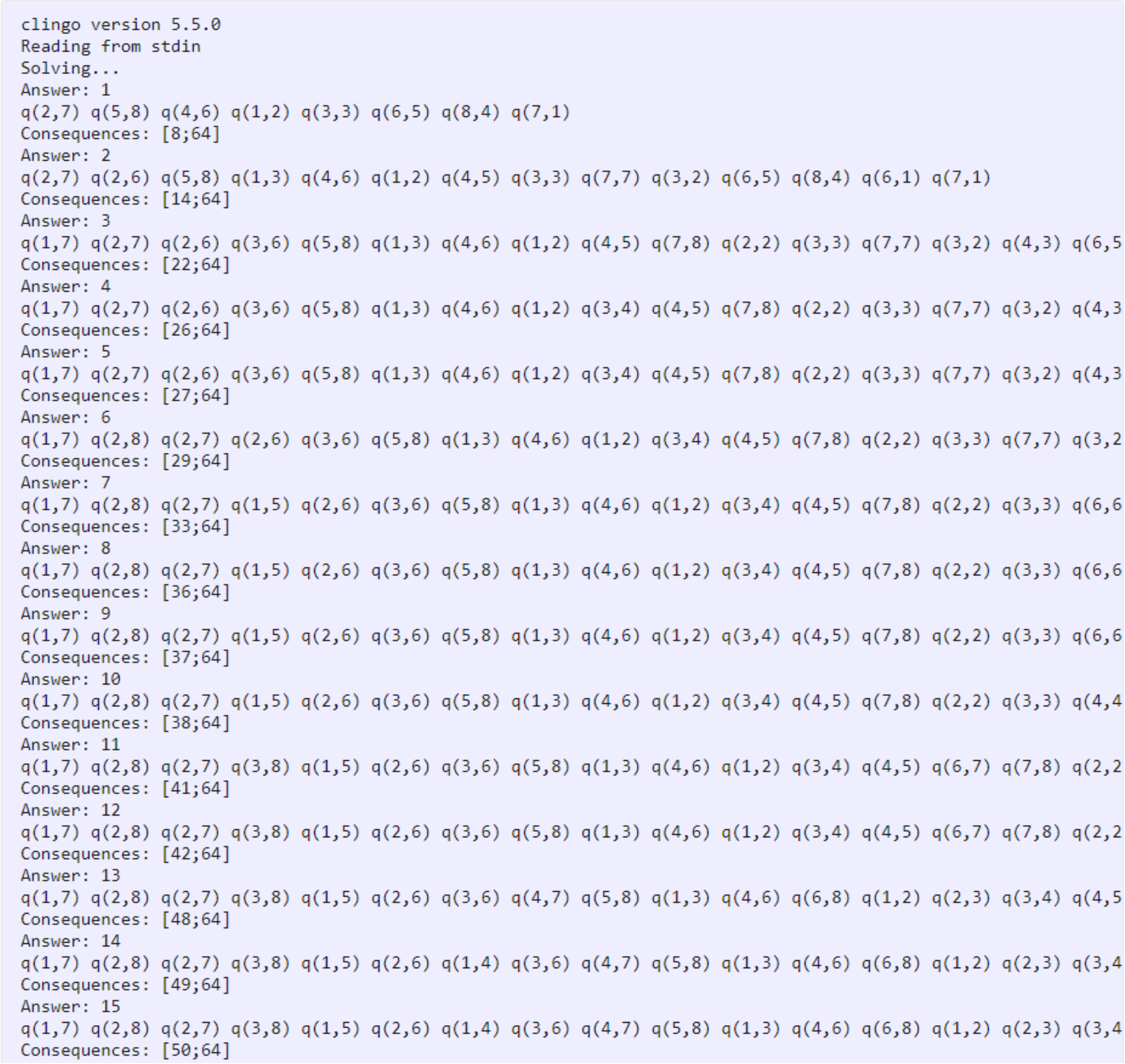
```
#const n = 8.

% n-Queens encoding

{ q(I,1..n) } == 1 :- I = 1..n.
{ q(1..n,J) } == 1 :- J = 1..n.
:- { q(D-J,J) } >= 2, D =   2..2*n.
:- { q(D+J,J) } >= 2, D = 1-n..n-1.
```

Default

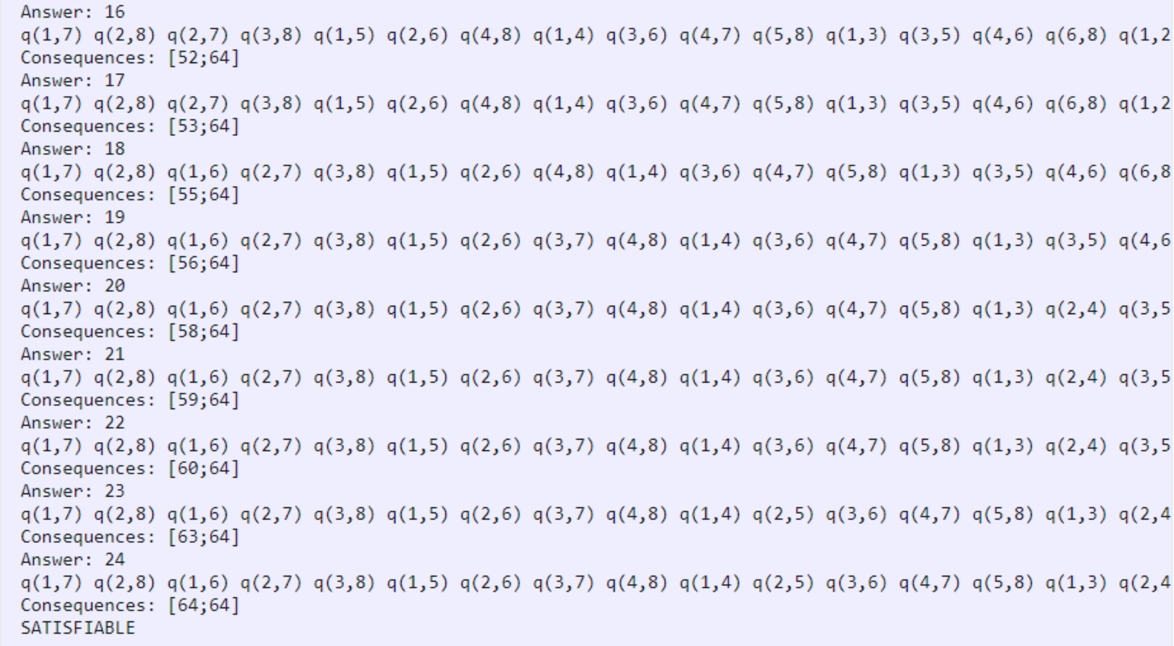


```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
q(2,7) q(5,8) q(4,6) q(1,2) q(3,3) q(6,5) q(8,4) q(7,1)
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.025s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 48
Conflicts    : 15       (Analyzed: 15)
Restarts     : 0
Model-Level  : 20.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 15       (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 15       (Average Length:   13.0 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 15       (Average:  1.87 Max:   8 Sum:     28)
  Executed   : 15       (Average:  1.87 Max:   8 Sum:     28 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 380      (Original: 148)
  Choice     : 16
Atoms        : 154
Bodies       : 243      (Original: 75)
  Count      : 22       (Original: 54)
Equivalences : 112      (Atom=Atom: 16 Body=Body: 0 Other: 96)
Tight        : Yes
Variables    : 80       (Eliminated:    0 Frozen:    64)
Constraints  : 178      (Binary:  78.7% Ternary:   0.0% Other:  21.3%)
```

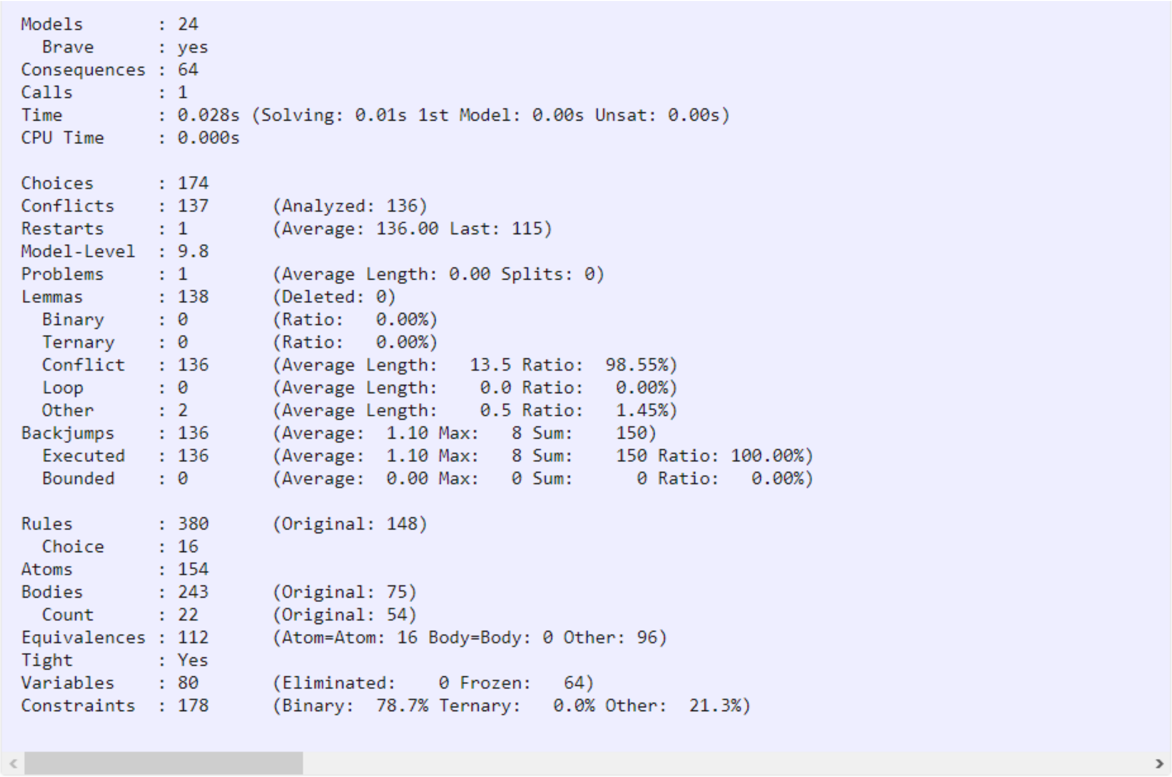
Brave



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
q(2,7) q(5,8) q(4,6) q(1,2) q(3,3) q(6,5) q(8,4) q(7,1)
Consequences: [8;64]
Answer: 2
q(2,7) q(2,6) q(5,8) q(1,3) q(4,6) q(1,2) q(4,5) q(3,3) q(7,7) q(3,2) q(6,5) q(8,4) q(6,1) q(7,1)
Consequences: [14;64]
Answer: 3
q(1,7) q(2,7) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(4,5) q(7,8) q(2,2) q(3,3) q(7,7) q(3,2) q(4,3) q(6,5
Consequences: [22;64]
Answer: 4
q(1,7) q(2,7) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(7,7) q(3,2) q(4,3
Consequences: [26;64]
Answer: 5
q(1,7) q(2,7) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(7,7) q(3,2) q(4,3
Consequences: [27;64]
Answer: 6
q(1,7) q(2,8) q(2,7) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(7,7) q(3,2
Consequences: [29;64]
Answer: 7
q(1,7) q(2,8) q(2,7) q(1,5) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(6,6
Consequences: [33;64]
Answer: 8
q(1,7) q(2,8) q(2,7) q(1,5) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(6,6
Consequences: [36;64]
Answer: 9
q(1,7) q(2,8) q(2,7) q(1,5) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(6,6
Consequences: [37;64]
Answer: 10
q(1,7) q(2,8) q(2,7) q(1,5) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(7,8) q(2,2) q(3,3) q(4,4
Consequences: [38;64]
Answer: 11
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(6,7) q(7,8) q(2,2
Consequences: [41;64]
Answer: 12
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(3,6) q(5,8) q(1,3) q(4,6) q(1,2) q(3,4) q(4,5) q(6,7) q(7,8) q(2,2
Consequences: [42;64]
Answer: 13
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(3,6) q(4,7) q(5,8) q(1,3) q(4,6) q(6,8) q(1,2) q(2,3) q(3,4) q(4,5
Consequences: [48;64]
Answer: 14
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(4,6) q(6,8) q(1,2) q(2,3) q(3,4
Consequences: [49;64]
Answer: 15
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(4,6) q(6,8) q(1,2) q(2,3) q(3,4
Consequences: [50;64]
```

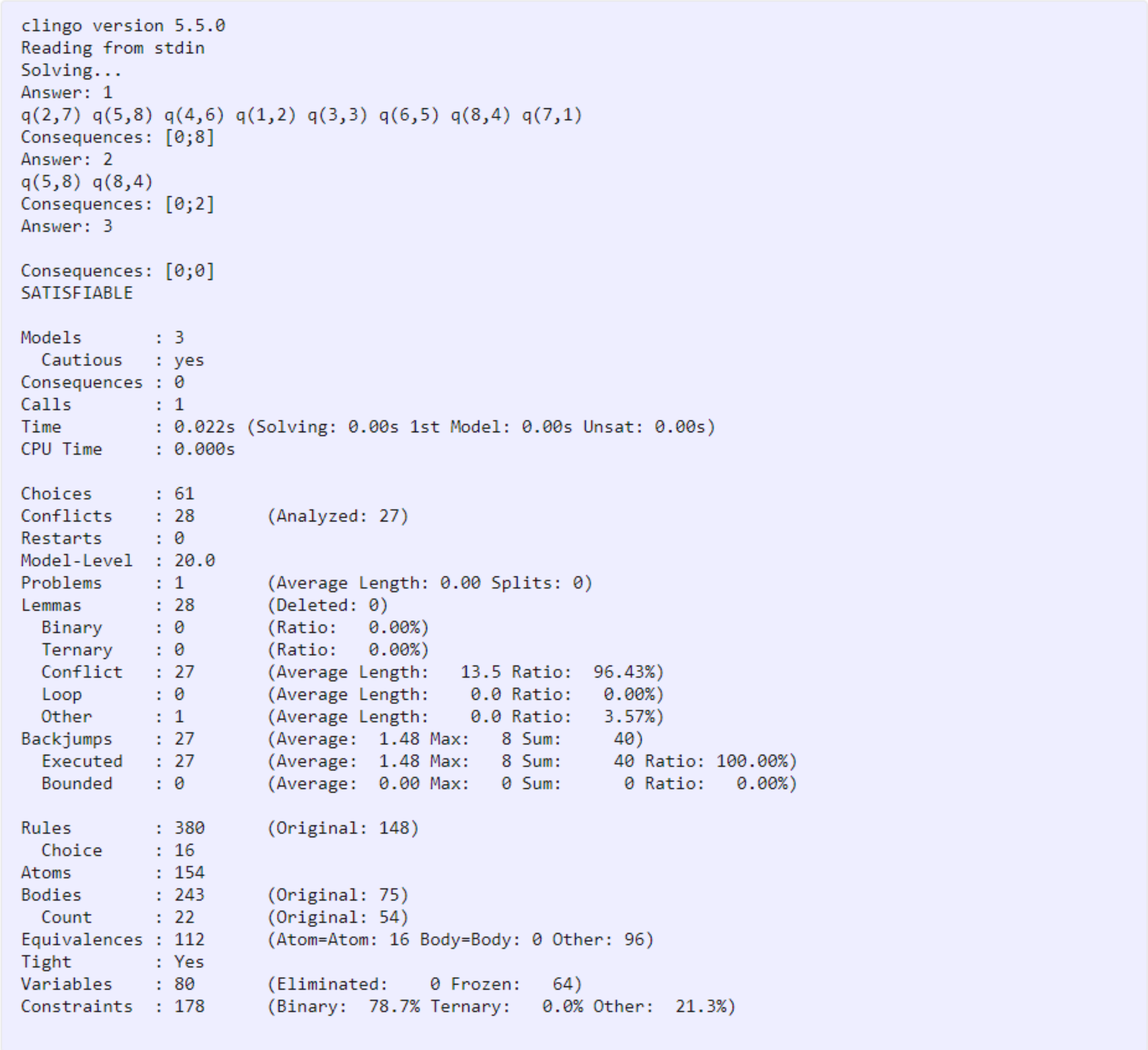


```
Answer: 16
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(3,5) q(4,6) q(6,8) q(1,2
Consequences: [52;64]
Answer: 17
q(1,7) q(2,8) q(2,7) q(3,8) q(1,5) q(2,6) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(3,5) q(4,6) q(6,8) q(1,2
Consequences: [53;64]
Answer: 18
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(3,5) q(4,6) q(6,8
Consequences: [55;64]
Answer: 19
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(3,7) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(3,5) q(4,6
Consequences: [56;64]
Answer: 20
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(3,7) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(2,4) q(3,5
Consequences: [58;64]
Answer: 21
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(3,7) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(2,4) q(3,5
Consequences: [59;64]
Answer: 22
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(3,7) q(4,8) q(1,4) q(3,6) q(4,7) q(5,8) q(1,3) q(2,4) q(3,5
Consequences: [60;64]
Answer: 23
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(3,7) q(4,8) q(1,4) q(2,5) q(3,6) q(4,7) q(5,8) q(1,3) q(2,4
Consequences: [63;64]
Answer: 24
q(1,7) q(2,8) q(1,6) q(2,7) q(3,8) q(1,5) q(2,6) q(3,7) q(4,8) q(1,4) q(2,5) q(3,6) q(4,7) q(5,8) q(1,3) q(2,4
Consequences: [64;64]
SATISFIABLE
```



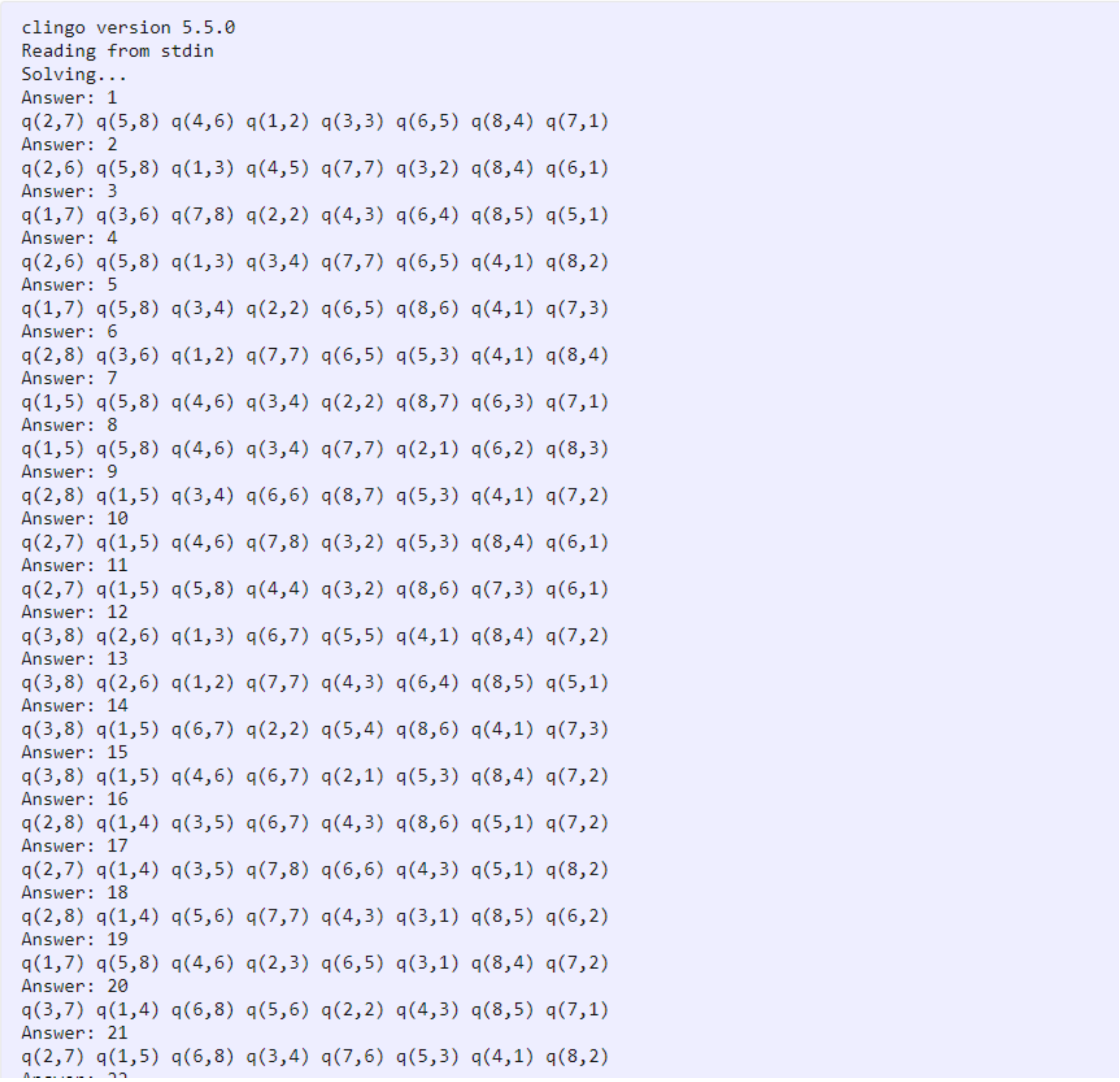
```
Models       : 24
  Brave      : yes
Consequences : 64
Calls        : 1
Time         : 0.028s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 174
Conflicts    : 137      (Analyzed: 136)
Restarts     : 1        (Average: 136.00 Last: 115)
Model-Level  : 9.8
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 138      (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 136      (Average Length:   13.5 Ratio:  98.55%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 2        (Average Length:    0.5 Ratio:   1.45%)
Backjumps    : 136      (Average:  1.10 Max:   8 Sum:     150)
  Executed   : 136      (Average:  1.10 Max:   8 Sum:     150 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:       0 Ratio:   0.00%)

Rules        : 380      (Original: 148)
  Choice     : 16
Atoms        : 154
Bodies       : 243      (Original: 75)
  Count      : 22       (Original: 54)
Equivalences : 112      (Atom=Atom: 16 Body=Body: 0 Other: 96)
Tight        : Yes
Variables    : 80       (Eliminated:    0 Frozen:   64)
Constraints  : 178      (Binary:  78.7% Ternary:   0.0% Other:  21.3%)
```

Cautious

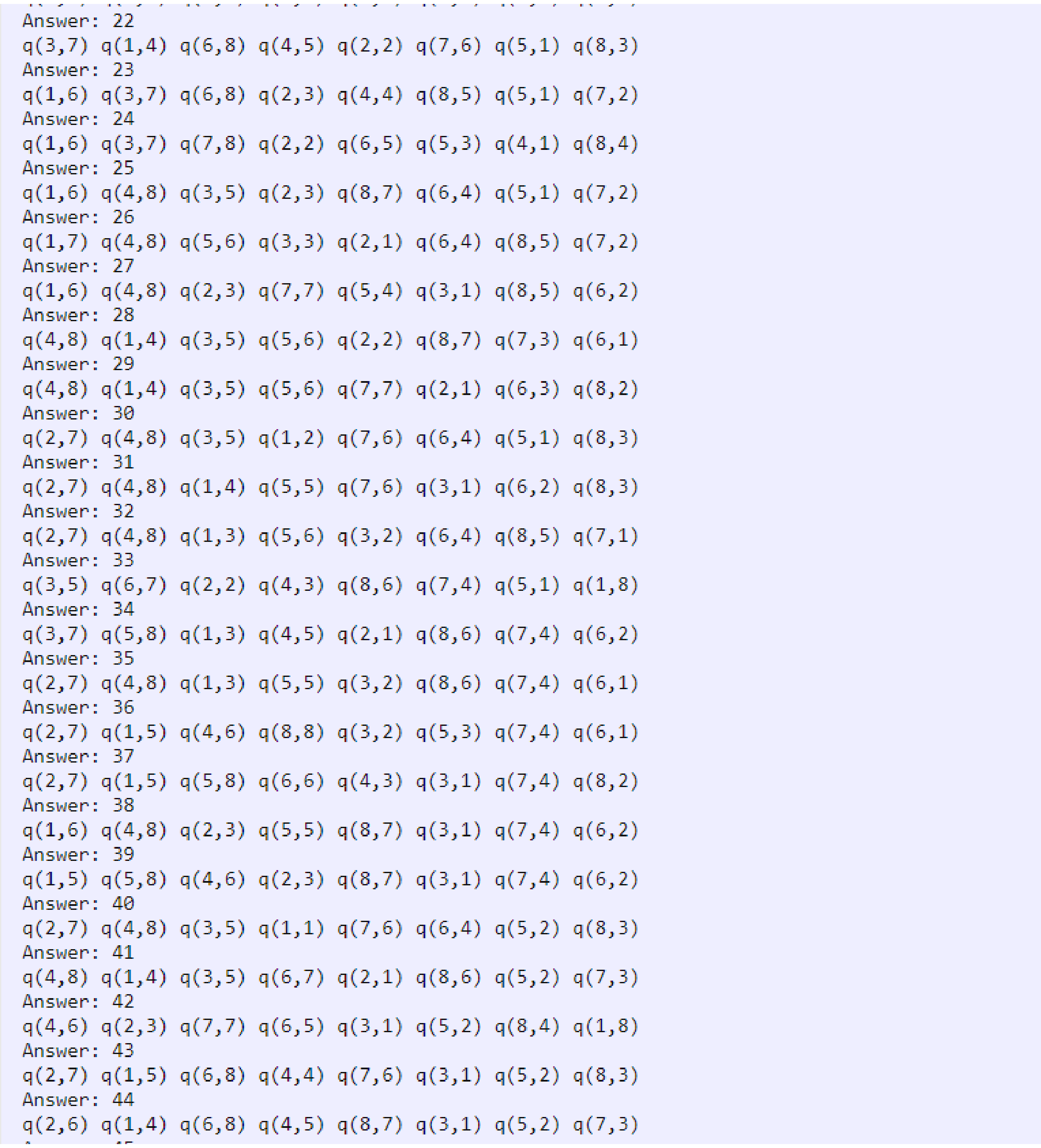


```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
q(2,7) q(5,8) q(4,6) q(1,2) q(3,3) q(6,5) q(8,4) q(7,1)
Consequences: [0;8]
Answer: 2
q(5,8) q(8,4)
Consequences: [0;2]
Answer: 3

Consequences: [0;0]
SATISFIABLE

Models       : 3
  Cautious   : yes
Consequences : 0
Calls        : 1
Time         : 0.022s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 61
Conflicts    : 28       (Analyzed: 27)
Restarts     : 0
Model-Level  : 20.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 28       (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 27       (Average Length:   13.5 Ratio:  96.43%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 1        (Average Length:    0.0 Ratio:   3.57%)
Backjumps    : 27       (Average:  1.48 Max:   8 Sum:     40)
  Executed   : 27       (Average:  1.48 Max:   8 Sum:     40 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 380      (Original: 148)
  Choice     : 16
Atoms        : 154
Bodies       : 243      (Original: 75)
  Count      : 22       (Original: 54)
Equivalences : 112      (Atom=Atom: 16 Body=Body: 0 Other: 96)
Tight        : Yes
Variables    : 80       (Eliminated:    0 Frozen:   64)
Constraints  : 178      (Binary:  78.7% Ternary:   0.0% Other:  21.3%)
```

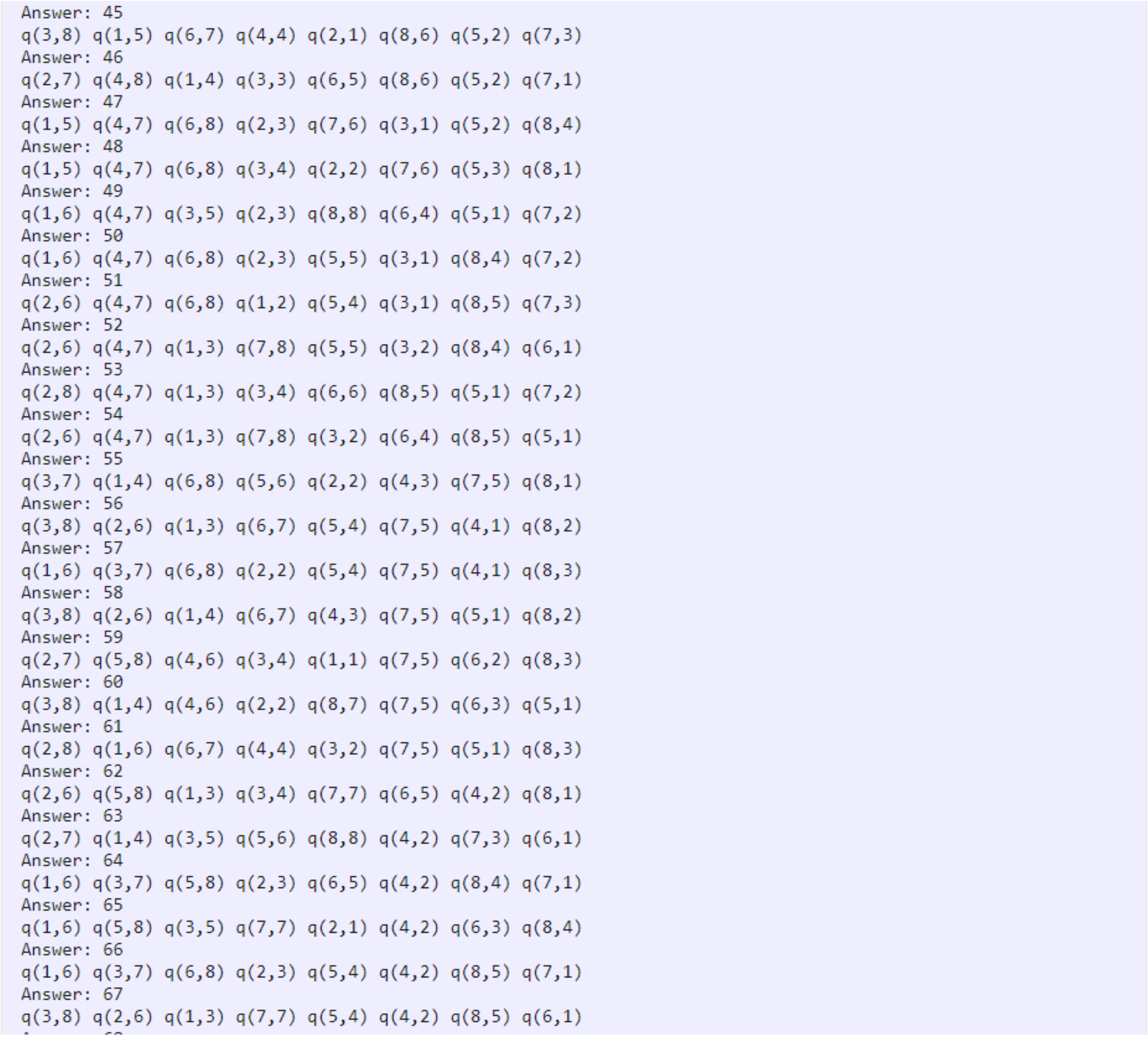
Enumerate All



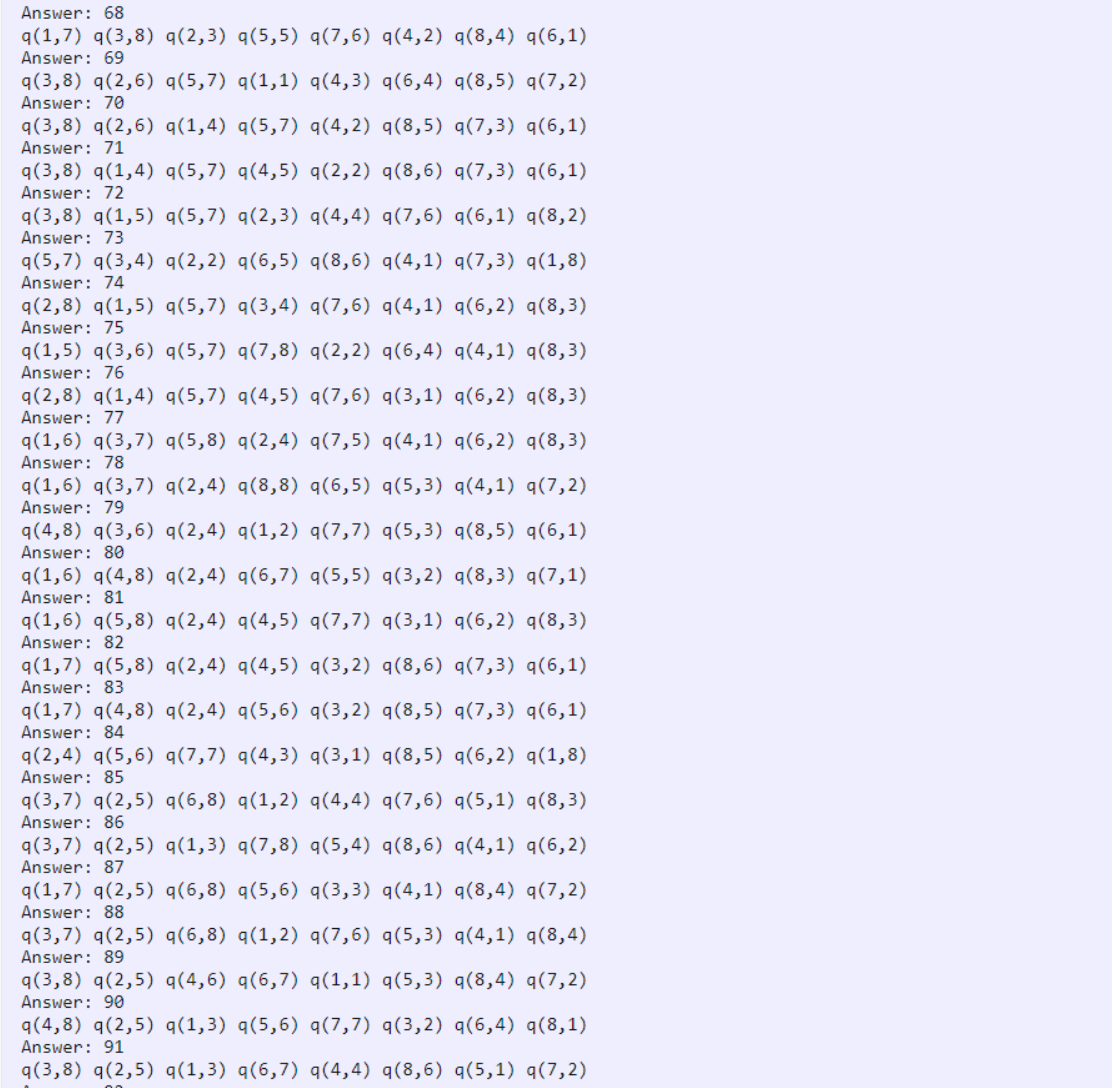
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
q(2,7) q(5,8) q(4,6) q(1,2) q(3,3) q(6,5) q(8,4) q(7,1)
Answer: 2
q(2,6) q(5,8) q(1,3) q(4,5) q(7,7) q(3,2) q(8,4) q(6,1)
Answer: 3
q(1,7) q(3,6) q(7,8) q(2,2) q(4,3) q(6,4) q(8,5) q(5,1)
Answer: 4
q(2,6) q(5,8) q(1,3) q(3,4) q(7,7) q(6,5) q(4,1) q(8,2)
Answer: 5
q(1,7) q(5,8) q(3,4) q(2,2) q(6,5) q(8,6) q(4,1) q(7,3)
Answer: 6
q(2,8) q(3,6) q(1,2) q(7,7) q(6,5) q(5,3) q(4,1) q(8,4)
Answer: 7
q(1,5) q(5,8) q(4,6) q(3,4) q(2,2) q(8,7) q(6,3) q(7,1)
Answer: 8
q(1,5) q(5,8) q(4,6) q(3,4) q(7,7) q(2,1) q(6,2) q(8,3)
Answer: 9
q(2,8) q(1,5) q(3,4) q(6,6) q(8,7) q(5,3) q(4,1) q(7,2)
Answer: 10
q(2,7) q(1,5) q(4,6) q(7,8) q(3,2) q(5,3) q(8,4) q(6,1)
Answer: 11
q(2,7) q(1,5) q(5,8) q(4,4) q(3,2) q(8,6) q(7,3) q(6,1)
Answer: 12
q(3,8) q(2,6) q(1,3) q(6,7) q(5,5) q(4,1) q(8,4) q(7,2)
Answer: 13
q(3,8) q(2,6) q(1,2) q(7,7) q(4,3) q(6,4) q(8,5) q(5,1)
Answer: 14
q(3,8) q(1,5) q(6,7) q(2,2) q(5,4) q(8,6) q(4,1) q(7,3)
Answer: 15
q(3,8) q(1,5) q(4,6) q(6,7) q(2,1) q(5,3) q(8,4) q(7,2)
Answer: 16
q(2,8) q(1,4) q(3,5) q(6,7) q(4,3) q(8,6) q(5,1) q(7,2)
Answer: 17
q(2,7) q(1,4) q(3,5) q(7,8) q(6,6) q(4,3) q(5,1) q(8,2)
Answer: 18
q(2,8) q(1,4) q(5,6) q(7,7) q(4,3) q(3,1) q(8,5) q(6,2)
Answer: 19
q(1,7) q(5,8) q(4,6) q(2,3) q(6,5) q(3,1) q(8,4) q(7,2)
Answer: 20
q(3,7) q(1,4) q(6,8) q(5,6) q(2,2) q(4,3) q(8,5) q(7,1)
Answer: 21
q(2,7) q(1,5) q(6,8) q(3,4) q(7,6) q(5,3) q(4,1) q(8,2)
Answer: 22
```



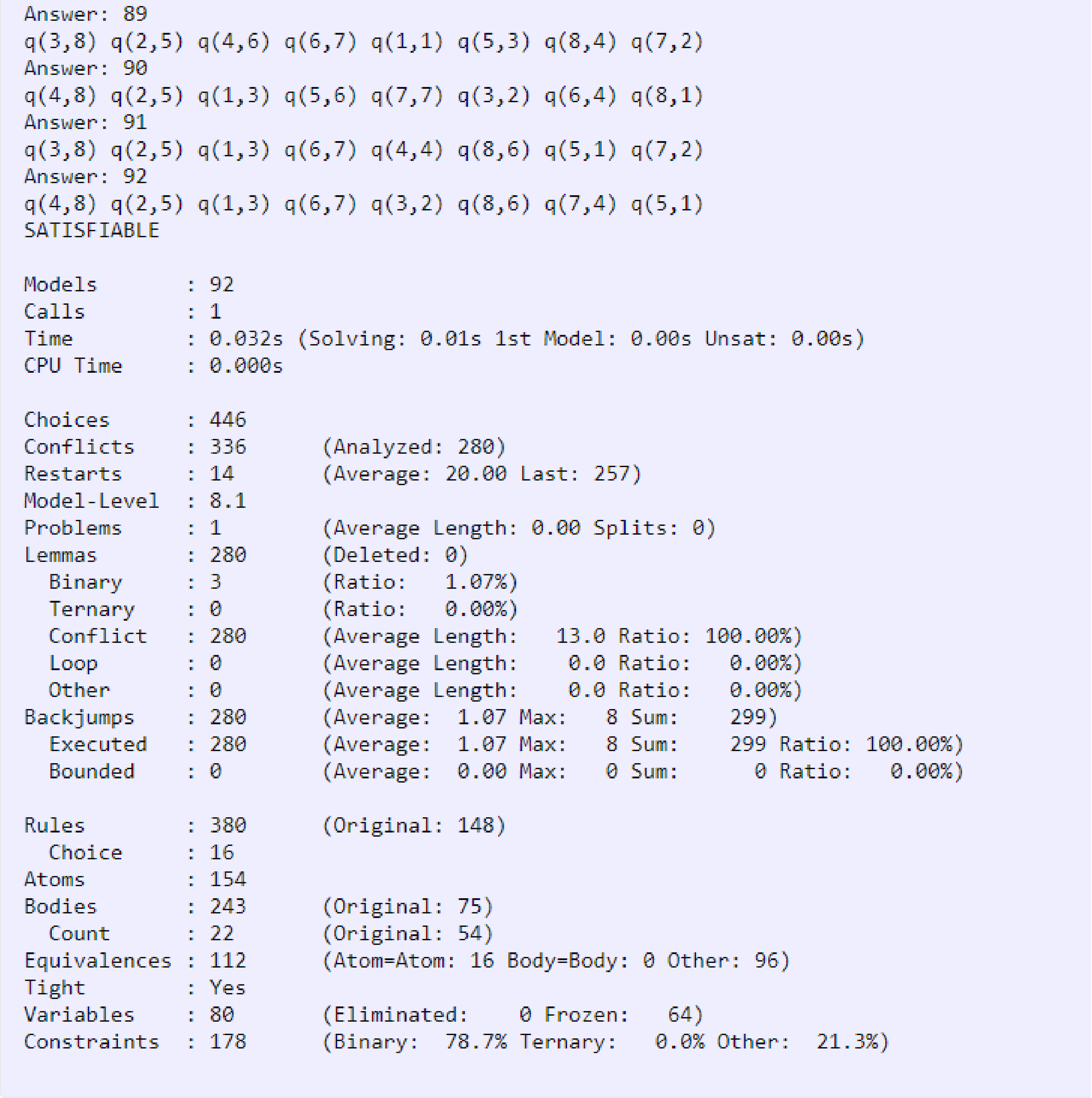
```
Answer: 22
q(3,7) q(1,4) q(6,8) q(4,5) q(2,2) q(7,6) q(5,1) q(8,3)
Answer: 23
q(1,6) q(3,7) q(6,8) q(2,3) q(4,4) q(8,5) q(5,1) q(7,2)
Answer: 24
q(1,6) q(3,7) q(7,8) q(2,2) q(6,5) q(5,3) q(4,1) q(8,4)
Answer: 25
q(1,6) q(4,8) q(3,5) q(2,3) q(8,7) q(6,4) q(5,1) q(7,2)
Answer: 26
q(1,7) q(4,8) q(5,6) q(3,3) q(2,1) q(6,4) q(8,5) q(7,2)
Answer: 27
q(1,6) q(4,8) q(2,3) q(7,7) q(5,4) q(3,1) q(8,5) q(6,2)
Answer: 28
q(4,8) q(1,4) q(3,5) q(5,6) q(2,2) q(8,7) q(7,3) q(6,1)
Answer: 29
q(4,8) q(1,4) q(3,5) q(5,6) q(7,7) q(2,1) q(6,3) q(8,2)
Answer: 30
q(2,7) q(4,8) q(3,5) q(1,2) q(7,6) q(6,4) q(5,1) q(8,3)
Answer: 31
q(2,7) q(4,8) q(1,4) q(5,5) q(7,6) q(3,1) q(6,2) q(8,3)
Answer: 32
q(2,7) q(4,8) q(1,3) q(5,6) q(3,2) q(6,4) q(8,5) q(7,1)
Answer: 33
q(3,5) q(6,7) q(2,2) q(4,3) q(8,6) q(7,4) q(5,1) q(1,8)
Answer: 34
q(3,7) q(5,8) q(1,3) q(4,5) q(2,1) q(8,6) q(7,4) q(6,2)
Answer: 35
q(2,7) q(4,8) q(1,3) q(5,5) q(3,2) q(8,6) q(7,4) q(6,1)
Answer: 36
q(2,7) q(1,5) q(4,6) q(8,8) q(3,2) q(5,3) q(7,4) q(6,1)
Answer: 37
q(2,7) q(1,5) q(5,8) q(6,6) q(4,3) q(3,1) q(7,4) q(8,2)
Answer: 38
q(1,6) q(4,8) q(2,3) q(5,5) q(8,7) q(3,1) q(7,4) q(6,2)
Answer: 39
q(1,5) q(5,8) q(4,6) q(2,3) q(8,7) q(3,1) q(7,4) q(6,2)
Answer: 40
q(2,7) q(4,8) q(3,5) q(1,1) q(7,6) q(6,4) q(5,2) q(8,3)
Answer: 41
q(4,8) q(1,4) q(3,5) q(6,7) q(2,1) q(8,6) q(5,2) q(7,3)
Answer: 42
q(4,6) q(2,3) q(7,7) q(6,5) q(3,1) q(5,2) q(8,4) q(1,8)
Answer: 43
q(2,7) q(1,5) q(6,8) q(4,4) q(7,6) q(3,1) q(5,2) q(8,3)
Answer: 44
q(2,6) q(1,4) q(6,8) q(4,5) q(8,7) q(3,1) q(5,2) q(7,3)
```



```
Answer: 45
q(3,8) q(1,5) q(6,7) q(4,4) q(2,1) q(8,6) q(5,2) q(7,3)
Answer: 46
q(2,7) q(4,8) q(1,4) q(3,3) q(6,5) q(8,6) q(5,2) q(7,1)
Answer: 47
q(1,5) q(4,7) q(6,8) q(2,3) q(7,6) q(3,1) q(5,2) q(8,4)
Answer: 48
q(1,5) q(4,7) q(6,8) q(3,4) q(2,2) q(7,6) q(5,3) q(8,1)
Answer: 49
q(1,6) q(4,7) q(3,5) q(2,3) q(8,8) q(6,4) q(5,1) q(7,2)
Answer: 50
q(1,6) q(4,7) q(6,8) q(2,3) q(5,5) q(3,1) q(8,4) q(7,2)
Answer: 51
q(2,6) q(4,7) q(6,8) q(1,2) q(5,4) q(3,1) q(8,5) q(7,3)
Answer: 52
q(2,6) q(4,7) q(1,3) q(7,8) q(5,5) q(3,2) q(8,4) q(6,1)
Answer: 53
q(2,8) q(4,7) q(1,3) q(3,4) q(6,6) q(8,5) q(5,1) q(7,2)
Answer: 54
q(2,6) q(4,7) q(1,3) q(7,8) q(3,2) q(6,4) q(8,5) q(5,1)
Answer: 55
q(3,7) q(1,4) q(6,8) q(5,6) q(2,2) q(4,3) q(7,5) q(8,1)
Answer: 56
q(3,8) q(2,6) q(1,3) q(6,7) q(5,4) q(7,5) q(4,1) q(8,2)
Answer: 57
q(1,6) q(3,7) q(6,8) q(2,2) q(5,4) q(7,5) q(4,1) q(8,3)
Answer: 58
q(3,8) q(2,6) q(1,4) q(6,7) q(4,3) q(7,5) q(5,1) q(8,2)
Answer: 59
q(2,7) q(5,8) q(4,6) q(3,4) q(1,1) q(7,5) q(6,2) q(8,3)
Answer: 60
q(3,8) q(1,4) q(4,6) q(2,2) q(8,7) q(7,5) q(6,3) q(5,1)
Answer: 61
q(2,8) q(1,6) q(6,7) q(4,4) q(3,2) q(7,5) q(5,1) q(8,3)
Answer: 62
q(2,6) q(5,8) q(1,3) q(3,4) q(7,7) q(6,5) q(4,2) q(8,1)
Answer: 63
q(2,7) q(1,4) q(3,5) q(5,6) q(8,8) q(4,2) q(7,3) q(6,1)
Answer: 64
q(1,6) q(3,7) q(5,8) q(2,3) q(6,5) q(4,2) q(8,4) q(7,1)
Answer: 65
q(1,6) q(5,8) q(3,5) q(7,7) q(2,1) q(4,2) q(6,3) q(8,4)
Answer: 66
q(1,6) q(3,7) q(6,8) q(2,3) q(5,4) q(4,2) q(8,5) q(7,1)
Answer: 67
q(3,8) q(2,6) q(1,3) q(7,7) q(5,4) q(4,2) q(8,5) q(6,1)
```

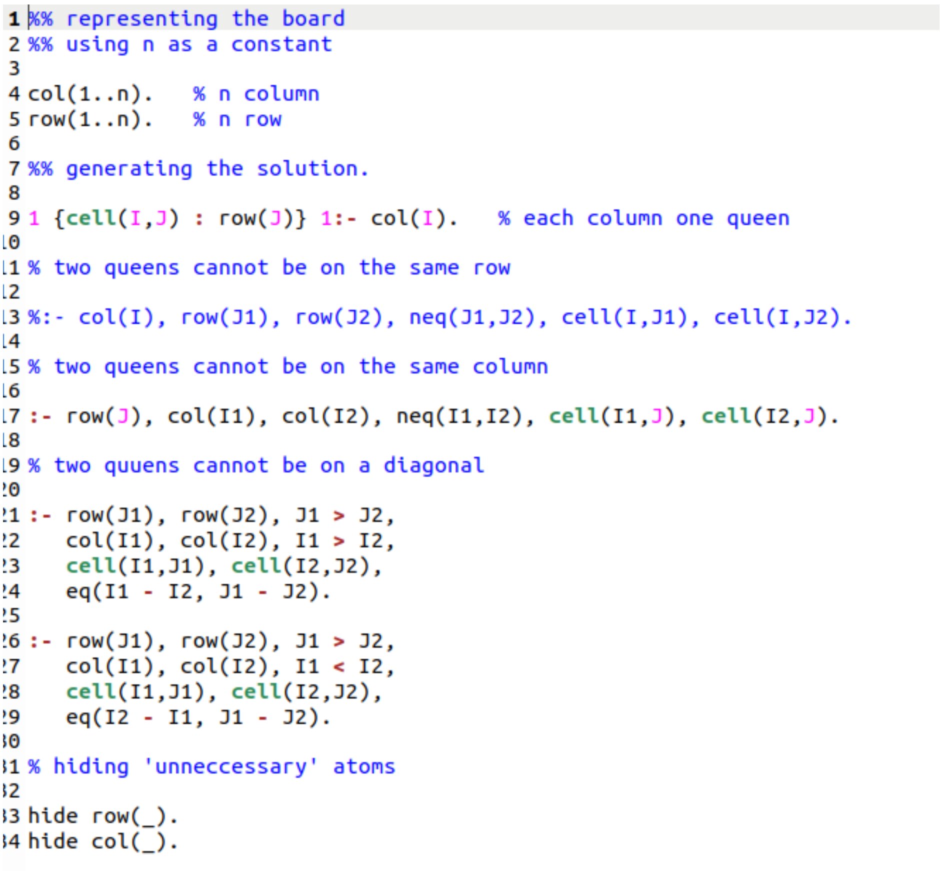


```
Answer: 68
q(1,7) q(3,8) q(2,3) q(5,5) q(7,6) q(4,2) q(8,4) q(6,1)
Answer: 69
q(3,8) q(2,6) q(5,7) q(1,1) q(4,3) q(6,4) q(8,5) q(7,2)
Answer: 70
q(3,8) q(2,6) q(1,4) q(5,7) q(4,2) q(8,5) q(7,3) q(6,1)
Answer: 71
q(3,8) q(1,4) q(5,7) q(4,5) q(2,2) q(8,6) q(7,3) q(6,1)
Answer: 72
q(3,8) q(1,5) q(5,7) q(2,3) q(4,4) q(7,6) q(6,1) q(8,2)
Answer: 73
q(5,7) q(3,4) q(2,2) q(6,5) q(8,6) q(4,1) q(7,3) q(1,8)
Answer: 74
q(2,8) q(1,5) q(5,7) q(3,4) q(7,6) q(4,1) q(6,2) q(8,3)
Answer: 75
q(1,5) q(3,6) q(5,7) q(7,8) q(2,2) q(6,4) q(4,1) q(8,3)
Answer: 76
q(2,8) q(1,4) q(5,7) q(4,5) q(7,6) q(3,1) q(6,2) q(8,3)
Answer: 77
q(1,6) q(3,7) q(5,8) q(2,4) q(7,5) q(4,1) q(6,2) q(8,3)
Answer: 78
q(1,6) q(3,7) q(2,4) q(8,8) q(6,5) q(5,3) q(4,1) q(7,2)
Answer: 79
q(4,8) q(3,6) q(2,4) q(1,2) q(7,7) q(5,3) q(8,5) q(6,1)
Answer: 80
q(1,6) q(4,8) q(2,4) q(6,7) q(5,5) q(3,2) q(8,3) q(7,1)
Answer: 81
q(1,6) q(5,8) q(2,4) q(4,5) q(7,7) q(3,1) q(6,2) q(8,3)
Answer: 82
q(1,7) q(5,8) q(2,4) q(4,5) q(3,2) q(8,6) q(7,3) q(6,1)
Answer: 83
q(1,7) q(4,8) q(2,4) q(5,6) q(3,2) q(8,5) q(7,3) q(6,1)
Answer: 84
q(2,4) q(5,6) q(7,7) q(4,3) q(3,1) q(8,5) q(6,2) q(1,8)
Answer: 85
q(3,7) q(2,5) q(6,8) q(1,2) q(4,4) q(7,6) q(5,1) q(8,3)
Answer: 86
q(3,7) q(2,5) q(1,3) q(7,8) q(5,4) q(8,6) q(4,1) q(6,2)
Answer: 87
q(1,7) q(2,5) q(6,8) q(5,6) q(3,3) q(4,1) q(8,4) q(7,2)
Answer: 88
q(3,7) q(2,5) q(6,8) q(1,2) q(7,6) q(5,3) q(4,1) q(8,4)
Answer: 89
q(3,8) q(2,5) q(4,6) q(6,7) q(1,1) q(5,3) q(8,4) q(7,2)
Answer: 90
q(4,8) q(2,5) q(1,3) q(5,6) q(7,7) q(3,2) q(6,4) q(8,1)
Answer: 91
q(3,8) q(2,5) q(1,3) q(6,7) q(4,4) q(8,6) q(5,1) q(7,2)
```



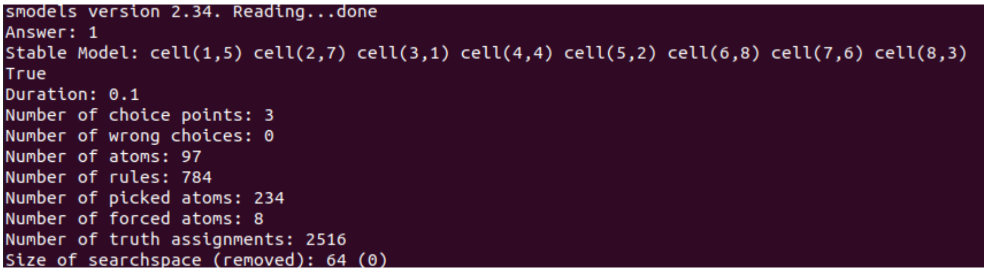
```
Answer: 89
q(3,8) q(2,5) q(4,6) q(6,7) q(1,1) q(5,3) q(8,4) q(7,2)
Answer: 90
q(4,8) q(2,5) q(1,3) q(5,6) q(7,7) q(3,2) q(6,4) q(8,1)
Answer: 91
q(3,8) q(2,5) q(1,3) q(6,7) q(4,4) q(8,6) q(5,1) q(7,2)
Answer: 92
q(4,8) q(2,5) q(1,3) q(6,7) q(3,2) q(8,6) q(7,4) q(5,1)
SATISFIABLE

Models       : 92
Calls        : 1
Time         : 0.032s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 446
Conflicts    : 336      (Analyzed: 280)
Restarts     : 14       (Average: 20.00 Last: 257)
Model-Level  : 8.1
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 280      (Deleted: 0)
  Binary     : 3        (Ratio:   1.07%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 280      (Average Length:   13.0 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 280      (Average:  1.07 Max:   8 Sum:    299)
  Executed   : 280      (Average:  1.07 Max:   8 Sum:    299 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 380      (Original: 148)
  Choice     : 16
Atoms        : 154
Bodies       : 243      (Original: 75)
  Count      : 22       (Original: 54)
Equivalences : 112      (Atom=Atom: 16 Body=Body: 0 Other: 96)
Tight        : Yes
Variables    : 80       (Eliminated:    0 Frozen:   64)
Constraints  : 178      (Binary:  78.7% Ternary:   0.0% Other:  21.3%)
```

Since SModels and Lparse are using a different syntax, We had to change the code.



```
%% representing the board
%% using n as a constant

col(1..n).   % n column
row(1..n).   % n row

%% generating the solution.

1 {cell(I,J) : row(J)} 1:- col(I).   % each column one queen

% two queens cannot be on the same row

%:- col(I), row(J1), row(J2), neq(J1,J2), cell(I,J1), cell(I,J2).

% two queens cannot be on the same column

:- row(J), col(I1), col(I2), neq(I1,I2), cell(I1,J), cell(I2,J).

% two quuens cannot be on a diagonal

:- row(J1), row(J2), J1 > J2,
   col(I1), col(I2), I1 > I2,
   cell(I1,J1), cell(I2,J2),
   eq(I1 - I2, J1 - J2).

:- row(J1), row(J2), J1 > J2,
   col(I1), col(I2), I1 < I2,
   cell(I1,J1), cell(I2,J2),
   eq(I2 - I1, J1 - J2).

% hiding 'unneccessary' atoms

hide row(_).
hide col(_).
```

SModels Output:



```
smodels version 2.34. Reading...done
Answer: 1
Stable Model: cell(1,5) cell(2,7) cell(3,1) cell(4,4) cell(5,2) cell(6,8) cell(7,6) cell(8,3)
True
Duration: 0.1
Number of choice points: 3
Number of wrong choices: 0
Number of atoms: 97
Number of rules: 784
Number of picked atoms: 234
Number of forced atoms: 8
Number of truth assignments: 2516
Size of searchspace (removed): 64 (0)
```

Four-Color Theorem Problem

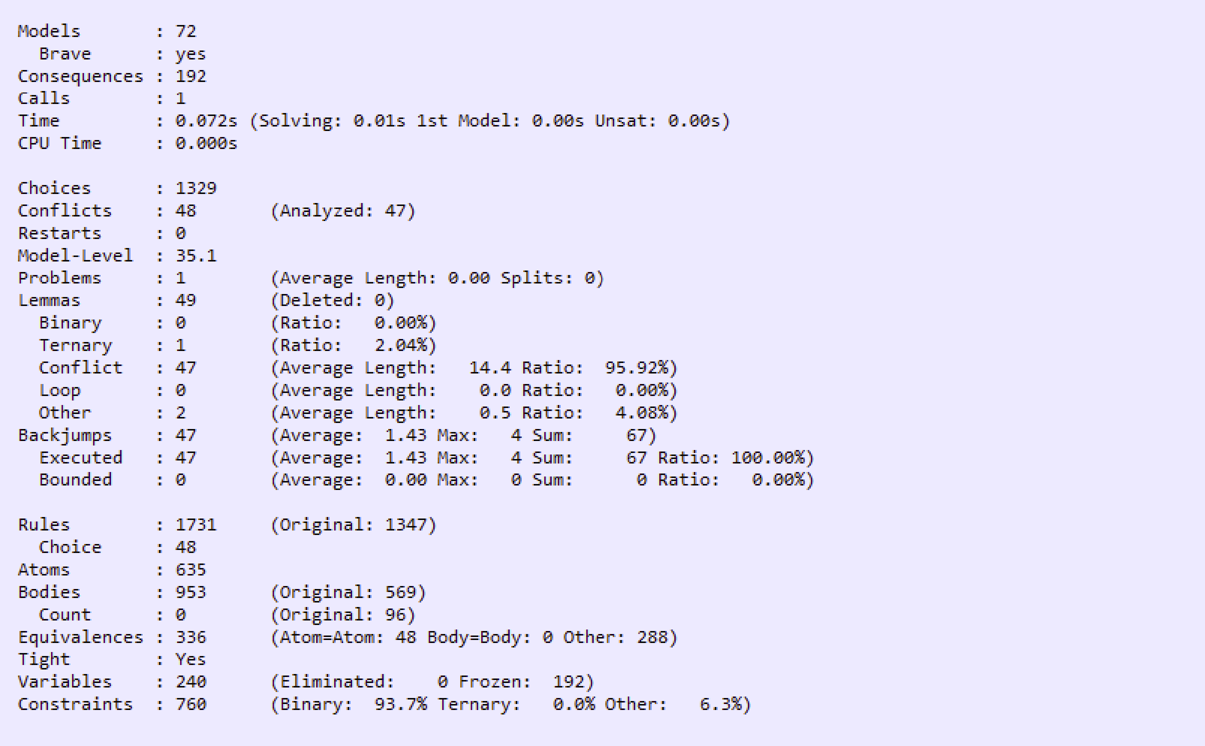
Problem

```
% Default
#const c = 4.
#const n = 48.
% Generate
{ color(X,1..c) } = 1 :- node(X).
% Test
:- edge(X,Y), color(X,C), color(Y,C).
% Nodes
node(1..n).
% (Directed) Edges
edge(1,(2;7)). edge(2,(1;5;3;7)). edge(3,(2;5;4)). edge(5,(2;3;7;6;4)).
edge(7,(1;2;5;8;9;6)). edge(8,(7;9;16;17)). edge(9,(7;8;6;10;15;16)).
edge(6,(5;7;9;4;10)). edge(4,(3;5;6;11)). edge(11,(4;10;13;12)).
edge(10,(9;6;11;15;14;13)). edge(15,(9;10;16;19;20;14)). edge(16,(8;9;15;17;18;19)).
edge(17,(8;16;18)). edge(18,(16;17;19;23)). edge(19,(15;16;18;20;25;23)).
edge(20,(15;19;14;13;21;29;25)). edge(14,(10;15;20;13)). edge(13,(11;10;14;20;12;21)).
edge(12,(11;13;22;21)). edge(22,(12;21;30)). edge(21,(20;13;12;22;30;29)). edge(30,(22;21;31;29)).
edge(31,(30;33;32;29)). edge(33,(31;32)). edge(32,(31;33;34;29;35)). edge(34,(32;35)).
edge(29,(20;21;30;31;32;34;35;28;36)). edge(35,(32;34;29;36)). edge(28,(20;29;25;26;27;37;36)).
edge(25,(19;20;12;23;26)). edge(23,(18;19;25)). edge(24,(26;27)). edge(26,(28;25;24;27)).
edge(27,(28;24;26;37;38)). edge(37,(28;27;36;48;38)). edge(36,(29;35;28;37;48)).
edge(48,(37;36;47;38)). edge(47,(48;46;38)). edge(46,(47;38;39)). edge(38,(27,37,48,47,46,39)).
edge(39,(46;38;40;43;45)). edge(40,(39;41;43)). edge(42,(41)). edge(41,(40;42;43)).
edge(43,(39;40;41;44;45)). edge(44,(43;45)). edge(45,(39;43;44)).
% Display
#show color/2.
```

Default

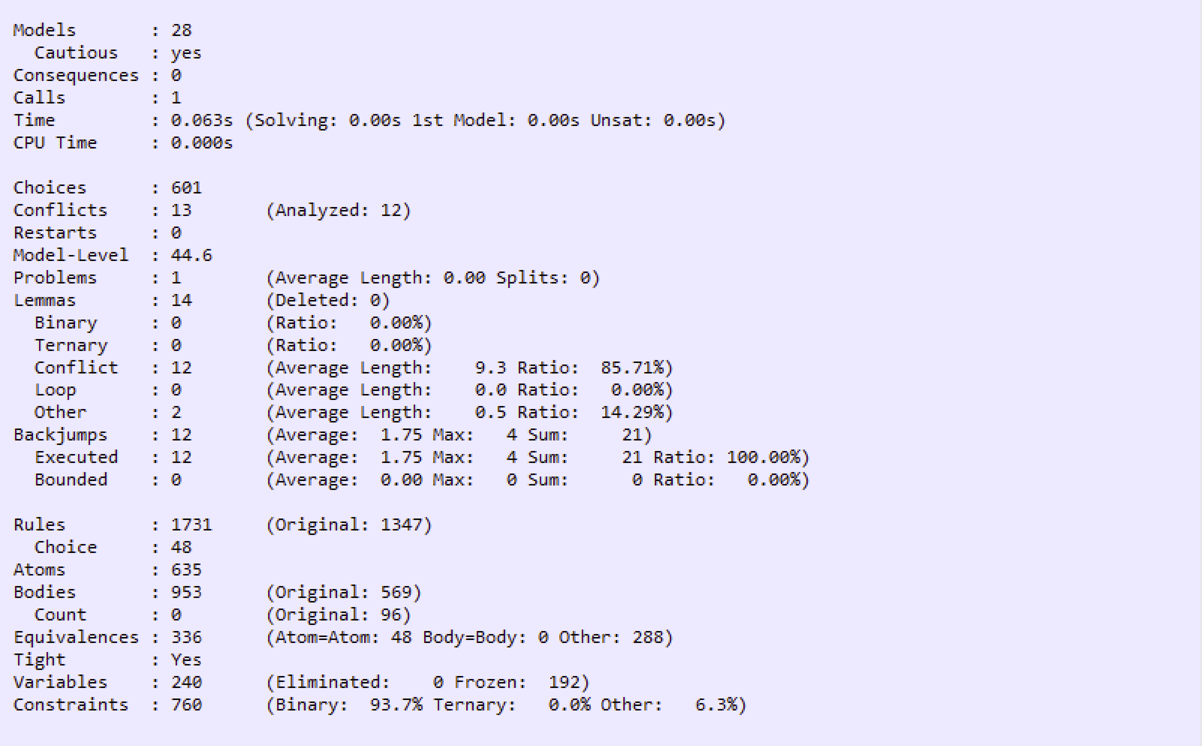
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
color(2,1) color(7,2) color(1,4) color(3,2) color(5,4) color(4,3) color(6,1) color(11,4) color(10,3) col
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.060s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 42
Conflicts    : 0        (Analyzed: 0)
Restarts     : 0
Model-Level  : 42.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 0        (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 0        (Average Length:    0.0 Ratio:   0.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 0        (Average:  0.00 Max:   0 Sum:      0)
  Executed   : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio: 100.00%)

Rules        : 1731     (Original: 1347)
  Choice     : 48
Atoms        : 635
Bodies       : 953      (Original: 569)
  Count      : 0        (Original: 96)
Equivalences : 336      (Atom=Atom: 48 Body=Body: 0 Other: 288)
Tight        : Yes
Variables    : 240      (Eliminated:    0 Frozen:    0)
Constraints  : 760      (Binary:  93.7% Ternary:   0.0% Other:   6.3%)
```

Brave



```
Models       : 72
  Brave      : yes
Consequences : 192
Calls        : 1
Time         : 0.072s (Solving: 0.01s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 1329
Conflicts    : 48       (Analyzed: 47)
Restarts     : 0
Model-Level  : 35.1
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 49       (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 1        (Ratio:   2.04%)
  Conflict   : 47       (Average Length:   14.4 Ratio:  95.92%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 2        (Average Length:    0.5 Ratio:   4.08%)
Backjumps    : 47       (Average:  1.43 Max:   4 Sum:     67)
  Executed   : 47       (Average:  1.43 Max:   4 Sum:     67 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 1731     (Original: 1347)
  Choice     : 48
Atoms        : 635
Bodies       : 953      (Original: 569)
  Count      : 0        (Original: 96)
Equivalences : 336      (Atom=Atom: 48 Body=Body: 0 Other: 288)
Tight        : Yes
Variables    : 240      (Eliminated:    0 Frozen:  192)
Constraints  : 760      (Binary:  93.7% Ternary:   0.0% Other:   6.3%)
```

Cautious



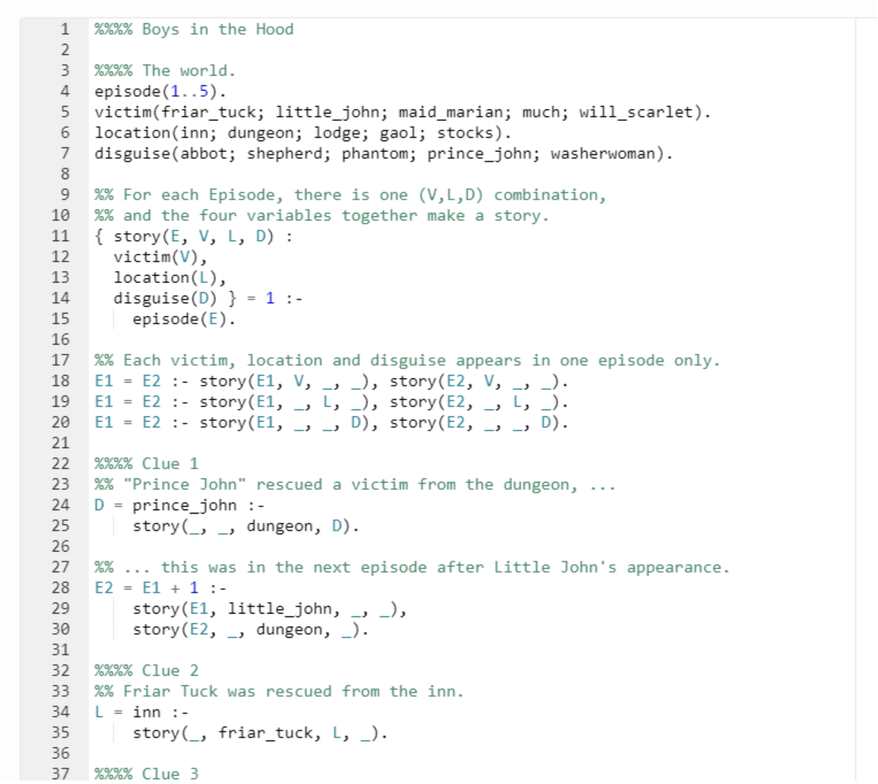
```
Models       : 28
  Cautious   : yes
Consequences : 0
Calls        : 1
Time         : 0.063s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 601
Conflicts    : 13       (Analyzed: 12)
Restarts     : 0
Model-Level  : 44.6
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 14       (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 12       (Average Length:    9.3 Ratio:  85.71%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 2        (Average Length:    0.5 Ratio:  14.29%)
Backjumps    : 12       (Average:  1.75 Max:   4 Sum:     21)
  Executed   : 12       (Average:  1.75 Max:   4 Sum:     21 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 1731     (Original: 1347)
  Choice     : 48
Atoms        : 635
Bodies       : 953      (Original: 569)
  Count      : 0        (Original: 96)
Equivalences : 336      (Atom=Atom: 48 Body=Body: 0 Other: 288)
Tight        : Yes
Variables    : 240      (Eliminated:    0 Frozen:  192)
Constraints  : 760      (Binary:  93.7% Ternary:   0.0% Other:   6.3%)
```

Enumerate All

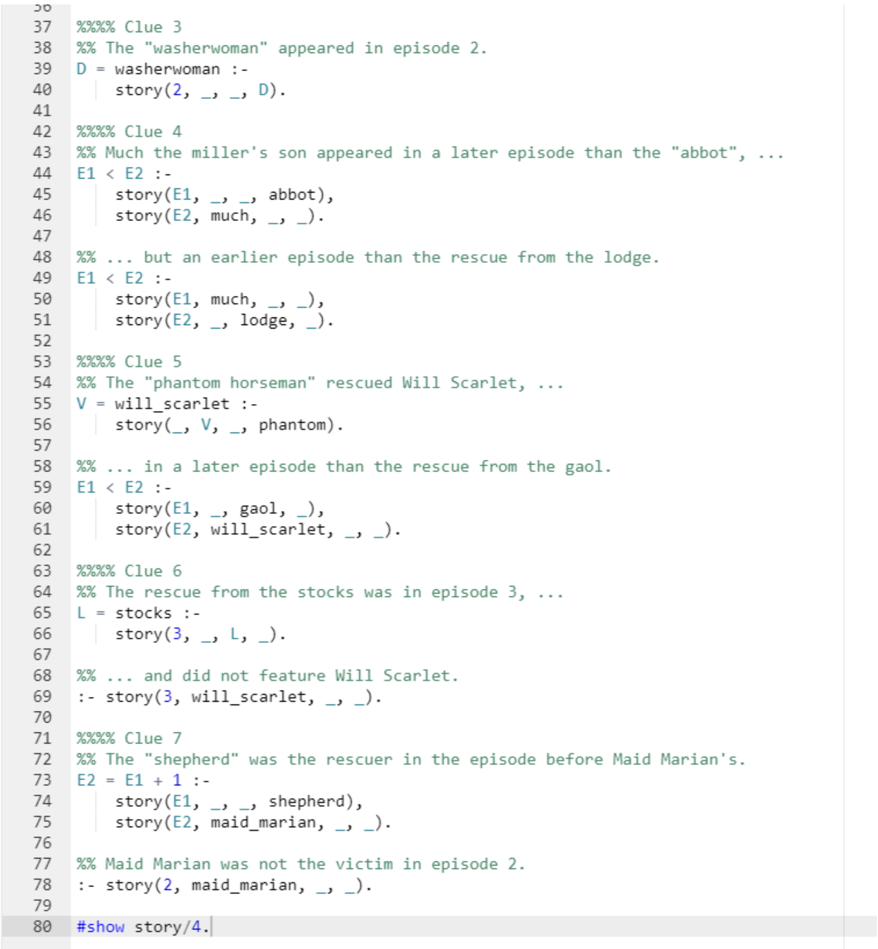
Browser crashes.

Logic Puzzle Problem #1

<u>Problem</u>

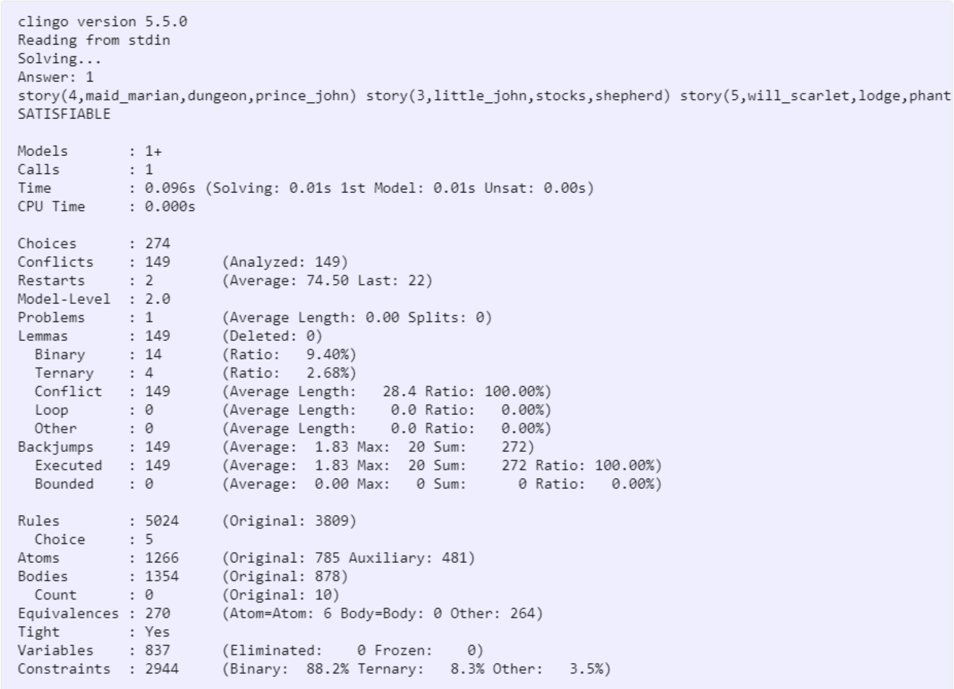


```
%%%% Boys in the Hood

%%%% The world.
episode(1..5).
victim(friar_tuck; little_john; maid_marian; much; will_scarlet).
location(inn; dungeon; lodge; gaol; stocks).
disguise(abbot; shepherd; phantom; prince_john; washerwoman).

%% For each Episode, there is one (V,L,D) combination,
%% and the four variables together make a story.
{ story(E, V, L, D) :
  victim(V),
  location(L),
  disguise(D) } = 1 :-
    episode(E).

%% Each victim, location and disguise appears in one episode only.
E1 = E2 :- story(E1, V, _, _), story(E2, V, _, _).
E1 = E2 :- story(E1, _, L, _), story(E2, _, L, _).
E1 = E2 :- story(E1, _, _, D), story(E2, _, _, D).

%%%% Clue 1
%% "Prince John" rescued a victim from the dungeon, ...
D = prince_john :-
    story(_, _, dungeon, D).

%% ... this was in the next episode after Little John's appearance.
E2 = E1 + 1 :-
    story(E1, little_john, _, _),
    story(E2, _, dungeon, _).

%%%% Clue 2
%% Friar Tuck was rescued from the inn.
L = inn :-
    story(_, friar_tuck, L, _).

%%%% Clue 3
```



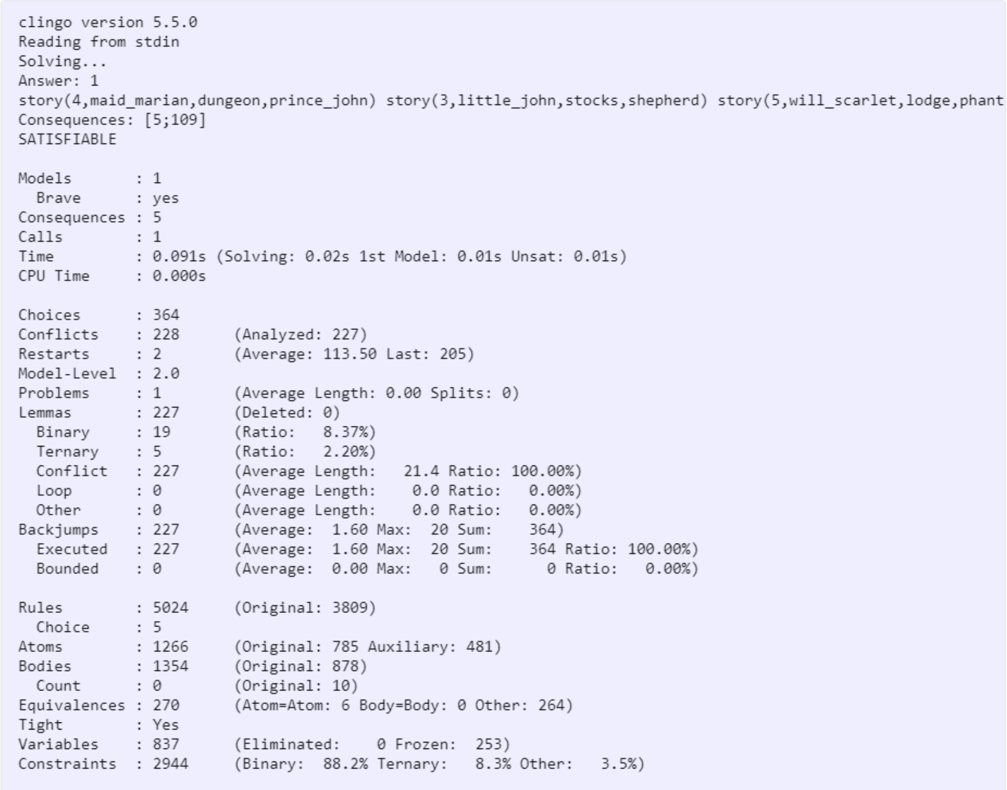
```
%%%% Clue 3
%% The "washerwoman" appeared in episode 2.
D = washerwoman :-
    story(2, _, _, D).

%%%% Clue 4
%% Much the miller's son appeared in a later episode than the "abbot", ...
E1 < E2 :-
    story(E1, _, _, abbot),
    story(E2, much, _, _).

%% ... but an earlier episode than the rescue from the lodge.
E1 < E2 :-
    story(E1, much, _, _),
    story(E2, _, lodge, _).

%%%% Clue 5
%% The "phantom horseman" rescued Will Scarlet, ...
V = will_scarlet :-
    story(_, V, _, phantom).

%% ... in a later episode than the rescue from the gaol.
E1 < E2 :-
    story(E1, _, gaol, _),
    story(E2, will_scarlet, _, _).

%%%% Clue 6
%% The rescue from the stocks was in episode 3, ...
L = stocks :-
    story(3, _, L, _).

%% ... and did not feature Will Scarlet.
:- story(3, will_scarlet, _, _).

%%%% Clue 7
%% The "shepherd" was the rescuer in the episode before Maid Marian's.
E2 = E1 + 1 :-
    story(E1, _, _, shepherd),
    story(E2, maid_marian, _, _).

%% Maid Marian was not the victim in episode 2.
:- story(2, maid_marian, _, _).

#show story/4.
```

Default



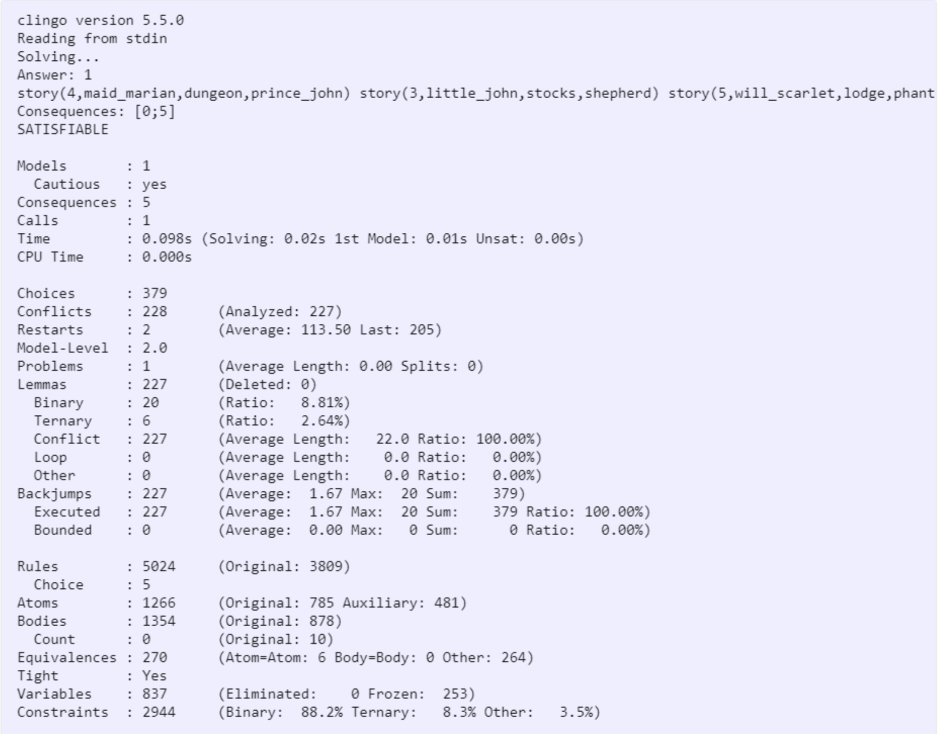
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
story(4,maid_marian,dungeon,prince_john) story(3,little_john,stocks,shepherd) story(5,will_scarlet,lodge,phant
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.096s (Solving: 0.01s 1st Model: 0.01s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 274
Conflicts    : 149      (Analyzed: 149)
Restarts     : 2        (Average: 74.50 Last: 22)
Model-Level  : 2.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 149      (Deleted: 0)
  Binary     : 14       (Ratio:   9.40%)
  Ternary    : 4        (Ratio:   2.68%)
  Conflict   : 149      (Average Length:   28.4 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 149      (Average:  1.83 Max:  20 Sum:    272)
  Executed   : 149      (Average:  1.83 Max:  20 Sum:    272 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 5024     (Original: 3809)
  Choice     : 5
Atoms        : 1266     (Original: 785 Auxiliary: 481)
Bodies       : 1354     (Original: 878)
  Count      : 0        (Original: 10)
Equivalences : 270      (Atom=Atom: 6 Body=Body: 0 Other: 264)
Tight        : Yes
Variables    : 837      (Eliminated:    0 Frozen:    0)
Constraints  : 2944     (Binary:  88.2% Ternary:   8.3% Other:   3.5%)
```

Brave



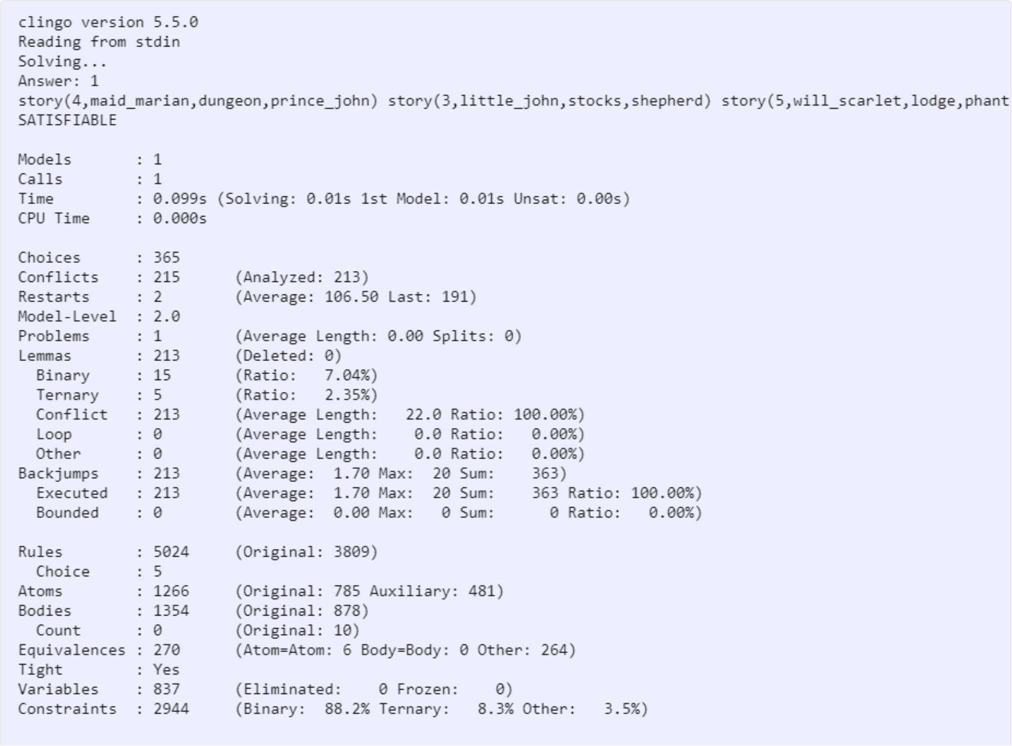
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
story(4,maid_marian,dungeon,prince_john) story(3,little_john,stocks,shepherd) story(5,will_scarlet,lodge,phant
Consequences: [5;109]
SATISFIABLE

Models       : 1
  Brave      : yes
Consequences : 5
Calls        : 1
Time         : 0.091s (Solving: 0.02s 1st Model: 0.01s Unsat: 0.01s)
CPU Time     : 0.000s

Choices      : 364
Conflicts    : 228      (Analyzed: 227)
Restarts     : 2        (Average: 113.50 Last: 205)
Model-Level  : 2.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 227      (Deleted: 0)
  Binary     : 19       (Ratio:   8.37%)
  Ternary    : 5        (Ratio:   2.20%)
  Conflict   : 227      (Average Length:   21.4 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 227      (Average:  1.60 Max:  20 Sum:    364)
  Executed   : 227      (Average:  1.60 Max:  20 Sum:    364 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 5024     (Original: 3809)
  Choice     : 5
Atoms        : 1266     (Original: 785 Auxiliary: 481)
Bodies       : 1354     (Original: 878)
  Count      : 0        (Original: 10)
Equivalences : 270      (Atom=Atom: 6 Body=Body: 0 Other: 264)
Tight        : Yes
Variables    : 837      (Eliminated:    0 Frozen:  253)
Constraints  : 2944     (Binary:  88.2% Ternary:   8.3% Other:   3.5%)
```

Cautious



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
story(4,maid_marian,dungeon,prince_john) story(3,little_john,stocks,shepherd) story(5,will_scarlet,lodge,phant
Consequences: [0;5]
SATISFIABLE

Models       : 1
  Cautious   : yes
Consequences : 5
Calls        : 1
Time         : 0.098s (Solving: 0.02s 1st Model: 0.01s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 379
Conflicts    : 228      (Analyzed: 227)
Restarts     : 2        (Average: 113.50 Last: 205)
Model-Level  : 2.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 227      (Deleted: 0)
  Binary     : 20       (Ratio:   8.81%)
  Ternary    : 6        (Ratio:   2.64%)
  Conflict   : 227      (Average Length:   22.0 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 227      (Average:  1.67 Max:  20 Sum:    379)
  Executed   : 227      (Average:  1.67 Max:  20 Sum:    379 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 5024     (Original: 3809)
  Choice     : 5
Atoms        : 1266     (Original: 785 Auxiliary: 481)
Bodies       : 1354     (Original: 878)
  Count      : 0        (Original: 10)
Equivalences : 270      (Atom=Atom: 6 Body=Body: 0 Other: 264)
Tight        : Yes
Variables    : 837      (Eliminated:    0 Frozen:  253)
Constraints  : 2944     (Binary:  88.2% Ternary:   8.3% Other:   3.5%)
```

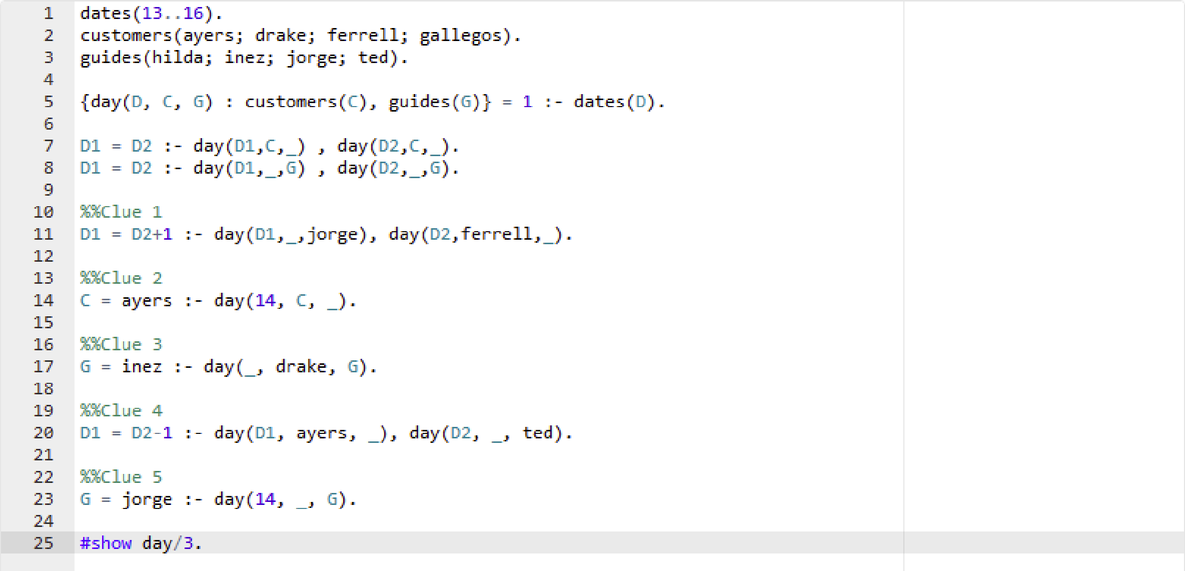
Enumerate All



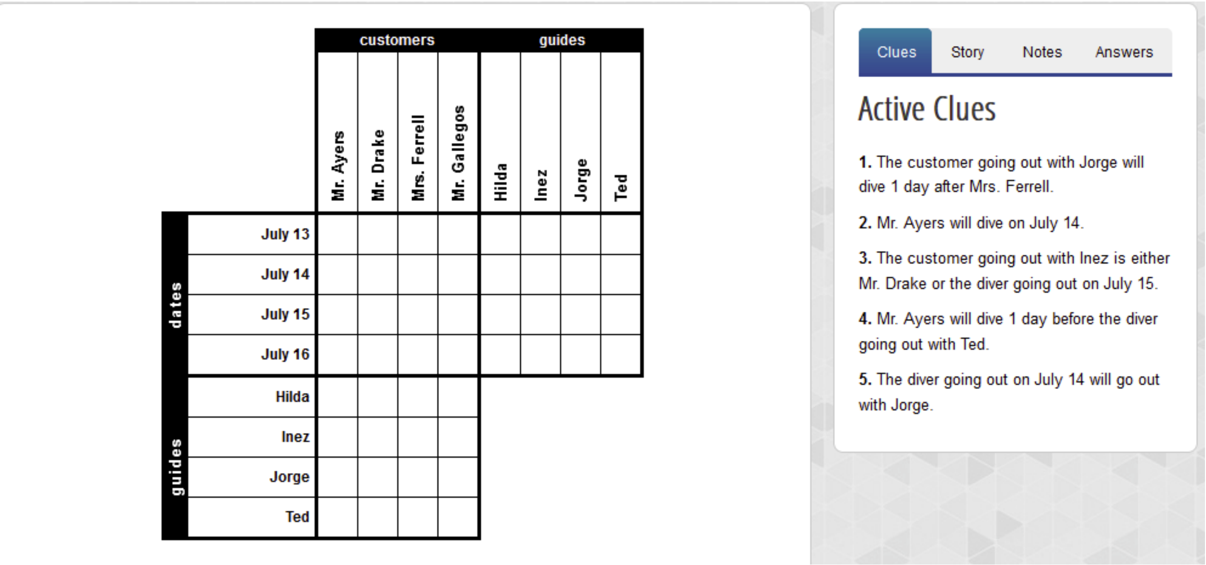
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
story(4,maid_marian,dungeon,prince_john) story(3,little_john,stocks,shepherd) story(5,will_scarlet,lodge,phant
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.099s (Solving: 0.01s 1st Model: 0.01s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 365
Conflicts    : 215      (Analyzed: 213)
Restarts     : 2        (Average: 106.50 Last: 191)
Model-Level  : 2.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 213      (Deleted: 0)
  Binary     : 15       (Ratio:   7.04%)
  Ternary    : 5        (Ratio:   2.35%)
  Conflict   : 213      (Average Length:   22.0 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 213      (Average:  1.70 Max:  20 Sum:    363)
  Executed   : 213      (Average:  1.70 Max:  20 Sum:    363 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 5024     (Original: 3809)
  Choice     : 5
Atoms        : 1266     (Original: 785 Auxiliary: 481)
Bodies       : 1354     (Original: 878)
  Count      : 0        (Original: 10)
Equivalences : 270      (Atom=Atom: 6 Body=Body: 0 Other: 264)
Tight        : Yes
Variables    : 837      (Eliminated:    0 Frozen:    0)
Constraints  : 2944     (Binary:  88.2% Ternary:   8.3% Other:   3.5%)
```

Logic Puzzle Problem #2

Problem



```
dates(13..16).
customers(ayers; drake; ferrell; gallegos).
guides(hilda; inez; jorge; ted).

{day(D, C, G) : customers(C), guides(G)} = 1 :- dates(D).

D1 = D2 :- day(D1,C,_) , day(D2,C,_).
D1 = D2 :- day(D1,_,G) , day(D2,_,G).

%%Clue 1
D1 = D2+1 :- day(D1,_,jorge), day(D2,ferrell,_).

%%Clue 2
C = ayers :- day(14, C, _).

%%Clue 3
G = inez :- day(_, drake, G).

%%Clue 4
D1 = D2-1 :- day(D1, ayers, _), day(D2, _, ted).

%%Clue 5
G = jorge :- day(14, _, G).

#show day/3.
```



		customers				guides			
		Mr. Ayers	Mr. Drake	Mrs. Ferrell	Mr. Gallegos	Hilda	Inez	Jorge	Ted
dates	July 13								
	July 14								
	July 15								
	July 16								
guides	Hilda				
	Inez				
	Jorge				
	Ted				

Active Clues

1. The customer going out with Jorge will dive 1 day after Mrs. Ferrell.

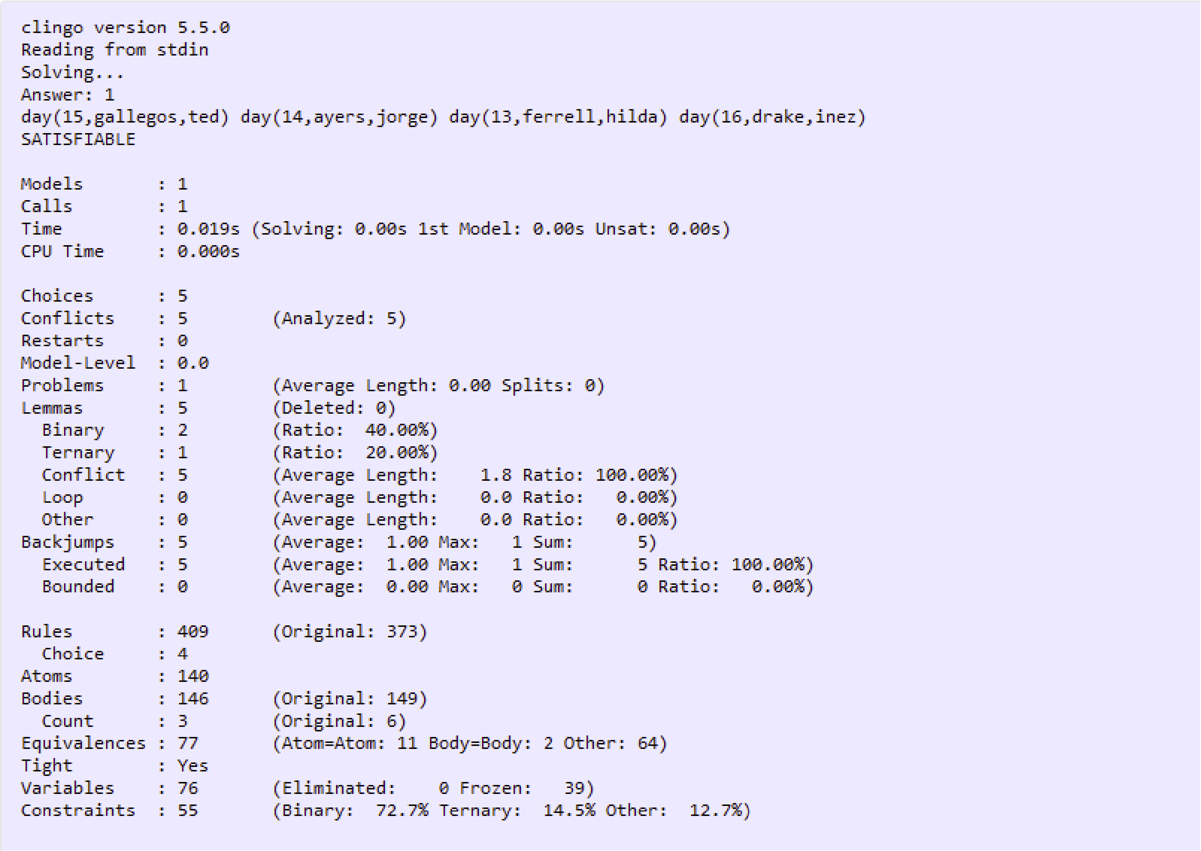
2. Mr. Ayers will dive on July 14.

3. The customer going out with Inez is either Mr. Drake or the diver going out on July 15.

4. Mr. Ayers will dive 1 day before the diver going out with Ted.

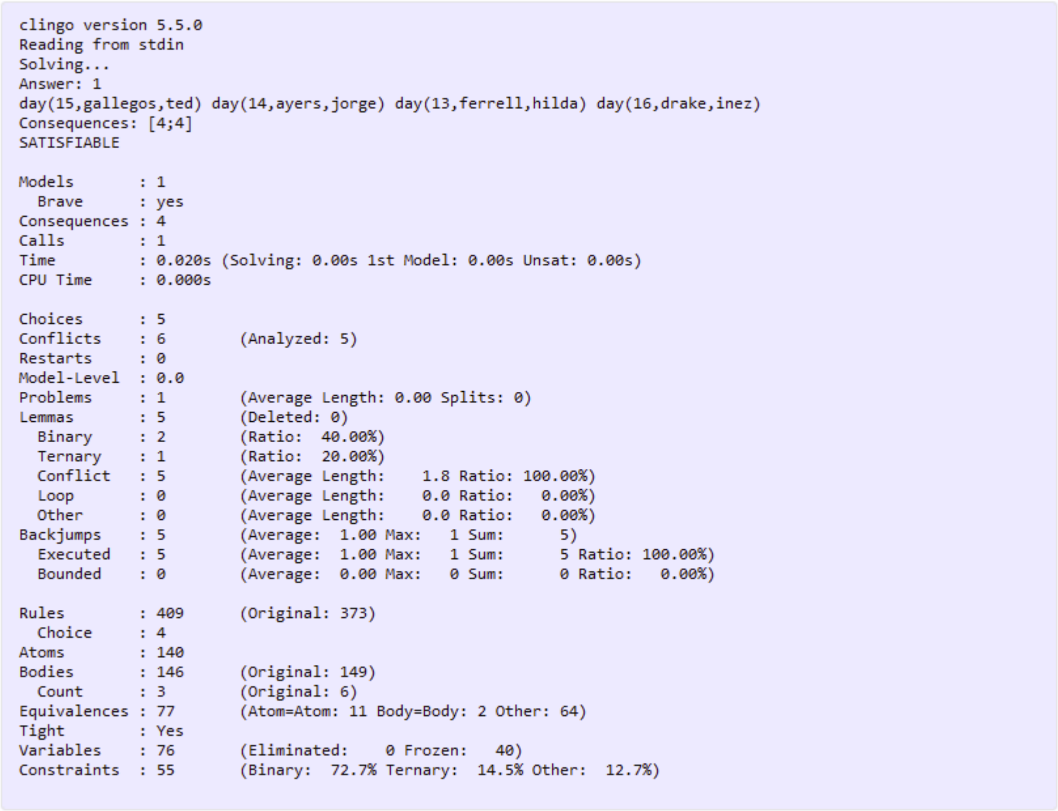
5. The diver going out on July 14 will go out with Jorge.

Default



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
day(15,gallegos,ted) day(14,ayers,jorge) day(13,ferrell,hilda) day(16,drake,inez)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.019s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 5
Conflicts    : 5        (Analyzed: 5)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 5        (Deleted: 0)
  Binary     : 2        (Ratio:  40.00%)
  Ternary    : 1        (Ratio:  20.00%)
  Conflict   : 5        (Average Length:    1.8 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 5        (Average:  1.00 Max:   1 Sum:      5)
  Executed   : 5        (Average:  1.00 Max:   1 Sum:      5 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 409      (Original: 373)
  Choice     : 4
Atoms        : 140
Bodies       : 146      (Original: 149)
  Count      : 3        (Original: 6)
Equivalences : 77       (Atom=Atom: 11 Body=Body: 2 Other: 64)
Tight        : Yes
Variables    : 76       (Eliminated:    0 Frozen:   39)
Constraints  : 55       (Binary:  72.7% Ternary:  14.5% Other:  12.7%)
```

Brave



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
day(15,gallegos,ted) day(14,ayers,jorge) day(13,ferrell,hilda) day(16,drake,inez)
Consequences: [4;4]
SATISFIABLE

Models       : 1
  Brave      : yes
Consequences : 4
Calls        : 1
Time         : 0.020s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 5
Conflicts    : 6        (Analyzed: 5)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 5        (Deleted: 0)
  Binary     : 2        (Ratio:  40.00%)
  Ternary    : 1        (Ratio:  20.00%)
  Conflict   : 5        (Average Length:    1.8 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 5        (Average:  1.00 Max:   1 Sum:      5)
  Executed   : 5        (Average:  1.00 Max:   1 Sum:      5 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 409      (Original: 373)
  Choice     : 4
Atoms        : 140
Bodies       : 146      (Original: 149)
  Count      : 3        (Original: 6)
Equivalences : 77       (Atom=Atom: 11 Body=Body: 2 Other: 64)
Tight        : Yes
Variables    : 76       (Eliminated:    0 Frozen:   40)
Constraints  : 55       (Binary:  72.7% Ternary:  14.5% Other:  12.7%)
```

Cautious

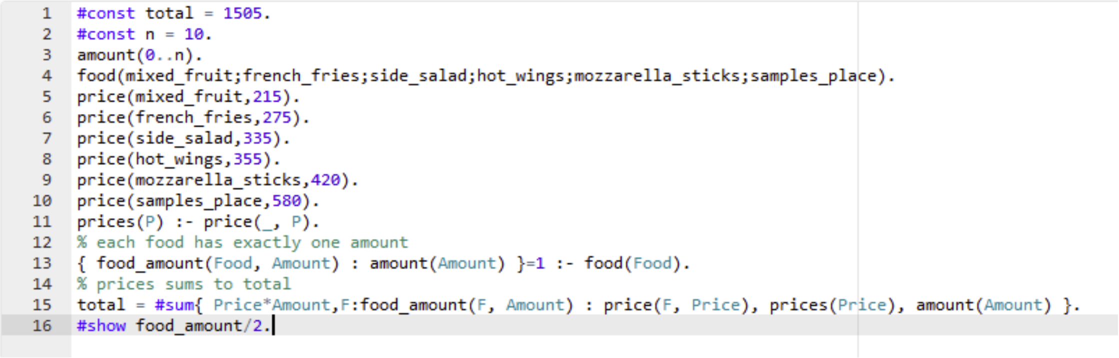
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
day(15,gallegos,ted) day(14,ayers,jorge) day(13,ferrell,hilda) day(16,drake,inez)
Consequences: [4;4]
SATISFIABLE

Models       : 1
  Cautious   : yes
Consequences : 4
Calls        : 1
Time         : 0.016s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 5
Conflicts    : 6        (Analyzed: 5)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 5        (Deleted: 0)
  Binary     : 2        (Ratio:  40.00%)
  Ternary    : 1        (Ratio:  20.00%)
  Conflict   : 5        (Average Length:    1.8 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 5        (Average:  1.00 Max:   1 Sum:      5)
  Executed   : 5        (Average:  1.00 Max:   1 Sum:      5 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 409      (Original: 373)
  Choice     : 4
Atoms        : 140
Bodies       : 146      (Original: 149)
  Count      : 3        (Original: 6)
Equivalences : 77       (Atom=Atom: 11 Body=Body: 2 Other: 64)
Tight        : Yes
Variables    : 76       (Eliminated:    0 Frozen:   40)
Constraints  : 55       (Binary:  72.7% Ternary:  14.5% Other:  12.7%)
```

Enumerate All

```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
day(15,gallegos,ted) day(14,ayers,jorge) day(13,ferrell,hilda) day(16,drake,inez)
SATISFIABLE

Models       : 1
Calls        : 1
Time         : 0.019s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 5
Conflicts    : 5        (Analyzed: 5)
Restarts     : 0
Model-Level  : 0.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 5        (Deleted: 0)
  Binary     : 2        (Ratio:  40.00%)
  Ternary    : 1        (Ratio:  20.00%)
  Conflict   : 5        (Average Length:    1.8 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 5        (Average:  1.00 Max:   1 Sum:      5)
  Executed   : 5        (Average:  1.00 Max:   1 Sum:      5 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 409      (Original: 373)
  Choice     : 4
Atoms        : 140
Bodies       : 146      (Original: 149)
  Count      : 3        (Original: 6)
Equivalences : 77       (Atom=Atom: 11 Body=Body: 2 Other: 64)
Tight        : Yes
Variables    : 76       (Eliminated:    0 Frozen:   39)
Constraints  : 55       (Binary:  72.7% Ternary:  14.5% Other:  12.7%)
```

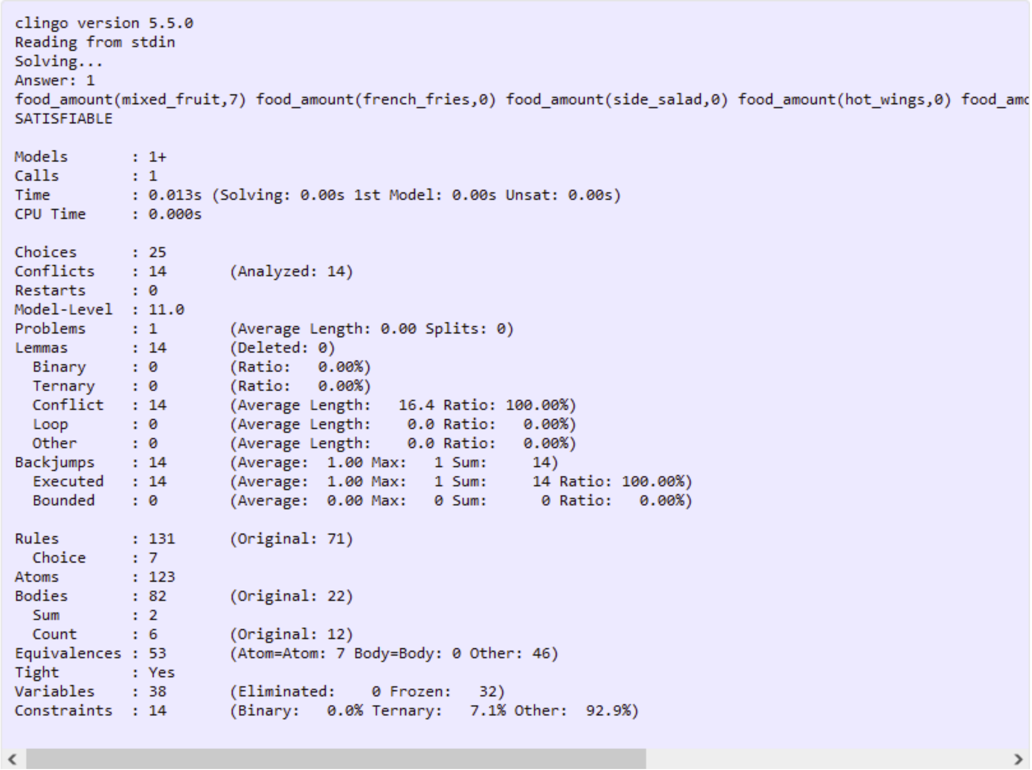
Restaurant Problem

Problem



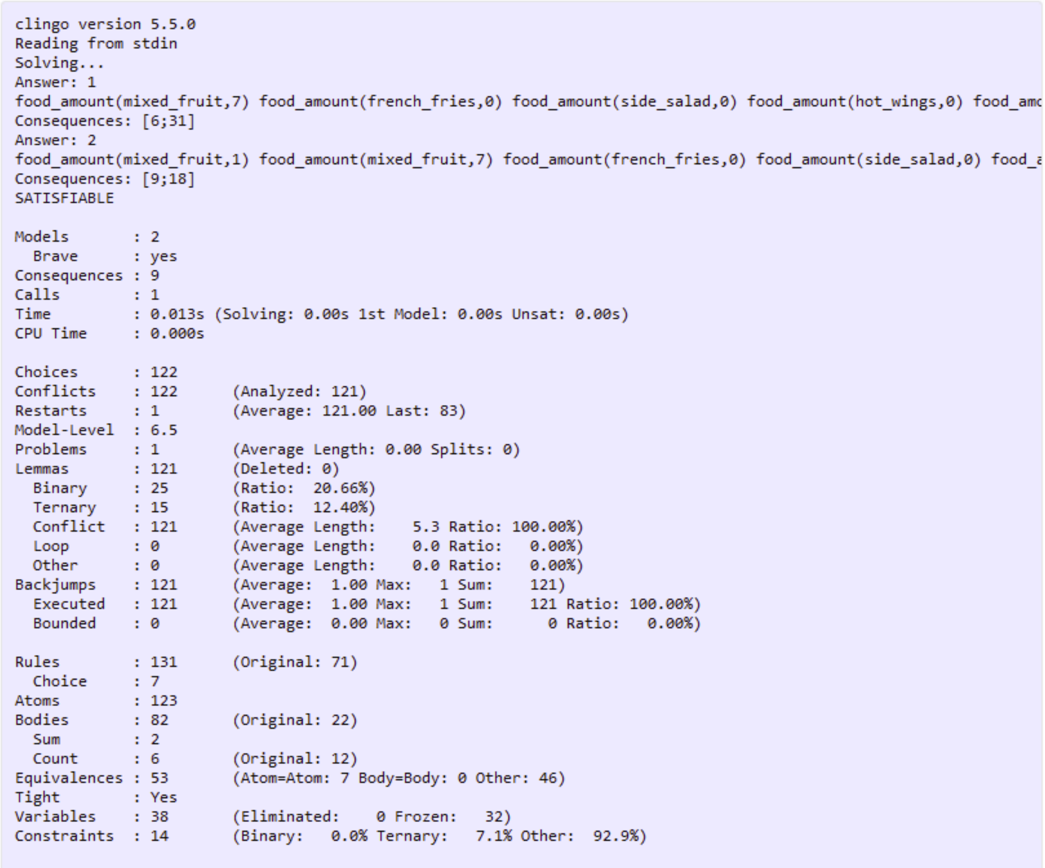
```
#const total = 1505.
#const n = 10.
amount(0..n).
food(mixed_fruit;french_fries;side_salad;hot_wings;mozzarella_sticks;samples_place).
price(mixed_fruit,215).
price(french_fries,275).
price(side_salad,335).
price(hot_wings,355).
price(mozzarella_sticks,420).
price(samples_place,580).
prices(P) :- price(_, P).
% each food has exactly one amount
{ food_amount(Food, Amount) : amount(Amount) }=1 :- food(Food).
% prices sums to total
total = #sum{ Price*Amount,F:food_amount(F, Amount) : price(F, Price), prices(Price), amount(Amount) }.
#show food_amount/2.
```

Default



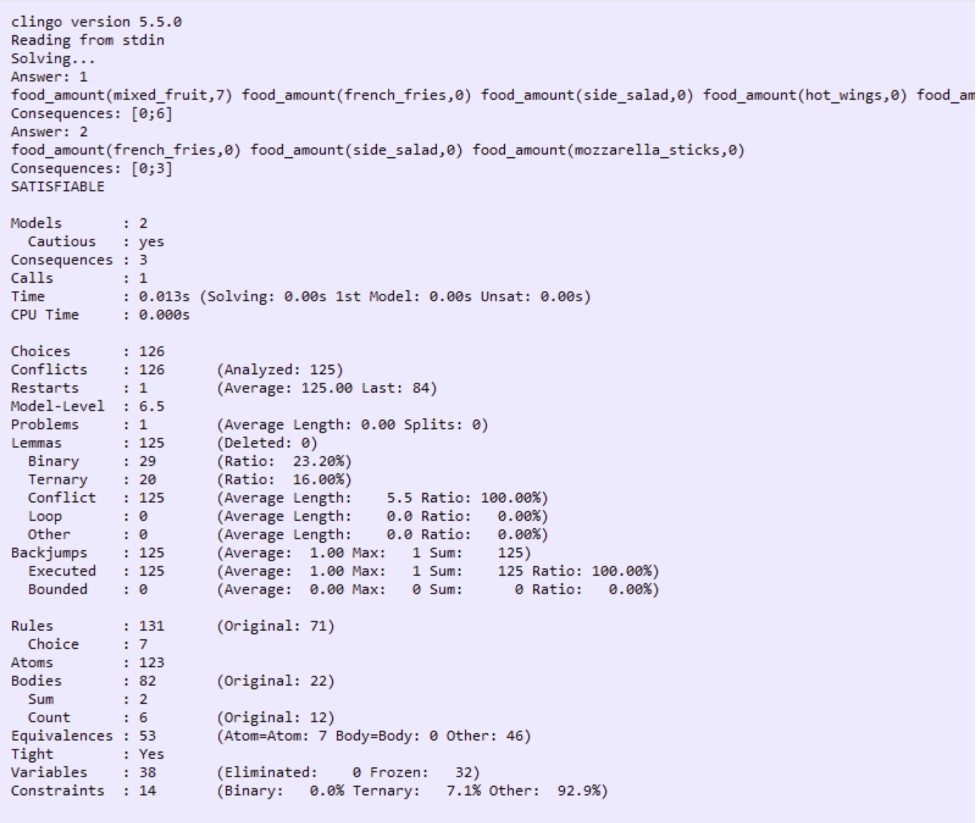
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
food_amount(mixed_fruit,7) food_amount(french_fries,0) food_amount(side_salad,0) food_amount(hot_wings,0) food_amo
SATISFIABLE

Models       : 1+
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 25
Conflicts    : 14       (Analyzed: 14)
Restarts     : 0
Model-Level  : 11.0
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 14       (Deleted: 0)
  Binary     : 0        (Ratio:   0.00%)
  Ternary    : 0        (Ratio:   0.00%)
  Conflict   : 14       (Average Length:   16.4 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 14       (Average:  1.00 Max:   1 Sum:     14)
  Executed   : 14       (Average:  1.00 Max:   1 Sum:     14 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 131      (Original: 71)
  Choice     : 7
Atoms        : 123
Bodies       : 82       (Original: 22)
  Sum        : 2
  Count      : 6        (Original: 12)
Equivalences : 53       (Atom=Atom: 7 Body=Body: 0 Other: 46)
Tight        : Yes
Variables    : 38       (Eliminated:    0 Frozen:   32)
Constraints  : 14       (Binary:   0.0% Ternary:   7.1% Other:  92.9%)
```

Brave



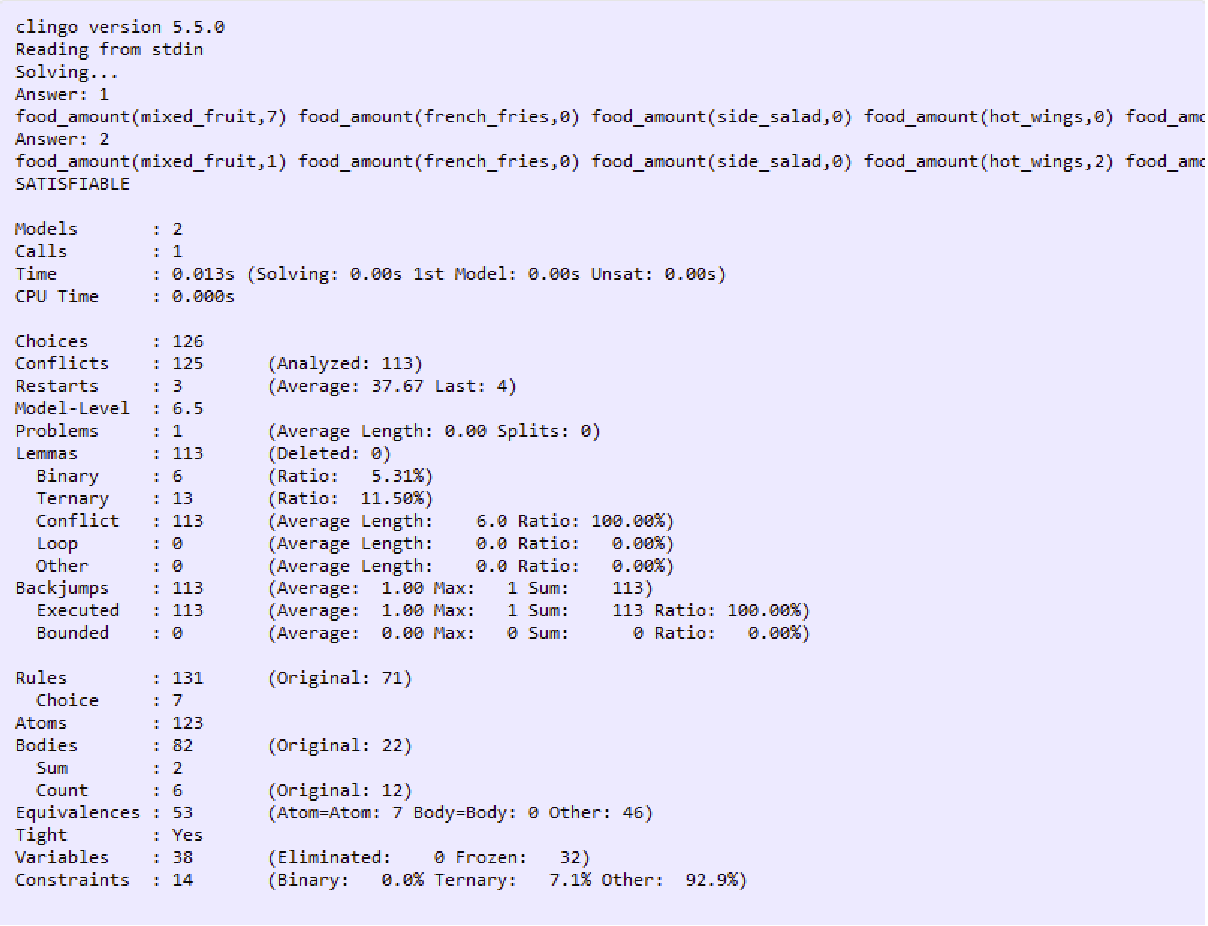
```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
food_amount(mixed_fruit,7) food_amount(french_fries,0) food_amount(side_salad,0) food_amount(hot_wings,0) food_amo
Consequences: [6;31]
Answer: 2
food_amount(mixed_fruit,1) food_amount(mixed_fruit,7) food_amount(french_fries,0) food_amount(side_salad,0) food_a
Consequences: [9;18]
SATISFIABLE

Models       : 2
  Brave      : yes
Consequences : 9
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 122
Conflicts    : 122      (Analyzed: 121)
Restarts     : 1        (Average: 121.00 Last: 83)
Model-Level  : 6.5
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 121      (Deleted: 0)
  Binary     : 25       (Ratio:  20.66%)
  Ternary    : 15       (Ratio:  12.40%)
  Conflict   : 121      (Average Length:    5.3 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 121      (Average:  1.00 Max:   1 Sum:    121)
  Executed   : 121      (Average:  1.00 Max:   1 Sum:    121 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 131      (Original: 71)
  Choice     : 7
Atoms        : 123
Bodies       : 82       (Original: 22)
  Sum        : 2
  Count      : 6        (Original: 12)
Equivalences : 53       (Atom=Atom: 7 Body=Body: 0 Other: 46)
Tight        : Yes
Variables    : 38       (Eliminated:    0 Frozen:   32)
Constraints  : 14       (Binary:   0.0% Ternary:   7.1% Other:  92.9%)
```

Cautious



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
food_amount(mixed_fruit,7) food_amount(french_fries,0) food_amount(side_salad,0) food_amount(hot_wings,0) food_am
Consequences: [0;6]
Answer: 2
food_amount(french_fries,0) food_amount(side_salad,0) food_amount(mozzarella_sticks,0)
Consequences: [0;3]
SATISFIABLE

Models       : 2
  Cautious   : yes
Consequences : 3
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 126
Conflicts    : 126      (Analyzed: 125)
Restarts     : 1        (Average: 125.00 Last: 84)
Model-Level  : 6.5
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 125      (Deleted: 0)
  Binary     : 29       (Ratio:  23.20%)
  Ternary    : 20       (Ratio:  16.00%)
  Conflict   : 125      (Average Length:    5.5 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 125      (Average:  1.00 Max:   1 Sum:    125)
  Executed   : 125      (Average:  1.00 Max:   1 Sum:    125 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 131      (Original: 71)
  Choice     : 7
Atoms        : 123
Bodies       : 82       (Original: 22)
  Sum        : 2
  Count      : 6        (Original: 12)
Equivalences : 53       (Atom=Atom: 7 Body=Body: 0 Other: 46)
Tight        : Yes
Variables    : 38       (Eliminated:    0 Frozen:   32)
Constraints  : 14       (Binary:   0.0% Ternary:   7.1% Other:  92.9%)
```

Enumerate All



```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
food_amount(mixed_fruit,7) food_amount(french_fries,0) food_amount(side_salad,0) food_amount(hot_wings,0) food_amo
Answer: 2
food_amount(mixed_fruit,1) food_amount(french_fries,0) food_amount(side_salad,0) food_amount(hot_wings,2) food_amo
SATISFIABLE

Models       : 2
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.000s

Choices      : 126
Conflicts    : 125      (Analyzed: 113)
Restarts     : 3        (Average: 37.67 Last: 4)
Model-Level  : 6.5
Problems     : 1        (Average Length: 0.00 Splits: 0)
Lemmas       : 113      (Deleted: 0)
  Binary     : 6        (Ratio:   5.31%)
  Ternary    : 13       (Ratio:  11.50%)
  Conflict   : 113      (Average Length:    6.0 Ratio: 100.00%)
  Loop       : 0        (Average Length:    0.0 Ratio:   0.00%)
  Other      : 0        (Average Length:    0.0 Ratio:   0.00%)
Backjumps    : 113      (Average:  1.00 Max:   1 Sum:    113)
  Executed   : 113      (Average:  1.00 Max:   1 Sum:    113 Ratio: 100.00%)
  Bounded    : 0        (Average:  0.00 Max:   0 Sum:      0 Ratio:   0.00%)

Rules        : 131      (Original: 71)
  Choice     : 7
Atoms        : 123
Bodies       : 82       (Original: 22)
  Sum        : 2
  Count      : 6        (Original: 12)
Equivalences : 53       (Atom=Atom: 7 Body=Body: 0 Other: 46)
Tight        : Yes
Variables    : 38       (Eliminated:    0 Frozen:   32)
Constraints  : 14       (Binary:   0.0% Ternary:   7.1% Other:  92.9%)
```

References

1. *Potassco*. [Online]. Available: https://potassco.org/. [Accessed: 07-Mar-2022].