Theory of Answer Set Programming

More about clingo Programs

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Use several directives of clingo programs

Objective

Represent definitions related to arithmetic in the language of clingo

Directives and Comments

Comment

> **Any text between the symbol % and the end of a line is a comment, disregarded by clingo.**

```
% This is comment blah, blah, blah…
```

#show

A #show directive instructs clingo to show some elements of the stable model and suppress the others

```
#show large/1.
```

/n refers to the arity of the predicate symbol

```
p. p(a).  p(a,b).
#show p/0. #show p/2.
```

#const

A #const directive allows us to use a symbolic constant as a placeholder

```
large(C) :- size(C,S1), size(uk,S2), S1 > S2.

#const c0=uk.
large(C) :- size(C,S1), size(c0,S2), S1 > S2.
```

Alternatively, the command line option `-c` can be used

- add in the command line

```
-c c0=uk
```

#include

```
% large.lp
large(C) :- size(C,S1), size(c0,S2), S1 > S2.
#show large/1.

% large_input.lp
#const c0=uk.
size(france,65; germany,83; italy,61; uk,64).
```

Multiple input files to clingo are concatenated

```
% clingo large.lp large_input.lp
```

Alternatively, we can include another file using #include

```
#include "large_input.lp".
```

Arithmetic in Clingo

Arithmetic

Complex terms can be built from constants and variables using the symbols

+ (addition) * (multiplication) ** (exponentiation)

/ (integer division) \ (remainder) | | (absolute value)

p(N,N*N+N+41) :- N=0..3.

reads

N and $N^2 + N + 41$ are in the relation p/2 if N is one of the numbers $0, \ldots, 3$.

Command: `clingo quadratic.lp`

Stable models: `p(0,41) p(1,43) p(2,47) p(3,53)`

Placeholders

```
p(N,a*N*N+b*N+c) :- N=0..n.
```

reads

N and $aN^2 + bN + c$ are in the relation p/2 if N is one of the numbers $0, \ldots, n$.

a,b,c,n are placeholders for any values that can be specified in the command

Command: `clingo -c a=1 -c b=1 -c c=41 -c n=10 quadratic.lp`

Alternatively, put in the file

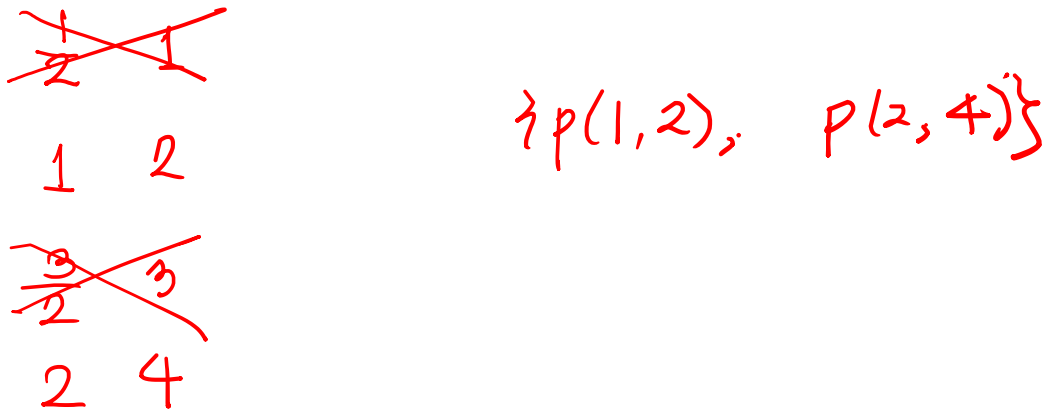
```
#const a=1. #const b=1. #const c=41. #const n=10.
```

Integer Arithmetic

The numbers that clingo knows about are integers

Q: What is the stable model of

```
p(M,N) :- N=1..4, N=2*M.
```



Intervals in Head

Intervals may be used not only in the bodies of rules but in the heads as well. For instance, a program may include the fact

```
p(0..3).
```

which has the same meaning as the set of 4 facts

```
p(0).  p(1).  p(2).  p(3).
```

This group of facts can be also abbreviated using pooling:

```
p(0; 1; 2; 3).
```

EXERCISE

Write a one-rule program that does not contain pooling and has the same stable model as

```
p(1,2; 2,4; 4,8; 8,16).
```

```
p(2**N, 2**(N+1)) :- N=0..3
```

EXERCISE

Write one-rule program that has the stable model

```
p(1,1)
p(2,1) p(2,2)
p(3,1) p(3,2) p(3,3)
p(4,1) p(4,2) p(4,3) p(4,4).
```

p(M, N) :- M=1..4, N=1..4, M >= N

Definitions Related to Arithmetic

Definition: Composite Numbers

An integer N is composite if

- it is divisible by some numbers from {2, …, N-1}

```
composite(N) :- N=1..n, I=2..N-1, N\I=0.
```

10

composite(5) :- T, I=2..4. 5\I=0

composite(4) :- T, I=2..3, 4\I=0

Definition: Fibonacci Numbers

Fibonacci Numbers

Fib(0) = 0

Fib(1) = 1

Fib(n+2) = Fib(n) + Fib(n+1) for $n \geq 0$

Fib(2) = Fib(0) + Fib(1) = 0 + 1 = 1

Fib(3) = Fib(1) + Fib(2) = 1 + 1 = 2

Fib(4) = = 3

Fib(5) = = 5

```
% fib(i,m): i-th Fibonacci number is m
fib(0,0).
fib(1,1).
fib(N+2,F1+F2) :- fib(N,F1),fib(N+1,F2), N=0..(n-2).
```

Definition: Factorials

Factorials

Fac(0) = 1

Fac(n+1) = (n+1) × Fac(n)

$0! = 1$

$(n+1)! = (n+1) \times n!$

```
% fac(i,m) : i-th factorial is m
fac(0,1).
fac(N+1,(N+1)*F) :- fac(N,F), N=1..n.
fac(F) :- fac(N,F).
#show fac/1.
```

should be N=0..n.

Lecture Wrap-Up