Practice of Answer Set Programming

Choice Rules and Constraints

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Compute stable models of programs containing choice rules and constraints by hand

Objective

Compute stable models of programs containing choice rules and constraints using clingo

Choice Rules

Choice Rule

Stable models of $p \vee \neg p$

$\emptyset$

$\{p\}$

Π

Stable models of $(p \vee \neg p) \wedge (q \vee \neg q)$

: $\emptyset$, $\{p\}$, $\{q\}$, $\{p, q\}$

$\emptyset \quad \Pi^{\emptyset} = (p \vee \top) \wedge (q \vee \top) \Leftrightarrow \top$

$\{p\} \quad \Pi^{\{p\}} = (p \vee \bot) \wedge (q \vee \top) \Leftrightarrow p$

$\{q\} \quad \Pi^{\{q\}} = \qquad \Leftrightarrow q$

$\{p, q\} \quad \Pi^{\{p,q\}} = (p \vee \bot) \wedge (q \vee \bot) \Leftrightarrow p \wedge q$

Choice Rule, cont'd

- Stable models of $(p_1 \vee \neg p_1) \wedge (p_2 \vee \neg p_2) \wedge \cdots \wedge (p_n \vee \neg p_n)$

  all subsets of $\{p_1, p_2, \cdots, p_n\}$

- We abbreviate the formula $(p_1 \vee \neg p_1) \wedge (p_2 \vee \neg p_2) \wedge \cdots \wedge (p_n \vee \neg p_n)$ as $\{p_1; \ldots; p_n\}$ and call it **choice rule**.

Choice Rules in Clingo

- Choice rule:

$$\{p(t_1, \ldots, t_k)\}.$$

- Its propositional image is the conjunction of all formulas of the form

$$p(v_1, \ldots, v_k) \vee \neg p(v_1, \ldots, v_k)$$

where v_i is a value of t_i

```
{p(1)}.
```

$p(1) \vee \neg p(1)$

S.M

$\emptyset$

$\{p(1)\}$

$\{1, 2, 3\}$

```
{p(1..3)}.
```

$(p(1) \vee \neg p(1)) \wedge (p(2) \vee \neg p(2))$

$\wedge (p(3) \vee \neg p(3))$

Choice Rules, cont'd

```
{p(a);q(b)}.
```

stands for

$$\big(p(a) \lor \neg\, p(a)\big) \land \big(q(b) \lor \neg\, q(b)\big)$$

```
% clingo choice.lp 0
Answer: 1
Answer: 2 q(b)
Answer: 3 p(a)
Answer: 4 p(a) q(b)
```

Choice Rules with Intervals and Pools

```
{p(1..3)}.
```

has the same meaning as

```
{p(1);p(2);p(3)}.
```

```
{p(a;b;c)}.
```

has the same meaning as

```
{p(a);p(b);p(c)}.
```

EXERCISE

For each of the given programs, what do you think is the number of its stable models?

(a) {p(1..3)}. {q(1..3)}.

{p(1); p(2); p(3)}. {q(1); q(2); q(3)}

64 8 x 8

(b) {p(1..3,1..3)}.

{p(1,1); p(1,2); p(1,3); p(2,1); p(2,2); p(2,3); p(3,1); p(3,2); p(3,3)}

$2^9 = 512$

Choice Rules with Cardinality Bounds

```
1 {p(1..3)} 2.
```

describes the subsets of {1,2,3} that consists of 1 or 2 elements.

```
Answer: 1 p(2)

Answer: 2 p(3)

Answer: 3 p(2) p(3)

Answer: 4 p(1)

Answer: 5 p(1) p(3)

Answer: 6 p(1) p(2)
```

EXERCISE

For each of the given programs, what do you think is the number of its stable models?

(a) `1{p(1..6)}.`

$63 = 2^6 - 1$

(b) `3{p(1..6)}3.`

$\binom{6}{3} = \frac{6 \times 5 \times 4}{3 \times 2} = 20$

EXERCISE

For each of the given rules, find a simpler rule that has the same meaning.

(a) `0{p(a)}.` $\{p(a)\}$

(b) `1{p(a)}.` $p(a)$

(c) `{p(a)} 1.` $\{p(a)\}$

Choice Rules with Variables

```
1 {p(X);q(X)} 1 :- X=1..2.
```

1 {p(1); q(1)} 1

1 {p(2); q(2)} 1

```
Answer: 1
q(1) p(2)
Answer: 2
q(1) q(2)
Answer: 3
p(1) p(2)
Answer: 4
p(1) q(2)
```

X is a global variable

Local vs. Global Variables

```
{p(I):I=1..7}.
```

- I is a local variable
- A local variable is a variable such that all its occurrences in the rule are in between { … }
- Other variables are global variables
- The rule expands into

```
{p(1); p(2); p(3); p(4); p(5); p(6); p(7)}.
```

- **Q: How many stable models are there?**

(a) 0 (b) 7 (c) 64 (d) 128

Local vs. Global Variables, cont'd

```
{p(I)} :- I=1..7.
```

- I is a global variable because it has an occurrence outside { … }

- The rule expands into

```
{p(1)}.
{p(2)}.
{p(3)}.
{p(4)}.
{p(5)}.
{p(6)}.
{p(7)}.
```

- **Q: How many stable models are there?**

  (a) 0 (b) 7 (c) 64 (d) 128

Local vs. Global Variables, cont'd

```
{q(I,J): J=1..3} :- I = 1..2.
```

{q(1,J) : J=1..3}

⇒ {q(1,1); q(1,2); q(1,3)}

Q: How many stable models are there?

(a) 6 (b) 8 (c) 12 (d) 64

{q(2,J) : J=1..3}

{q(2,1); q(2,2); q(2,3)}

The rule expands into

```
{q(1,1);q(1,2);q(1,3)}.
{q(2,1);q(2,2);q(2,3)}.
```

Constraints

- A **constraint** is a rule that has no head, e.g.,   :- p(1)

  which can be understood as $\bot \leftarrow p(1)$

- Constraints are often used with choice rules to weed out “undesirable” stable models, for which the constraint is “violated.”

```
{p(X): X=1..3}.
:- p(1).
```

∅
~~{p(1)}~~
{p(2)}
{p(3)}
~~{p(1), p(2)}~~
{p(2), p(3)}
~~{p(1), p(3)}~~
~~{p(1), p(2), p(3)}~~

```
{p(X): X=1..3}.
:- p(1).
```

not

{p(1)}
{p(1), p(2)}
{p(1), p(3)}
{p(1), p(2), p(3)}

```
{p(X): X=1..3}.
:- not p(1), not p(2).
```

{p(1)}
{p(2)}
{p(1), p(2)}
{p(2), p(3)}
{p(1), p(3)}
{p(1), p(2), p(3)}

```
{p(X): X=1..3}.
:- p(1), not p(2).
```

Cardinality Bounds vs. Constraints

- **Cardinality bounds in a choice rule can be sometimes replaced by constraints.**

- ```
  {p(a);q(b)}1.
  ```

  has the same meaning as the pair of rules

  ```
  {p(a);q(b)}.
  :- p(a), q(b).
  ```

- **Exercise:** **Find a similar transformation for the rule**

```
1{p(a);q(b)}.
```

{p(a); q(b)}.

:- not p(a), not q(b).
```

Lecture Wrap-Up