



Ayudantía 2

Modelación en Clingo

Por Trinidad Bravo y Pablo González

29 de agosto de 2025



Repaso Ayudantía Pasada



Repaso

Átomos/Proposiciones

- Definen propiedades o reglas que pueden ser **verdaderas** o **falsas**
- Un mismo predicado puede definir múltiples proposiciones, si se definen con la misma palabra pero distinta aridad.

`p.`
`p(q).`

`aprende(estudiante).`
`aprende(estudiante, profesor).`

Modelo

- Es la **solución** del programa lógico
- Es un **conjunto minimal** de átomos que satisfacen las condiciones lógicas
- Pueden existir **varios**, así como **ninguno**

Minimalidad

- Sólo son modelos aquellos conjuntos con la **mínima cantidad** posible de átomos
- De lo contrario, podrían existir infinitos modelos
- Por ejemplo, si **{p, r, q}** es un modelo, **{p, r, q, s}** no puede serlo



Modelo - Ejercicio

Considera el siguiente conjunto de reglas:

```
p.  
q :- p.  
s :- q.
```

¿Cuál es el conjunto mínimo de hechos que hace verdadero a **s**?

- A) { p }
- B) { p, q }
- C) { p, q, s }
- D) { s }



Modelo - Ejercicio

Considera el siguiente conjunto de reglas:

```
p.  
q :- p.  
s :- q.
```

¿Cuál es el conjunto mínimo de hechos que hace verdadero a **s**?

A) { p }

B) { p, q }

C) { p, q, s }

D) { s }



Repaso

Predicados

- Representan una propiedad/relación
- **Siempre** comienzan con **minúscula**

Aridad

- Corresponde al **número de términos** que reciben

Reglas

- De la forma

$$\textit{Head} \leftarrow \textit{Body}$$

- Si *Body* es verdadero, entonces **algo** en el *Head* también debe serlo
- Si no existe *Body*, lo llamamos **hecho**
- Si el *Body* tiene varios átomos, están en **conjunción**
- Si el *Head* tiene varios átomos, están en **disyunción**



Restricciones de Cardinalidad

Repaso y profundización



Restricciones de Cardinalidad

- En el contexto de la Head de una regla, estas permiten elegir **distintas combinaciones** de átomos o predicados para que aparezcan en los modelos.
- Por ejemplo, para el programa:

```
p.  
{q; r} :- p.      % Si p está en el modelo,  
                  % alguna combinación entre q y r también lo está
```

¿Qué combinaciones de átomos pueden generarse desde la restricción?



Restricciones de Cardinalidad

- En el contexto de la Head de una regla, estas permiten elegir **distintas combinaciones** de átomos o predicados para que aparezcan en los modelos.
- Por ejemplo, para el programa:

```
p.  
{q; r} :- p.      % Si p está en el modelo,  
                  % alguna combinación entre q y r también lo está
```

Las combinaciones pueden ser **{p}**, **{p,q}**, **{p,r}** y **{p,q,r}**.



Restricciones de Cardinalidad

Limitando combinaciones

- Por defecto, Clingo prueba con todas las combinaciones posibles.
- Puede limitarse el número de elementos a incluir rodeando con números el conjunto de la restricción:

```
p.  
1{q; r; s}2 :- p.      % Si p está en el modelo, alguna combinación  
                        % de 1 a 2 elementos entre q, r y s  
                        % también lo está
```

¿Cuántos modelos genera este programa?



Restricciones de Cardinalidad

Limitando combinaciones

- Por defecto, Clingo prueba con todas las combinaciones posibles.
- Puede limitarse el número de elementos a incluir rodeando con números el conjunto de la restricción:

```
p.  
1{q; r; s}2 :- p.      % Si p está en el modelo, alguna combinación  
                        % de 1 a 2 elementos entre q, r y s  
                        % también lo está
```

Ahora, las combinaciones pueden ser **{p;q}**, **{p;r}**, **{p;s}**, **{p,q;r}**, **{p;r;s}** y **{p;q;s}** (6 modelos).



Restricciones de Cardinalidad

Ejercicio

- Supongamos que tenemos un programa con N líneas del tipo:

```
p.  
1 {a_1, b_1} 2 :- p.  
1 {a_2, b_2} 2 :- p.  
(...)  
1 {a_n, b_n} 2 :- p.
```

¿Cuántos modelos genera este programa?



Restricciones de Cardinalidad

Ejercicio

- Supongamos que tenemos un programa con N líneas del tipo:

```
p.  
1 {a_1, b_1} 2 :- p.  
1 {a_2, b_2} 2 :- p.  
(...)  
1 {a_n, b_n} 2 :- p.
```

$\left\{ \begin{array}{l} \{p, a_i\} \\ \{p, b_i\} \\ \{p, a_i, b_i\} \end{array} \right\} \times N \text{ veces } *$

El programa genera **3^N modelos distintos**

* Para cada línea, tres combinaciones posibles



Restricciones de Cardinalidad

Condiciones dentro de las restricciones

- Al poner el carácter ":" es posible crear condiciones dentro de las restricciones de cardinalidad para generar reglas más complejas.

```
num(0..5).  
3{seleccionado(X) : num(X)}3.      % selecciona 3 tal que sean num  
#show seleccionado/1.              % muestra los seleccionados
```

¿Cuántos modelos hay?



Restricciones de Cardinalidad

Condiciones dentro de las restricciones

- Al poner el carácter ":" es posible crear condiciones dentro de las restricciones de cardinalidad para generar reglas más complejas.

```
num(0..5).  
3{seleccionado(X) : num(X)}3.      % selecciona 3 tal que sean num  
#show seleccionado/1.              % muestra los seleccionados
```

Hay 20 modelos posibles.



Restricciones de Cardinalidad

Condiciones dentro de las restricciones

```
num(0..5).  
3{seleccionado(X) : num(X)}3 :- seleccionado.  
#show seleccionado/1.
```

Modelo vacío.



Restricciones de Cardinalidad

Condiciones dentro de las restricciones

```
seleccionado.  
num(0..5).  
3{seleccionado(X) : num(X)}3 :- seleccionado.  
#show seleccionado/1.
```

Hay 20 modelos posibles.



Restricciones de Cardinalidad

Condiciones dentro de las restricciones

```
#const n = 5.  
  
tiempo(1..n).  
persona(pedro).  
  
1{estudia(P, T) : tiempo(T)}5 :- persona(P).
```

¿Qué simula el programa anterior?



Restricciones de Cardinalidad

Condiciones dentro de las restricciones

```
#const n = 5.  
  
tiempo(1..n).  
persona(pedro).  
  
1{estudia(P, T) : tiempo(T)}5 :- persona(P).
```

**¡Incluye todos los modelos en los que
Pedro estudia entre los tiempos 1 y 5!**



Restricciones de Cardinalidad

```
#const n = 5.
```

```
tiempo(1..n).  
persona(pedro).
```

```
1{estudia(P, T) : tiempo(T)}5 :- persona(P).
```

¡Son equivalentes!

```
persona(pedro).
```

```
1{estudia(P, 1);estudia(P, 2);estudia(P,  
3);estudia(P, 4);estudia(P, 5);}5 :-  
persona(P).
```



Restricciones de Cardinalidad

Ejercicio

Interpreta el siguiente código de Clingo:

```
linterna(0..5).  
0{encendida(L)}1 :- linterna(L).
```



Negación



Negación

La palabra **not** indica la **ausencia de un átomo en un modelo**.
En la lógica formal, not p. equivale a escribir $\neg p$

Not p.

$\neg p.$

En la ayudantía pasada a este tipo de regla lo llamamos **Filtro**. Ahora lo veremos como una forma de negación



Negación

Not p .

p **no pertenece** al modelo, o su equivalente:

$\text{:- } p$.

$p \text{ :- not } q$.

p pertenece al modelo si es que q no pertenece a este



Negación

- Podemos formar nuevos predicados:

```
desmayarse(P):- not desayunar(P), not dormir(P).
```

- Podemos evitar redundancias:

```
catedra(X,Y):- profesor(X), profesor(Y), not catedra(Y,X), X!=Y.
```

IMPORTANTE: Muchas veces al usar **not** junto con variables, deberemos agregar otro predicado que “haga segura” nuestra variable. En el último ejemplo, esto se hace al agregar **profesor(X)** y **profesor(Y)**.



Negación

- Podemos realizar restricciones más complejas

```
:- notregar_arbol(A), notabonar_arbol(A), sano_arbol(A).
```

Esto se traduce a que si hay un árbol sano, este debe haber sido regado y abonado. En otras palabras, no puede existir un modelo en donde un árbol esté sano, y no se haya regado y abonado.



Negación

```
persona(trini).  
persona(pablo).  
vegetarian(trini).  
come_empanada_pino(P):- persona(P), not vegetarian(P).
```

¿Qué modelo resulta?



Modelación



Modelación

Algunos tips

- Dado que Clingo es un lenguaje declarativo, pensar en el problema **resuelto**, no en cómo resolverlo.
- Probar que las reglas funcionan **individualmente** sirve para entender qué funciona y qué no.
- Soltar la mano, especialmente pasando predicados lógicos a Clingo.
- ¡Ejercitar! Hay muchos ejemplos básicos, medios y avanzados.



Quiz en Menti



Modelación

Ejercicio

Se tienen 8 jugadores de tenis que quieren jugar un campeonato de dobles. Modela un programa que indique los posibles equipos.



Modelación

Ejercicio

Se tiene un curso de 20 alumnos que deben repartirse en 6 grupos para el proyecto final. Los grupos deben ser de 3 o 4 personas. Modela un programa que indique los posibles grupos



Ayudantía 2

Modelación en Clingo

Por Trinidad Bravo y Pablo
González

29 de agosto de 2025