3 PREGUNTA 3

PARTE 1

En este problema se debe demostrar que dados dos programas tales que $\Pi \subseteq \Pi'$ y M modelo de Π entonces existe un modelo M' de Π tal que $M \subseteq M'$. Para esto, demostraremos en primer lugar la existencia, y después la contención.

Existencia: Sea I el conjunto potencia de los átomos del programa Π notemos que como $\Pi \subseteq \Pi'$, entonces $I \subseteq I'$ con I' el conjunto de átomos de Π'. Luego, es posible definir Π' como:

$$\Pi = \{I', r' : body'(r') \cap I' \neq \emptyset\}$$

con r' = todas las reglas de la forma $Head \leftarrow Body$ con $|Head| = 1$. Por lo tanto, es posible definir un conjunto $M' \subseteq I'$ tal que $body'(r') \subseteq M'$ junto con $Body'(r') \cap M' \neq \emptyset$ lo que implica que $head'(r') \cap M' \neq \emptyset$ lo que es precisamente la definición de modelo, lo que implica que M' es modelo de Π'.

Contención: Para esta demostraremos por contradicción que $M \subseteq M'$. Supongamos que la contención que estamos buscando no se cumple, es decir, $M' \subseteq M$. Con M modelo de Π y M' modelo de Π'.

Como $M' \subseteq M$, $Head'(r') \cap M \neq \emptyset$ y $Body'(r') \subseteq M$ con $r' \in \Pi$ para todo átomo de Π'. Luego, M es modelo de Π', pero esto es una contradicción debido a que viola el principio de minimalidad de modelo ya que ya existe un M' minimal, luego se debe cumplir que $M' \subseteq M$.

PARTE 2

En esta demostración haremos induccion en base al numero de reglas. Sea Π un programa sin negacion y con reglas con la aridad de *head* menor o igual a 1.

Si el numero de reglas (n) es igual a cero notemos que el conjunto de reglas $\mathcal{R}$ del programa es vacio, con ello, $\{\emptyset\} \in \Pi$ y $\{\emptyset\} \subseteq M$, pero notemos que $head \cap M = \{\emptyset\} \cap M = \emptyset$. Luego, el pograma no tiene conjunto de respuesta distinto al vacío y se cumple con lo pedido.

Ahora asumamos, que el programa tiene a lo más un modelo para las primeras n reglas con $|Head| = 0$ o $|Head| = 1$.

Si le agregamos una regla al programa, notemos que esta puede ser:

$$Head \leftarrow \{\} \tag{1}$$

$$Head \leftarrow Body_1 \wedge ... \wedge Body_k \tag{2}$$

$$Head \leftarrow Body_1 \vee ... \vee Body_k \tag{3}$$

con $\{i..k\}$ una enumeración de algún conjunto *Body*. Si la regla que se agrega es como (1), entonces es trivial que exista solo un modelo minimal, el cual es el que contiene al mismo *Head*. Si la regla que se agrega es como (2), si los átomos del body, son distintos a los ya existentes en las n reglas, por hipótesis inductiva, ya existirá un solo modelo minimal que siempre cumpla

Head. Si los átomos, son subconjunto de una regla ya existente con las mismas características, entonces existe un modelo minimal y este será el compuesto por los átomos de las reglas recién agregadas.

Si la regla que se agrega es como (3) y el programa esta compuesto por reglas de conjunción y disyunción, entonces por hipótesis inductiva el programa contiene un solo modelo minimal, compuesto por los átomos cuya aridad de regla de conjunción es menor.

Si el programa está compuesto solo por reglas de conjunción, y si los átomos de (3) son subconjunto de los átomos del programa, entonces, existe un solo modelo minimal y es el que está compuesto por estos átomos recién agregados, en caso contrario, por hipótesis inductiva, existe un modelo minimal.

PARTE 3

El algoritmo que revuelve los programas con ngación y retorna los modelos es el siguiente:

1. $M := \varnothing$
2. $N := \varnothing$
3. Seleccionar una regla de Π tal que los átomos sin negación del *Body*, pertenezcan a M y los atómos con negacion del *Body*, pertenezcan a N.
4. Si *Head* no está contenido en M entonces hacer $M := Head \cup M$
5. En caso de encontrar una regla, volver al paso 3, de lo contrario, continuar al siguiente paso.
6. Si todas las reglas con el mismo *Head* satisfacen que los átomos sin negación del $Body \cap N \neq \varnothing$ o los átomos con negación del $Body \cap M \neq \varnothing$, y si *Head* no está contenido en N entonces hacer $N := Head \cup N$
7. En caso de encontrar una regla, volver al paso 6, de lo contrario continuar al paso 8.
8. Escoger un $\beta = \text{new\_atom}$ que no esté en M ni en N. Si este átomo existe, ir al paso 9, de lo contrario, retornar la unión de todos los M de las recursiones hechas por $(M \cup \beta, N)$ y $(M, N \cup \beta)$.
9. Volver al paso 3 con $(M \cup \beta, N)$ y con $(M, N \cup \beta)$ por separado.

Este algoritmo, queda mejor explicado con el siguiente ejemplo:

Listing 3.1: Ejemplo problema ASP

```
1 r :- p, not s.
2 s :- q, not r.
3 p.
4 q.
```

Dado este problema el algoritmo hara lo siguiente:

- Comenzamos con los primeros elementos instanciados, $M := \varnothing$, $N := \varnothing$.
- Dado el paso 3, seleccionamos la regla 3, p. Y se continua al paso 4.

- Como $M = \varnothing$ hacemos $M := Head \cup M$. Y volvemos al paso 3. ($M = \{p\}$ $N = \{\varnothing\}$)
- Ahora seleccionamos la regla 4 que cumple con las condiciones dadas por el paso 3.
- como $M = \{p\}$ hacemos $M := Head \cup M$. Y volvemos al paso 3. ($M = \{p, q\}$ $N = \{\varnothing\}$)
- Como en el paso 3 no se encuentras más reglas, se va al paso 6.
- Notemos que en el paso 6 no existe ninguna regla con el mismo *Head* que satisfaga las condiciones necesarias, por lo mismo se continua al paso 8.
- Escogemos r que no está en ninguno de los conjuntos N ni M, y volvemos al paso 3 con ($M \cup r$, N) y con (M, $N \cup r$) por separado.

($M \cup r$, N):

- Ahora se vuelve al paso 3, pero no existen reglas que cumplan con la condición, con ello se continua al paso 6.
- Notemos que la regla 2, cumple con que los atomos con negacion del *Body* se encuentra en M.
- Como $N = \varnothing$ hacemos $N := Head \cup N$. Y volvemos al paso 6. ($M = \{p, q, r\}$ $N = \{ s\}$).
- Como no se encuentran mas reglas que cumplan con las condiciones, se vuelve al paso 8, pero no existen más atomos que no sean comunes, con lo que se debe retornar, ($M \cup \beta$, N) $\cup$ (M, $N \cup \beta$).

(M, $N \cup r$):

- De forma análoga a lo anterior se llega a que $M = \{p, q, s\}, N = \{r\}$.

De esta forma se consigue que los modelos del programa instanciado son $M_1 = \{p, q, s\}$ y $M_2 = \{p, q, r\}$.

Finalmente, cabe recalcar que este algoritmo es correcto debido a que busca todos los posibles modelos que pueden resultar de una posible aparición de átomos en negación. En donde se busca de manera recursiva todos los átomos que pueden pertenecer a un modelo u otro. Finalmente, cuando no se encuentran más átomos, comunes entre modelo, se retorna la unión de todos los conjuntos encontrados en todas las recursiones del algoritmo.