UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Tarea Examen 1: Lógica Proposicional

ALUMNO

Carlos Emilio Castañon Maldonado

PROFESOR

Javier Enríquez Mendoza

AYUDANTES

Kevin Axel Prestegui Ramos
Karla Denia Salas Jiménez
Ramón Arenas Ayala
Oscar Fernando Millán Pimentel

Lógica Computacional

Tarea Examen 1: Lógica Proposicional

1 (**1 pt.**) Demuestre que las siguientes fórmulas son equivalentes utilizando interpretaciones
$\varphi_1 \leftrightarrow \varphi_2 \qquad (\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$.

$\varphi_1 \leftrightarrow \varphi_2$
Sea I_1 de $\varphi_1 \leftrightarrow \varphi_2$ donde:
$\varphi_1 = 1$ y $\varphi_2 = 1$
Tendremos:
$1 \leftrightarrow 1 = 1$
Sea I_2 de $\varphi_1 \leftrightarrow \varphi_2$ donde:
$\varphi_1 = 0$ y $\varphi_2 = 0$
Tendremos:
$0 \leftrightarrow 0 = 1$
Sea I_3 de $\varphi_1 \leftrightarrow \varphi_2$ donde:
$\varphi_1 = 0$ y $\varphi_2 = 1$
Tendremos:
$0 \leftrightarrow 1 = 0$
Sea I_4 de $\varphi_1 \leftrightarrow \varphi_2$ donde:
$\varphi_1 = 1$ y $\varphi_2 = 0$
Tendremos:
$1 \leftrightarrow 0 = 0$

$(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$
Sea I_1 de $(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$ donde:
$\varphi_1 = 1$ y $\varphi_2 = 1$
Tendremos:
$(1 \wedge 1) \vee (\neg 1 \wedge \neg 1) = (1) \vee (0 \wedge 0) = 1$
Sea I_2 de $(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$ donde:
$\varphi_1 = 0$ y $\varphi_2 = 0$
Tendremos:
$(0 \wedge 0) \vee (\neg 0 \wedge \neg 0) = (0) \vee (1 \wedge 1) = 1$
Sea I_3 de $(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$ donde:
$\varphi_1 = 0$ y $\varphi_2 = 1$
Tendremos:
$(0 \wedge 1) \vee (\neg 0 \wedge \neg 1) = (0) \vee (1 \wedge 0) = 0$
Sea I_4 de $(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$ donde:
$\varphi_1 = 1$ y $\varphi_2 = 0$
Tendremos:
$(1 \wedge 0) \vee (\neg 1 \wedge \neg 0) = (0) \vee (0 \wedge 1) = 0$

$$\therefore \varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \neg\varphi_2)$$

2 (**2 pts.**) Decida si el siguiente argumento es correcto utilizando resolución binaria.
Indique el significado de las variables proposicionales usadas y muestre a detalle las transformaciones a la forma clausular.

Petronilo o Quelónico cometieron un crimen.
Petronilo estuvo fuera de la tienda donde se cometió el crimen.
Si Petronilo estaba fuera de la tienda entonces no estaba en la escena del crimen.
Si Petronilo no estaba en la escena del crimen entonces él no pudo haber cometido el crimen.
Por lo tanto Quelónico debió haber cometido el crimen.

Como primer paso, pasamos los enunciados a su forma *prop*
- Petronilo o Quelónico cometieron un crimen
$a =$ Petronilo cometió un crimen
$b =$ Quelónico cometió un crimen
$\therefore a \vee b$

- Petronilo estuvo fuera de la tienda donde se cometió el crimen.
$c =$ Petronilo estuvo fuera de la tienda donde se cometió el crimen.
$\therefore c$

- Si Petronilo estaba fuera de la tienda entonces no estaba en la escena del crimen.
$c =$ Petronilo estaba fuera de la tienda
$\rightarrow =$ entonces
$\neg =$ no
$d =$ estaba en la escena del crimen
$\therefore c \rightarrow \neg d$

- Si Petronilo no estaba en la escena del crimen entonces él no pudo haber cometido el crimen.
$\neg d =$ Petronilo no estaba en la escena del crimen
$\rightarrow =$ entonces
$\neg =$ no

$a =$ (Petronilo) pudo haber cometido el crimen
$\therefore \neg d \rightarrow \neg a$

- Por lo tanto Quelónico debió haber cometido el crimen.
$b =$ Quelónico debió haber cometido el crimen
$\therefore b$

Como podemos observar tenemos un argumento lógico de la forma:

$$\begin{array}{l} \text{a} \vee b \\ \text{c} \\ \text{c} \rightarrow \neg d \\ \neg d \rightarrow \neg a \\ \hline \therefore\ b \end{array}$$

Ahora procedemos a pasar a su forma **FNN** cada una de las anteriores:

	$\text{a} \vee b$	c	$\text{c} \rightarrow \neg d$	$\neg d \rightarrow \neg a$
FNN	$\text{a} \vee b$	c	$\neg c \vee \neg d$	$\text{d} \vee \neg a$

Ahora procedemos a pasar a su forma **FNC** cada una de las anteriores:

	$\text{a} \vee b$	c	$\neg c \vee \neg d$	$\text{d} \vee \neg a$
FNC	$\text{a} \vee b$	c	$\neg c \vee \neg d$	$\text{d} \vee \neg a$

Ahora procedemos a aplicar **resolución binaria**:

$$[[a \vee\ b], [c], [\neg c \vee \neg d], [d \vee \neg a]]$$

$$[a, b, c, \neg c, \neg d, d, \neg a]$$

$$[a, b, \neg d, d, \neg a]$$

$$[a, b, \neg a]$$

$$[b]$$

$\therefore$ El argumento lógico descrito es correcto, además de que como hemos llegado a la conclusión de antes, podemos decir sin lugar a dudas que Quelónico fue quien cometió el crimen.

3 (**1.5 pts.**)La función traducción negativa de Gödel-Gentzen se define como sigue: $gg(\varphi)$ es la fórmula obtenida a partir de φ al colocar dos negaciones enfrente de cada variable proposicional y enfrente de cada disyunción. Por ejemplo

$$\begin{aligned} gg\Big(p \wedge q \to r \vee \neg s\Big) &= \neg\neg p \wedge \neg\neg q \to \neg\neg(\neg\neg r \vee \neg\neg\neg s) \\ gg\Big(\neg(\neg q \wedge (s \vee \bot))\Big) &= \neg(\neg\neg\neg q \wedge \neg\neg(\neg\neg s \vee \bot)) \end{aligned}$$

(a) Defina gg recursivamente y muestre que su definición es correcta aplicando su definición en el segundo ejemplo anterior.

Procedemos a definir gg recursivamente de la siguiente manera usando `Haskell`.

```haskell
data Prop = Var String | Cons Bool| Not Prop
      | And Prop Prop | Or Prop Prop
      | Impl Prop Prop | Syss Prop Prop
      deriving (Eq,Show)

-- Funcion de traduccion negativa gg
gg :: Prop -> Prop
gg (Var p) = Not (Not (Var p))
gg (Cons b) = Cons b
gg (Not p) = Not (gg p)
gg (And p q) = And (gg p) (gg q)
gg (Or p q) = Not (Not (Or (gg p) (gg q)))
gg (Impl p q) = Impl (gg p) (gg q)
gg (Syss p q) = Syss (gg p) (gg q)
```

Uso en el segundo ejemplo:

$$gg\Big(\neg(\neg q \wedge (s \vee \bot))\Big)$$

$$\Big(\neg gg((\neg q \wedge (s \vee \bot)))\Big) \quad \text{Aquí se aplica el caso de } gg(Not\ p)$$

$$\Big(\neg(\neg gg(q) \wedge gg((s \vee \bot)))\Big) \quad \text{Aquí se aplica el caso de } gg(And\ p\ q)$$

$$\Big(\neg(\neg\neg\neg q \wedge \neg\neg(gg(s) \vee gg(\bot)))\Big) \quad \text{Aquí se aplican los casos de } gg(Not\ p) \text{ y } gg(Or\ p\ q)$$

$$\Big(\neg(\neg\neg\neg q \wedge \neg\neg(\neg\neg s \vee \bot))\Big) \quad \text{Aquí se aplican los casos de } gg(Var\ p) \text{ y } gg(Cons\ b)$$

$$\therefore \neg(\neg\neg\neg q \wedge \neg\neg(\neg\neg s \vee \bot)) = gg\Big(\neg(\neg q \wedge (s \vee \bot))\Big)$$

(b) Demuestre que para cualquier fórmula φ, se cumple que $\varphi \equiv gg(\varphi)$.
Para esto haremos uso de la **inducción estructural**
Caso Base:

$$\begin{aligned} & gg(\varphi) \\ & \neg\neg\varphi \\ & \varphi && \text{Por definición de doble negación} \\ \therefore\ & \varphi \equiv gg(\varphi) \end{aligned}$$

φ	$\neg\varphi$	$\neg(\neg\varphi)$
1	0	1
0	1	0

Hipótesis Inductiva:
Supondremos que la equivalencia $\varphi \equiv gg(\varphi)$ se cumple para una fórmula ϕ
Paso Inductivo:
Como queremos demostrar que la equivalencia $\varphi \equiv gg(\varphi)$ se cumple para cualquier formula, entonces queremos demostrar que se cumple para la negación, conjunción, disyunción, implicación y equivalencia lógica, por lo que procedemos a demostrarlo:
Sean ϕ y ψ fórmulas pertenecientes a prop, tendremos que queremos demostrar las siguientes:

$\neg\phi$ $\quad$ $\neg\psi$ $\quad$ $\phi \wedge \psi$ $\quad$ $\phi \vee \psi$ $\quad$ $\phi \to \psi$ $\quad$ $\phi \leftrightarrow \psi$

Para la formula $\neg\phi$ veamos lo siguiente:

$$\begin{aligned} & gg(\neg\phi) \\ & \neg(\neg\neg\phi) && \text{Por } gg(Not\ p) \\ & \neg(\phi) && \text{Por } H.I \text{ y definición de doble negación} \\ \therefore\ & \neg\phi \equiv gg(\neg\phi) \end{aligned}$$

Para la formula $\neg\psi$ veamos lo siguiente:

$$\begin{aligned} & gg(\neg\psi) \\ & \neg(\neg\neg\psi) && \text{Por } gg(Not\ p) \\ & \neg(\psi) && \text{Por } H.I \text{ y definición de doble negación} \\ \therefore\ & \neg\psi \equiv gg(\neg\psi) \end{aligned}$$

Para la formula $\phi \wedge \psi$ veamos lo siguiente:

$$\begin{aligned} & gg(\phi \wedge \psi) \\ & (gg(\phi) \wedge gg(\psi)) && \text{Por } gg(And\ p\ q) \\ & (gg(\phi) \wedge gg(\psi)) && \text{Por } H.I \text{ y definición de doble negación} \\ & (\neg\neg\phi \wedge \neg\neg\psi) && \text{Por } gg(q) \\ & (\phi \wedge \psi) && \text{Por } H.I \text{ y definición de doble negación} \\ \therefore\ & \phi \wedge \psi \equiv gg(\phi \wedge \psi) \end{aligned}$$

Para la formula $\phi \vee \psi$ veamos lo siguiente:

$$\begin{aligned} & gg(\phi \vee \psi) \\ & \neg\neg(gg(\phi) \vee gg(\psi)) && \text{Por } gg(Or\ p\ q) \\ & (gg(\phi) \vee gg(\psi)) && \text{Por } H.I \text{ y definición de doble negación} \\ & (\neg\neg\phi \vee \neg\neg\psi) && \text{Por } gg(p) \\ & (\phi \vee \psi) && \text{Por } H.I \text{ y definición de doble negación} \\ \therefore\ & \phi \vee \psi \equiv gg(\phi \vee \psi) \end{aligned}$$

Para la formula $\phi \to \psi$ veamos lo siguiente:

$$\begin{array}{rl}
gg(\phi \to \psi) & \\
(gg(\phi) \to gg(\psi)) & \text{Por } gg(Impl\ p\ q) \\
(\neg\neg\phi \to \neg\neg\psi) & \text{Por } gg(p) \\
(\phi \to \psi) & \text{Por } H.I \text{ y definición de doble negación} \\
\therefore \phi \to \psi \equiv gg(\phi \to \psi) &
\end{array}$$

Para la formula $\phi \leftrightarrow \psi$ veamos lo siguiente:

$$\begin{array}{rl}
gg(\phi \leftrightarrow \psi) & \\
(gg(\phi) \leftrightarrow gg(\psi)) & \text{Por } gg(Syss\ p\ q) \\
(\neg\neg\phi \leftrightarrow \neg\neg\psi) & \text{Por } gg(p) \\
(\phi \leftrightarrow \psi) & \text{Por } H.I \text{ y definición de doble negación} \\
\therefore \phi \leftrightarrow \psi \equiv gg(\phi \leftrightarrow \psi) &
\end{array}$$

$\therefore$ Por el principio de inducción queda demostrado que para cualquier formula φ se cumple que $\varphi \equiv gg(\varphi)$.

4 (**2pts.**) Decidir si el siguiente argumento es correcto utilizando el algoritmo DPLL. Debes mostrar los pasos detalladamente y en caso de que exista, mostrar y verificar el modelo obtenido.

$$\{p \vee r,\ q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ r,\ p \vee \neg q \vee \neg r \vee \neg s\} \models p$$
$$\{p \vee r,\ q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ r,\ p \vee \neg q \vee \neg r \vee \neg s,\ \neg p\}$$

$$\begin{array}{rll}
\cdot \models & p \vee r,\ q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ r,\ p \vee \neg q \vee \neg r \vee \neg s,\ \neg p & \text{Unit } \neg p \\
\neg p \models & p \vee r,\ q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ r,\ p \vee \neg q \vee \neg r \vee \neg s & \text{Unit } r \\
\neg p, r \models & p \vee r,\ q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ p \vee \neg q \vee \neg r \vee \neg s & \text{Red } \neg p \\
\neg p, r \models & r,\ q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ \neg q \vee \neg r \vee \neg s & \text{Elim } r \\
\neg p, r \models & q \vee \neg r \vee s,\ \neg q \vee \neg r \vee s,\ \neg q \vee \neg r \vee \neg s & \text{Red } r \\
\neg p, r \models & q \vee s,\ \neg q \vee s,\ \neg q \vee \neg s & \text{Split } q \\
\neg p, r, q \models & q \vee s,\ \neg q \vee s,\ \neg q \vee \neg s & \\
\neg p, r, \neg q \models & q \vee s,\ \neg q \vee s,\ \neg q \vee \neg s & \textbf{Escogemos esta} \\
\neg p, r, \neg q \models & q \vee s,\ \neg q \vee s,\ \neg q \vee \neg s & \text{Red } \neg q \\
\neg p, r, \neg q \models & s,\ \neg q \vee s,\ \neg q \vee \neg s & \text{Unit } s \\
\neg p, r, \neg q, s \models & \neg q \vee s,\ \neg q \vee \neg s & \text{Elim } \neg q \\
\neg p, r, \neg q, s \models & s, \neg s & \text{Red } s \\
\neg p, r, \neg q, s \models & s & \text{Elim } s \\
\neg p, r, \neg q, s \models & \varnothing & \text{Succes}
\end{array}$$

$p\ =\ 0 \quad r\ =\ 1 \quad q\ =\ 0 \quad s\ =\ 1$

$\{p{\vee}r,\ q{\vee}\neg r{\vee}s,\ \neg q{\vee}\neg r{\vee}s,\ r,\ p{\vee}\neg q{\vee}\neg r{\vee}\neg s,\ \neg p\}$ $\quad \therefore$ El conjunto es satisfacible

$\{0{\vee}1,\ 0{\vee}\neg(1){\vee}1,\ \neg(0){\vee}\neg(1){\vee}1,\ 1,\ 0{\vee}\neg(0){\vee}\neg(1){\vee}\neg(1),\ \neg(0)\}$

$\{1,\ 0 \vee 0 \vee 1,\ 1 \vee 0 \vee 1,\ 1,\ 0 \vee 1 \vee 0 \vee 0,\ 1\}$

$\{1, 1, 1, 1, 1, 1\}$ $\quad \therefore$ El argumento es incorrecto

5 (**1.5 pts.**) *El piojo y la pulga se van a casar y han invitado a las nupcias a la mosca, el escarabajo y el abejorro. Estos finos invitados deben sentarse juntos en el banquete pero debido a viejos pleitos y supersticiones sabemos que:*

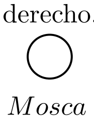
- *La mosca no quiere sentarse junto al abejorro.*
- *La mosca no quiere sentarse en la silla de la izquierda.*
- *El escarabajo no quiere sentarse a la derecha del abejorro.*

Represente esta información como una instancia del problema SAT, es decir mediante formas normales conjuntivas. Debe incluir la información implícita necesaria como que una silla sólo puede ocuparla un bicho, que cada bicho debe estar sentado y que ninguno ocupa más de una silla.

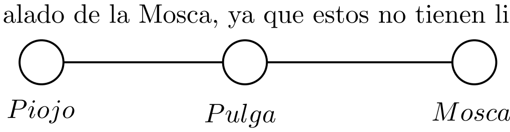
Este pareciera ser un problema trivial pero no lo es, es por eso que he decidido retomar mis conocimientos previos de **(Probabilidad) Combinatoria** y **Teoría de Gráficas** para poder modelar este problema en una instancia de SAT.
Primero analizamos el problema y nos damos cuenta de los siguientes factores importantes:
Son 5 los integrantes de la boda, el piojo, la pulga, la mosca, el escarabajo y el abejorro.
El escarabajo no quiere sentarse a la derecha del abejorro (Lo cual implica que pueden sentarse juntos cumpliendo con esta condición)
La mosca no quiere sentarse junto al abejorro.
La mosca no quiere sentarse en la silla de la izquierda.
Además de que una silla sólo puede ocuparla un bicho, que cada bicho debe estar sentado y que ninguno ocupa más de una silla.

Ahora, si tratamos de sacar el numero de combinaciones posibles para acomodar a estos 5 bichos en estos 5 lugares nos daríamos cuenta que eso es 5! lo cual es $5! = 5 * 4 * 3 * 2 * 1 = 120$ (Esto sin considerar nuestras limitaciones) como este es un numero muy grande como para representarlo en predicados, procedemos a modelar una gráfica que cumpla con las características antes dichas.

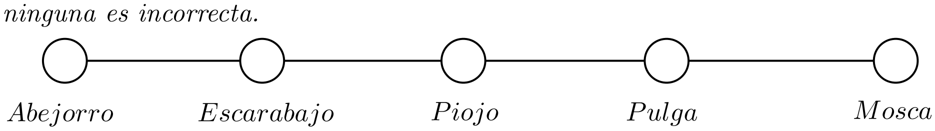
Dicho esto podemos modelar una gráfica donde sea G la mesa donde estarán, V los vértices los cuales estarán dados por **cada bicho sentado en la mesa en una silla única y no repetible** y E las cuales serán dadas si un vértice tiene conexión con otro (que un bicho este sentado junto a otro), dicho esto empezamos a graficar nuestra gráfica G con nuestro bicho con mas limitaciones, la mosca a la cual sentaremos en el extremo derecho.



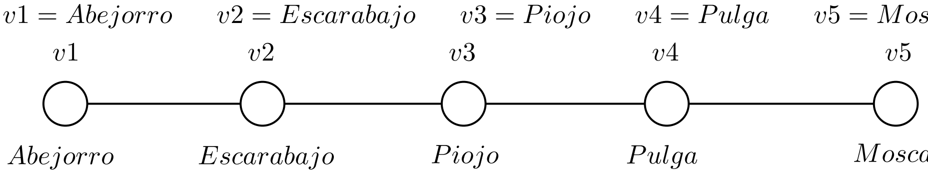
Ahora, dentro de los 4 integrantes de la boda que nos quedan escogemos a los novios para que se sienten alado de la Mosca, ya que estos no tienen limitaciones.



Ahora solo nos queda por asignarle lugar al abejorro y al escarabajo
Nota: *El predicado el escarabajo no quiere sentarse a la derecha del abejorro no implica una posición específica para el escarabajo y el abejorro en relación a una línea horizontal o vertical. En otras palabras, la posición relativa entre el escarabajo y el abejorro dependerá de la perspectiva desde la cual se está viendo la situación. Por lo tanto representar esto como escarabajo-abejorro o como abejorro-escarabajo ,son formas posibles y ninguna es incorrecta.*



Notemos que la gráfica G tendrá estas propiedades , $V = \{v1, v2, v3, v4, v5\}$ y $E = \{v1v2, v2v3, v3v4, v4v5\}$
En donde:
$v1 = Abejorro$ $v2 = Escarabajo$ $v3 = Piojo$ $v4 = Pulga$ $v5 = Mosca$



Traducción a prop

Para poder lograr esta traducción deberemos tener los siguientes predicados:
$p1 =$ Abejorro esta sentado en la mesa
$p2 =$ Escarabajo esta sentado en la mesa
$p3 =$ Piojo esta sentado en la mesa
$p4 =$ Pulga esta sentado en la mesa
$p5 =$ Mosca esta sentado en la mesa

$q1 =$ Abejorro esta sentado a su izquierda
$q2 =$ Escarabajo esta sentado a su izquierda
$q3 =$ Piojo esta sentado a su izquierda
$q4 =$ Pulga esta sentado a su izquierda
$q5 =$ Mosca esta sentado a su izquierda

$q`1 =$ Abejorro esta sentado a su derecha
$q`2 =$ Escarabajo esta sentado a su derecha
$q`3 =$ Piojo esta sentado a su derecha
$q`4 =$ Pulga esta sentado a su derecha
$q`5 =$ Mosca esta sentado a su derecha

$s1 =$ Abejorro esta sentado en la silla de la izquierda
$s2 =$ Escarabajo esta sentado en la silla de la izquierda
$s3 =$ Piojo esta sentado en la silla de la izquierda
$s4 =$ Pulga esta sentado en la silla de la izquierda
$s5 =$ Mosca esta sentado en la silla de la izquierda

$s`1 =$ Abejorro esta sentado en la silla de la derecha
$s`2 =$ Escarabajo esta sentado en la silla de la derecha
$s`3 =$ Piojo esta sentado en la silla de la derecha
$s`4 =$ Pulga esta sentado en la silla de la derecha
$s`5 =$ Mosca esta sentado en la silla de la derecha

Comenzamos con que el escarabajo no quiere sentarse a la derecha del abejorro, a lo que tendremos por comodidad:
$p2 \leftrightarrow \neg s`1$

Ahora con el enunciado, la mosca no quiere sentarse junto al abejorro, tendremos:
$p5 \leftrightarrow \neg(q1 \wedge q`1)$

Ahora, con nuestra ultima restricción/enunciado, la mosca no quiere sentarse en la silla de la izquierda, tendremos:
$p5 \leftrightarrow s5$

Ahora retomando la gráfica de la hoja anterior (y recordando el concepto de vecindad en teoría de gráficas) tendremos que los vértices anteriores estaban definidos como:

$v1 \in G \leftrightarrow N(v1) = \{v2\}$
$v2 \in G \leftrightarrow N(v2) = \{v1, v3\}$
$v3 \in G \leftrightarrow N(v3) = \{v2, v4\}$
$v4 \in G \leftrightarrow N(v4) = \{v3, v5\}$
$v5 \in G \leftrightarrow N(v5) = \{v4\}$

En donde $N(u)$ es la forma de decir quienes son los vecinos (vértices contiguos a un vértice) de un vértice u Sin embargo esta lista solo funciona para la gráfica que hemos definido , no para todas las posibles combinaciones de bichos sentados siguiendo las reglas, es por eso que aquí es donde juntaremos nuestras reglas para saber las combinaciones de lo vértices anteriormente dados y de esa forma dar con sus equivalentes proposicionales.

Nuestros nuevos vértices con sus posibles vecinos quedarían como:

$v1 \in G \leftrightarrow N(v1) = \{v2\} \cup \{v2, v3\} \cup \{v2, v4\} \cup \{v3, v2\} \cup \{v2, v4\} \cup \{v4, v2\} \cup \{v4, v3\}$
$v2 \in G \leftrightarrow N(v2) = \{v1, v3\} \cup \{v3, v4\} \cup \{v3, v5\} \cup \{v4, v3\} \cup \{v4, v5\}$
$v3 \in G \leftrightarrow N(v3) = \{v2, v4\} \cup \{v1, v2\} \cup \{v1, v4\} \cup \{v1, v5\} \cup \{v2, v1\} \cup \{v2, v4\} \cup \{v2, v5\} \cup \{v4, v1\} \cup \{v4, v2\} \cup \{v4, v5\}$
$v4 \in G \leftrightarrow N(v4) = \{v3, v5\} \cup \{v1, v2\} \cup \{v1, v3\} \cup \{v1, v5\} \cup \{v2, v1\} \cup \{v2, v3\} \cup \{v2, v5\} \cup \{v3, v1\} \cup$

$\{v3, v2\} \cup \{v3, v5\}$
$v5 \in G \leftrightarrow N(v5) = \{v4\} \cup \{v2\} \cup \{v3\}$

Nota: En v1 No se incluye v5 por que ese es la mosca, en v2 no se incluye el caso de el abejorro por que ese ya estaba contemplado mas adelante con $p2 \leftrightarrow \neg s`1$, en v5 solo consideramos a todos los vértices (excepto el de el abejorro) para que sean sus vecinos derechos para que se cumpla el que la mosca no este en una silla de la izquierda y por ende solo este en la silla de la derecha

Notese que en cada una de las combinaciones anteriores no se considera al $v5$ (la mosca) en alguna otra posición que no sea sentada a la derecha, para que entonces se cumpla lo de que no se quiere sentar en una silla izquierda, además de que para cada uno de los vértices es importante su orden ya que por ejemplo en los casos de $v1$ (el abejorro) exponemos dos posibles vecinos a este, $\{v2, v4\}$ y $\{v4, v2\}$, en el primero estamos diciendo que el escarabajo esta sentado a su izquierda y la pulga a su derecha, en cambio en el otro caso estamos intercambiando los lugares de ambos.

Una vez dicho lo anterior procedemos a traducir cada uno a su forma proposicional

Abejorro
$p1 \leftrightarrow ((q`2) \vee (q2 \wedge q`3) \vee (q2 \wedge q`4) \vee (q3 \wedge q`2) \vee (q4 \wedge q`2) \vee (q4 \wedge q`3))$

Escarabajo
$p2 \leftrightarrow ((q1 \wedge q`3) \vee (q3 \wedge q`4) \vee (q3 \wedge q`5) \vee (q4 \wedge q`3) \vee (q4 \wedge q`5))$

Piojo
$p3 \leftrightarrow ((q2\wedge q`4)\vee(q1\wedge q`2)\vee(q1\wedge q`4)\vee(q1\wedge q`5)\vee(q2\wedge q`1)\vee(q2\wedge q`4)\vee(q2\wedge q`5)\vee(q4\wedge q`1)\vee(q4\wedge q`2)\vee(q4\wedge q`5))$

Pulga
$p4 \leftrightarrow ((q3\wedge q`5)\vee(q1\wedge q`2)\vee(q1\wedge q`3)\vee(q1\wedge q`5)\vee(q2\wedge q`1)\vee(q2\wedge q`3)\vee(q2\wedge q`5)\vee(q3\wedge q`1)\vee(q3\wedge q`2)\vee(q3\wedge q`5))$

Mosca
$p5 \leftrightarrow ((q4) \vee (q2) \vee (q3)) \wedge p5 \leftrightarrow s5$

Forma Normal Negativa

Abejorro
$((\neg p1 \vee (((((q`2 \vee (q2 \wedge q`3)) \vee (q2 \wedge q`4)) \vee (q3 \wedge q`2)) \vee (q4 \wedge q`2)) \vee (q4 \wedge q`3))) \wedge ((((((\neg q`2 \wedge (\neg q2 \vee \neg q`3)) \wedge (\neg q2 \vee \neg q`4)) \wedge (\neg q3 \vee \neg q`2)) \wedge (\neg q4 \vee \neg q`2)) \wedge (\neg q4 \vee \neg q`3)) \vee p1))$

Escarabajo
$((\neg p2 \vee (((((q1 \wedge q`3) \vee (q3 \wedge q`4)) \vee (q3 \wedge q`5)) \vee (q4 \wedge q`3)) \vee (q4 \wedge q`5))) \wedge ((((((\neg q1 \vee \neg q`3) \wedge (\neg q3 \vee \neg q`4)) \wedge (\neg q3 \vee \neg q`5)) \wedge (\neg q4 \vee \neg q`3)) \wedge (\neg q4 \vee \neg q`5)) \vee p2))$

Piojo
$((\neg p3 \vee ((((((((((q2 \wedge q`4) \vee (q1 \wedge q`2)) \vee (q1 \wedge q`4)) \vee (q1 \wedge q`5)) \vee (q2 \wedge q`1)) \vee (q2 \wedge q`4)) \vee (q2 \wedge q`5)) \vee (q4 \wedge q`1)) \vee (q4 \wedge q`2)) \vee (q4 \wedge q`5))) \wedge (((((((((((\neg q2 \vee \neg q`4) \wedge (\neg q1 \vee \neg q`2)) \wedge (\neg q1 \vee \neg q`4)) \wedge (\neg q1 \vee \neg q`5)) \wedge (\neg q2 \vee \neg q`1)) \wedge (\neg q2 \vee \neg q`4)) \wedge (\neg q2 \vee \neg q`5)) \wedge (\neg q4 \vee \neg q`1)) \wedge (\neg q4 \vee \neg q`2)) \wedge (\neg q4 \vee \neg q`5)) \vee p3))$

Pulga
$((\neg p4 \vee ((((((((((q3 \wedge q`5) \vee (q1 \wedge q`2)) \vee (q1 \wedge q`3)) \vee (q1 \wedge q`5)) \vee (q2 \wedge q`1)) \vee (q2 \wedge q`3)) \vee (q2 \wedge q`5)) \vee (q3 \wedge q`1)) \vee (q3 \wedge q`2)) \vee (q3 \wedge q`5))) \wedge (((((((((((\neg q3 \vee \neg q`5) \wedge (\neg q1 \vee \neg q`2)) \wedge (\neg q1 \vee \neg q`3)) \wedge (\neg q1 \vee \neg q`5)) \wedge (\neg q2 \vee \neg q`1)) \wedge (\neg q2 \vee \neg q`3)) \wedge (\neg q2 \vee \neg q`5)) \wedge (\neg q3 \vee \neg q`1)) \wedge (\neg q3 \vee \neg q`2)) \wedge (\neg q3 \vee \neg q`5)) \vee p4))$

Mosca
$((\neg p5 \vee ((q4 \vee q2) \vee q3)) \wedge (((\neg q4 \wedge \neg q2) \wedge \neg q3) \vee p5) \wedge ((\neg p5 \vee s5) \wedge (\neg s5 \vee p5)))$

Como podemos apreciar hemos logrado representar la información en una instancia de SAT, sobre cuales son las condiciones suficientes y necesarias para que cada bicho pueda estar presente en el casamiento de una manera en la que todos se encuentren sentados además de que cada uno de ellos solo ocupe su silla y no la compartan con alguien mas, además de claro seguir las peticiones de cada invitado para evitar conflictos en la fiesta.

6 (**2 pts.**) Una cláusula $\mathcal{C} = \ell_1 \vee \ldots \vee \ell_k$ es de Horn si a lo más una de sus literales ℓ_i es positiva.
Por ejemplo $\neg t \vee p \vee \neg q \vee \neg s$ es de Horn y $\neg t \vee p \vee q$ no es de Horn.

(a) ¿ existen cláusulas de Horn unitarias? en caso afirmativo, de qué forma son.
Sí existen cláusulas de Horn unitarias, ya que por definicion de clausula, esta puede albergar desde una literal hasta un numero n de ellas, entonces, en este caso particular de las clausulas de Horn tendríamos que las clausulas de Horn unitarias tendrían que verse forzosamente como una clausula donde solo se encuentre una l, la cual forzosamente no es negativa para cumplir que sea de Horn.

(b) ¿ existen cláusulas de Horn positivas? en caso afirmativo, de qué forma son.
Si existen las cláusulas de Horn positivas, por la definicion dada, las clausulas de Horn deben tener a lo mas una de sus literales de forma positiva, sin embargo si invertimos esta regla para que entonces, tengamos una clausula que tenga a lo mas una literal negativa estaríamos ante algo diferente a una clausula de Horn, estaríamos ante una clausula positiva de Horn.

(c) ¿ existen cláusulas de Horn negativas? en caso afirmativo, de qué forma son.
Si existen las cláusulas de Horn negativas ya que por la definicion de clausula de Horn tenemos que es de Horn **si a lo mas una de sus literales es positiva**, por lo que podemos tener una clausula que no tenga literales positivas ya que la definicion no no lo impide y eso lo volvería una clausula de Horn negativa

(d) Una fórmula es de Horn si en su forma normal conjuntiva todas las cláusulas son de Horn.
Dada una fórmula cualquiera A, ¿ es posible hallar una fórmula de Horn equivalente a A?
Si es posible ya que para haber llegado a la forma normal conjuntiva de esas clausulas debimos haber pasado antes por la forma normal negativa y a ambas se llega por medio de equivalencias, por lo que si, debe existir una equivalencia a A

(e) Demuestre o refute la siguiente propiedad: un resolvente de dos cláusulas de Horn es necesariamente una cláusula de Horn.
Esta afirmación no es verdadera necesariamente por que si tomamos una clausula de Horn Positiva y una normal , su resolvente no sera una clausula de Horn

7 Punto Extra (**hasta 2 pts**)
Sea H un conjunto de cláusulas de Horn no positivas.
Muestre que H es satisfacible.

Si H es un conjunto de cláusulas de Horn no positivas, entonces cada cláusula en H debe ser una cláusula de Horn negativa o una cláusula de Horn unitaria.
Recordando las respuestas a las preguntas anteriores, cada cláusula de Horn negativa puede contener un solo literal negativo, y cada cláusula de Horn unitaria puede contener un solo literal positivo.

Por lo que si tenemos a un conjunto S (el cual sera el conjunto de variables proposicionales que aparecen en las cláusulas de H), tendremos que si una cláusula de H es una cláusula de Horn negativa, entonces su literal negativa debe ser una variable proposicional en S, por consiguiente si una cláusula de H es una cláusula de Horn unitaria, entonces su literal positivo también debe ser una variable proposicional en S.

Ahora, si para cada variable proposicional p en S, asignamos un valor de verdad arbitrario, ya sea verdadero o falso y luego, para cada cláusula de Horn negativa en H que contenga la variable proposicional, asignamos el valor de verdad falso, de esta forma, cada cláusula de Horn negativa se vuelve verdadera, ya que su único literal negativo se evalúa como falso.
Para cada cláusula de Horn unitaria en H que contenga la variable proposicional en cuestión como un literal positivo, le asignamos el valor de verdad.
De esta forma, cada cláusula de Horn unitaria se vuelve verdadera, ya que su único literal se evalúa como verdadero.

Observemos que esta asignación de verdad satisface todas las cláusulas en H.
Por lo tanto, si H es un conjunto de cláusulas de Horn no positivas, entonces H es satisfacible.

¿Es esto cierto si existen cláusulas positivas en H ?

Si llegasen a existir cláusulas de Horn positivas en H, entonces no necesariamente es cierto que H sea satisfacible, ya que , la presencia de cláusulas de Horn positivas en H puede hacer que H sea insatisfacible, por tomar un ejemplo, si H contiene tanto una cláusula de Horn positiva como su negación, entonces H no puede ser satisfacible.