Lógica Computacional 2019-2
Proyecto 3

Profesor: Fernando Abigail Galicia Mendoza
Ayudante: Leonardo Hernández Cano
Laboratorista: Francisco Emmanuel Anaya González

Licenciatura en Ciencias de la Computación
Facultad de Ciencias, UNAM

Hernández Leyva Mirén Jessamyn
Ruiz López Jorge Antonio

UNIFICACIÓN

Al igual que en la lógica de proposiciones puede ser que necesitemos saber la validez de un argumento, así que como antes tenemos los Tableux, pero necesitamos más formas de saber la validez o correctud de un argumento. ¿Que tal si adaptamos el método de resolución binaria para lógica de predicados? ¿Es posible?.

Para resolver problemas como el anterior necesitamos conocer más sobre ésta lógica.

Primero retomemos el tema de sustituciones que se definió anteriormente. Como vimos en algunas ocasiones necesitaos aplicar sustituciones, pero ¿Es posible encontrar una sustitución que haga que todo una conjunto Γ sea igual entre cada una de sus literales?, por ejemplo:

Sea $\Gamma = \{Pxy, Pwy, Pwfx\}$ donde $P^{(2)}, f^{(1)}$
entonces buscamos tener una sustitución que nos deje a todas las literales iguales como sigue: $\{Pwfx, Pwfx, Pwfx\}$ donde la sustitución sería $[x := w, y := fx]$

¿Por qué es más complejo que en lógica propocicional?

En lógica de predicados como acabamos de ver, no es tan sencillo pues a diferencia de las literales p, $\neg p$ en lógica proposicional, las literales Pax, $\neg Pzb$ no pueden resolverse aunque tengan el mismo símbolo de predicado, pues no son contrarias estrictamente.

¿Qué es una **sustitución**?

Definición 1. Una sustitución en lógica de predicados es una función $sigma : Var \to Term$ con la propiedad de que $\sigma(x) = x$ para casi todas las variables. Es decir $\sigma(x_i) \neq x_i$ únicamente para un número finito de variables x_i con $1 \leq i \leq n$ (que tenga soporte finito).

Se denota cómo sigué:

$$\sigma = [x1 := t_1, x_2 := t_2, ..., x_n := t_sn]$$
$$\text{o bien: } \sigma = [\vec{x} := \vec{t}]$$

Ejemplo:

Sean $\sigma = [x := gxy, z := w]$ con g^2 una función y $\varphi = Pzfxy \wedge Qz$ una formula con P^2 y Q^1 predicados y f^2 una función. Aplicaremos la sustitución σ sobre φ, es decir:
$\varphi\sigma = (Pzfxy \wedge Qz)[x := gxy, z := w]$
$\varphi\sigma = (Pwfgxyy \wedge Qw)$

¿Qué es la **composición de sustituciones**? ¿Para que sirve?

Proposición 1.
Sean $\sigma = [x_1 := t_1, ..., x_n := t_n]$ y $\tau = [y_1 := s_1, ..., y_m := s_m]$ dos sustituciones. La composición, denotada $\sigma \circ \tau$ o simplemente $\sigma\tau$, es una sustitución definida por:

$$\sigma\tau = [z_1 := t_1\tau, ..., z_k := t_k\tau, w_1 := r_1, ..., w_l := r_l]$$

donde $k \leq n$, $l \leq m$, las variables z_i son exactamente aquellas x_p tales que $x_p \neq t_p\tau$, obsérvese que son distintos sintácticamente, es decir, no confundir el operador de sustitución := con el de igualdad =, y los pares $w_j := r_j$ son aquellos pares $y_q := s_q$ tales que $y_q \notin \{x_1, ..., x_n\}$, es decir aquellos donde no se puede aplicar la sustitución.

La composisción de sustituciones es asociativa, i.e. $(\sigma\rho)\tau = \sigma(\rho\tau)$.

Demostración.

Consideremos una variable cualquiera u. P.D. $u(\sigma\rho)\tau = u\sigma(\rho\tau)$
Sabemos que $u(\sigma\rho)\tau = [u(\sigma\rho)]\tau = [u(\sigma)\rho]\tau$ ... (1)
y por otra parte, $u\sigma(\rho\tau) = (u\sigma)(\rho\tau) = [(u\sigma)(\rho\tau)] = [(u\sigma)\rho]\tau$ .. (2) y como podemos notar (1) es igual a (2), es decir, $[(u\sigma)\rho]\tau = [(u\sigma)\rho]\tau$.

Ejemplo:

Sean

$$\sigma = [x := f(y), y := u], \rho = [x := g(u), w := r], \tau = [u := h(r), z := v]$$

$$\sigma\rho = [x := f(u), y := u, w := r]$$

$$\rho\tau = [x := g(h(r)), w := r, u := h(r), z := v]$$
$$(\sigma\rho)\tau = [x := f(h(r)), y := h(r), w := r, u := h(r), z := v]$$
$$\sigma(\rho\tau) = [x := f(h(r)), y := h(r), w := r, u := h(r), z := v]$$

Notemos que en el caso de x, como se define antes una sustitución lo que se hace es siempre "ignorar la segunda".

Ahora que definimos las sustituciones y basandonos en el ejemplo anterior donde buscabamos que todas las literales fueran iguales entonces podemos decir que aquella sustitución σ que hace que las literales de un conjunto W sean todas iguales entre ellas, es decir W consta de un solo elemento, la llamaremos *Unificador.*

¿Qué son los **unificadores**?

Todas las variables están cuantificadas universalmente de manera implícita, es decir en el caso de $\neg Pzb$ y Pax en realidad tenemos $\forall x\ Pax$ y $\forall x\ \neg Pzb$ por lo que podemos aplicar la sustitución $\sigma = [x, z := b, a]$ a las fórmulas sin cuantificadores, obteniendo como resultado Pab y $\neg Pab$ que sí son literales contrarias.

El proceso de unificación consiste en encontrar, dado un conjunto de literales o términos W, una sustitución de tal forma que el conjunto resultante W conste de un solo elemento.

Definición 2. Sea W un conjunto no vacío de términos. Un unificador de W es una sustitución σ tal que $|W\sigma| = 1$, es decir el conjunto $W\sigma$ resultante de aplicar σ a todos los elementos de W consta de un mismo elemento. Si W tiene un unificador decimos que W es unificable.

Un conjunto de términos puede tener una infinidad de unificadores o ninguno, dado un conjunto finito de fórmulas W, es decidible mediante un algoritmo si W es unificable o no; si W es unificable el algoritmo proporciona un unificador que llamaremos el unificador más general.

Ejemplo:

Consideremos el siguiente conjunto $W = \{Pxfya, Pbfca\}$, donde $P^{(2)}, f^{(2)}$ y a, b, c son ctes., entonces un unificador σ para W que cumple que $|W\sigma| = 1$ es:

$\sigma = [x := b, y := c]$

el cual al aplicarlo nos deja a W como: $\{Pbfca, Pbfca\}$, del cual podemos ver que $|W\sigma| = 1$. Por lo tanto σ es un unificador para W.

¿Qué es el **unificador más general**? ¿Por qué son importantes?

Definición 3. Un unificador σ de un conjunto de términos W, se llama unificador más general (umg) si para cada unificador τ de W, existe una sustitución ϑ, tal que $\sigma\vartheta = \tau$.

La importancia de los unificadores más generales es que nos permiten representar de manera finita un número infinito de sustituciones.

Ejemplo:

Sea $W = \{gxfcv, ghubfcw\}$ donde $g^{(2)}, f^{(2)}, h^{(2)}$ y a, b, c son constantes. $\tau = [x := hbb, u := b, v := w]$ y $\sigma = [x := hub, v := w]$ así podemos ver que en particular si tenemos un $\rho = [u := b]$ entonces $\tau\rho = \sigma$, de esta manera es facil ver que σ es el **umg**.

Entonces definiremos el algoritmo de unificación de términos para poder generalizarlo a literales. A lo largo de la ejecución del algoritmo notaremos una pequeña sutileza en su ejecución, la cual veremos al momento de definirlo.

El Algoritmo Martelli-Montanari sirve para ayudarnos a encontrar el unificador mas general de un conjunto de literales, o bien saber si este no es unificable.

MARTELLI-MONTANARI

Daremos una idea intuitiva del algoritmo de forma sencilla como se sigue:

Entrada: Un conjunto de ecuaciones $\{s_1 = r_1, ..., s_k = r_k\}$ tales que se desea unificar simultáneamente, con $r_i \neq r_j$, con $1 \leq i \leq k$.

Salida: Un unificador más general μ tal que $s_i\mu = r_i\mu$ para toda $1 \leq i \leq n$.

Ahora daremos la definicion formal de el Algoritmo de Martelli-Montanari:

Nombre de la regla	$P_{t1...tn} = Q_{t1...tn}$	Acción
[DESC] Descomposición	$f_{s1...sn} = f_{t1...tn}$	sustituir $s_i = t_i$
[DESC_EX] Desc. extendida	$P_{s1...sn} = Q_{t1...tn}$	sustituir $s_i = t_i$
[DFALLA] Desc. fallida	$f_{s1...sn} = g_{t1...tn}$ con $f \neq g$	falla
[DFALLA_EX] Desc. ex. fallida	$P_{s1...sn} = Q_{t1...tn}$ con $P \neq Q$	falla
[ELIM] Eliminación	x = x	eliminar
[SWAP] Intercambio	t = x donde t no es variable	sustituir x por t
[SUST] Sustitución	x = t donde x no figura en t	eliminar x = t y aplicar la sustitución [x:= t]
[SFALLA] Sust. fallida	x = t, x figura en t y $x \neq t$	falla

El algoritmo para términos lo ejecutamos como sigue:

Nombre de la regla	$t1 = t2$	Acción
[DESC] Descomposición	$f_{s1...sn} = f_{t1...tn}$	sustituir $s_i = t_i$
[DFALLA] Des. fallida	$f_{s1...sn} = g_{t1...tn}$ con $f \neq g$	falla
[ELIM] Eliminación	x = x	eliminar
[SWAP] Intercambio	t = x donde t no es una variable	sustituir x por t
[SUST] Sustitución	x = t donde x no figura en t	eliminar x = t y aplicar la sustitución [x:= t]
[SFALLA] Sust. fallida	x = t, x figura en t y $x \neq t$	falla

La sutileza entre el algoritmo de unificación y el algoritmo de MM generalizado es, como podemos ver, que en el generalizado tenemos una descomposición que si considera las literales y no solamente los términos como en el algoritmo de unificación.

Ejemplo:

Dado el siguiente conjunto $\{Pfxy, Pfxa\}$ encontrar el unificador mas general.
Usamos Martelli-Montanari para esto:

1. $\{Pfxy = Pfxa\}$
2. $\{fx = fx, y = a\}[DESC_E X]$
3. $\{x = x, y = a\}[DESC]$
4. $\{y = a\}[ELIM]$
5. $\emptyset[SUST][y := a]$

El unificador se obtiene al componer las sustituciones utilizadas desde el inicio:
$\mu = [y := a]$ $\{Pfxy, Pfxa\}\mu = \{Pfxa\}$ es el umg.

Ejemplo de uso:

El algoritmo W, que se infiere del del algoritmo Hindley-Milner, el cual es de importancia en la inferencia de tipos para calculo Lambda, pues es el comunmente utilizado. Este hace uso de un umg, y aun que hay varios algoritmos para hacerlo,el algoritmo de unificación de Martelli-Montanari es uno de los más eficientes.

Una vez que conseguimos unificar términos y literales consiguiendo el umg podemos pensar de nuevo en resolución binaria para la lógica de predicados pero antes deberíamos hacernos algunas preguntas:

¿Cómo podemos aplicarla si las fórmulas están cuantificadas? ¿Qué pasa si tenemos multiples cuantificadores diferentes?

FORMAS NORMALES

En la lógica proposicional teníamos algo llamado Formas Normales (negativa y conjuntiva), las cuales nos servían para poder aplicar el algortimo de resolución binaria, entonces con base en esto podríamos definir las Formas Normales necesarias para poder ejecutar el algoritmo de resolusión binaria o algoritmo de Robinson en la lógica de predicados.

Definición 4. Una fórmula φ está **rectificada** si y solo si se cumplen las siguientes condiciones:

1. φ no tiene presencias libres y ligadas de la misma variable, es decir, $FV(\varphi) \cap BV(\varphi) = \phi$.
2. φ no tiene cuantificadores ajenos de la misma variable con alcances ajenos.
3. φ no tiene cuantificadores multiples de la misma variable, es decir, subfórmulas de la fórmula $\forall x \forall x \psi$, $\exists x \exists x \psi$, $\forall x \exists x \psi$, $\exists x \forall x \psi$.
4. φ no tiene cuantificadores vácuos (tontos), es decir, cuantificadores de la forma $\forall xpsi$, $\exists x \psi$ donde $x \notin FV(\varphi)$.

   NOTA: Cuando decimos que $x \notin FV(\varphi)$ nos podemos referir a:

   (a) $x \notin vars(\varphi)$, es decir, x no aparece en las variables de φ.

   (b) Si somos estrictos con la definición, entonces podría interpretarse que como $x \notin FV(\varphi)$ entonces $x \in BV(\varphi)$, es decir x estaría ligada en φ.

   Entonces a menos que se especifique que $x \in BV(\varphi)$, cuando digamos que $x \notin FV(\varphi)$, *nos referiremos a que x no figura en φ.*

Como el objetivo principal de esto es mostrar que cualquier fórmula es rectificable entonces podemos apoyarnos de lo siguiente.

Lema 1. Se cumplen las siguientes equivalencias lógicas:

1. Eliminación de cuantificadores multiples:

- $\forall x \forall x \varphi \equiv \forall x \varphi$
- $\exists x \exists x \varphi \equiv \exists x \varphi$
- $\exists x \forall x \varphi \equiv \forall x \varphi$
- $\forall x \exists x \varphi \equiv \exists x \varphi$

2. Renombre de variables. Si $y \notin FV(\varphi)$ entonces:

   - $\forall x \varphi \equiv \forall y(\varphi[x := y])$
   - $\exists x \varphi \equiv \exists y(\varphi[x := y])$

   Si $y \notin FV(\varphi)$, entonces: $\forall x \varphi \equiv \forall y(\varphi[x := y])$
   Demostración:

   Sea $M = \langle M, I \rangle$ y $\sigma : Var \to M$ **P.D.** $\models \forall x \varphi \leftrightarrow \forall y(\varphi[x := y])$
   P.D. $I_\sigma(\forall x \varphi) = 1$ syss $I_\sigma(\forall y(\varphi[x := y])) = 1$

   $\to$) Si $I_\sigma(\forall x \varphi) = 1$ entonces $\forall m \in M$ $I_{\sigma[x/m]}(\varphi) = 1$ y como $y \notin FV(\varphi)$, es decir, $y \notin vars(\varphi)$ podemos aplicar una sustitución, en particular podemos aplicar $[x := y]$ así obtenemos $I_{\sigma[x/y]}(\varphi) = 1 = I_\sigma(\forall y([x := y])) = 1$

   $\leftarrow$) Si $I_\sigma(\forall y(\varphi[x := y])) = 1$ entonces $\forall m \in M$ $I_{\sigma[y/m]}(\forall y \varphi) = 1$ $= I_\sigma(\forall x(\varphi[x := y])) = 1$ y como por hipótesis tenemos que $y \notin FV(\varphi)$ entonces tenemos $I_{\sigma[x/m]}(\varphi) = 1 = I_\sigma(\forall x \varphi) = 1$

3. Eliminación de cuantificadores vácuos. Si $x \notin FV(\varphi)$ entonces:

   - $\forall x \varphi \equiv \varphi$
   - $\exists x \equiv \varphi$

Ejemplo:
Se la formula $\varphi = \forall p \forall p(Qp \land \forall p \exists q(Qrq \lor p \forall r(Mr))$ buscamos su forma rectificada.

- Eliminamos los cuantificadores multiples:
  $\varphi = \forall p \forall p(Qp \land \forall p \exists q(Qrq \lor p \forall r(Mr))$

- Renombramos variables
  $FV(\varphi) = \{r\}$ y $BV(\varphi) = \{p, q, r\}$ entonces $FV(\varphi) \cap BV(\varphi) \neq \emptyset$ por lo que hay que renombrar r variable libre $[r := s]$ y optenemos $\varphi = \ p(Qp \wedge \forall p \exists q(Qsq \vee p\forall r(Mr))$

- Quitamos los cuantificadores vacuos.
  $\varphi = \forall p(Qp \wedge \exists q(Qsq \vee p\forall r(Mr))$ la cual es la formula ya rectificada.

Definición 5. Forma Normal Negativa (FNN)

Una fórmula está en Forma Normal Negativa syss se cumplen las siguientes condiciones:

1. φ no contiene símbolos de equivalencia ni implicaciones y
2. las negaciones que figuran en φ afectan sólo a fórmulas atómicas.

Como la transformación a la FNN en la lógica de predicados en bastante similar a la de la lógica proposicional entonces solo bastaria agregar los casos donde la negación afecta a un cuantificador, entonces las leyes resutarían como sigue:

- Doble Negación: $\neg\neg\varphi \equiv \varphi$
- De Morgan: $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
- De Morgan: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$
- $\neg(\varphi \rightarrow \psi) \equiv \varphi \wedge \neg\psi$
- $\neg(\varphi \leftrightarrow \psi) \equiv \neg\varphi \leftrightarrow \psi \equiv \varphi \leftrightarrow \neg\psi$
- $\neg\forall x\varphi \equiv \exists x\neg\varphi$
- $\neg\exists x\varphi \equiv \forall x\neg\varphi$

Mostraremos la siguiente equivalecia: $\neg\exists x\varphi \equiv \forall x\neg\varphi$

Demostración:

Sean $M = \langle M, I \rangle, \sigma : Var \to M$ **P.D.** $\models \neg\exists x\varphi \leftrightarrow \forall x\neg\varphi$
P.D. $I_\sigma(\neg\exists x\varphi) = 1$ syss $I_\sigma(\forall x\neg\varphi) = 1$

Si $I_\sigma(\neg\exists x\varphi) = 1$ por definición esto se cumple syss $\exists\sigma : Var \to M$ tal que $I_\sigma(\neg\exists x\varphi) = 1$ syss $\neg\exists m \in M$ tal que $I_{\sigma[x/m]}(\varphi) = 1$ syss $\neg\exists m \in M$ tal que $I_{\sigma[x/m]}(\neg\varphi) = 0$, *"no hay ninguno tal que su negación sea cero"*, syss $\forall m \in M$ $I_{\sigma[x/m]}(\varphi) = 0$ syss $\forall m \in M$ $I_{\sigma[x/m]}(\neg\varphi) = 1$ syss $I_\sigma(\forall x\neg\varphi) = 1$

Notemos que al ser un camino de syss's no es necesario hacer "ida y vuelta".

Definición 6. Forma Normal Prenex (FNP)

Una fórmula φ esta en Forma Normal Prenexa si está escrita encabezada por una cadena de cuantificadores existenciales o universales, seguidos por una fórmula sin cuantificadores lógicos.

Es decir, es de la forma:

$$Q_1x_1, Q_2, x_2, ..., Q_nx_n\varphi \text{ donde } Q_i \in \{\forall, \exists\} \text{ para toda } 1 \leq i \leq n$$

En tal caso, la cadena de cuantificadores $Q_1x_1, ..., Q_nx_n$ se conoce como el prefijo de φ mientras que ψ es la matriz de φ.

Lema 2. Se cumplen las siguientes equivalencias.

1. Si $x \notin FV(\phi)$:
   - $\varphi \wedge \forall x\psi \equiv \forall x(\varphi \wedge \psi)$
   - $\varphi \wedge \exists x\psi \equiv \exists x(\varphi \wedge \psi)$
   - $\varphi \vee \forall x\psi \equiv \forall x(\varphi \vee \psi)$
   - $\varphi \vee \exists x\psi \equiv \exists x(\varphi \vee \psi)$
   - $\varphi \to \forall x\psi \equiv \forall x(\varphi \to \psi)$
   - $\varphi \to \exists x\psi \equiv \exists x(\varphi \to \psi)$
2. Si $x \notin FV(\psi)$:

- $\forall x\varphi \to \psi \equiv \exists x(\varphi \to \psi)$
- $\exists x\varphi \to \psi \equiv \forall x(\varphi \to \psi)$

Demostraremos que **si x no figura libre en φ, entonces:** $\forall x\varphi \to \psi \equiv \exists x(\varphi \to \psi)$

Demostración:

Por hipotesis $x \notin FV(\varphi)$.
Sea $M =< M, \mathcal{I}, >$ una $\mathcal{L}$-estructura y $\sigma : Var \to |M|$.

$$\text{P.D. } \mathcal{I}_\sigma(\forall x \to \psi) = 1 \text{ syss } \mathcal{I}_\sigma(\exists x(\varphi \to \psi)) = 1.$$

Supongamos que $\mathcal{I}_\sigma(\forall x\varphi) = 1$ entonces $\mathcal{I}_\sigma(\psi) = 1$

$$\text{P.D. } \mathcal{I}_\sigma(\exists x(\varphi \to \psi)) = 1.$$

Es decir, hay que demostrar que $\exists m \in M$ tal que $\mathcal{I}_{\sigma[x/m]}(\varphi \to \psi) = 1$.

Procedemos por contradicción:

Supongamos que $\forall m \in M$ $\mathcal{I}_{\sigma[x/m]}(\varphi \to \psi) = 0$, esto ocurre syss

$$\mathcal{I}_{\sigma[x/m]}(\varphi) = 1 \text{ y } \mathcal{I}_{\sigma[x/m]}(\psi) = 0$$

(Llegamos a una contradición ya que que por (1) sabemos que $\mathcal{I}_{\sigma[x/m]}(\psi) = 1$), por lo tanto $\mathcal{I}_\sigma(\exists x(\varphi \to \psi)) = 1$.

Así $\forall x\varphi \to \psi \equiv \exists x(\varphi \to \psi)$

Definición 7. Forma Normal Skolem (FNS)

Una enunciado de la forma $\forall x_1, ..., \forall x_n\varphi$ donde φ no tiene cuantificadores y además está en forma normal conjuntiva es una fórmula en forma normal de Skolem.

Es decir buscamos que φ este encabezada solo por cuantificadores universales ($\forall$), así que buscamos eliminar los existenciales ($\exists$), la estrategia a seguir para esto es la siguiente:

$$\exists x\varphi \text{ se sustituye por } \varphi[x := c]$$
$$\forall x_1, ..., \forall x_n \exists y\varphi \text{ se sustituye por } \forall x_1, \forall x_n\varphi[y := gx_1...x_n]$$

Donde c es un símbolo de constante nuevo, llamado constante de Skolem y $g^{(n)}$ es un símbolo de función nuevo llamado función de Skolem. Con $Sko(\varphi)$ denotamos a la skolemización de φ que se obtiene al eliminar mediante este proceso, de izquierda a derecha, todos los cuantificadores existenciales de la fórmula φ.

Este tipo de formulas son equisatisfacibles.

Definición 8. Equisatisfacibilidad.

Sean φ, ψ dos fórmulas, decimos que φ, ψ son equisatisfacibles cuando se cumple que φ es satisfacible si y sólo si ψ es satisfacible.
Esta propiedad será denotada con $\varphi \sim_{sat} \psi$.

Proposición 5.

Si φ es una fórmula entonces $\varphi \sim_{sat} Sko(\varphi)$

Demostración:

Supongamos que está en FNP, lo cual solo nos dejaría los cuantifadores juntos y eliminaría aquellos repetidos o que simplemente cuantifican variables que no aparecen en φ. Entonces $\exists M, M \models \varphi$

P.D. $\exists M, M \models \varphi_{FNS}$

Caso 1) Si φ no tiene cuantificadores $\exists$, entonces como $\exists M, M \models \varphi$ podemos decir que φ_{FNS} queda igual pues no hay sustituciones por aplicar para quitar el cuantificador $\exists$ porque no hay ninguno, así podemos decir que $\exists M, M \models \varphi_{FNS}$.

Caso 2) Si φ si tiene cuantificadores $\exists$, entonces φ puede ser de la siguiente forma:

- $\varphi =_{def} \exists x\varphi$ (φ es de la forma $\exists x\varphi$), entonces $Sko(\varphi) = \varphi[x := c]$ entonces como $M \models \exists x\varphi$ tenemos que $M \models \varphi[x := c]$ pues quitamos el $\exists$, pero también las variables que cuantificaba.

- $\varphi =_{def} \forall x_1...\forall x_n \exists y\varphi$ entonces $Sko(\varphi) = \forall x_1...\forall x_n\varphi[y := gx_1...x_n], g^{(n)}$ y tenemos que $M \models \varphi$, entonces podemos fijarnos en la parte del $\exists$, y como $M \models \varphi$, pero como $Sko(\varphi) = \forall x_1...\forall x_n\varphi[y := gx_1...x_n], g^{(n)}$ y como $M \models \forall x_1...\forall x_n\exists y\varphi$ tenemos que $M \models \forall x_1...\forall x_n\varphi[y := gx_1...x_n]$, pues la variable que era cuantificada por el existencial desaparece y se pone como función de cada variable del $\forall$ y así solo nos quedamos con variables cuantificadas con el universal, por lo tanto $M \models \varphi_{FNS}$

Por lo tanto $\varphi \sim_{sat} Sko(\varphi)$.

Definición 9. Forma Normal Clausular (FNC)

Sean φ una fórmula y fns$(\varphi) = \forall x_1...\forall x_n\psi$ la forma normal de Skolem de φ. Si ψ es $C_1 \wedge .. \wedge C_n$ entonces la forma clausular de φ, denotada $Cl(\varphi)$ es la secuencia de cláusulas

$$Cl(\varphi) = C_1, .., C_n$$

donde sin perder generalidad las variables en cualesquiera dos cláusulas son distintas. Es decir, cualesquiera dos cláusulas tienen variables ajenas.

El algoritmo que **dada una fórmula φ cualquiera devuelva su FNC**, es el siguiente:

1. Rectificar φ

2. Dar la FNN de φ

3. Prenexar φ

4. Skolemizar φ

5. Transformar ψ (la matriz de la skolemización anterior) a FNC obteniendo $fnc(\psi) = C_1 \wedge C_2 \wedge ... \wedge C_k$

6. Renombrar las variables de las clausulas para que no cincidan entre clausulas
7. La FNC es $\{C_1', ..., C_n'\}$

Ejemplo:

Dada la formula $\varphi = \forall x(\neg Pxy \wedge \forall x \forall y(Qyz \rightarrow \exists y \exists z Rz))$ dar su FNC. Procederemos paso a paso.

1. **Rectificar φ:**

   - $FV(\varphi) \cap BV(\varphi) \neq \emptyset$ pues $FV(\varphi) = \{y, z\}$ y $BV(\varphi) = \{x, y, x\}$ por lo que hay que renombramos las variables que nos causan problema (y y z) cómo sigue:

     [y := w]
     z := v

     y obtenemos: $\varphi = \forall x(\neg Pxw \wedge \forall x \forall y(Qyv \rightarrow \exists y \exists z Rz))$
   - Hay cuantificadores sobre la misma variable pero con distinto alcance por lo que procedemos a quitarlos y obtenemos:

     $$\varphi = \forall x(\neg Pxw \wedge \forall y(Qyv \rightarrow \exists y \exists z Rz))$$

   - Quitamos los cuantificadores tontos.

     $$\varphi = \forall x(\neg Pxw \wedge \forall y(Qyv \rightarrow \exists z Rz))$$

2. **Dar la FNN de φ**

   Usando De Morgan ($\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$) obtenemos:

   $$\varphi = \forall x(\neg Pxw \wedge \forall y(\neg Qyv \vee \exists z Rz))$$

3. **Prenexar φ**
   Utilizando las equivalencias:

   - $\varphi \wedge \forall x \psi \equiv \forall x(\varphi \wedge \psi), ..., (1)$
   - $\varphi \vee \exists x \psi \equiv \exists x(\varphi \vee \psi), ...(2)$

obtenemos:
$\varphi = \forall x \forall y(\neg Pxw \wedge (\neg Qyv \vee \exists z Rz))$ aplicando (1)
$\varphi = \forall x \forall y \exists z(\neg Pxw \wedge (\neg Qyv \vee Rz))$ aplicando (2)

4. **Skolemizar φ**
Utilizamos la regla $\forall x_1, ..., \forall x_n \exists y \varphi$ se sustituye por $\forall x_1, \forall x_n \varphi[y := gx_1...x_n]$ donde para nuestro caso se va a sustituir [z := gxy] obteniendo:
$\varphi = \forall x \forall y(\neg Pxw \wedge (\neg Qyv \vee Rgxy))$

5. **Transformar ψ a FNC**
Obtenemos $Cl(\varphi) = \neg Pxw$,$\neg Qyv \vee Rgxy$

6. **Renombrar las variables de las clausulas para que no cincidan entre clausulas** Las variables que coinciden son x y por lo que aplicaremos las siguientes sustituciones a la segunda formula:

   [x := u]
   y := m

   obteniendo: $\neg Qmv \vee Rgum$

7. **y así obtenemos la FNC:** $\{\neg Pxw$,$\neg Qmv \vee Rgum\}$

CONCLUSIONES

Cómo pudimos notar a lo largo del proyecto, los procesos de unificación y para convertir a formas normales no solo son más complicados en lógica de predicados que en lógica proposicional, si no que además tienen muchisimo más poder:

La unificación de formulas es un proceso que puede ser muy complicado, se ha logrado facilitar gracias al algoritmo Martelli-Montanari.

Por otro lado tenemos la FNC la cual facilita aún más los procesos pues reduce las formulas a clausulas mucho más simples de manejar y nos permite, de la misma manera que en la lógica de proposiciones usar algoritmos como el de resolusión binaria o de Robinson.

Tambien pudimos notar la importancia y utilidad de los metodos descritos para diferentes aplicaciones como satisfacibilidad, validez y consecuencia lógica.