Universidad de Los Andes

http://cesimo.ing.ula.ve

Unidad 1: Introducción a la lógica matemática

Jacinto Dávila

mailto:jacinto@ula.ve

Centro de Simulación y Modelos (CESIMO)

Proposiciones

¿Qué es una proposición?

Una proposición es un decir asociado a un valor de verdad.

Los valores de verdad, normalmente, son verdadero (V) y falso (F).

Actividad: En la tabla siguiente, asigne valores de verdad a las proposiciones:

q= "3 es mayor que 2"	**q=V**
p= "El cielo está nublado"	**p=__**
s= "Estaba vivo pocos minutos antes de morir"	**s=__**
t= "Por favor, hable un poco más lento"	**t=__**
z= "Ojalá que llueva"	**z=__**
r= "¿Quién ha llegado?"	**r=__**
u= "La proposición u es falsa"	**u=__**
v= "El rey de Venezuela no es ateo"	**v=__**

¿Qué tan libres somos para asignar esos valores?

Proposiciones simples y compuestas

¿Cómo asignamos los valores de verdad en estos casos:?

p="El perro está durmiendo y el gato está despierto".

q="Si un entero es múltiplo de 6 entonces es múltiplo de 3".

r="Si llega Pedro entonces se hace la reunión".

s="Lo dijo o no lo dijo".

t="Un pan es un pan o Ecuador está en Europa".

u="Es un satélite si y sólo si gira en torno a otro astro".

v="Cayó del árbol y se fracturó un hueso".

w="Se fracturó un hueso y cayó del árbol".

x="3 más 4 es 7 y 8 es divisible por 5".

y="No ha llegado"

z="No es cierto que Miguel ha llegado"

m="Llueve, implica que esta nublado"

¿De qué dependen los valores de verdad?

La semántica del lenguaje proposicional

Esas formas de dependencias están asociadas a ciertas palabras que conectan o modifican las proposiciones simples

Es posible construir una **tabla de la verdad** mostrando esas dependencias *funcionales*. Acá las presentamos con los símbolos normalmente usados entre los lógicos.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \equiv q$
V	V	F	V	V	V	V
V	F	F	F	V	F	F
F	V	V	F	V	V	F
F	F	V	F	F	V	V

Actividad: Observe cuidadosamente el caso de $p \rightarrow q$

Fórmulas proposicionales

La sintáxis del lenguaje proposicional se construye así:

Toda proposición es una fórmula.

Si X e Y son fórmulas entonces

$$(\neg X), (X \wedge Y), (X \vee Y), (X \supset Y), (X \equiv Y), (X)$$

son también fórmulas.

Aplicando las reglas anteriores un número finito de veces se obtienen todas las fórmulas. En algunos textos las fórmulas así formadas se denominan **fórmulas bien formadas**.

Interpretaciones, modelos y fórmulas consistentes

Una interpretación de una fórmula es una asignación de un valor de verdad a cada una de sus componentes.

Un modelo es un interpretación que hace cierta a la fórmula.

Una fórmula es consistente si puede ser verdadera para alguna interpretación.

La fórmula del ejemplo anterior es consistente. La formula $p \wedge \neg p$ no es consistente.

Una fórmula que no es verdadera para ninguna interpretación se llama inconsistente.

Actividad: Averigüe qué es una Tautología.

¿En qué consiste la deducción?

Contigencia, validez y falsabilidad

Una fórmula se dice válida si es verdadera para toda interpretación posible. $p \vee \neg p$ **es válida**.

Las fórmulas válidas se llaman también **tautologías**.

Una fórmula consistente pero no válida se denomina contingente.

Una fórmula contingente es falsable.

Actividad: Revise los titulares de prensa de hoy e indique cuáles se pueden asociar a proposiciones falsables.

Consecuencia lógica

Si S es un conjunto de fórmulas y A una fórmula entonces

$$S \models A$$

Esto significa que todas las interpretaciones que hacen verdaderas a **todas** las fórmulas que pertenecen a S también hacen verdadera a A.

Se dice también que A es **consecuencia lógica** de las fórmulas de S.

Para que no lo sea tiene que existir una interpretación que haga verdaderas todas las fórmulas de S y haga falsa a A.

Consistencia de un conjunto de fórmulas

Un conjunto S de fórmulas es consistente si todos sus componentes se hacen verdaderos para alguna interpretación. Para esa interpretación se dice que el conjunto es **consistente**.

Nótese que la consistencia del conjunto equivale a que sea consistente la fórmula constituída por la conjunción de todas sus fórmulas.

Según esta definición de consecuencia lógica, si un conjunto de fórmulas $\{S_1, S_2, .., S_n\}$ es inconsistente, cualquier fórmula Q es su consecuencia lógica[a].

Actividad: asegúrese de entender la última afirmación.

Para indicar que una fórmula inconsistente es consecuencia lógica del conjunto de fórmulas S se indica $S \models Falso$.

[a] pués la implicación $(S_1 \wedge S_2 \wedge \; .. \; \wedge S_n) \rightarrow Q$ es una tautología

El principio de deducción.

C **es consecuencia lógica del conjunto de fórmulas** S **si y sólo si el conjunto de fórmulas** $H = S \cup \{\neg C\}$ **es inconsistente, es decir F para toda interpretación**.

Esto se ve notando que, o bien S es inconsistente con lo cual C (o cualquier otra fórmula) es su consecuencia lógica, o bien S es consistente por lo cual, para toda interpretación que hace S verdadero, para hacer H falso debe ser $\neg C$ falso o sea C verdadero.

A este principio de deducción se llama en Matemática **demostración por el absurdo (reducción al absurdo)**.

Las partes de una proposición y los límites del lenguaje proposicional

Actividad: consideremos las proposiciones siguientes:

p="Todo mamífero es vertebrado"

q="Algún mamífero es vertebrado"

h="Ningún mamífero es vertebrado"

r="Ningún mamífero es no vertebrado"

s="Algún no mamífero es vertebrado"

t="Existe algún mamífero no vertebrado"

u="Todo mamífero es no vertebrado"

v="Todo vertebrado es mamífero"

w="Alg'un vertebrado es no mamífero"

Si p es cierta, ¿qué se puede decir acerca de las otras proposiciones?.

El lenguaje proposicional es menos expresivo que el lenguaje natural.

Como se aprecia en la lámina anterior, las variables proposicionales ocultan relaciones lógicas que son muy importante en nuestra forma de razonar.

Actividad: Discuta cómo esto limita la "expresividad" del lenguaje.

¿Qué es expresividad?.

En busca de un nuevo lenguaje

inferencias posibles en la lógica de predicados

Actividad: Considere estos ejemplos:

¿Se pueden expresar en lógica proposicional?.

"París está en Francia"

"Francia está en Europa"

"París está en Europa"

"Juan es padre de Adolfo"

"Adolfo es hijo de Juan"

"Todos los perros ladran"

"Merlin es un perro"

"Merlin ladra"

Actividad: Cuál es el problema

Elementos del cálculo o lógica de predicados

1. **Constantes**: romeo, julieta, ...
2. **Variables**: X, Y, Ahora, Luego, Jacinto, ... (según cierta convención).
3. **Funciones**: f(X), .(), [H—T], ...
4. **Símbolos de predicados**: ama_a/2, mortal/1, ...

Estos son los elementos básicos de la lógica ahora llamada clásica

Definiciones sintácticas en el cálculo de predicados

constantes , nombre de función , predicados, variables y símbolo no-lógicos

Definición lm1.1 : Un término se define así :

- Una constante o una variable es un término.
- Si $t_{1.}$,.....t_n son términos y p es el nombre de una función n-aria entonces p (t_1 ,....., t_n) es un término.

Definición lm1.2: Si t_1,,.., t_n son términos y q es un predicado n-ario , entonces q (t_1 ,....,t_n) es un *átomo.*

Definición lm1.3: formúlas bién formadas del cálculo de primer orden con igualdad , se definen asi:

- Un átomo es una fórmula bién formada .
- Si t_1y t_2 son términos , $t_1 = t_2$ es una fórmula bién formulada .
- Si Φ_1 y Φ_2 son f.b.f., entonces :

  $(\neg\ \Phi_1)$, $(\ \Phi_1 \wedge \Phi_2\ )$, $(\Phi 1 \vee \Phi_2)$, $(\ \Phi_1 \leftarrow \Phi_2\ )$ $(\ \Phi_1 \rightarrow \Phi_2\ )$ y $(\ \Phi_1 \leftrightarrow \Phi_2\ )$

  son f.b.f.
- Si Φ es una f.b.f. y X es una variable entonces $(\forall\ X\ \ \Phi)$ y $(\exists X\ \ \Phi\ )$ son f.b.f..

Definición lm1.4: Un lenguaje del cálculo de primer orden con igualdad es el conjunto de todas las fórmulas bién formadas del *cálculo de primer orden con igualdad* que se pueden construir con un conjunto dado de símbolo numéricos.

La lógica y el amor

Romeo ama a Julieta	$ama\_a(romeo, julieta)$
Julieta no ama a nadie que am$_a^e$ a Romeo.	$\forall X(ama\_a(X, romeo) \rightarrow \neg ama\_a(julieta, X))$
Todo el mundo ama a alguien.	$(\forall X)(\exists Y)(ama\_a(X, Y))$
Teresa ama a todo el mundo.	$\forall X(ama\_a(teresa, X))$
Existe alguien que ama a todo el mundo.	$(\exists Y)\,(\forall X)\,(ama\_a(Y, X))$
Todo el mundo ama a a un amante.	$\forall X\ \forall Y\ (\exists Z\ ama\_a\,(Y, Z) \rightarrow ama\_a\,(X, Y))$

La semántica del cálculo de predicado

Definición lm1.5: Una interpretación de un lenguaje L es una tripleta ⟨D,F,P⟩ donde: D es un conjunto no vacío de objetos. F es una función que "conecta" cada nombre de constante en L y cada variable en L con un elemento de D y que conecta a cada nombre de función en L con una función de D^n a D. Por su parte, P es una función que conecta cada predicado en L con un subconjunto de D^n. A este conjunto se le conoce como la extensión de ese predicado.

Actividad: Revise cuidadosamente este definición

¿Qué es peculiar en las funciones involucradas?

Actividad: Piense en el dominio y rango de las funciones en matemática elemental

Reglas para construir una interpretación

- Si X es una variable libre, $I(X) = I_v(X)$
- Si f es un nombre de función n-aria y $t_1, t_2, ..., t_n$ son términos (variables o formas funcionales) la interpretación de la fórmula f es el valor de la función correspondiente en la interpretación cuyos argumentos son las interpretaciones de los términos, es decir:

$$I(f(t_1, t_2, ..., t_n)) = I_c(f)(I_v(t_1), I_v(t_2), ..., I_v(t_n))$$

- Si P es un predicado n-ario y $t_1, t_2, ..., t_n$ son términos, su interpretación es:

$$I(P(t_1, t_2, ..., t_n)) = I_c(P)(I_v(t_1), I_v(t_2), ..., I_v(t_n))$$

- Si t_1, t_2 son términos $I(t_1 = t_2)$ es verdadera si $I(t_1) = I(t_2)$. Si no es falsa.

- Si A y B son fórmulas entonces los valores de $\neg A$, $A \vee B$, $A \wedge B$, $A \leftarrow B$, $A \equiv B$ se definen como en álgebra de proposiciones.

- Para interpretar los cuantificadores introducimos nueva notación: $I_{X/d}$ es una interpretación como la $I = (D, I_c, I_v)$ pero que asigna a la variable X la interpretación d con $d \in D$. Entonces:
  - Si A es una fórmula y X la variable ligada al quantificador, se interpreta (se dice): $I(\forall A)$ = verdadero si $I_{X/d}(A)$ = verdadero para **todo** d que pertenece a D. Es falso en los demás casos.
  - Si A es una fórmula y X la variable ligada se interpreta: $I(\exists A)$ = verdadero si $I_{X/d}(A)$ = verdadero para **algún** d que pertenece a D. Es falso si no hay tal d.

Actividad: Por favor, revise cuidadosamente el ejemplo que le dará el profesor.

Consistencia formal

y otros conceptos similarmente tratados

Una fórmula se dice:

válida si es verdadera para toda interpretación.

consistente si es verdadera para alguna interpretación.

inconsistente si es falsa para toda interpretación.

contingente si es consistente pero no válida.

Actividad: Construya un ejemplo con una fórmula falsable

¿Para qué podrán servirnos estas formalidades?

Formas prenex

Para razonar con más facilidad, los lógicos suelen transformar sus fórmulas a formas bien conocidas.

Llamamos **forma prenex** a una matriz de fórmulas sin cuantificadores, precedida de un *prefijo* que es una sucesión finita de cuantificadores.

Teorema lm1.1 Para toda fórmula existe una fórmula prenex equivalente.

La forma clausal es una forma prenex.

Convirtiendo oraciones lógicas a la forma clausal

Para convertir una fórmulada dada a su forma clausal:

1 Reescriba toda subfórmula

$$\text{W1} \leftrightarrow \text{W2 como (W1} \leftarrow \text{W2)} \wedge \text{(W2} \leftarrow \text{W1)}$$

2 Reescriba toda subfórmula

$$\text{W1} \leftarrow \text{W2 como W1} \vee \neg \text{ W2}$$

3 Distribuya tanto como pueda todas las ocurrencias de $\neg$, reescribiendo

$$\neg\ (\exists X)\ \text{W como } (\forall X)\ \neg\ \text{W}$$

$$\neg\ (\forall X)\ \text{W como } (\exists X)\ \neg\ \text{W}$$

$$\neg\ (\text{ W1 } \vee \text{ W2) como } \neg \text{ W1} \wedge \neg \text{ W2}$$

$$\neg\ (\text{ W1 } \wedge \text{ W2) como } \neg \text{ W1} \vee \neg \text{ W2}$$

$$\neg\neg \text{ W como W}$$

4 Distribuya tanto como pueda todas las ocurrencias de "$\vee$ " reescribiendo

$$\text{W} \vee (\text{ W1 } \wedge \text{ W2}) \text{ como } (\text{ W } \vee \text{ W1}) \wedge (\text{ W } \vee \text{ W2})$$

$$\text{W1} \vee (\forall X)\ \text{W2 como } \forall X\ (\text{ W1 } \vee \text{ W2 })$$

$$\text{W1} \vee (\exists X)\ \text{W2 como } \exists X\ (\text{ W1 } \vee \text{ w2 })$$

5 Distribuya tanto como pueda todas las ocurrencias de "$\forall$ " reescribiendo

$$\forall X\ (\text{ W1 } \wedge \text{ W2 }) \text{ como } (\forall X)\ \text{W1} \wedge (\forall X)\ \text{W2}$$

δ: Si en este punto las oraciones *No* contienen cuantificadores existenciales el proceso está casi completo.

Si hay cuantificadores existenciales reemplace cada *fórmula cerrada* con la transformación que se conoce como *Skolemización*.

- Si se puede aplicar el paso 5 *después* de la skolemización hágalo. Para presentar la conjunción final como un conjunto de cláusulas con "cuantificación implícita", elimine todos los cuantificadores y todas las "$\wedge$ ".

Recuerde que si S es la fórmula original y C el nuevo conjunto de cláusulas,:

i C puede ser satisfecho ssi S puede ser satisfecho.

ii (*ii*) C $\models$ S.

No olvide las suposiciones iniciales respecto a los quantificadores y las variables:

Las variables y sus quantificadores están claramente asociados. Sin ambigüedades.

Ejercicios

LPO (Lógica de primer orden o FOL en inglés, es la lógica de predicados, restringida a tener variables que solamente representan objetos del universo.

En las lógicas de orden superior, las variables representan, además de objetos, relaciones. Es decir, es posible cuantificar sobre las relaciones.

Actividad: Traduzca las siguientes oraciones a LPO y luego a forma clausal, usando como predicados los nombres que se enfatizan (use la cantidad de argumentos que le parezca más conveniente):

i “Si X es *padre* o X es *madre* entonces X es el *padre* de alguien”.

ii “Si X es *progenitor* y X es del sexo *femenino* entonces X es la *madre* de alguien”.

iii “Si X es *padre* o X es *madre* entonces alguien es el *padre* de alguien ”.

iv “Si alguien es un *progenitor* entonces alguien es un *hijo*”.

v “Todas las familias *felices* son *iguales* , pero una familia infeliz es infeliz a *su* manera” (Tolstoi).

Programación lógica

Actividad: Investigue los siguientes conceptos:

- **nombres de constantes, nombres de funciones, nombres de variables, términos y predicados.**
- **Fórmulas atómicas y oraciones lógicas.**
- **Cuantificador universal y cuantificador existencial.**
- **Variables libres y ligadas (ó atadas).**
- **Literal.**
- **Término** *grounded* **(básico) y fórmula básica.**
- **Antecedentes y consecuencias.**
- **Una instancia de una fórmula.**
- **Reemplazos y sustituciones.**
- **Instanciar e instanciaciones.**

¿Cómo se les llama en inglés?

¿Qué es la programación lógica?

Es un formalismo computacional que:

- usa lógica para representar el conocimiento.
- usa inferencias lógicas para manipular ese conocimiento.

La programación lógica puede ser considerada un lenguaje pues:

1. usa una sintáxis bién definida (La forma clausal de la lógica de primer orden).
2. posee *muchas* semánticas bién definidas (e.g. Modelos mínimos).
3. usas reglas de inferencia bién conocidas y con propiedades computacionales atractivas (e.g. Resolución).

..y, entonces, ¿qué es el PROLOG?

PROLOG es un lenguaje para programación lógica. Pero la programación lógica **no** es solamente PROLOG.

$$PROLOG = \textit{forma clausal} + \textit{Resolución} + \textit{estrategia de control.}$$

Actividad: Considere el siguiente grupo de cláusulas:

```
vota( jacinto, X ) si salva_a( X, venezuela ).
salva_a( X , venezuela ) si noble( X ).
noble( madre_teresa ).
```

1. **¿Cómo podría ud. inferir que *Jacinto vota por ud.*?**
2. **¿Puede ud. inferir que *Jacinto no es noble*?**
3. **¿Qué proposiciones básicas se pueden inferir a partir de esas cláusulas?**

Las virtudes de la P.L. (según sus defensores)

1 Programación lógica ≡ Programación basada en conocimientos.

2 Permite definir con, precisión matemática:

- Los programas y las respuestas computadas con ellos.
- Los programas y sus especificaciones.
- La diferencia entre programas.

3 La programación lógica separa el que del cómo. El conocimiento se separa del cómo se le usa (ver adelante la definicion del predicado "agrega").

4 La programación lógica ofrece un paradigma uniforme para la tecnología de software :

- un programa lógico $\stackrel{?}{\equiv}$ programa convencional.
- un programa lógico ≡ especificación
- un programa lógico ≡ una base de datos (¿deductiva?).

5 La programación lógica sigue evolucionando:

- Meta programación lógica.
- Programación lógica con restricciones (CLP).
- Programación lógica concurrente y paralela (Concurrent LP).
- Programación lógica inductiva (ILP).
- Programación lógica abductiva (ALP).
- Agentes en programación lógica.

6 La programación lógica puede ser usado en otros contextos distintos a computación:

- Razonamiento legal.
- Teoría de argumentación
- y ahora, localmente, en *modelado de sistemas!*

Los defectos de programación lógica (también según sus defensores)

1. PROLOG se queda corto. (unificación sintáctica, búsqueda simple y "ciertos problemas" con la negación).
2. No hay variables globales en PROLOG.
3. PROLOG usa un único procedimiento de prueba para todos los propósitos.

Elementos de la teoría de pruebas

Si pretendemos manipular símbolos necesitamos normar todas sus tranformaciones. En el caso de las formas lógicas, a ese conjunto de reglas de transformación se les denomina teoría de pruebas. Si principal concepto es la relación de derivabilidad entre un conjunto de fórmulas, S, y alguna fórmula, s, particular.

Así se define una relación de derivabilidad:

$$\vdash = \{< S, s > \mid S \text{ es un subconjunto de L, } s \text{ pertenece a L y } s \text{ es derivable a partir de } S \text{ usando } R\ \}$$

Actividad: Investigue que son axiomas, teoremas y reglas de inferencia:

Esta es la regla de inferencia más popular:

$$\{(\ \mathbf{A} \text{ si } \mathbf{B}), \mathbf{B}\ \} \vdash \mathbf{A} \ \ \textbf{.....} \ \ \textit{modus ponens}$$

Actividad: Usela para probar algo.

Una secuencia <s_1, s_2, ..., s_n > ..... es una prueba de s_n, siempre que los s_i se puedan "encadenar" correctamente.

¿Cúal es la relación entre pruebas y argumentos?

¿Cómo caracterizamos los elementos de un argumento en español?

Lógica para resolver problemas

Si S expresa correctamente las suposiciones de un problema entonces s, su consecuencia, expresa correctamente la solución.

Con una interpretación (pretendida) **I**, podemos decir:

Si **I** es un modelo para S entonces **I** es un modelo para s.

¿Cómo podemos asegurar que el computador "entiende la interpretación (pretendida)?

Solución de problemas y prueba de teoremas

Probar teoremas (una conclusión anticipada) puede servir para resolver problemas.

Eso es siempre que el probador cumpla con ciertas propiedades ideales:

Sanidad (correctitud):

$$\textbf{Para todo S y s. S} \models \textbf{s si S} \vdash \textbf{s}$$

Completitud:

$$\textbf{Para todo S y s, S} \vdash \textbf{s si S} \models \textbf{s}$$

¿Son estas propiedades indispensables?

¿Existen sistemas inferenciales completos?

Para lógica de proposiciones, la respuesta es SI.

Para la lógica de predicados de primer orden, la respuesta es SI.

Para la lógica de predicados de segundo orden, la respuesta es NO.

Procedimiento de prueba: Un algoritmo producto del acoplamiento entre un sistema inferencial y una estrategia para aplicar las reglas de inferencia.

Implicación y validez

Para todo conjunto de oraciones S = { s_1, ..., s_n } y cualquier oración s.

$$S \models s \text{ si y solo si } S\text{-}\{s_i\} \models (s \text{ si } s_i) \text{ [para todo i]}$$

$$S \models s \text{ si y solo si } (s \text{ si } s_1 \text{ y } s_2 \text{ y .. } s_n) \text{ es válida}$$

Esta es la conexión ideal entre teoría de pruebas y teoría de modelos.

El dominio de Herbrand y los conceptos asociados en la teoría de modelos

El dominio de Herbrand, $\mathbf{H}_L$, son todos los términos básicos que pueden construirse usando los nombres de constantes y nombres de función disponibles en el alfabeto (vocabulario) de L, el lenguaje con el que se construyen los programas y las preguntas.

Ejemplo: Sea L el lenguaje con la constante 0 y las funciones s y p. El dominio de Herbrand, H_L, es:

$$\{\ 0,\ s(0),\ p(0),\ s(s(0)),\ s(p(0)),\ p(s(0)),\ p(p(0)),\ ...\}$$

La base de Herbrand

La base de Herbrand de L es el conjunto de todos los átomos básicos que pueden escribirse en L.

Normalmente se habla de la base de Herbrand de un programa lógico dado P, B(P), restringiendo el lenguaje a los predicados que aparecen en P.

Actividad: Note que los términos no se restringen de igual manera.

Si HL = {tu, yo, este_curso**} y P es:**

> le_agrada(yo, Cualquiera) si asiste(Cualquiera, este_curso).
> asiste(Cualquiera, este_curso) si astuto(Cualquiera).
> astuto(tu).

¿Cómo es B(P)?

Sistemas inferenciales para lógica clausal

La instanciación básica de P, G(P), es el conjunto de todas las instancias básicas de todas las cláusulas de P. Sea P:

entiende(bob, X) si entiende(X, logica).
entiende(jacinto, logica).

con H = {bob, jacinto, logica}.

Tenemos que G(P) es:

entiende(bob, bob) si entiende(bob, logica).
entiende(bob, jacinto) si entiende(jacinto, logica).
entiende(bob, logica) si entiende(logica, logica).
entiende(jacinto, logica).

Actividad: Investigue (recuerde) estos conceptos importantes: transportación de literales, modus ponens, modus tollens.

Resolución en lógica proposicional

Un paso de resolución

$A \vee literales$	$\neg A \vee otros$	Padres
$\searrow$	$\swarrow$	
$literales \vee otros$		Resolvente

Propiedades de resolución:

1.- Es una regla sana(correcta): Todo resolvente es implicado por sus padres.

2.- La cláusula vacía sólo se puede obtener de cláusulas unitarias.

3.- Para cláusulas de Horn, la cláusula vacía es derivable si el conjunto de cláusulas no puede ser satisfecho.

(Resolución es completa para refutación) .

$P \models A$ ssi $P \cup \{\neg$ A$\}$ no puede ser satisfecho.

$P \models A$ ssi $P \cup \{\neg$ A$\} \vdash$ la cláusula vacía usando resolución.

Sustituciones

Una substitución es un conjunto finito de reemplazos, cada uno de los cuales toma la forma U/t, donde U es una variable y t es un término:

$$\sigma = \{U_1/t_1, U_2/t_2, ..., U_m/t_m\}$$

Propiedad de funcionalidad: Los Ui son diferentes.

Propiedad de idempotencia: $W\sigma = (W\sigma)\sigma$

Composición de sustituciones

$$\sigma_1 = \{U_1/t_1, U_2/t_2, ..., U_m/t_m\}$$

$$\sigma_2 = \{V_1/s_1, V_2/s_2, ..., V_n/s_n\}$$

$$\sigma_1 \circ \sigma_2 = \{U_1/t_1\sigma_2, ..., U_m/t_m\sigma_2\} \cup \{V_j/s_j | V_j \notin \{U_1, ..U_m\}\}$$

Resolución en lógica de predicados

$$(\forall X)W(X) \vdash (\forall Y_o)....(\forall Y_m)W(X)\{X/t\}$$

donde t es cualquier término y las Y_o......Y_m son las variables, si existen, del término t.

Esta es la regla de instanciación universal.

Resolución en programación lógica

El programa lógico.

C_1 : `agrega( nulo, W,W )`

C_2 : `agrega( U.X, Y, U.Z ) si agrega( X, Y, Z ).`

La pregunta

$$? \ agrega \ (\ t_1 \ , \ t_2 \ , \ t_3 \ )$$

$$\neg\exists \ agrega \ (\ t_1 \ , t_2 \ , \ t_3 \ )$$

$$\forall \neg \ agrega \ (\ t_1 \ , \ t_2 \ , \ t_3 \ )$$

Q_1 : ? agrega(V.L_1, L_2, V.a.nulo).

Actividad: No olvide renombrar las variables en las cláusulas resolventes al usarlas.

$Q_1: \quad ?$

$agrega(V.L_1, L_2, V.a.nulo)$

$\downarrow \quad \theta_1 = \left\{ \begin{array}{c} \mathrm{U_1/V,\ X_1/L_1,} \\ \mathrm{Y_1/L_2,\ Z_1/a\ .nulo} \end{array} \right\}$

$Q_2 \quad : \quad ?$

$agrega(L_1, L_2, a.nulo)$

$\swarrow \searrow$

$\theta_2 = \left\{ \begin{array}{c} \mathrm{L_1/nulo,} \\ \mathrm{L_2/a.nulo} \end{array} \right\}$

$\square$

$\theta_2 = \left\{ \begin{array}{c} \mathrm{L_1/U_2.X_2,\ Y_2/L_2,} \\ \mathrm{U_2/a,\ Z_2/nulo} \end{array} \right\}$

$Q_3 \quad : \quad ?$

$agrega(X_2, L_2, nulo)$

$\downarrow \quad \theta_4 = \left\{ \begin{array}{c} \mathrm{X_2/nulo\ ,} \\ \mathrm{L_2/nulo} \end{array} \right\}$

$\square$

Computaciones y respuestas

Una de las virtudes de la formalización es la posibilidad de definir con precisión conceptos claves como COMPUTACION y RESPUESTA.

Una derivación que comienza con la pregunta raíz es una computación a partir de esa pregunta.

Un árbol de derivación puede contener computaciones finitas e infinitas (ciclos).

Las computaciones finitas son de dos tipos: computaciones exitosas ($\Box$) y computación con falla finita ($\blacksquare$).

Una pregunta raíz tiene la forma $\neg\exists(A_1 \wedge ... \wedge A_n)$

si logramos refutarla tenemos, por lo menos, $P \models \exists((A_1 \wedge ... \wedge A_n)\theta^*)$

Existe un subconjunto significativo de θ^* que contribuye con los valores finales de las variables en la pregunta y que se denomina la: ***sustitución respuesta computada***.

Ese el mínimo subconjunto de θ^* que satisface:

$$((A_1 \wedge ... \wedge A_n)\Phi\theta^*) = (A_1 \wedge ... \wedge A_n)\Phi$$

Como ven, en programación lógica establecemos, con precisión matemática, que es una respuesta o salida de un programa.

Finalmente..

Actividad: Para su uso personal, prepare un resumen de la unidad y consulte con el profesor cualquier duda. Los comentarios también son bienvenidos.

Actividad: Revise las bitácoras cuando realice el resumen o cuandoquiera aclarar estas láminas.

Nuestro agradecimiento al Prof. Carlos Domingo quien cedió material para las primeras láminas de esta unidad.