Paradigmas de Programación

Programación Lógica: Lógica Proposicional

Universidad Tecnológica Nacional
Facultad Regional Resistencia

UTN
FRRe

Contenidos

- Introducción
  - Problema
  - Sistemas de Producción
  - Búsqueda
- ***Programación con Lógica Proposicional***
  - ***Sintaxis***
  - ***Consecuencia lógica***
  - ***Resolución***
  - ***Refutación y Deducción***
  - ***Árbol de Resolución***
  - ***Negación***
  - ***Resolución SLD***
- Programación con Lógica de Predicados
  - Sintaxis
  - Cláusulas
  - Unificación
  - Resolución
- ProLog
  - Ejemplos

Lógica. Definición.

- En la **Lógica Formal** se estudian los principios y métodos a través de los cuales podemos determinar la *validez de argumentos, desde el punto de vista solamente de su estructura*, sin tomar en cuenta el contenido semántico de las expresiones de los argumentos.

- Se puede decir que la Lógica es una herramienta para el análisis de la veracidad de argumentos en base sólo a la estructura correcta del razonamiento, su validez lógica, más que la verdad o falsedad de las proposiciones que la componen.

- El análisis de la veracidad de argumentos se realiza mediante ciertos axiomas básicos y ciertas reglas de deducción.

Lógica II

Todos los hombres son mortales

Todos los investigadores son hombres

Todos los investigadores son mortales

Argentina está en África o Argentina está en Asia.

Argentina no está en Asia

En consecuencia, Argentina está en África.

Lógica III

Si el tren llega tarde y no hay taxis en la estación, entonces Juan se retrasará a su cita. Juan no está retrasado para su cita. El tren llegó tarde. Por lo tanto hay taxis en la estación.

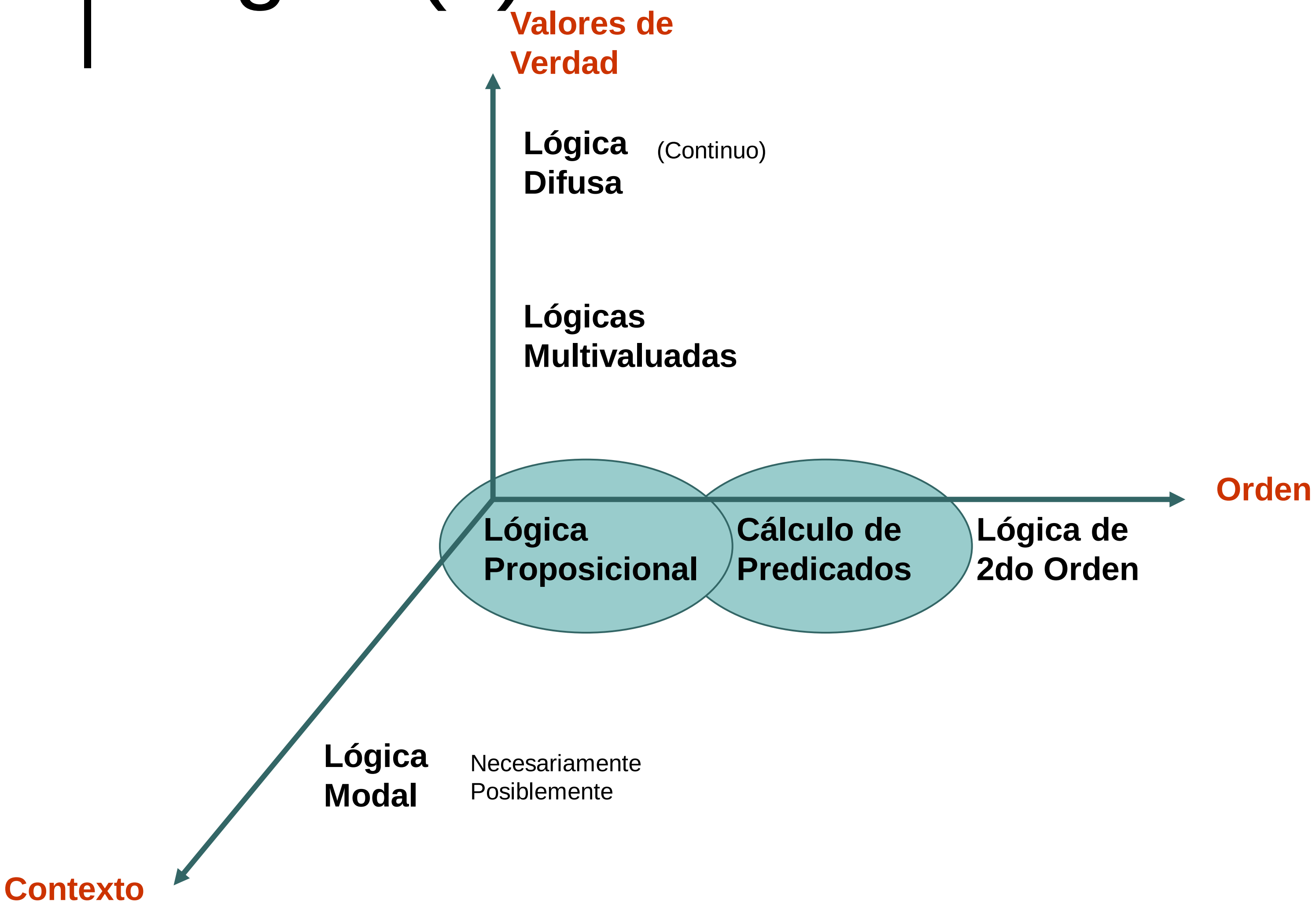
Si está lloviendo y María no tiene su paraguas, entonces se mojará. María no está mojada. Está lloviendo. Por lo tanto María tiene su paraguas.

p
q
r

p
q
r

Si p y no q, entonces r. No r. p. Por lo tanto, q.

Lógica(s)


Valores de
Verdad
Lógica
Difusa
(Continuo)
Lógicas
Multivaluadas
Orden
Lógica
Proposicional
Cálculo de
Predicados
Lógica de
2do Orden
Lógica
Modal
Necesariamente
Posiblemente
Contexto

Proposiciones

- Las proposiciones son expresiones que pueden evaluarse a V o F.
- Son expresiones declarativas y nunca interrogativas o imperativas.

Son Proposiciones

- Hay vida en la tierra.
- La suma entre 2 y 5 es 7.
- Todos los hombres de Colombia son solteros.
- Todo número natural mayor que 2 es la suma de dos números primos.
- Los cantantes no duermen.
- Comer mucho, engorda.
- El hombre desciende del elefante.

No Son Proposiciones

- Puedes pasarme la sal?
- Teclea exit
- Arriba Argentina!
- ¡Levántate temprano!
- ¿Has entendido lo que es una proposición?
- ¡Estudia esta lección!
- Preparados, listos, ya!

Proposiciones

- Estudiaremos los principios para determinar la validez de argumentos conformados con proposiciones.

  - **Proposiciones simples o átomos:** **p**, **q**

  - **Proposiciones compuestas:** formadas por proposiciones simples a través de conectores lógicos. p or q, r and s

Conectivos Lógicos

Operadores unarios:

NEGACION: not, $\neg$

Operadores binarios:

CONJUNCION: and, $\wedge$

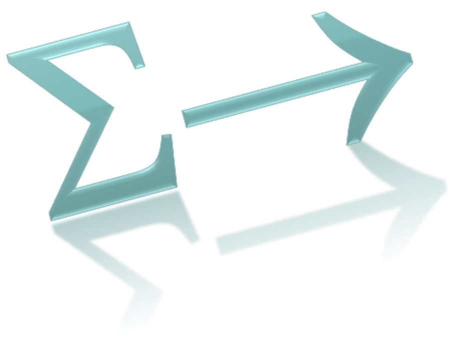
DISYUNCION: or, $\vee$

CONDICIONAL: implicación, $\Rightarrow$

BICONDICIONAL: doble implicación $\Leftrightarrow$

Formulas Bien Formadas

- Una fórmula bien formada (fbf) es una expresión que representa una proposición simple o compuesta, la cual esta bien escrita de acuerdo con determinada sintaxis.



Sintaxis

$$\text{fbf} \rightarrow \text{atomo}$$
$$\text{fbf} \rightarrow \neg\text{fbf}$$
$$\text{fbf} \rightarrow \text{fbf} \wedge \text{fbf}$$
$$\text{fbf} \rightarrow \text{fbf} \vee \text{fbf}$$
$$\text{fbf} \rightarrow \text{fbf} \Rightarrow \text{fbf}$$
$$\text{fbf} \rightarrow \text{fbf} \Leftrightarrow \text{fbf}$$

Interpretación

- Asignación de valores de verdad para las proposiciones de una expresión.

- El significado de una formula proposicional se puede expresar por medio de una función:

  ϖ: prop $\rightarrow$ {verdadero, falso}

  La función ϖ es una *función de interpretación* que satisface:

p	q	¬p	p ∧ q	p ∨ q	p ⇒ q	p ⇔ 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

- ***Ejemplo****: Juan llora y Florencia se ríe.*

- *Para n proposiciones atómicas distintas, existen 2^n interpretaciones posibles.*

Interpretación 2

- Una fórmula se dice **válida** si es verdadera bajo cualquier interpretación (tautología).
- Una fórmula es **inválida** si no es válida.
- Una fórmula es **insatisfascible** o inconsistente si es falsa bajo cualquier interpretación (contradicción), sino es **satisfascible** o consistente.

Válido	**Inválido**	
Siempre verdadero	A veces V o F	Siempre falso
Satisfacible		**Insatisfacible**

Leyes de Equivalencia

Dadas dos fbf F y G, son equivalentes si sus tablas de verdad tienen los mismos valores de verdad:

- $\neg(\neg p) \equiv p$
- $p \wedge q \equiv q \wedge p$
- $p \vee q \equiv q \vee p$

- $(p \wedge q) \wedge r \equiv q \wedge (p \wedge r)$
- $(p \vee q) \vee r \equiv q \vee (p \vee r)$

- $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

- $p \Leftrightarrow q \equiv p \Rightarrow q \wedge q \Rightarrow p$

- $p \Rightarrow q \equiv \neg p \vee q$

- $\neg(p \wedge q) \equiv \neg p \vee \neg q$
- $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Reglas de Inferencia

- Las reglas de inferencia dan la posibilidad de hallar la verdad de proposiciones a partir de otras.

- La regla más famosa es el MODUS PONENS.

$$\{p \rightarrow q, p\} \Rightarrow q$$

Consistencia Lógica

- Es una propiedad de un conjunto de axiomas.
- Se dice que un conjunto de axiomas es consistente si a partir de él no puede deducirse simultáneamente una proposición (p) y su contraria ($\neg p$).

Consecuencia Lógica

- Dadas las fórmulas $F_1, \ldots, F_n$ y una fórmula G, se dice que G es una consecuencia lógica de $F_1, \ldots, F_n$, si y solo si para cualquier interpretación I, en la cual $F_1, \ldots, F_n$ es verdadera, G también lo es.

Pruebas

Dadas las fórmulas $F_1, F_2, \ldots, F_n$ y una fórmula G, G es una consecuencia lógica de $F_1, F_2, \ldots, F_n$, si y solo si:

Método Directo

$(F_1 \wedge F_2 \wedge \ldots \wedge F_n) \Rightarrow G$ es válida (Tautología).

Método Indirecto

$(F_1 \wedge F_2 \wedge \ldots \wedge F_n) \wedge \neg G$ es insatisfacible (Contradicción).

Definiciones

- **Literal Proposicional**

  Es una variable proposicional o la negación de una variable proposicional.

  Un *literal proposicional positivo* es simplemente una variable proposicional,

  Un *literal proposicional negativo* es la negación de una variable proposicional.

- **Cláusula proposicional**

  Es una disyunción de literales proposicionales.

  Es una formula proposicional donde el número de literales proposicionales están conectados con el operador $\vee$.

Cláusula de Horn

- **Cláusula Proposicional de Horn**

  Es una cláusula proposicional con un literal positivo como máximo:

  - $\mathbf{q}$ (clausula unidad)
  - $\neg \mathbf{p_1} \vee \ldots \vee \neg \mathbf{p_n} \vee \mathbf{q}$

  $\left.\right\}$ cláusulas de programa

  - $\neg \mathbf{p_1} \vee \ldots \vee \neg \mathbf{p_n}$ cláusula meta

  - donde $p_1 \ldots p_n$, q son variables proposicionales.

Deducción y Contradicción

Refutación - Reducción al Absurdo

Dado un programa lógico P y una consulta Q, puede establecerse la deducción

$P \vdash_{Res} Q$?

Reducción al absurdo.

Para demostrar la existencia de algo (Q) a partir de un programa P:

- *Asumimos lo contrario* ($\neg Q$)
- *Se usa un proceso de deducción*
- *Si eso resulta un absurdo, se prueba Q.*

Deducción y Contradicción

Refutación - Reducción al Absurdo

1. Dado un Programa P. (Conjunto de cláusulas de programa).

2. Agregaremos $\neg Q$ como una hipótesis y usaremos resolución para establecer una contradicción en la forma de cláusula vacía.
   - Una consulta (query) es una conjunción de variables proposicionales y tiene la siguiente forma:

   $q_1 \wedge \ldots \wedge q_n$ cuya negación es: $\neg (q_1 \wedge \ldots \wedge q_n)$ ó $\neg q_1 \vee \ldots \vee \neg q_n$

3. Usando resolución, la única manera de obtener la cláusula vacía, es aplicando la regla de resolución a la variable proposicional p y $\neg p$, en la cual tenemos una contradicción.
   - La cláusula vacía es denotado por $\square$.

Regla de Resolución
(Robinson)

Es una **regla de inferencia** que se utiliza en la programación lógica para realizar deducciones (es completa):

$$C_1 \vee p, \quad C_2 \vee \neg p \quad \vdash \quad C_1 \vee C_2$$

Tomemos por ejemplo una cláusula C1 de tipo (3) y C2 de tipo (2):

$$C1 = \neg q_1 \vee \ldots \vee \neg q_n$$
$$C2 = \neg r_1 \vee \ldots \vee \neg r_m \vee s$$

Podemos tener dos cláusulas:

$$C1 \vee p$$
$$C2 \vee \neg p$$

Podemos escribirlas como:

$$(q_1 \wedge \ldots \wedge q_n) \Rightarrow p$$
$$(r_1 \wedge \ldots \wedge r_m \wedge p) \Rightarrow s$$

Lo que significa:

$$(q_1 \wedge \ldots \wedge q_n \wedge r_1 \wedge \ldots \wedge r_m) \Rightarrow s$$
$$\neg q_1 \vee \ldots \vee \neg q_n \vee \neg r_1 \neg \vee \ldots \vee \neg r_m \vee s$$
$$C1 \vee C2$$

Mundo Cerrado (closed-world assumption)

Consideremos los siguientes ejemplos:

Un señor se encuentra con dos señoritas en la parada de colectivos y le dice a una de ellas: “Usted es muy bonita”

La otra mujer tiene derecho a sentirse “Menos bonita”?

En la clase de Paradigmas, el profesor dice: “Todos los Jueves hay examen”.

Significa esto que ningún otro día habrá examen?

Ejemplo 1

Dado el Programa:

- $\neg p \vee \neg q \vee r$
- p
- q

Es posible deducir r?.

Para ello agregamos la consulta $\neg r$ como hipótesis agregada.

1.	$\neg p \vee \neg q \vee r$	
2.	p	Hipótesis
3.	q	
4.	$\neg r$	Hipótesis agr.
5.	$\neg p \vee \neg q$	Res por 4 y 1
6.	$\neg q$	Res por 5 y 2
7.	$\square$	Res por 6 y 3

Ejemplo 2

Dado el programa P. Podemos deducir **s** del mismo?

$(p \wedge q \wedge r) \Rightarrow s$	$\neg p \vee \neg q \vee \neg r \vee s$
$(t \wedge w) \Rightarrow r$	$\neg t \vee \neg w \vee r$
q	q
$(v \wedge r) \Rightarrow p$	$\neg v \vee \neg r \vee p$
t	t
v	v
$v \Rightarrow w$	$\neg v \vee w$

Ejemplo 2

1.	$\neg p \vee \neg q \vee \neg r \vee s$	
2.	$\neg t \vee \neg w \vee r$	
3.	q	
4.	$\neg v \vee \neg r \vee p$	Hipótesis
5.	t	
6.	v	
7.	$\neg v \vee w$	
8.	$\neg s$	Hipótesis agregada
9.	$\neg p \vee \neg q \vee \neg r$	Res por 1 y 8
10.	$\neg p \vee \neg q \vee \neg t \vee \neg w$	Res por 2 y 9
11.	$\neg p \vee \neg t \vee \neg w$	Res por 3 y 10
12.	$\neg t \vee \neg w \vee \neg v \vee \neg r$	Res por 4 y 11
13.	$\neg w \vee \neg v \vee \neg r$	Res por 5 y 12
14.	$\neg w \vee \neg r$	Res por 6 y 13
15.	$\neg v \vee \neg r$	Res por 7 y 14
16.	$\neg t \vee \neg w \vee \neg v$	Res por 2 y 15
17.	$\neg w \vee \neg v$	Res por 5 y 16
18.	$\neg v$	Res por 7 y 17
19.	$\square$	Res por 6 y 18

Árbol de Resolución

- Cada literal positivo de la consulta se considera como una submeta, las cuales son derivadas en turno.
- Cada submeta coincidirá con una o más cláusulas de programa.
- La cola de cada cláusula elegida nos da mas submetas para derivar

a) $(p \wedge q \wedge r) \Rightarrow s$

b) $(t \wedge w) \Rightarrow r$

c) q

d) $(v \wedge r) \Rightarrow p$

e) t

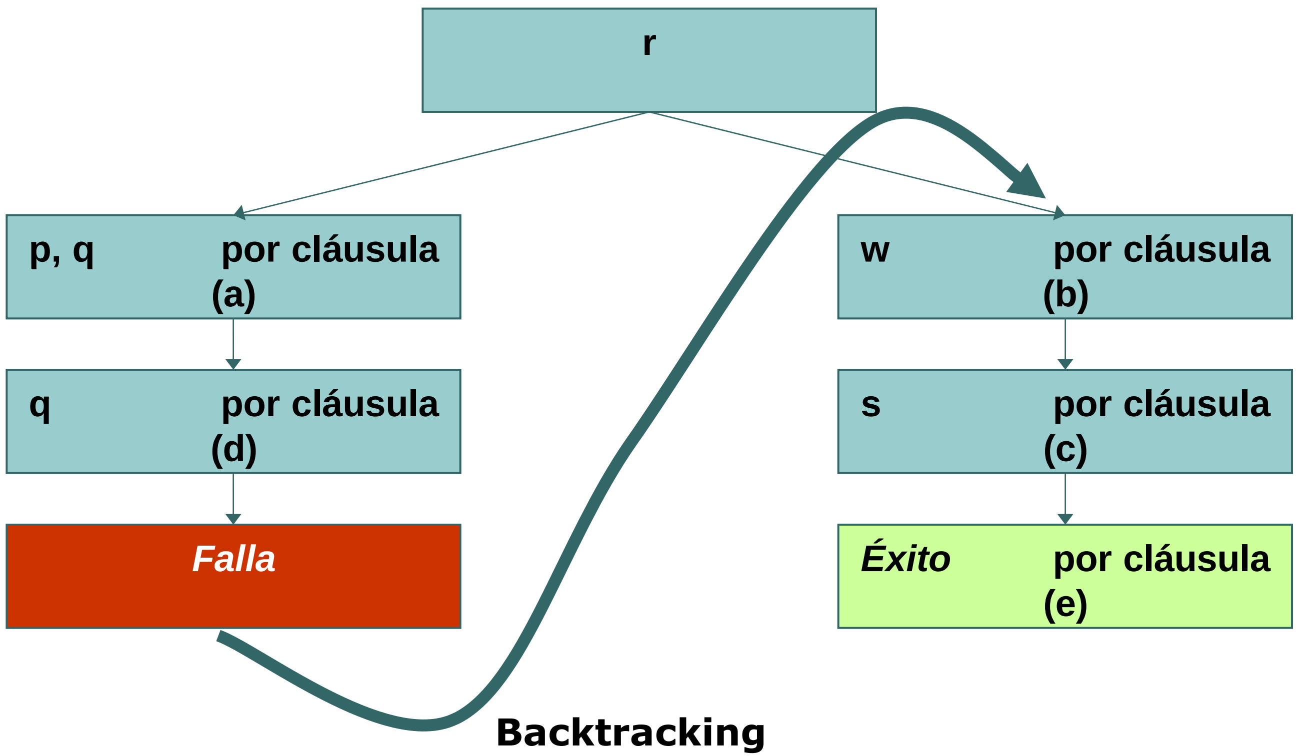
f) v

g) $v \Rightarrow w$

- s
- p, q, r — por cláusula (a)
- p, q, t, w — por cláusula (b)
- p, t, w — por cláusula (c)
- v, r, t, w — por cláusula (d)
- v, r, w — por cláusula (e)
- r, w — por cláusula (f)
- r, v — por cláusula (g)
- t, w, v — por cláusula (b)
- w, v — por cláusula (e)
- v, v — por cláusula (g)
- v — por cláusula (f)
- *Éxito* — por cláusula (f)

Árbol de Resolución

a) $(p \wedge q) \Rightarrow r$
b) $w \Rightarrow r$
c) $s \Rightarrow w$
d) p
e) s



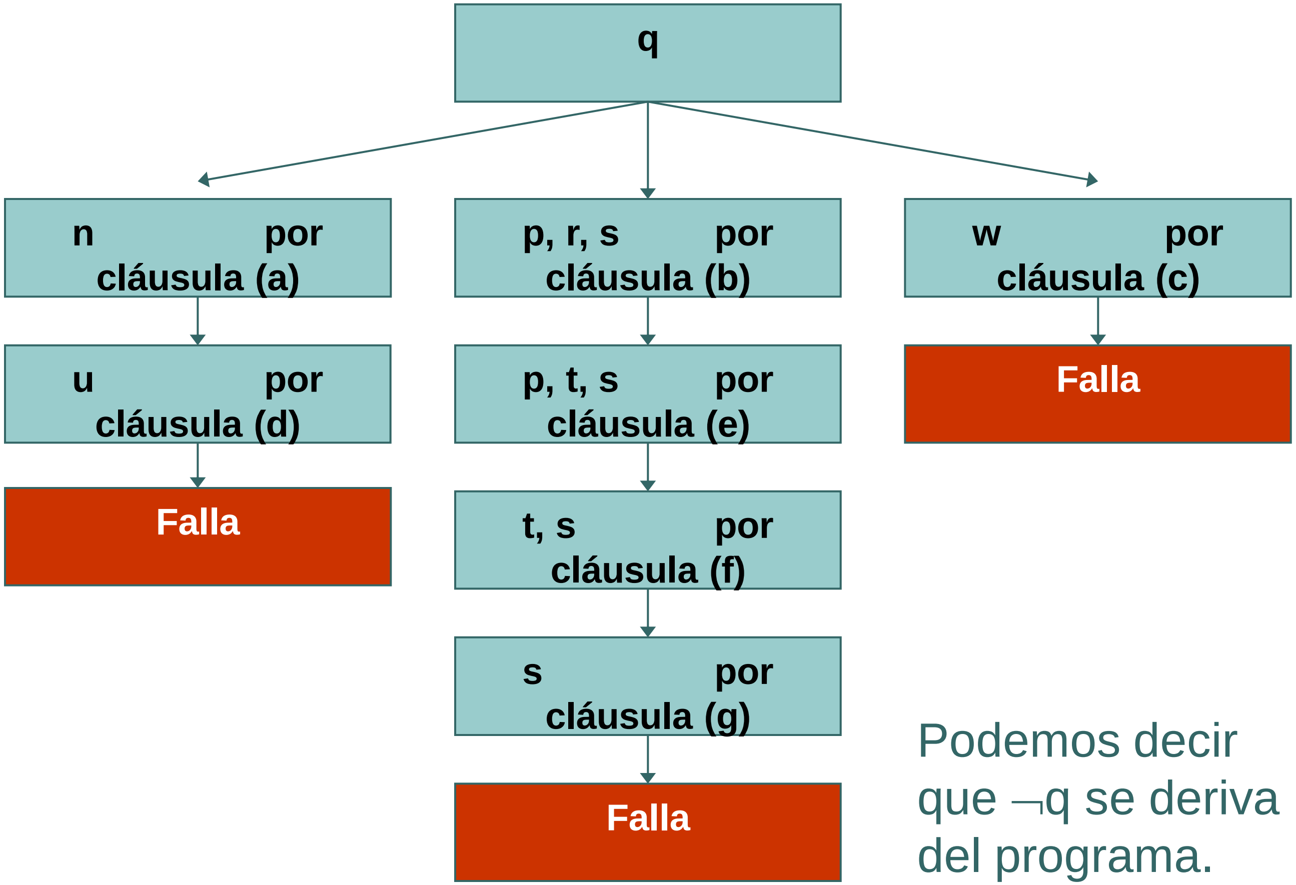
Negación en Programación Lógica

- Como vimos, los programas lógicos solo expresan información positiva.
- No es posible mostrar que un literal negativo es consecuencia de un programa lógico.
- Consideraremos cualquier consulta como falsa si no puede derivarse del programa.
- Esto se conoce con el nombre de *Negación por falla* o por *mundo cerrado*.
- Una sentencia que no fue considerada, puede no necesariamente ser falsa aunque la técnica la considere falsa.

"Si P es un programa proposicional, q es una consulta y $P \vdash q$ *no puede establecerse, podemos deducir que* $P \vdash \neg q$*."*

Negación en Programación Lógica: Ejemplo

a) $n \Rightarrow q$
b) $(p \wedge r \wedge s) \Rightarrow q$
c) $w \Rightarrow q$
d) $u \Rightarrow n$
e) $t \Rightarrow r$
f) p
g) t



Podemos decir que $\neg q$ se deriva del programa.

Resolución SLD

- El uso de la función de selección con resolución se denomina Resolución SLD (Selective Linear for Definite clauses)
  - **Seleccction Rule**
  - **Linear Resolution**
  - **Definite Clauses (Horn clauses)**
- Es una estrategia de resolución en la que se establece qué cláusula y qué literal serán seleccionados.
  - **La regla de selección** determina qué fórmula atómica del objetivo utilizar a la hora de generar un nuevo nivel en el árbol SLD. (izq – der, der – izq.)
  - **La regla de búsqueda** determina cómo se seleccionan las cláusulas que unifican con una fórmula atómica dada. (arr – abj, abj – arr)

Resolución SLD: Ejemplo

1. $(p \wedge r \wedge s) \Rightarrow q$
2. $(p \wedge w) \Rightarrow q$
3. $w \Rightarrow n$
4. $(p \wedge t) \Rightarrow n$
5. $s \Rightarrow w$
6. $t \Rightarrow r$
7. p
8. t
9. s