Inteligencia Artificial

Lógica Proposicional
y
Razonamiento Automático

Edgar Andrade, Ph.D.

Matemáticas Aplicadas y Ciencias de la computación

Última revisión: Febrero de 2023

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Contenido

Contenido

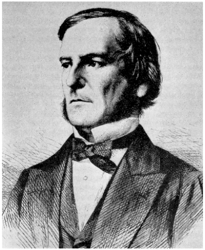
Influencias históricas (1/5)

Gottfried Leibniz
(1646–1716)
☞ Calculus Ratiocinator

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

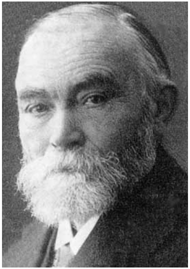
Influencias históricas (1/5)

Gottfried Leibniz
(1646–1716)
☞ Calculus Ratiocinator



George Boole
(1815–1864)
☞ Las facultades mentales son operaciones simbólicas

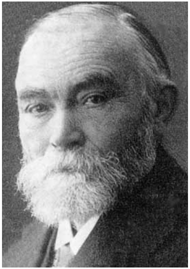
Influencias históricas (2/5)



Gotlob Frege
(1848–1925)

☞ Las matemáticas se fundamentan en la lógica

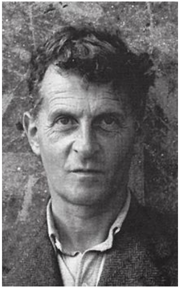
Influencias históricas (2/5)



Gotlob Frege
(1848–1925)
☞ Las matemáticas se fundamentan en la lógica

David Hilbert
(1862–1943)
☞ Las matemáticas y la manipulación de símbolos

Influencias históricas (3/5)

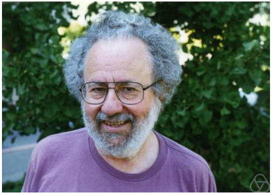


Ludwig Wittgenstein (1889–1951)

Emile Post (1897–1954)

☞ Procedimiento de las tablas de verdad

Influencias históricas (4/5)

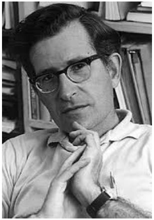


Martin Davis
(1928–)

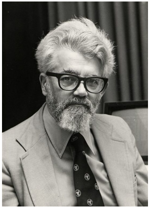
Hilary Putnam
(1926–2016)

☞ Algoritmo eficiente para buscar un modelo

Influencias históricas (5/5)



Noam Chomsky
(1928–)



John McCarthy
(1927–2011)

☞ Formalización del lenguaje y del sentido común

MACC Matemáticas Aplicadas y Ciencias de la Computación

Contenido

El lenguaje como representación

Nairo

monta

bicicleta

Las palabras se relacionan con aspectos del mundo y su combinación representa situaciones.

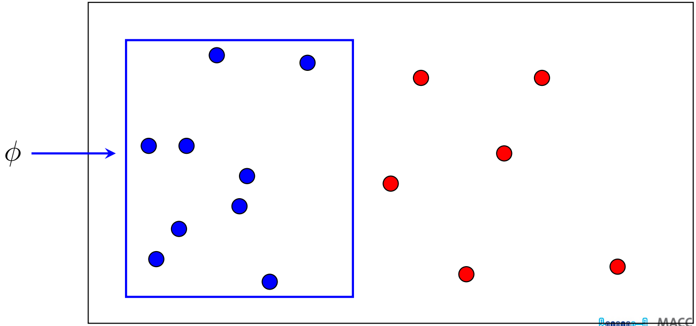
MACC
Matemáticas Aplicadas y
Ciencias de la Computación

El lenguaje como restricciones

Situaciones en las que el agente afirma que "Nairo monta bicicleta".

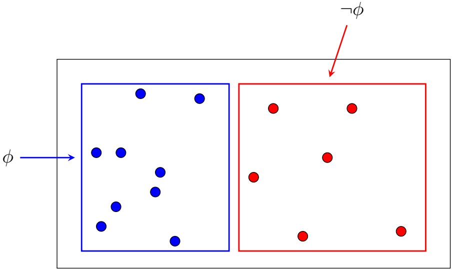
Proposiciones

Una proposición ϕ puede verse como una colección de situaciones: aquellas en las que el agente afirma que ϕ.



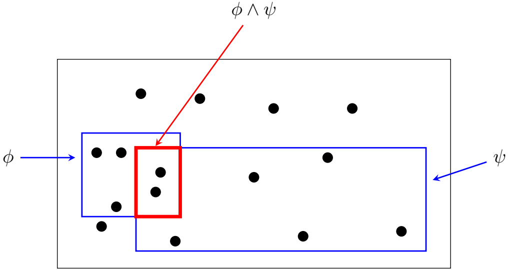
Constantes lógicas

- Negación: $\neg$



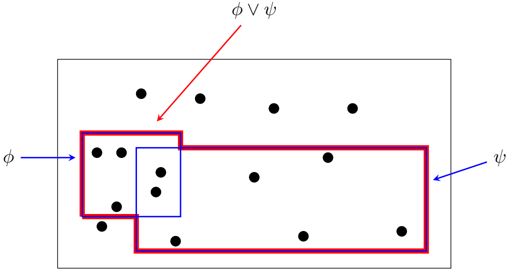
Constantes lógicas

- Negación: $\neg$
- Conjunción: $\wedge$



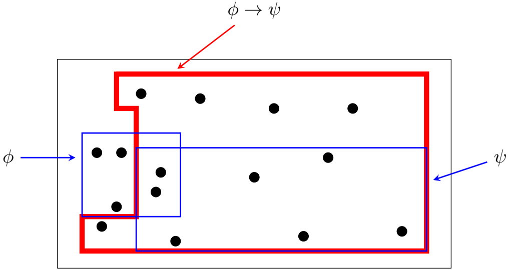
Constantes lógicas

- Negación: $\neg$
- Conjunción: $\wedge$
- Disyunción: $\vee$



Constantes lógicas

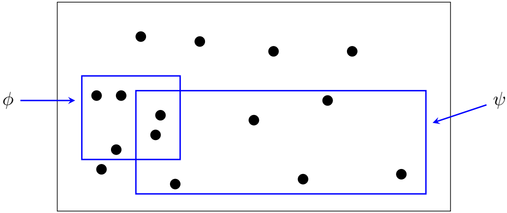
- Negación: $\neg$
- Conjunción: $\wedge$
- Disyunción: $\vee$
- Implicación: $\rightarrow$



MACC
Matemáticas Aplicadas y Ciencias de la Computación

Constantes lógicas

- Negación: $\neg$
- Conjunción: $\wedge$
- Disyunción: $\vee$
- Implicación: $\rightarrow$
- Doble implicación: $\leftrightarrow$



¿Cómo se visualiza $\phi \leftrightarrow \psi$?

Ejercicios de representación (1/2)

- Tanto los liberales como los socialistas favorecieron la moción.

Ejercicios de representación (1/2)

- Tanto los liberales como los socialistas favorecieron la moción.

  Respuesta: $p \wedge q$

p : Los liberales favorecieron la moción.

q : Los socialistas favorecieron la moción.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejercicios de representación (1/2)

- Tanto los liberales como los socialistas favorecieron la moción.

  Respuesta: $p \wedge q$

p : Los liberales favorecieron la moción.

q : Los socialistas favorecieron la moción.

- Juan está de mal humor sólo si se acaba de levantar.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejercicios de representación (1/2)

- Tanto los liberales como los socialistas favorecieron la moción.

  Respuesta: $p \wedge q$

p : Los liberales favorecieron la moción.

q : Los socialistas favorecieron la moción.

- Juan está de mal humor sólo si se acaba de levantar.

  Respuesta: $q \rightarrow p$

p : Juan se acaba de levantar.

q : Juan está de mal humor.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejercicios de representación (2/2)

- Para que el partido funcione mejor, es suficiente que Smith sea excluido.

Ejercicios de representación (2/2)

- Para que el partido funcione mejor, es suficiente que Smith sea excluido.

  Respuesta: $p \to q$

p : Smith es excluido del partido.

q : El partido funciona mejor.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejercicios de representación (2/2)

- Para que el partido funcione mejor, es suficiente que Smith sea excluido.

  Respuesta: $p \rightarrow q$

p : Smith es excluido del partido.

q : El partido funciona mejor.

- No es el caso que Jorge venga si Pedro o Jaime vienen.

Ejercicios de representación (2/2)

- Para que el partido funcione mejor, es suficiente que Smith sea excluido.

  Respuesta: $p \to q$

p : Smith es excluido del partido.

q : El partido funciona mejor.

- No es el caso que Jorge venga si Pedro o Jaime vienen.

  Respuesta: $(p \vee q) \to \neg r$

p : Pedro viene.

q : Jaime viene.

r : Jorge viene.

Contenido

Fórmulas

Gramática

$$
\begin{aligned}
S &\rightarrow Atomica \mid Compleja \\
Atomica &\rightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\rightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Fórmulas

Gramática

$$
\begin{aligned}
S &\rightarrow Atomica \mid Compleja \\
Atomica &\rightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\rightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \Rightarrow Compleja$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow Atomica \mid Compleja \\
Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\longrightarrow \neg S \mid (S \land S) \mid (S \lor S) \mid (S \to S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg S$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow Atomica \mid Compleja \\
Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg Compleja$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow Atomica \mid Compleja \\
Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg(S \rightarrow S)$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow Atomica \mid Compleja \\
Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg(Compleja \rightarrow S)$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\rightarrow Atomica \mid Compleja \\
Atomica &\rightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\rightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg((S \wedge S) \rightarrow S)$$

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow \mathit{Atomica} \mid \mathit{Compleja} \\
\mathit{Atomica} &\longrightarrow \mathit{verdadero} \mid \mathit{falso} \mid p \mid q \mid r \mid \ldots \\
\mathit{Compleja} &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg((\mathit{Atomica} \wedge S) \rightarrow S)$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow Atomica \mid Compleja \\
Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \stackrel{*}{\Rightarrow} \neg((p \wedge S) \rightarrow S)$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Fórmulas

Gramática

$$\begin{aligned} S &\longrightarrow Atomica \mid Compleja \\ Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\ Compleja &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S) \end{aligned}$$

Ejemplo

$$S \stackrel{*}{\Rightarrow} \neg((p \wedge Atomica) \rightarrow S)$$

MACC Matemáticas Aplicadas y Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow \mathit{Atomica} \mid \mathit{Compleja} \\
\mathit{Atomica} &\longrightarrow \mathit{verdadero} \mid \mathit{falso} \mid p \mid q \mid r \mid \ldots \\
\mathit{Compleja} &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \to S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \overset{*}{\Rightarrow} \neg((p \wedge q) \to S)$$

Fórmulas

Gramática

$$
\begin{aligned}
S &\longrightarrow Atomica \mid Compleja \\
Atomica &\longrightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\longrightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \stackrel{*}{\Rightarrow} \neg((p \wedge q) \rightarrow Atomica)$$

MACC Matemáticas Aplicadas y Ciencias de la Computación

Fórmulas

Gramática

$$
\begin{aligned}
S &\rightarrow Atomica \mid Compleja \\
Atomica &\rightarrow verdadero \mid falso \mid p \mid q \mid r \mid \ldots \\
Compleja &\rightarrow \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \rightarrow S) \mid (S \leftrightarrow S)
\end{aligned}
$$

Ejemplo

$$S \stackrel{*}{\Rightarrow} \neg((p \wedge q) \rightarrow r)$$

MACC Matemáticas Aplicadas y Ciencias de la Computación

Uso de paréntesis

Economía de paréntesis

1. Omitiremos paréntesis exteriores.
2. Omitiremos paréntesis según la precedencia: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
3. Múltiples $\wedge$ y $\vee$ sin paréntesis.

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Uso de paréntesis

Economía de paréntesis

1. Omitiremos paréntesis exteriores.
2. Omitiremos paréntesis según la precedencia: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
3. Múltiples $\wedge$ y $\vee$ sin paréntesis.

Ejemplos

$(p \rightarrow r)$	☞	$p \rightarrow r$
$\neg((p \wedge q) \rightarrow r)$	☞	$\neg(p \wedge q \rightarrow r)$
$((p \wedge q) \wedge r)$	☞	$p \wedge q \wedge r$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Contenido

Objetivo de la lógica

☞ La lógica estudia las deducciones (o razonamientos deductivos).

Objetivo de la lógica

☞ La lógica estudia las deducciones (o razonamientos deductivos).

Una deducción es un discurso que lleva de unas premisas a una conclusión.

$$\frac{\text{Premisa}_1, \ldots, \text{Premisa}_n}{\text{Conclusión}}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Deducciones

La deducción es una lista de pasos. Cada paso debe estar justificado por una razón.

Deducciones

La deducción es una lista de pasos. Cada paso debe estar justificado por una razón.

1. Fórmula 1 Razón

Deducciones

La deducción es una lista de pasos. Cada paso debe estar justificado por una razón.

1.	Fórmula 1	Razón
2.	Fórmula 2	Razón

Deducciones

La deducción es una lista de pasos. Cada paso debe estar justificado por una razón.

1.	Fórmula 1	Razón
2.	Fórmula 2	Razón
	$\vdots$	$\vdots$

MACC Matemáticas Aplicadas y Ciencias de la Computación

Deducciones

La deducción es una lista de pasos. Cada paso debe estar justificado por una razón.

1.	Fórmula 1	Razón
2.	Fórmula 2	Razón
	⋮	⋮
n.	Conclusión	Razón

MACC Matemáticas Aplicadas y Ciencias de la Computación

Deducciones

La deducción es una lista de pasos. Cada paso debe estar justificado por una razón.

1.	Fórmula 1	Razón
2.	Fórmula 2	Razón
	⋮	⋮
n.	Conclusión	Razón

☞ Las posibles razones son:

- Ser una premisa.
- Obtenerse mediante una regla de las fórmulas anteriores.
- Abre una caja de suposición.

Reglas de deducción (1/4)

Eliminación de $\wedge$

$$\frac{\alpha \wedge \beta}{\alpha}$$

1.	$\alpha \wedge \beta$	Razón
2.	α	RE$\wedge$1

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Reglas de deducción (1/4)

Eliminación de $\wedge$

$$\frac{\alpha \wedge \beta}{\alpha}$$

1.	$\alpha \wedge \beta$	Razón
2.	α	RE$\wedge$1

También funciona

$$\frac{\alpha \wedge \beta}{\beta}$$

1.	$\alpha \wedge \beta$	Razón
2.	β	RE$\wedge$1

Reglas de deducción (1/4)

Eliminación de $\wedge$

$$\frac{\alpha \wedge \beta}{\alpha}$$

1.	$\alpha \wedge \beta$	Razón
2.	α	RE$\wedge$1

También funciona

$$\frac{\alpha \wedge \beta}{\beta}$$

1.	$\alpha \wedge \beta$	Razón
2.	β	RE$\wedge$1

Introducción de $\wedge$

$$\frac{\alpha, \beta}{\alpha \wedge \beta}$$

1.	α	Razón
2.	β	Razón
3.	$\alpha \wedge \beta$	RI$\wedge$1,2

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Conmutatividad de $\wedge$

A partir de $\alpha \wedge \beta$ se deduce $\beta \wedge \alpha$. Es decir,

$$\frac{\alpha \wedge \beta}{\beta \wedge \alpha}$$

Ejemplo — Conmutatividad de $\wedge$

A partir de $\alpha \wedge \beta$ se deduce $\beta \wedge \alpha$. Es decir,

$$\frac{\alpha \wedge \beta}{\beta \wedge \alpha}$$

1.	$\alpha \wedge \beta$	Premisa

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Conmutatividad de $\wedge$

A partir de $\alpha \wedge \beta$ se deduce $\beta \wedge \alpha$. Es decir,

$$\frac{\alpha \wedge \beta}{\beta \wedge \alpha}$$

1.	$\alpha \wedge \beta$	Premisa
2.	β	RE$\wedge$ 1

Ejemplo — Conmutatividad de $\wedge$

A partir de $\alpha \wedge \beta$ se deduce $\beta \wedge \alpha$. Es decir,

$$\frac{\alpha \wedge \beta}{\beta \wedge \alpha}$$

1.	$\alpha \wedge \beta$	Premisa
2.	β	RE$\wedge$ 1
3.	α	RE$\wedge$ 1

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Conmutatividad de $\wedge$

A partir de $\alpha \wedge \beta$ se deduce $\beta \wedge \alpha$. Es decir,

$$\frac{\alpha \wedge \beta}{\beta \wedge \alpha}$$

1.	$\alpha \wedge \beta$	Premisa
2.	β	RE$\wedge$ 1
3.	α	RE$\wedge$ 1
4.	$\beta \wedge \alpha$	RI$\wedge$ 2,3

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Reglas de deducción (2/4)

Eliminación de $\rightarrow$

$$\frac{\alpha \rightarrow \beta, \alpha}{\beta}$$

1.	$\alpha \rightarrow \beta$	Razón
2.	α	Razón
3.	β	RE$\rightarrow$1,2

Introducción de $\rightarrow$

1.	$\boxed{\alpha}$	Suposición
2.	$\boxed{\beta}$	Razón
3.	$\alpha \rightarrow \beta$	RI$\rightarrow$1–2

☞ ¡Caja de suposición!

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Ejemplo — Transitividad de $\to$

A partir de $\alpha \to \beta$ y $\beta \to \gamma$ se deduce $\alpha \to \gamma$. Es decir,

$$\frac{\alpha \to \beta, \beta \to \gamma}{\alpha \to \gamma}$$

Ejemplo — Transitividad de $\to$

A partir de $\alpha \to \beta$ y $\beta \to \gamma$ se deduce $\alpha \to \gamma$. Es decir,

$$\frac{\alpha \to \beta, \beta \to \gamma}{\alpha \to \gamma}$$

1.	$\alpha \to \beta$	Premisa

MACC Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Transitividad de $\rightarrow$

A partir de $\alpha \rightarrow \beta$ y $\beta \rightarrow \gamma$ se deduce $\alpha \rightarrow \gamma$. Es decir,

$$\frac{\alpha \rightarrow \beta, \beta \rightarrow \gamma}{\alpha \rightarrow \gamma}$$

1.	$\alpha \rightarrow \beta$	Premisa
2.	$\beta \rightarrow \gamma$	Premisa

Ejemplo — Transitividad de $\to$

A partir de $\alpha \to \beta$ y $\beta \to \gamma$ se deduce $\alpha \to \gamma$. Es decir,

$$\frac{\alpha \to \beta, \beta \to \gamma}{\alpha \to \gamma}$$

1.	$\alpha \to \beta$	Premisa
2.	$\beta \to \gamma$	Premisa
3.	α	Supuesto

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Transitividad de $\rightarrow$

A partir de $\alpha \rightarrow \beta$ y $\beta \rightarrow \gamma$ se deduce $\alpha \rightarrow \gamma$. Es decir,

$$\frac{\alpha \rightarrow \beta, \beta \rightarrow \gamma}{\alpha \rightarrow \gamma}$$

1.	$\alpha \rightarrow \beta$	Premisa
2.	$\beta \rightarrow \gamma$	Premisa
3.	α	Supuesto
4.	β	RE$\rightarrow$ 1,3

Ejemplo — Transitividad de $\to$

A partir de $\alpha \to \beta$ y $\beta \to \gamma$ se deduce $\alpha \to \gamma$. Es decir,

$$\frac{\alpha \to \beta, \beta \to \gamma}{\alpha \to \gamma}$$

1.	$\alpha \to \beta$	Premisa
2.	$\beta \to \gamma$	Premisa
3.	$\boxed{\alpha}$	Supuesto
4.	$\boxed{\beta}$	RE$\to$ 1,3
5.	$\boxed{\gamma}$	RE$\to$ 2,4

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Transitividad de $\rightarrow$

A partir de $\alpha \rightarrow \beta$ y $\beta \rightarrow \gamma$ se deduce $\alpha \rightarrow \gamma$. Es decir,

$$\frac{\alpha \rightarrow \beta, \beta \rightarrow \gamma}{\alpha \rightarrow \gamma}$$

1.	$\alpha \rightarrow \beta$	Premisa
2.	$\beta \rightarrow \gamma$	Premisa
3.	α	Supuesto
4.	β	RE$\rightarrow$ 1,3
5.	γ	RE$\rightarrow$ 2,4
6.	$\alpha \rightarrow \gamma$	RI$\rightarrow$ 3–5

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Cajas de suposición

- Un supuesto abre una caja de suposición.
- Una suposición solo se quita con una regla:
    - Introducción de la $\rightarrow$.
    - Eliminación de $\neg$ (ver adelante)
- No se puede afirmar la conclusión sin haber cerrado todas las cajas de suposición.

1.	$\alpha \rightarrow \beta$	Premisa
2.	$\beta \rightarrow \gamma$	Premisa
3.	α	Supuesto
4.	β	RE$\rightarrow$ 1,3
5.	γ	RE$\rightarrow$ 2,4
6.	$\alpha \rightarrow \gamma$	RI$\rightarrow$ 3–5

MACC Matemáticas Aplicadas y Ciencias de la Computación

Reglas de deducción (3/4)

Eliminación de $\vee$

$$\frac{\alpha \vee \beta, \neg\alpha}{\beta}$$

1.	$\alpha \vee \beta$	Razón
2.	$\neg\alpha$	Razón
3.	β	RE$\vee$1,2

Introducción de $\vee$

$$\frac{\alpha}{\alpha \vee \beta}$$

1.	α	Razón
2.	$\alpha \vee \beta$	RI$\vee$1

También funciona

$$\frac{\beta}{\alpha \vee \beta}$$

1.	β	Razón
2.	$\alpha \vee \beta$	RI$\vee$1

Reglas de deducción (4/4)

Eliminación de $\neg$

1.	$\boxed{\neg\alpha}$	Suposición
2.	$\boxed{\beta \wedge \neg\beta}$	Razón
3.	α	RE$\neg$1–2

Introducción de $\neg$

$$\frac{\alpha}{\neg\neg\alpha}$$

1.	α	Razón
2.	$\neg\neg\alpha$	RI$\neg$1

MACC Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Eliminación de $\neg\neg$

A partir de $\neg\neg\alpha$ se deduce α. Es decir,

$$\frac{\neg\neg\alpha}{\alpha}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Eliminación de $\neg\neg$

A partir de $\neg\neg\alpha$ se deduce α. Es decir,

$$\frac{\neg\neg\alpha}{\alpha}$$

1.	$\neg\neg\alpha$	Premisa

Ejemplo — Eliminación de $\neg\neg$

A partir de $\neg\neg\alpha$ se deduce α. Es decir,

$$\frac{\neg\neg\alpha}{\alpha}$$

1.	$\neg\neg\alpha$	Premisa
2.	$\neg\alpha$	Supuesto

MACC Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Eliminación de $\neg\neg$

A partir de $\neg\neg\alpha$ se deduce α. Es decir,

$$\frac{\neg\neg\alpha}{\alpha}$$

1.	$\neg\neg\alpha$	Premisa
2.	$\neg\alpha$	Supuesto
3.	$\neg\neg\alpha \wedge \neg\alpha$	RI$\wedge$ 1,2

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Ejemplo — Eliminación de $\neg\neg$

A partir de $\neg\neg\alpha$ se deduce α. Es decir,

$$\frac{\neg\neg\alpha}{\alpha}$$

1.	$\neg\neg\alpha$	Premisa
2.	$\neg\alpha$	Supuesto
3.	$\neg\neg\alpha \wedge \neg\alpha$	RI$\wedge$ 1,2
4.	α	RE$\neg$ 2–3

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1. $\alpha \to \beta$ Premisa

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1.	$\alpha \to \beta$	Premisa
2.	$\neg\beta$	Premisa

MACC Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1.	$\alpha \to \beta$	Premisa
2.	$\neg\beta$	Premisa
3.	$\neg\neg\alpha$	Supuesto

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1.	$\alpha \to \beta$	Premisa
2.	$\neg\beta$	Premisa
3.	$\neg\neg\alpha$	Supuesto
4.	α	Eliminación $\neg\neg$ 3

MACC Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1.	$\alpha \to \beta$	Premisa
2.	$\neg\beta$	Premisa
3.	$\neg\neg\alpha$	Supuesto
4.	α	Eliminación $\neg\neg$ 3
5.	β	RE$\to$ 1,4

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1.	$\alpha \to \beta$	Premisa
2.	$\neg\beta$	Premisa
3.	$\neg\neg\alpha$	Supuesto
4.	α	Eliminación $\neg\neg$ 3
5.	β	RE$\to$ 1,4
6.	$\neg\beta \wedge \beta$	RI$\wedge$ 2,5

MACC Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo — Modus tollendo tollens

A partir de $\alpha \to \beta$ y $\neg\beta$ se deduce $\neg\alpha$. Es decir,

$$\frac{\alpha \to \beta, \neg\beta}{\neg\alpha}$$

1.	$\alpha \to \beta$	Premisa
2.	$\neg\beta$	Premisa
3.	$\neg\neg\alpha$	Supuesto
4.	α	Eliminación $\neg\neg$ 3
5.	β	RE$\to$ 1,4
6.	$\neg\beta \wedge \beta$	RI$\wedge$ 2,5
7.	$\neg\alpha$	RE$\neg$ 3–6

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Ejercicios

1. $$\frac{\alpha \wedge (\beta \wedge \gamma)}{\gamma \wedge \alpha}$$

2. $$\frac{\alpha \to (\beta \to \gamma)}{(\alpha \wedge \beta) \to \gamma}$$

3. $$\frac{\neg\alpha \wedge \neg\beta}{\neg(\alpha \vee \beta)}$$

4. $$\frac{\alpha \vee \beta, \alpha \to \gamma, \beta \to \gamma}{\gamma}$$

MACC Matemáticas Aplicadas y Ciencias de la Computación

Contenido

Base lógica de conocimiento

¿Qué es una base lógica de conocimiento?

Hechos $+$ **Reglas**

Base lógica de conocimiento

¿Qué es una base lógica de conocimiento?

Hechos	+	**Reglas**
Átomos		Implicaciones

Base lógica de conocimiento

¿Qué es una base lógica de conocimiento?

Hechos	+	**Reglas**
Átomos		Implicaciones

☞ Los hechos ayudan a representar una situación. Cuantos más hechos se conozcan, más precisa es la situación conocida.

MACC Matemáticas Aplicadas y Ciencias de la Computación

Base lógica de conocimiento

¿Qué es una base lógica de conocimiento?

Hechos	+	**Reglas**
Átomos		Implicaciones

☞ Los hechos ayudan a representar una situación. Cuantos más hechos se conozcan, más precisa es la situación conocida.

☞ Las implicaciones representan conocimiento lógico, de sentido común y de dominio.

Reglas (1/2)

Definiciones:

- Un literal es un átomo (literal positivo) o la negación de un átomo (literal negativo).

Reglas (1/2)

Definiciones:

- Un literal es un átomo (literal positivo) o la negación de un átomo (literal negativo).
- Una situación es la conjunción de uno o más literales.

Reglas (2/2)

Definiciones:
Una regla es la implicación de una situación (antecedente) a un literal (consecuente).

Reglas (2/2)

Definiciones:
Una regla es la implicación de una situación (antecedente) a un literal (consecuente).

$$\underbrace{p \wedge q}_{antecedente} \quad \rightarrow \quad \underbrace{r}_{consecuente}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Reglas (2/2)

Definiciones:
Una regla es la implicación de una situación (antecedente) a un literal (consecuente).

$$\underbrace{p \wedge q}_{antecedente} \quad \rightarrow \quad \underbrace{r}_{consecuente}$$

☞ Vamos a ver algoritmos eficientes para deducciones que solo incluyen reglas.

Reglas (formal)

Gramática

$$
\begin{aligned}
Atomica &\longrightarrow\ p \mid q \mid r \mid \ldots \\
Literal &\longrightarrow Atomica \mid \neg Atomica \\
Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\
Regla &\longrightarrow Situacion \rightarrow Literal
\end{aligned}
$$

Reglas (formal)

Gramática

$$
\begin{aligned}
Atomica &\longrightarrow\ p \mid q \mid r \mid \ldots \\
Literal &\longrightarrow Atomica \mid \neg Atomica \\
Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\
Regla &\longrightarrow Situacion \rightarrow Literal
\end{aligned}
$$

Ejemplo

$$Regla \overset{*}{\Rightarrow} Literal \wedge Situacion \rightarrow Literal$$

Reglas (formal)

Gramática

$$
\begin{aligned}
Atomica &\longrightarrow \; p \mid q \mid r \mid \ldots \\
Literal &\longrightarrow Atomica \mid \neg Atomica \\
Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\
Regla &\longrightarrow Situacion \to Literal
\end{aligned}
$$

Ejemplo

$$Regla \stackrel{*}{\Rightarrow} Atomica \wedge Situacion \to Literal$$

Reglas (formal)

Gramática

$$\begin{aligned} Atomica &\longrightarrow \; p \mid q \mid r \mid \ldots \\ Literal &\longrightarrow Atomica \mid \neg Atomica \\ Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\ Regla &\longrightarrow Situacion \to Literal \end{aligned}$$

Ejemplo

$$Regla \overset{*}{\Rightarrow} p \wedge Situacion \to Literal$$

Reglas (formal)

Gramática

$$
\begin{aligned}
Atomica &\longrightarrow\ p \mid q \mid r \mid \ldots \\
Literal &\longrightarrow Atomica \mid \neg Atomica \\
Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\
Regla &\longrightarrow Situacion \rightarrow Literal
\end{aligned}
$$

Ejemplo

$$
Regla \stackrel{*}{\Rightarrow} p \wedge Literal \rightarrow Literal
$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Reglas (formal)

Gramática

$$
\begin{aligned}
Atomica &\longrightarrow\ p \mid q \mid r \mid \ldots \\
Literal &\longrightarrow Atomica \mid \neg Atomica \\
Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\
Regla &\longrightarrow Situacion \rightarrow Literal
\end{aligned}
$$

Ejemplo

$$Regla \stackrel{*}{\Rightarrow} p \wedge Atomica \rightarrow Literal$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Reglas (formal)

Gramática

$$\begin{aligned} Atomica &\longrightarrow \; p \mid q \mid r \mid \ldots \\ Literal &\longrightarrow Atomica \mid \neg Atomica \\ Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\ Regla &\longrightarrow Situacion \rightarrow Literal \end{aligned}$$

Ejemplo

$$Regla \overset{*}{\Rightarrow} p \wedge q \rightarrow Literal$$

Reglas (formal)

Gramática

$$\begin{aligned} \mathit{Atomica} &\longrightarrow\ p \mid q \mid r \mid \ldots \\ \mathit{Literal} &\longrightarrow \mathit{Atomica} \mid \neg \mathit{Atomica} \\ \mathit{Situacion} &\longrightarrow \mathit{Literal} \mid \mathit{Literal} \wedge \mathit{Situacion} \\ \mathit{Regla} &\longrightarrow \mathit{Situacion} \to \mathit{Literal} \end{aligned}$$

Ejemplo

$$\mathit{Regla} \overset{*}{\Rightarrow} p \wedge q \to \neg \mathit{Atomica}$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Reglas (formal)

Gramática

$$
\begin{aligned}
Atomica &\longrightarrow\ \ p \mid q \mid r \mid \ldots \\
Literal &\longrightarrow Atomica \mid \neg Atomica \\
Situacion &\longrightarrow Literal \mid Literal \wedge Situacion \\
Regla &\longrightarrow Situacion \rightarrow Literal
\end{aligned}
$$

Ejemplo

$$Regla \overset{*}{\Rightarrow} p \wedge q \rightarrow \neg r$$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Estrategias de deducción

Objetivo: Un literal que representa un hecho que queremos conocer si es cierto.

Estrategias de deducción

Objetivo: Un literal que representa un hecho que queremos conocer si es cierto.

Forward chaining

Partir de los literales en la base e ir aplicando reglas para obtener nuevos literales, hasta que se pueda llegar al objetivo.

Estrategias de deducción

Objetivo: Un literal que representa un hecho que queremos conocer si es cierto.

Forward chaining

Partir de los literales en la base e ir aplicando reglas para obtener nuevos literales, hasta que se pueda llegar al objetivo.

Backward chaining

Partir del objetivo y buscar hacia atrás si hay alguna regla con consecuente igual al objetivo. Los literales del antecedente de la regla encontrada los buscamos en la base o los tratamos de deducir de igual manera que el objetivo.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo = 'r'
Base = ['$q \rightarrow r$', '$p \wedge s \rightarrow r$', 's', 'p']

`conteo` $\leftarrow$

☞ Diccionario donde `conteo`$[r]$ es el número de átomos en el antecedente de la regla r.

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $= [$'$q \to r$', '$p \wedge s \to r$', 's', 'p'$]$

$$\texttt{conteo} \leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}2\}$$

☞ Diccionario donde $\texttt{conteo}[r]$ es el número de átomos en el antecedente de la regla r.

Forward chaining - Ejemplo

Objetivo = 'r'
Base = ['$q \rightarrow r$', '$p \wedge s \rightarrow r$', 's', 'p']

$$\texttt{conteo} \leftarrow \{\text{'}q \rightarrow r\text{'}{:}1,\ \text{'}p \wedge s \rightarrow r\text{'}{:}2\}$$
$$\texttt{deducidos} \leftarrow [\text{'}s\text{'}, \text{'}p\text{'}]$$

☞ Lista de átomos que ya se sabe se deducen de la base.

Forward chaining - Ejemplo

Objetivo = ‘r’
Base = [‘$q \to r$’, ‘$p \wedge s \to r$’, ‘s’, ‘p’]

$$
\begin{aligned}
\texttt{conteo} &\leftarrow \{‘q \to r’{:}1,\ ‘p \wedge s \to r’{:}2\} \\
\texttt{deducidos} &\leftarrow [‘s’, ‘p’] \\
\texttt{pila} &\leftarrow [‘s’, ‘p’]
\end{aligned}
$$

☞ Lista de átomos que ya se sabe se deducen de la base.

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$
\begin{aligned}
\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}:1,\ \text{'}p \wedge s \to r\text{'}:2\} \\
\texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\
\texttt{pila} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\
\texttt{a} &\leftarrow \texttt{pila}.pop()
\end{aligned}
$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\begin{aligned}\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}:1,\ \text{'}p \wedge s \to r\text{'}:2\}\\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}]\\ \texttt{pila} &\leftarrow [\text{'}s\text{'}]\\ \texttt{a} &\leftarrow \text{'}p\text{'}\end{aligned}$$

Forward chaining - Ejemplo

Objetivo = 'r'
Base = ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\texttt{conteo} \leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}2\}$$
$$\texttt{deducidos} \leftarrow [\text{'}s\text{'}, \text{'}p\text{'}]$$
$$\texttt{pila} \leftarrow [\text{'}s\text{'}]$$
$$\texttt{a} \leftarrow \text{'}p\text{'}$$

☞ ¿a es el objetivo?

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\begin{aligned} \texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}2\} \\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\ \texttt{pila} &\leftarrow [\text{'}s\text{'}] \\ \texttt{a} &\leftarrow \text{'}p\text{'} \end{aligned}$$

☞ No

MACC Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$
\begin{aligned}
\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}:1,\ \text{'}p \wedge s \to r\text{'}:\textcolor{red}{1}\} \\
\texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\
\texttt{pila} &\leftarrow [\text{'}s\text{'}] \\
\texttt{a} &\leftarrow \text{'}p\text{'}
\end{aligned}
$$

☞ Disminuimos el conteo de las reglas con a en el antecedente.

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $= [\text{'}q \rightarrow r\text{'}, \text{'}p \wedge s \rightarrow r\text{'}, \text{'}s\text{'}, \text{'}p\text{'}]$

$$
\begin{aligned}
\texttt{conteo} &\leftarrow \{\text{'}q \rightarrow r\text{'}:1,\ \text{'}p \wedge s \rightarrow r\text{'}:1\} \\
\texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\
\texttt{pila} &\leftarrow [\text{'}s\text{'}] \\
\texttt{a} &\leftarrow \texttt{pila}.pop()
\end{aligned}
$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\begin{aligned}\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}1\}\\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}]\\ \texttt{pila} &\leftarrow []\\ \texttt{a} &\leftarrow \text{'}s\text{'}\end{aligned}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ ‘r’
Base $=$ [‘$q \to r$’, ‘$p \land s \to r$’, ‘s’, ‘p’]

$$\begin{aligned}\texttt{conteo} &\leftarrow \{\text{‘}q \to r\text{’}{:}1,\ \text{‘}p \land s \to r\text{’}{:}1\}\\ \texttt{deducidos} &\leftarrow [\text{‘}s\text{’}, \text{‘}p\text{’}]\\ \texttt{pila} &\leftarrow []\\ \texttt{a} &\leftarrow \text{‘}s\text{’}\end{aligned}$$

☞ ¿a es el objetivo?

Forward chaining - Ejemplo

Objetivo = 'r'
Base = ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\begin{aligned} \texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}1\} \\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\ \texttt{pila} &\leftarrow [] \\ \texttt{a} &\leftarrow \text{'}s\text{'} \end{aligned}$$

☞ No

Forward chaining - Ejemplo

Objetivo = 'r'
Base = ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\texttt{conteo} \leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}\textcolor{red}{0}\}$$
$$\texttt{deducidos} \leftarrow [\text{'}s\text{'}, \text{'}p\text{'}]$$
$$\texttt{pila} \leftarrow []$$
$$\texttt{a} \leftarrow \text{'}s\text{'}$$

☞ Disminuimos el conteo de las reglas con a en el antecedente.

MACC Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$
\begin{aligned}
\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1,\ \text{'}p \wedge s \to r\text{'}{:}0\} \\
\texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\
\texttt{pila} &\leftarrow [] \\
\texttt{c} &\leftarrow \text{'}r\text{'}
\end{aligned}
$$

☞ Encontramos el consecuente de la regla.

Forward chaining - Ejemplo

Objetivo = 'r'
Base = ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\begin{aligned} \texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1\} \\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}] \\ \texttt{pila} &\leftarrow [] \\ \texttt{c} &\leftarrow \text{'}r\text{'} \end{aligned}$$

☞ Quitamos la regla y preguntamos: ¿El consecuente ha sido deducido?

Forward chaining - Ejemplo

$$\text{Objetivo} = \text{`}r\text{'}$$
$$\text{Base} = [\text{`}q \to r\text{'}, \text{`}p \wedge s \to r\text{'}, \text{`}s\text{'}, \text{`}p\text{'}]$$

$$\begin{aligned} \texttt{conteo} &\leftarrow \{\text{`}q \to r\text{'}{:}1\} \\ \texttt{deducidos} &\leftarrow [\text{`}s\text{'}, \text{`}p\text{'}] \\ \texttt{pila} &\leftarrow [] \\ \texttt{c} &\leftarrow \text{`}r\text{'} \end{aligned}$$

☞ No

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\texttt{conteo} \leftarrow \{\text{'}q \to r\text{'}{:}1\}$$
$$\texttt{deducidos} \leftarrow [\text{'}s\text{'}, \text{'}p\text{'}, \text{'}r\text{'}]$$
$$\texttt{pila} \leftarrow [\text{'}r\text{'}]$$

☞ Incluimos el consecuente en `deducidos` y en `pila`.

MACC Matemáticas Aplicadas y Ciencias de la Computación

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $= [$'$q \to r$', '$p \wedge s \to r$', 's', 'p'$]$

$$\begin{aligned}\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}:1\}\\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}, \text{'}r\text{'}]\\ \texttt{pila} &\leftarrow [\text{'}r\text{'}]\\ \texttt{a} &\leftarrow \texttt{pila}.pop()\end{aligned}$$

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$
\begin{aligned}
\texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1\} \\
\texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}, \text{'}r\text{'}] \\
\texttt{pila} &\leftarrow [] \\
\texttt{a} &\leftarrow \text{'}r\text{'}
\end{aligned}
$$

☞ ¿a es el objetivo?

Forward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']

$$\begin{aligned} \texttt{conteo} &\leftarrow \{\text{'}q \to r\text{'}{:}1\} \\ \texttt{deducidos} &\leftarrow [\text{'}s\text{'}, \text{'}p\text{'}, \text{'}r\text{'}] \\ \texttt{pila} &\leftarrow [] \\ \texttt{a} &\leftarrow \text{'}r\text{'} \end{aligned}$$

☞ Sí

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Pseudo código Forward chaining

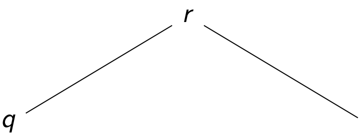
```
Algorithm 1: forward
  Input:
  * Una base de conocimiento base
  * Un literal objetivo
  Output:
  * True o False
  /* --- Inicio ---
reglas ← copia de base.reglas          /* Lista
conteo ← {}          /* Diccionario (tabla hash)
for cada regla en base.reglas do
    conteo[regla] ← len(regla.antecedente);
end
pila ← copia de base.hechos          /* Lista
deducidos ← copia de base.hechos          /* Lista
while len(pila) > 0 do
    if objetivo está en deducidos then
        return True;
    end
    a ← pila.pop();
    eliminaciones ← []          /* Lista
    for cada i desde 0 hasta len(reglas) do
        regla ← reglas[i];
        if a está en regla.antecedente then
            conteo[regla] ← conteo[regla] − 1;
            if conteo[regla] = 0 then
                c ← regla.consecuente;
                if c no está en deducidos then
                    Incluir c en pila;
                    Incluir c en deducidos;
                end
                Incluir i en eliminaciones;
            end
        end
    end
    for cada i en eliminaciones do
        regla ← reglas[i];
        Eliminar conteo[regla];
        Eliminar regla[i];
    end
end
return False;
/* --- Fin ---
```

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Backward chaining - Ejemplo

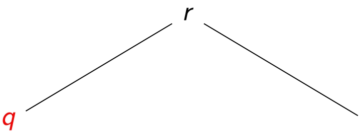
Objetivo $=$ 'r'

Base $= [$'$q \to r$', '$p \wedge s \to r$', 's', 'p'$]$



Backward chaining - Ejemplo

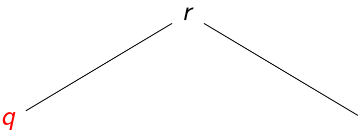
Objetivo $=$ 'r'
Base $= [$'$q \rightarrow r$', '$p \wedge s \rightarrow r$', 's', 'p'$]$



☞ ¿q está en la base?

Backward chaining - Ejemplo

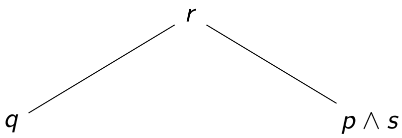
Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']



☞ No

Backward chaining - Ejemplo

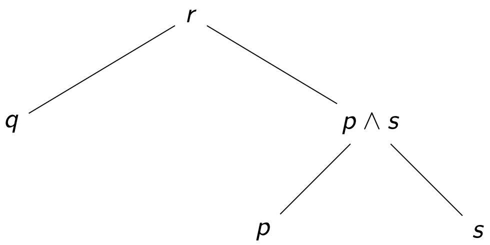
Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \wedge s \to r$', 's', 'p']



Backward chaining - Ejemplo

Objetivo = 'r'

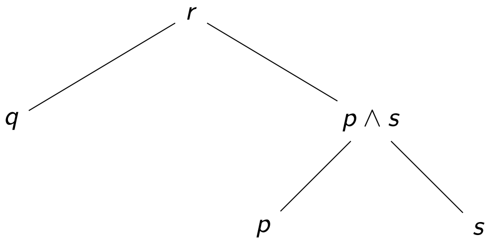
Base = ['$q \to r$', '$p \wedge s \to r$', 's', 'p']



☞ ¿p está en la base y s está en la base?

Backward chaining - Ejemplo

Objetivo $=$ 'r'
Base $=$ ['$q \rightarrow r$', '$p \wedge s \rightarrow r$', 's', 'p']

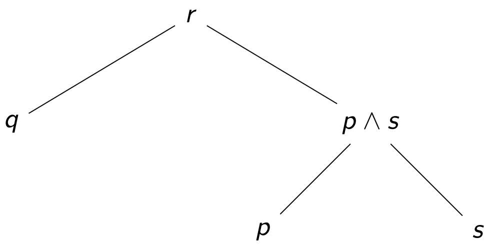


☞ Sí y ¿s está en la base?

Backward chaining - Ejemplo

Objetivo $=$ ‘r’

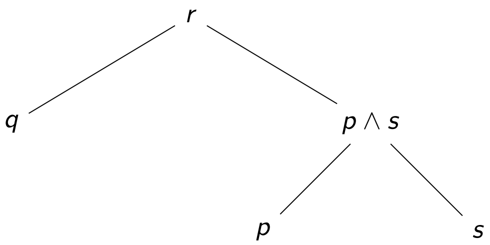
Base $= [$‘$q \rightarrow r$’, ‘$p \wedge s \rightarrow r$’, ‘s’, ‘p’$]$



☞ Sí y sí

Backward chaining - Ejemplo

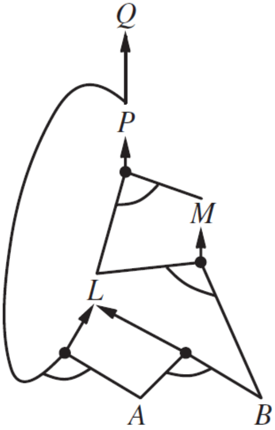
Objetivo $=$ 'r'
Base $=$ ['$q \to r$', '$p \land s \to r$', 's', 'p']



☞ ¡Deducimos r!

Otro ejemplo

$$P \to Q$$
$$L \wedge M \to P$$
$$B \wedge L \to M$$
$$A \wedge P \to L$$
$$A \wedge B \to L$$
$$A$$
$$B$$



Pseudo código Backward chaining (1/2)

function AND_OR_GRAPH_SEARCH(*objetivo*, *base*) **return** *success* or *failure*
  return OR_SEARCH(*objetivo*, *base*, [])

```
Algorithm 2: or_search
  Input:
  * Una base de conocimiento base
  * Un literal consecuente
  * Una lista camino
  Output:
  * True o False
  /* --- Inicio ---
if consecuente está en base.hechos then
  | return True;
end
if consecuente está en camino then
  | return False;
end
candidatos ← base.reglas_aplicables(consecuente);
if candidatos está vacía then
  | return False;
end
for cada regla en candidatos do
  | antecedente ← regla.antecedente;
  | Incluir consecuente en camino;
  | resultado ← and_search(antecedente, camino);
  | if resultado es True then
  |   | return True;
  | end
end
/* --- Fin ---
```

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Pseudo código Backward chaining (2/2)

Algorithm 3: `and_search`

```
Input:
* Una base de conocimiento base
* Una lista literales de literales
* Una lista camino
Output:
* True o False
/* --- Inicio ---                                        */
for cada p en literales do
    resultado ← or_search(p, camino);
    if resultado es False then
        return False;
    end
end
return True;
/* --- Fin ---                                           */
```

Deducción automática

☞ Tanto Forward chaining como Backward chaining corren en tiempo lineal respecto al tamaño de la base.

Deducción automática

☞ Tanto Forward chaining como Backward chaining corren en tiempo lineal respecto al tamaño de la base.

☞ Es posible que no todas las fórmulas relevantes para un problema se puedan escribir como reglas, por lo que en este caso no sería posible usar estas herramientas de deducción automática.

Deducción automática

☞ Tanto Forward chaining como Backward chaining corren en tiempo lineal respecto al tamaño de la base.

☞ Es posible que no todas las fórmulas relevantes para un problema se puedan escribir como reglas, por lo que en este caso no sería posible usar estas herramientas de deducción automática.

☞ Alternativas: DPLL, WalkSAT.

En esta sesión usted aprendió

- Un poco de historia sobre la lógica proposicional.
- El lenguaje y la deducción natural de la lógica proposicional.
- Los componentes de una base lógica de conocimiento.
- El algoritmo de deducción mediante forward-chaining.
- El algoritmo de deducción mediante backward-chaining.