

Fórmulas de la lógica proposicional

Sesión 3

Edgar Andrade, PhD

Última revisión: Enero 2020

Matemáticas Aplicadas y Ciencias de la Computación



En esta sesión estudiaremos:

1. Fórmulas como árboles
2. Funciones recursivas sobre fórmulas

1 Fórmulas como árboles

2 Funciones recursivas sobre fórmulas

Definimos un árbol como un objeto con tres atributos

TREE:

.label \Leftarrow Puede ser un átomo o un conector

Definimos un árbol como un objeto con tres atributos

TREE:

.label \Leftarrow Puede ser un átomo o un conectivo

.left \Leftarrow Árbol hijo a la izquierda o NULL

Definimos un árbol como un objeto con tres atributos

TREE:

.label	⇐ Puede ser un átomo o un conectivo
.left	⇐ Árbol hijo a la izquierda o NULL
.right	⇐ Árbol hijo a la derecha o NULL

Ejemplos de fórmulas como árboles (1/2)

$\text{Tree}(q, \text{NULL}, \text{NULL})$

q

Ejemplos de fórmulas como árboles (1/2)

$\text{Tree}(q, \text{NULL}, \text{NULL})$

q

$\text{Tree}(\neg, \text{NULL}, \text{Tree}(q, \text{NULL}, \text{NULL}))$

\neg

q

Ejemplos de fórmulas como árboles (1/2)

$\text{TREE}(q, \text{NULL}, \text{NULL})$

q

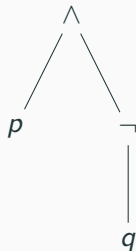
$\text{TREE}(\neg, \text{NULL}, \text{TREE}(q, \text{NULL}, \text{NULL}))$

\neg

q

Ejemplos de fórmulas como árboles (2/2)

$\text{TREE}(\wedge,$
 $\text{TREE}(p, \text{NULL}, \text{NULL}),$
 $\text{TREE}(\neg, \text{NULL}, \text{TREE}(q, \text{NULL}, \text{NULL})))$



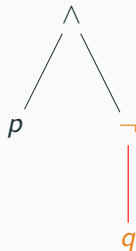
Ejemplos de fórmulas como árboles (2/2)

$\text{TREE}(\wedge,$
 $\text{TREE}(p, \text{NULL}, \text{NULL}),$
 $\text{TREE}(\neg, \text{NULL}, \text{TREE}(q, \text{NULL}, \text{NULL})))$



Ejemplos de fórmulas como árboles (2/2)

$\text{TREE}(\wedge,$
 $\text{TREE}(p, \text{NULL}, \text{NULL}),$
 $\text{TREE}(\neg, \text{NULL}, \text{TREE}(q, \text{NULL}, \text{NULL}))$
)



1 Fórmulas como árboles

2 Funciones recursivas sobre fórmulas

Funciones recursivas (1/3)

Objetivo: Definimos una función que encuentre el conjunto de átomos de una fórmula.

Funciones recursivas (1/3)

Objetivo: Definimos una función que encuentre el conjunto de átomos de una fórmula.

Ejemplos:

Argumento		Resultado
q	\mapsto	$\{q\}$

Funciones recursivas (1/3)

Objetivo: Definimos una función que encuentre el conjunto de átomos de una fórmula.

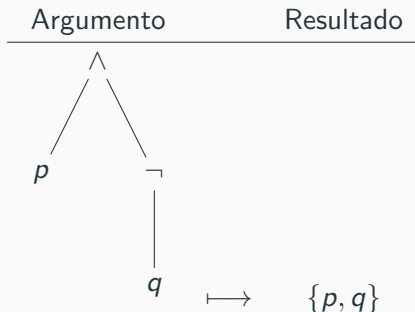
Ejemplos:

Argumento		Resultado
\neg		
$ $		
q	\mapsto	$\{q\}$

Funciones recursivas (1/3)

Objetivo: Definimos una función que encuentre el conjunto de átomos de una fórmula.

Ejemplos:



Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

 SI f .RIGHT == NULL:

 RETORNAR $\{f$.LABEL $\}$

 :

Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

⋮

La definición continúa
en un momento

Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

⋮

La condición se cumple
sii f es un átomo

Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

\vdots

Ej:

$f = \text{Tree}(q, \text{null}, \text{null})$

Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

⋮

Ej:

$f = \text{Tree}(q, \text{null}, \text{null})$

Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

⋮

Ej:

f = Tree(q , null, null)

ATOMOS(f) = $\{f$.label}

Funciones recursivas (1/3)

Sea f un árbol arbitrario:

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

\vdots

Ej:

$f = \text{Tree}(q, \text{null}, \text{null})$

ATOMOS(f) = $\{q\}$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR f .LABEL

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

⋮

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

SI NO, SI f .LABEL == \neg :
 RETORNAR ATOMOS(f .RIGHT)

⋮

La condición se cumple sii la raíz de f es \neg

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

\vdots

Ej: $f = \text{TREE}(\neg, \text{NULL}, \text{Tree}(q, \text{null}, \text{null}))$

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

\vdots

Ej: $f = \text{TREE}(\neg, \text{NULL}, \text{Tree}(q, \text{null}, \text{null}))$

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

⋮

Ej: Observe que f .right=Tree(q ,null,null)

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

⋮

Ej: $f = \text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null}))$

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

\vdots

Ej: Luego $\text{Atomos}(f) = \text{Atomos}(f.\text{right})$

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

⋮

Ej: Luego $\text{Atomos}(f) = \text{Atomos}(\text{Tree}(q, \text{null}, \text{null}))$

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $f$ .LABEL
```

```
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
      RETORNAR ATOMOS( $f$ .RIGHT)
```

⋮

Ej: Luego $\text{Atomos}(f) = \{q\}$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR { $f$ .LABEL}  
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR ATOMOS( $f$ .RIGHT)
```

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:
 RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

La condición se cumple sii la raíz de f es un conector que no es \neg

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f$ .LABEL}  
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR ATOMOS( $f$ .RIGHT)
```

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:
 RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Ej: $f = \text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})))$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Ej: Observe que f .left = Tree(p , null, null)

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f$ .LABEL $\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR ATOMOS( $f$ .RIGHT)
```

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:
 RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Ej: $f = \text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})))$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL}

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Ej: Y que f .right = Tree(\neg , null, Tree(q , null, null))

Funciones recursivas (1/3)

```
DEF ATOMOS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f$ .LABEL}  
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR ATOMOS( $f$ .RIGHT)
```

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:
 RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Ej: $f = \text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})))$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Luego $\text{Atomos}(f) = \text{Atomos}(f.\text{left}) \cup \text{Atomos}(f.\text{right})$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Luego Atomos(f) = $\{p\} \cup \text{Atomos}(f.\text{right})$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Luego Atomos(f) = $\{p\} \cup \{q\}$

Funciones recursivas (1/3)

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Es decir, $\text{Atomos}(f) = \{p, q\}$

Funciones recursivas (1/3)

Función que devuelve el conjunto de átomos de una fórmula f

DEF ATOMOS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f$.LABEL $\}$

SI NO, SI f .LABEL == \neg :

RETORNAR ATOMOS(f .RIGHT)

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR ATOMOS(f .LEFT) \cup ATOMOS(f .RIGHT)

Funciones recursivas (2/3)

Función que devuelve el conjunto de subfórmulas de una fórmula f

DEF SUBFORMS(f):

SI f .RIGHT == NULL:

RETORNAR $\{f\}$

SI NO, SI f .LABEL == \neg :

RETORNAR $\{f\} \cup \text{SUBFORMS}(f.\text{RIGHT})$

SI NO, SI f .LABEL $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR $\{f\} \cup \text{SUBFORMS}(f.\text{LEFT}) \cup \text{SUBFORMS}(f.\text{RIGHT})$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{RIGHT})$   
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{LEFT}) \cup \text{SUBFORMS}(f.\text{RIGHT})$ 
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f$ .RIGHT)  
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f$ .LEFT)  $\cup \text{SUBFORMS}(f$ .RIGHT)
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

$\text{Subforms}(f) =$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{RIGHT})$   
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{LEFT}) \cup \text{SUBFORMS}(f.\text{RIGHT})$ 
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

$= \text{Subforms}(\text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null}))))$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR { $f$ }  
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR { $f$ }  $\cup$  SUBFORMS( $f$ .RIGHT)  
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR { $f$ }  $\cup$  SUBFORMS( $f$ .LEFT)  $\cup$  SUBFORMS( $f$ .RIGHT)
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

$$= \{f\} \cup \text{Subforms}(\text{Tree}(p, \text{null}, \text{null})) \cup \text{Subforms}(\text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})))$$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f$ .RIGHT)  
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f$ .LEFT)  $\cup \text{SUBFORMS}(f$ .RIGHT)
```

Ejemplo: Paso a paso de Subforms(f) para $f = p \wedge \neg q$:

$$= \{f\} \cup \{p\} \cup \text{Subforms}(\text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})))$$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{RIGHT})$   
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{LEFT}) \cup \text{SUBFORMS}(f.\text{RIGHT})$ 
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

$$= \{f\} \cup \{p\} \cup \{\neg q\} \cup \text{Subforms}(\text{Tree}(q, \text{null}, \text{null}))$$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{RIGHT})$   
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f.\text{LEFT}) \cup \text{SUBFORMS}(f.\text{RIGHT})$ 
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

$$= \{f\} \cup \{p\} \cup \{\neg q\} \cup \{q\}$$

Funciones recursivas (2/3)

```
DEF SUBFORMS( $f$ ):  
  SI  $f$ .RIGHT == NULL:  
    RETORNAR  $\{f\}$   
  SI NO, SI  $f$ .LABEL ==  $\neg$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f$ .RIGHT)  
  SI NO, SI  $f$ .LABEL  $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ :  
    RETORNAR  $\{f\} \cup \text{SUBFORMS}(f$ .LEFT)  $\cup \text{SUBFORMS}(f$ .RIGHT)
```

Ejemplo: Paso a paso de $\text{Subforms}(f)$ para $f = p \wedge \neg q$:

$$\text{Subforms}(p \wedge \neg q) = \{p \wedge \neg q, p, \neg q, q\}$$

Funciones recursivas (3/3)

Función que sustituye una subfórmula A de una fórmula B por una fórmula A' :

DEF SUST[B, A, A']:

SI $A \notin \text{SUBFORMS}[B]$:

RETORNAR B

SI NO, SI $B == A$:

RETORNAR A'

SI NO, SI $B.\text{LABEL} == \neg$:

RETORNAR TREE(\neg , NULL, SUST[$B.\text{RIGHT}$, A , A'])

SI NO, SI $B.\text{LABEL} \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$:

RETORNAR TREE($B.\text{LABEL}$, SUST[$B.\text{LEFT}$, A , A'], SUST[$B.\text{RIGHT}$, A , A'])

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$$\text{Sust}[p \wedge \neg q, q, \neg r] =$$

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$= \text{Tree}(B.\text{label}, \text{Sust}[B.\text{left}, A, A'], \text{Sust}[B.\text{right}, A, A'])$

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨ →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$= \text{Tree}(\wedge, \text{Sust}[\text{Tree}(p, \text{null}, \text{null}), q, \neg r], \text{Sust}[\text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})), q, \neg r])$

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$= \text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Sust}[\text{Tree}(\neg, \text{null}, \text{Tree}(q, \text{null}, \text{null})), q, \neg r])$

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$= \text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Tree}(\neg, \text{null}, \text{Sust}[\text{Tree}(q, \text{null}, \text{null}), q, \neg r]))$

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$= \text{Tree}(\wedge, \text{Tree}(p, \text{null}, \text{null}), \text{Tree}(\neg, \text{null}, \neg r))$

Funciones recursivas (3/3)

```
DEF SUST[B, A, A']:  
  SI A ∉ SUBFORMS[B]:  
    RETORNAR B  
  SI NO, SI B == A:  
    RETORNAR A'  
  SI NO, SI B.LABEL == ¬:  
    RETORNAR TREE(¬, NULL, SUST[B.RIGHT, A, A'])  
  SI NO, SI B.LABEL ∈ {∧, ∨, →, ↔}:  
    RETORNAR TREE(B.LABEL, SUST[B.LEFT, A, A'], SUST[B.RIGHT, A, A'])
```

Ejemplo: Paso a paso de $\text{Sust}[p \wedge \neg q, q, \neg r]$:

$$\text{Sust}[p \wedge \neg q, q, \neg r] = p \wedge \neg \neg r$$

Fin de la sesión 3

En esta sesión usted ha aprendido:

1. Representar las fórmulas como árboles
2. Encontrar el paso a paso de una función recursiva sobre una fórmula dada