The language of ‘if’, ‘not’, ‘and’, ‘or’

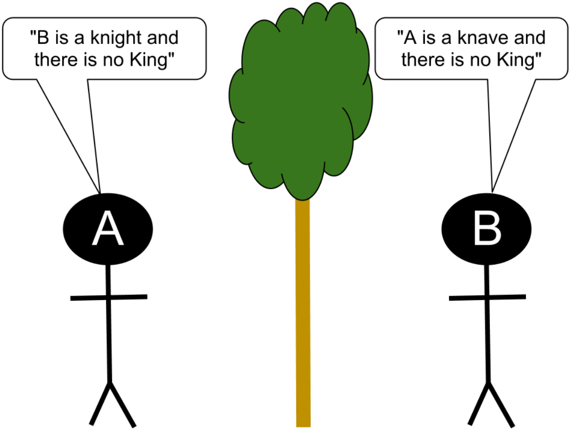
| University of Edinburgh | PHIL08004 |

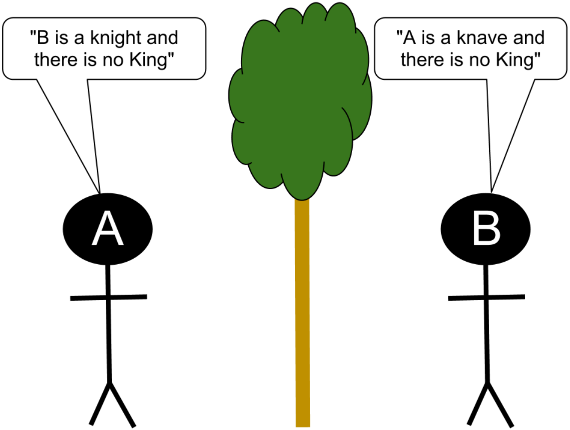
$$(P \wedge Q).\ (R \vee {\sim}Q)\ \therefore R$$

Puzzle. Suppose you are wondering whether or not there is a King of the Island of Knights and Knaves. You come across two inhabitants A and B, who make the following statements:

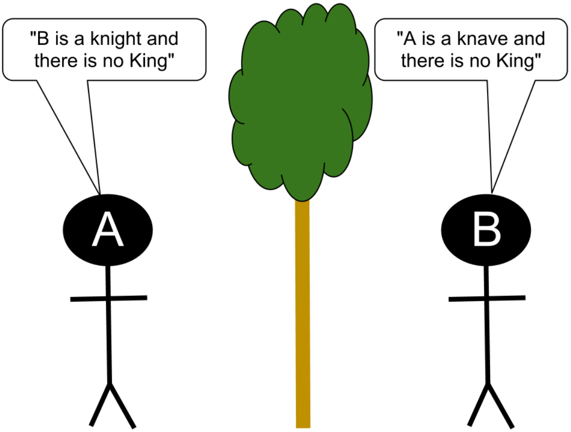
A: B is a knight and there is no King.
B: A is a knave and there is no King.

Does the island have a King?

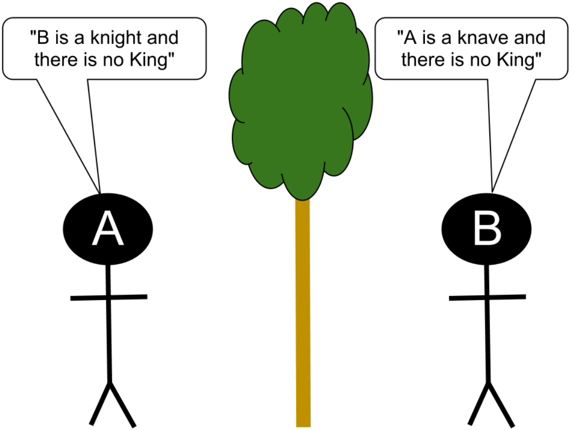

"B is a knight and there is no King"
"A is a knave and there is no King"
A
B


"B is a knight and there is no King"
A
"A is a knave and there is no King"
B

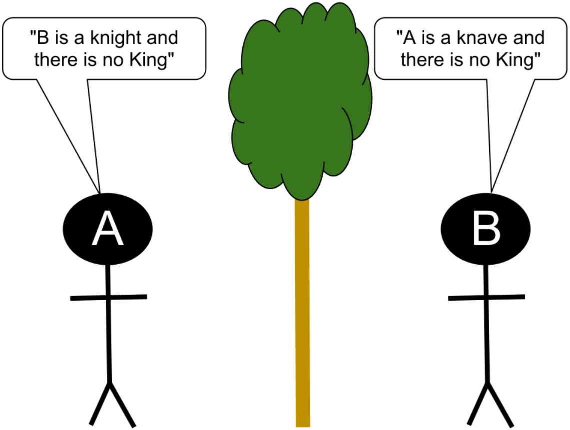
Knight?



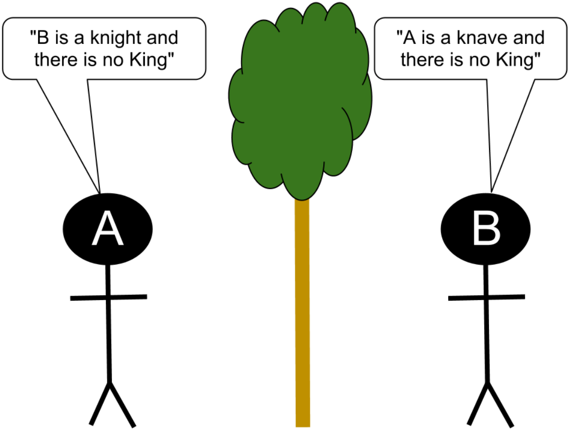
Knight?	Knight?


"B is a knight and there is no King"
"A is a knave and there is no King"
A
B

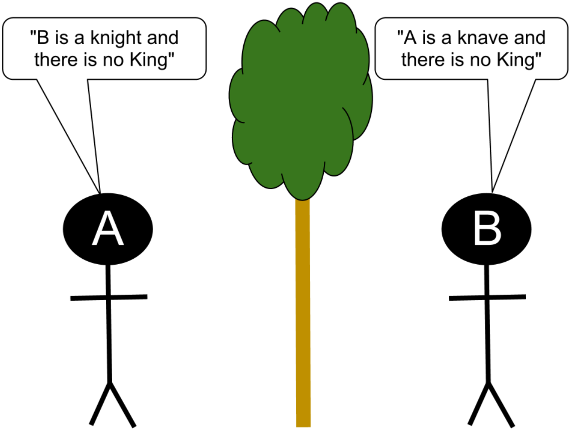
Knave



	Knight?
Knave	


"B is a knight and there is no King"
"A is a knave and there is no King"
A
B

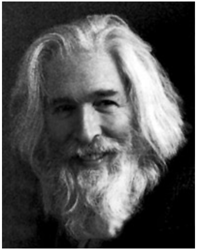
Knave	Knave



Knave Knave

But is there a king?

YES



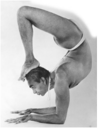
Alfred reads books and Alfred does yoga. $\therefore$ Alfred does yoga.

Alfred reads books and Alfred does yoga. $\therefore$ Alfred does yoga.

$P \;\therefore Q$

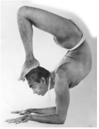
Alfred reads books and Alfred does yoga. ∴ Alfred does yoga.

- R: Alfred reads books.
- Y: Alfred does yoga.



Alfred reads books and Alfred does yoga. $\therefore$ Alfred does yoga.

- R: Alfred reads books.
- Y: Alfred does yoga.



Alfred reads books and Alfred does yoga. $\therefore$ Alfred does yoga.

- R: Alfred reads books.
- Y: Alfred does yoga.

Alfred reads books and Alfred does yoga. $\therefore$ Alfred does yoga.

- R: Alfred reads books.
- Y: Alfred does yoga.

$\neg(R \rightarrow \neg Y) \therefore Y$

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T		
T	F		
F	T		
F	F		

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	
T	F		
F	T		
F	F		

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	
T	F	F	
F	T		
F	F		

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	
T	F	F	
F	T	F	
F	F		

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	
T	F	F	
F	T	F	
F	F	F	

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	T
T	F	F	
F	T	F	
F	F	F	

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	T
T	F	F	F
F	T	F	
F	F	F	

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	

R	Y	$\neg(R \to \neg Y)$	R and Y
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

Functional completeness

- the set of connectives $\{\neg, \rightarrow\}$ is functionally complete
  - other pairs are as well, e.g. $\{\neg, \wedge\}$, $\{\neg, \vee\}$
- for any possible truth-function there is a formula of $\mathscr{L}_1$ that expresses it
- in fact we can get by with a single connective:
  - the Sheffer stroke (`NAND`, `NOR`)

Functional completeness

- the set of connectives $\{\neg, \rightarrow\}$ is functionally complete
  - other pairs are as well, e.g. $\{\neg, \wedge\}$, $\{\neg, \vee\}$
- for any possible truth-function there is a formula of $\mathscr{L}_1$ that expresses it
- in fact we can get by with a single connective:
  - the Sheffer stroke (`NAND`, `NOR`)

Functional completeness

- the set of connectives $\{\neg, \rightarrow\}$ is functionally complete
  - other pairs are as well, e.g. $\{\neg, \wedge\}$, $\{\neg, \vee\}$
- for any possible truth-function there is a formula of $\mathscr{L}_1$ that expresses it
- in fact we can get by with a single connective:
  - the Sheffer stroke (NAND, NOR)

Functional completeness

- the set of connectives $\{\neg, \rightarrow\}$ is functionally complete
  - other pairs are as well, e.g. $\{\neg, \wedge\}$, $\{\neg, \vee\}$
- for any possible truth-function there is a formula of $\mathscr{L}_1$ that expresses it
- in fact we can get by with a single connective:
  - the Sheffer stroke (`NAND`, `NOR`)

Functional completeness

- the set of connectives $\{\neg, \rightarrow\}$ is functionally complete
  - other pairs are as well, e.g. $\{\neg, \wedge\}$, $\{\neg, \vee\}$
- for any possible truth-function there is a formula of $\mathscr{L}_1$ that expresses it
- in fact we can get by with a single connective:
  - the Sheffer stroke (`NAND`, `NOR`)

Truth-functional logic

- Sentence letters are assigned **truth-values**
- Truth-values: T, F
- The sentential connectives, $\neg, \rightarrow$, are interpreted as **functions**
  - they map truth-values to truth-values

Truth-functional logic

- Sentence letters are assigned **truth-values**
- Truth-values: T, F
- The sentential connectives, $\neg, \rightarrow$, are interpreted as **functions**
  - they map truth-values to truth-values

Truth-functional logic

- Sentence letters are assigned **truth-values**
- Truth-values: T, F
- The sentential connectives, $\neg, \rightarrow$, are interpreted as **functions**
  - they map truth-values to truth-values

Truth-functional logic

- Sentence letters are assigned **truth-values**
- Truth-values: T, F
- The sentential connectives, $\neg, \rightarrow$, are interpreted as **functions**
  - they map truth-values to truth-values

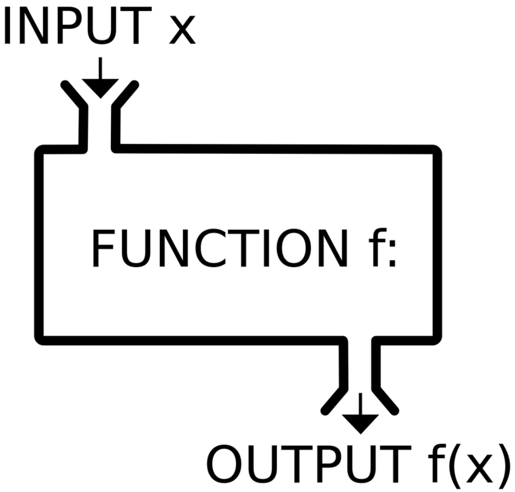
- Sentence letters are assigned **truth-values**
- Truth-values: T, F
- The sentential connectives, $\neg, \rightarrow$, are interpreted as **functions**
  - they map truth-values to truth-values

Truth-functional logic

- Sentence letters are assigned **truth-values**
- Truth-values: T, F
- The sentential connectives, $\neg, \rightarrow$, are interpreted as **functions**
  - they map truth-values to truth-values

What is a function?

What is a function?



Functions

- What is a **function**?

Functions

- What is a **function**?

Functions

- What is a **function**?
- A function takes things of some specified type and via some procedure outputs things of some specified type.

Functions

- What is a **function**?
- A function takes things of some specified type and via some procedure outputs things of some specified type.
- Examples of function:

Functions

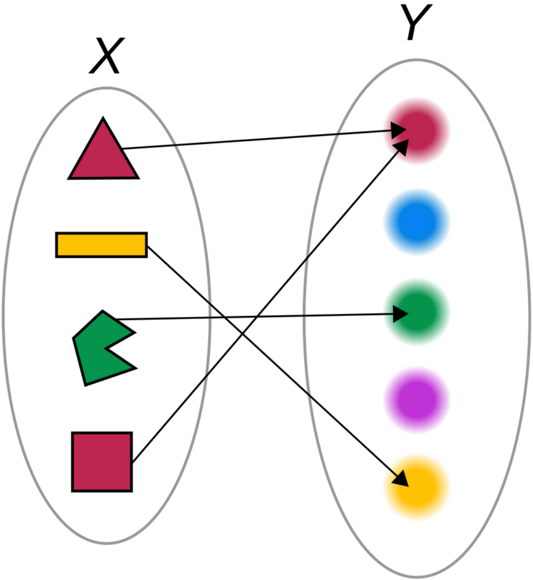
- What is a **function**?
- A function takes things of some specified type and via some procedure outputs things of some specified type.
- Examples of function:
  - Addition: Input (2,3), output 5

Functions

- What is a **function**?
- A function takes things of some specified type and via some procedure outputs things of some specified type.
- Examples of function:
  - Addition: Input (2,3), output 5
  - Student-to-birthday function: Input student, output birthday

Functions

- What is a **function**?
- A function takes things of some specified type and via some procedure outputs things of some specified type.
- Examples of function:
  - Addition: Input (2,3), output 5
  - Student-to-birthday function: Input student, output birthday
- Function: an input type, an output type, and a special mapping.


X
Y

Truth functions

- Inputs: $\{T, F\}$
- Outputs: $\{T, F\}$

- $g(T) = F$
- $g(F) = T$

Truth functions

- Inputs: $\{T, F\}$
- Outputs: $\{T, F\}$
- $g(T) = F$
- $g(F) = T$

Truth functions

- Inputs: $\{T, F\}$
- Outputs: $\{T, F\}$

- $g(T) = F$
- $g(F) = T$

Truth functions

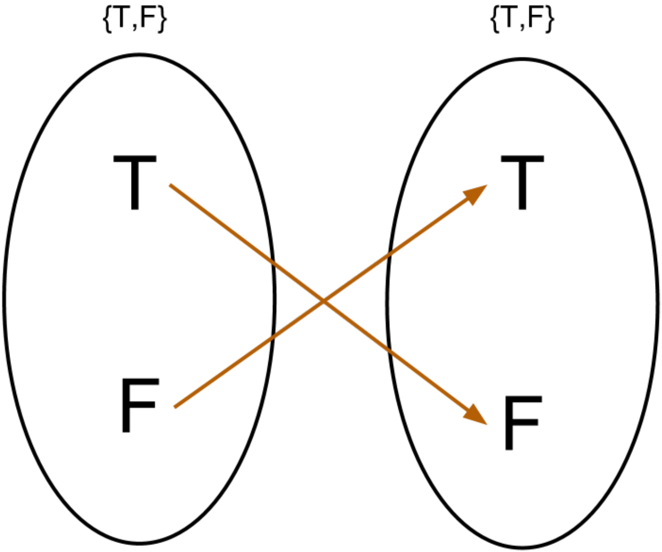
- Inputs: $\{T, F\}$
- Outputs: $\{T, F\}$

- $g(T) = F$
- $g(F) = T$

Truth functions

- Inputs: $\{T, F\}$
- Outputs: $\{T, F\}$

- $g(T) = F$
- $g(F) = T$


{T,F}
{T,F}
T
F
T
F

Truth functions

- Inputs: $\{(T,T),(T,F),(F,T),(F,F)\}$
- Outputs: $\{T,F\}$

- $h(T,T) = T$
- $h(T,F) = F$
- $h(F,T) = F$
- $h(F,F) = F$

Truth functions

- Inputs: $\{(T, T), (T, F), (F, T), (F, F)\}$
- Outputs: $\{T, F\}$
- $h(T, T) = T$
- $h(T, F) = F$
- $h(F, T) = F$
- $h(F, F) = F$

Truth functions

- Inputs: $\{(T, T), (T, F), (F, T), (F, F)\}$
- Outputs: $\{T, F\}$
- $h(T, T) = T$
- $h(T, F) = F$
- $h(F, T) = F$
- $h(F, F) = F$

Truth functions

- Inputs: $\{(T,T),(T,F),(F,T),(F,F)\}$
- Outputs: $\{T,F\}$

- $h(T,T)=T$
- $h(T,F)=F$
- $h(F,T)=F$
- $h(F,F)=F$

Truth functions

- Inputs: $\{(T, T), (T, F), (F, T), (F, F)\}$
- Outputs: $\{T, F\}$

- $h(T, T) = T$
- $h(T, F) = F$
- $h(F, T) = F$
- $h(F, F) = F$

Truth functions

- Inputs: $\{(T, T), (T, F), (F, T), (F, F)\}$
- Outputs: $\{T, F\}$

- $h(T, T) = T$
- $h(T, F) = F$
- $h(F, T) = F$
- $h(F, F) = F$

Truth functions

- Inputs: $\{(T, T), (T, F), (F, T), (F, F)\}$
- Outputs: $\{T, F\}$

- $h(T, T) = T$
- $h(T, F) = F$
- $h(F, T) = F$
- $h(F, F) = F$

Many, many, truth-functions

- 4 unary truth functions
- 16 binary truth functions
- 256 ternary truth functions
- ...
- $2^{(2^n)}$ n-ary truth-functions

Many, many, truth-functions

- 4 unary truth functions
- 16 binary truth functions
- 256 ternary truth functions
- ...
- $2^{(2^n)}$ n-ary truth-functions

Many, many, truth-functions

- 4 unary truth functions
- 16 binary truth functions
- 256 ternary truth functions
- ...
- $2^{(2^n)}$ n-ary truth-functions

Many, many, truth-functions

- 4 unary truth functions
- 16 binary truth functions
- 256 ternary truth functions
- ...
- $2^{(2^n)}$ n-ary truth-functions

Many, many, truth-functions

- 4 unary truth functions
- 16 binary truth functions
- 256 ternary truth functions
- ...
- $2^{(2^n)}$ n-ary truth-functions

Many, many, truth-functions

- 4 unary truth functions
- 16 binary truth functions
- 256 ternary truth functions
- ...
- $2^{(2^n)}$ *n*-ary truth-functions

‘and’, ‘or’

- Alfred either read a book or Alfred did yoga. Alfred didn’t do yoga. So Alfred read a book.
- Alfred is smart. Alfred is overconfident. So, Alfred is smart but overconfident.

‘and’, ‘or’

- Alfred either read a book or Alfred did yoga. Alfred didn’t do yoga. So Alfred read a book.
- Alfred is smart. Alfred is overconfident. So, Alfred is smart but overconfident.

‘and’, ‘or’

- Alfred either read a book or Alfred did yoga. Alfred didn’t do yoga. So Alfred read a book.
- Alfred is smart. Alfred is overconfident. So, Alfred is smart but overconfident.

$\mathscr{L}_2$: The language of ‘if’, ‘not’, ‘and’, ‘or, and ‘iff’

- Basic symbols:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- Formation rules:
  - Any sentence letter is a sentence
  - If $\Box$ is a sentence, then $\neg\Box$ is a sentence
  - If $\Box$ and $\bigcirc$ are a sentences, then:
    - $(\Box \rightarrow \bigcirc)$ is a sentence
    - $(\Box \wedge \bigcirc)$ is a sentence
    - $(\Box \vee \bigcirc)$ is a sentence
    - $(\Box \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of ‘if’, ‘not’, ‘and’, ‘or, and ‘iff’

- Basic symbols:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- Formation rules:
  - Any sentence letter is a sentence
  - If $\Box$ is a sentence, then $\neg\Box$ is a sentence
  - If $\Box$ and $\bigcirc$ are a sentences, then:
    - $(\Box \rightarrow \bigcirc)$ is a sentence
    - $(\Box \wedge \bigcirc)$ is a sentence
    - $(\Box \vee \bigcirc)$ is a sentence
    - $(\Box \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of ‘if’, ‘not’, ‘and’, ‘or, and ‘iff’

- BASIC SYMBOLS:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- FORMATION RULES:
  - Any sentence letter is a sentence
  - If $\square$ is a sentence, then $\neg\square$ is a sentence
  - If $\square$ and $\bigcirc$ are a sentences, then:
    - $(\square \rightarrow \bigcirc)$ is a sentence
    - $(\square \wedge \bigcirc)$ is a sentence
    - $(\square \vee \bigcirc)$ is a sentence
    - $(\square \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- Basic symbols:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- Formation rules:
  - Any sentence letter is a sentence
  - If $\square$ is a sentence, then $\neg\square$ is a sentence
  - If $\square$ and $\bigcirc$ are a sentences, then:
    - $(\square \rightarrow \bigcirc)$ is a sentence
    - $(\square \wedge \bigcirc)$ is a sentence
    - $(\square \vee \bigcirc)$ is a sentence
    - $(\square \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- **Basic symbols:**
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- **Formation rules:**
  - Any sentence letter is a sentence
  - If $\Box$ is a sentence, then $\neg\Box$ is a sentence
  - If $\Box$ and $\bigcirc$ are a sentences, then:
    - $(\Box \rightarrow \bigcirc)$ is a sentence
    - $(\Box \wedge \bigcirc)$ is a sentence
    - $(\Box \vee \bigcirc)$ is a sentence
    - $(\Box \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: `The language of ‘if’, ‘not’, ‘and’, ‘or, and ‘iff’`

- Basic symbols:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- Formation rules:
  - Any sentence letter is a sentence
  - If $\Box$ is a sentence, then $\neg\Box$ is a sentence
  - If $\Box$ and $\bigcirc$ are a sentences, then:
    - $(\Box \rightarrow \bigcirc)$ is a sentence
    - $(\Box \wedge \bigcirc)$ is a sentence
    - $(\Box \vee \bigcirc)$ is a sentence
    - $(\Box \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- Basic symbols:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- Formation rules:
  - Any sentence letter is a sentence
  - If $\square$ is a sentence, then $\neg\square$ is a sentence
  - If $\square$ and $\bigcirc$ are a sentences, then:
    - $(\square \rightarrow \bigcirc)$ is a sentence
    - $(\square \wedge \bigcirc)$ is a sentence
    - $(\square \vee \bigcirc)$ is a sentence
    - $(\square \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- BASIC SYMBOLS:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- FORMATION RULES:
  - Any sentence letter is a sentence
  - If □ is a sentence, then ¬□ is a sentence
  - If □ and ○ are a sentences, then:
    - (□ → ○) is a sentence
    - (□ ∧ ○) is a sentence
    - (□ ∨ ○) is a sentence
    - (□ ↔ ○) is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- BASIC SYMBOLS:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- FORMATION RULES:
  - Any sentence letter is a sentence
  - If $\square$ is a sentence, then $\neg\square$ is a sentence
  - If $\square$ and $\bigcirc$ are a sentences, then:
    - $(\square \rightarrow \bigcirc)$ is a sentence
    - $(\square \wedge \bigcirc)$ is a sentence
    - $(\square \vee \bigcirc)$ is a sentence
    - $(\square \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- **Basic symbols:**
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- **Formation rules:**
  - Any sentence letter is a sentence
  - If $\Box$ is a sentence, then $\neg\Box$ is a sentence
  - If $\Box$ and $\bigcirc$ are a sentences, then:
    - $(\Box \rightarrow \bigcirc)$ is a sentence
    - $(\Box \wedge \bigcirc)$ is a sentence
    - $(\Box \vee \bigcirc)$ is a sentence
    - $(\Box \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

$\mathscr{L}_2$: The language of 'if', 'not', 'and', 'or, and 'iff'

- Basic symbols:
  - Sentence letters: $P, \ldots, Z$
    - (including numerical subscripts, e.g. P_4, Q_{23}, etc.)
  - Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
  - Punctuation:),(
- Formation rules:
  - Any sentence letter is a sentence
  - If $\square$ is a sentence, then $\neg\square$ is a sentence
  - If $\square$ and $\bigcirc$ are a sentences, then:
    - $(\square \rightarrow \bigcirc)$ is a sentence
    - $(\square \wedge \bigcirc)$ is a sentence
    - $(\square \vee \bigcirc)$ is a sentence
    - $(\square \leftrightarrow \bigcirc)$ is a sentence
  - Nothing else is a sentence of $\mathscr{L}_2$ unless it can be constructed by means of these rules.

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
    - A conjunction says that both of the propositions are true
- P and Q
    - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- “James is angry and Ann is happy”
- ‘and’ is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

Conjunction

- "James is angry and Ann is happy"
- 'and' is a connective used to combine two propositions
  - A conjunction says that both of the propositions are true
- P and Q
  - P, Q are **conjuncts**
- $(P \wedge Q)$

∧

- Truth table for conjunction

$\square$	$\bigcirc$	$(\square \wedge \bigcirc)$
T	T	T
T	F	F
F	T	F
F	F	F

$\wedge$

- Truth table for conjunction

$\square$	$\bigcirc$	$(\square \wedge \bigcirc)$
T	T	T
T	F	F
F	T	F
F	F	F

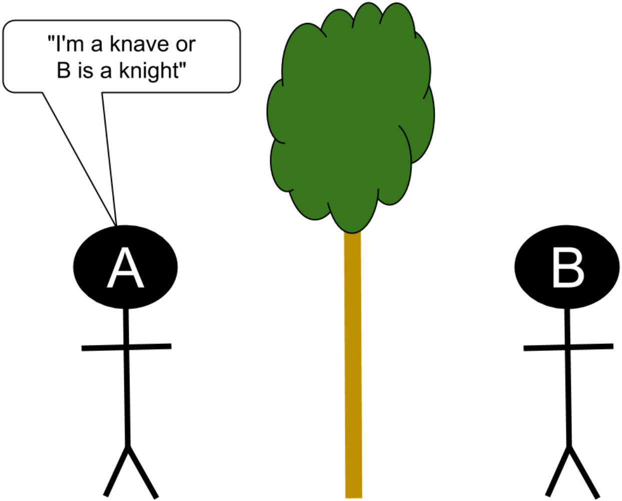
$\wedge$

- Truth table for conjunction

$\square$	$\bigcirc$	$(\square \wedge \bigcirc)$
T	T	T
T	F	F
F	T	F
F	F	F

Puzzle. Two of the inhabitants—A and B—are standing together under a tree. A makes the following statement: "Either I am a knave or B is a knight."

What are A and B?


"I'm a knave or
B is a knight"
A
B

OR

- **Exclusive OR**: "I'll either marry Elle or Ruby (but not both)"
- **Inclusive OR**: "Must either pass a logic course or a foreign language course (or both)"

P	Q	P XOR Q
true	*true*	*false*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

P	Q	$P \vee Q$
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

OR

- **Exclusive OR**: "I'll either marry Elle or Ruby (but not both)"
- **Inclusive OR**: "Must either pass a logic course or a foreign language course (or both)"

P	Q	P XOR Q
true	*true*	*false*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

P	Q	$P \vee Q$
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

OR

- **Exclusive OR**: "I'll either marry Elle or Ruby (but not both)"
- **Inclusive OR**: "Must either pass a logic course or a foreign language course (or both)"

P	Q	P XOR Q
true	*true*	*false*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

P	Q	$P \vee Q$
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

OR

- **Exclusive OR**: "I'll either marry Elle or Ruby (but not both)"
- **Inclusive OR**: "Must either pass a logic course or a foreign language course (or both)"

P	Q	$P \text{ XOR } Q$
true	*true*	*false*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

P	Q	$P \vee Q$
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

OR

- **Exclusive OR**: "I'll either marry Elle or Ruby (but not both)"
- **Inclusive OR**: "Must either pass a logic course or a foreign language course (or both)"

P	Q	P XOR Q
true	*true*	*false*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

P	Q	$P \vee Q$
true	*true*	*true*
true	*false*	*true*
false	*true*	*true*
false	*false*	*false*

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

Disjunction

- "James is angry or Ann is happy"
- 'or' is a connective used to combine two propositions
  - A disjunction says that **at least one** of two propositions is true (inclusive OR)
- P or Q
  - P, Q are **disjuncts**
- $(P \vee Q)$

$\vee$

- Truth-table for disjunction

$\square$	$\bigcirc$	$(\square \vee \bigcirc)$
T	T	T
T	F	T
F	T	T
F	F	F

$\vee$

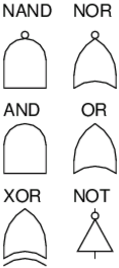
- Truth-table for disjunction

$\Box$	$\bigcirc$	$(\Box \vee \bigcirc)$
T	T	T
T	F	T
F	T	T
F	F	F

$\vee$

- Truth-table for disjunction

$\square$	$\bigcirc$	$(\square \vee \bigcirc)$
T	T	T
T	F	T
F	T	T
F	F	F


NAND
NOR
AND
OR
XOR
NOT