‘And’, ‘Or’ and ‘Not’

January 22nd, 2004

$\wedge$, $\vee$, $\neg$

- ‘$\wedge$’ is the *conjunction symbol* — read ‘and’.
- ‘$\vee$’ is the *disjunction symbol* — read ‘or’ or ‘either...or...’
- ‘$\neg$’ is the *negation symbol* — read ‘not’ or ‘it is not the case that’.
    - Collectively, these are known as the *Boolean connectives*, after the 19th century logician George Boole

Syntax

- '$\wedge$' and '$\vee$' are *binary connectives*: they take two sentences and make a new sentence.
  - Whenever P and Q are sentences, (P) $\wedge$ (Q) and (P) $\vee$ (Q) are also sentences.
  - Contrast the English words 'and' and 'or', which can also conjoin verb phrases ('Lassie barked and howled') and noun phrases ('Dean or Kerry will win').
- '$\neg$' is a *unary connective:* it takes one sentence and makes a new sentence.
  - Whenever P is a sentence, $\neg$(P) is a sentence.
  - Contrast the English word 'not', which can occur in all sorts of places.

- Thus, we can build up arbitrarily complicated sentences:
  - $\neg(White(snow))$
  - $(\neg(White(snow))) \vee (Green(grass))$
  - $(\neg(White(snow))) \vee (Green(grass)) \wedge (\neg(\neg(\neg(Red(blood)))))$
  - etc.

Semantics

- $(P) \wedge (Q)$ is true when both P and Q are true; it is false when either P or Q is false.
  - Unlike the English 'and', there's never any suggestion about temporal order.
- $(P) \vee (Q)$ is true when one or both of P and Q is true; it is false when both of them are false.
  - Unlike the English 'or', there's never any suggestion that it's not the case that both P and Q are true. ($\vee$ is 'inclusive' rather than 'exclusive')
- $\neg(P)$ is true when P is false; it is false when P is true.

- We can sum this up in the form of *truth-tables*:

P	Q	$(P) \wedge (Q)$	$(P) \vee (Q)$
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

P	$\neg(P)$
TRUE	FALSE
FALSE	TRUE

Parentheses

- The notation I've just described is quite hard to read because of all the parentheses.
- But we can't just leave out the parentheses altogether. This would leave us with apparently ambiguous sentences like
  - State(guam) $\wedge$ State(puertorico) $\vee$ State(pennsylvania).
- We'll adopt a more liberal notation that requires parentheses only when they're required to avoid ambiguity.

- We allow '¬' to be followed by a sentence not in parentheses. In these circumstances '¬' takes *narrowest scope*:
  - ¬State(guam) ∧ State(idaho) is equivalent to ¬(State(guam)) ∧ State(idaho).
  - Like '-' in algebra.
- We allow '∧' and '∨' to link sentences not in parentheses, when those sentences start with the negation symbol.
- We allow multiple sentences to be joined by '∧' and '∨', thus:
  - State(guam) ∨ State(puertorico) ∨ State(alaska)
  - Just like '+' and '×' in alegebra.

The Boolean connectives in *Tarski's World*

The Henkin-Hintikka game

- The point of this is to give you another way to understand why your sentences get the truth-values they do in *TW*.
- If you think a sentence should be true in a world, but *TW* says it's false, you can play the game to figure out where you're going wrong.

For next week

- Read: sections 3.1-3.8, and if you want to read ahead, 4.1 -4.4; do the *You try it* exercises.
- Do: exercises 2.22, 2.24 - 2.27, 3.6, 3.9, 3.13 - 3.15 (10% each)