The Language of PC

Syntax

Symbols

Non-logical
Sentence letters: ‘A’, ‘B’, ‘C’, ‘P_1’, ‘P_2’, ‘R’, …
Logical
Connectives: ‘$\neg$’ (or ‘~’), ‘&’ (or ‘•’, ‘$\wedge$’), ‘$\vee$’, ‘$\rightarrow$’ (or ‘$\supset$’), ‘$\leftrightarrow$’ (or ‘$\equiv$’).
Punctuation
Brackets: ‘[’ , ‘]’ (or ‘(’, ‘)’).

Well-formed formulae

(1) Every sentence letter is a wff.
(2) If φ is a wff then so is ‘$\neg\varphi$’.
(3) If φ and ψ are wffs then so are ‘[φ & ψ]’, ‘[$\varphi \vee \psi$]’, ‘[$\varphi \rightarrow \psi$]’, and ‘[$\varphi \leftrightarrow \psi$]’.
(4) Nothing else is a wff.

Because the wffs are specified recursively in this manner, each wff has a constituent structure which can be represented as a tree. Of particular interest are the *immediate* constituents of a wff.

Example: ‘[A & [B $\vee$ C]]’

Semantics

We give meaning to the language (interpret it) by giving a meaning to each wff. This is not done case-by-case; rather, it is done compositionally by giving rules for how the meaning of a wff is determined from the meaning of its immediate constituents. The rules we use are:

(1) The meaning of a sentence letter is a truth value, T or F.
(2) m(‘$\neg$’) = the f such that $f(x)$ = T iff x = F.
(3) m(‘&’) = the f such that $f(x, y)$ = T iff x = T and y = T.
(4) m(‘$\vee$’) = the f such that $f(x, y)$ = T iff x = T or y = T.
(5) m(‘$\rightarrow$’) = the f such that $f(x, y)$ = T iff x = F or y = T.
(6) m(‘$\leftrightarrow$’) = the f such that $f(x, y)$ = T iff x = T and y = T or x = F and y = F.
(7) If C is a 1-place connective then m(‘Cφ’) = m(C)(m(φ)) (function application)
(8) If C is a 2-place connective then m(‘[φ C ψ]’ = m(C)(m(φ), m(ψ))

Note: the meaning of the 1-place connective (‘$\neg$’) is a 1-place function from truth values to truth values; the meaning of each 2-place connective (‘&’, ‘$\vee$’, ‘$\rightarrow$’ , ‘$\leftrightarrow$’) is a 2-place function from pairs of truth values to truth values. We can represent these functions using *truth tables*:

x	y	m('¬')(x)	m('&')(x, y)	m('∨')(x, y)	m('→')(x, y)	m('↔')(x, y)
T	T	F	T	T	T	T
T	F		F	T	F	F
F	T	T	F	T	T	F
F	F		F	F	T	T

The connectives are always interpreted in the same way. So interpreting the language amounts to interpreting just the sentence letters – that is, assigning a truth value to each.

Entailment. If Γ is a set of wffs (possibly empty) and φ is a wff, then define 'Γ ⊨ φ' to mean that there is no interpretation on which each wff in Γ is true and yet φ is false. If Γ ⊨ φ, say that Γ *entails* φ, or that φ *follows from* Γ, or that φ *is a consequence of* Γ. Also say that 'Γ ⊨ φ' is a *correct semantic sequent*.

Examples of correct semantic sequents: ¬A ⊨ ¬[A & B]; A, [A → B], [B → C] ⊨ C; ⊨ [A → [B → A]]; ⊨ [A ∨ ¬A]; [A → B] ⊨ [¬B → ¬A]; ⊨ [¬A → [A → B]]; ¬A ⊨ [A → B].

Examples of incorrect semantic sequents: [A ∨ B] ⊨ A; [A ∨ B], A ⊨ ¬B; [A → B], B ⊨ A; [A → B], ¬A ⊨ ¬B.

Deductive system

If Γ is a set of wffs (possibly empty) and φ is a wff then define 'Γ ⊢ φ' to mean that φ can be deduced from Γ using the rules of deduction of the system. If Γ ⊢ φ then say that 'Γ ⊢ φ' is a *correct syntactic sequent*.

Rules of deduction:

Modus Ponens (MP)	From [φ → ψ] and φ deduce ψ
Modus Tollens (MT)	From [φ → ψ] and ¬ψ deduce ¬φ
Hypoth. Syllogism (HS)	From [φ → ψ] and [ψ → ξ] deduce [φ → ξ]
Disjunctive Syllogism (DS)	From [φ ∨ ψ] and ¬φ deduce ψ
Constructive Dilemma (CD)	From [[φ → ψ] & [ξ → σ]] and [φ ∨ ξ] deduce [ψ ∨ σ]
Absorption (Abs)	From [φ → ψ] deduce [φ → [φ & ψ]]
Simplification (Simp)	From [φ & ψ] deduce φ
Conjunction (Conj)	From φ and ψ deduce [φ & ψ]
Addition (Add)	From φ deduce [φ ∨ ψ]
De Morgan's Theorems (DeM)	From ¬[φ & ψ] deduce [¬φ ∨ ¬ψ] (and vice-versa) From ¬[φ ∨ ψ] deduce [¬φ & ¬ψ] (and vv)
Commutation (Com)	From [φ & ψ] deduce [ψ & φ] (and vv) From [φ ∨ ψ] deduce [ψ ∨ φ] (and vv)
Association (Assoc)	From [φ & [ψ & ξ]] deduce [[φ & ψ] & ξ] (and vv) From [φ ∨ [ψ ∨ ξ]] deduce [[φ ∨ ψ] ∨ ξ] (and vv)
Distribution (Dist)	From [φ & [ψ ∨ ξ]] deduce [[φ & ψ] ∨ [φ & ξ]] (and vv) From [φ ∨ [ψ & ξ]] deduce [[φ ∨ ψ] & [φ ∨ ξ]] (and vv)
Double Negation (DN)	From φ deduce ¬¬φ (and vv)

Transposition (Trans)	From [φ → ψ] deduce [¬ψ → ¬φ] (and vv)
Material Implication (Impl)	From [φ → ψ] deduce [¬φ ∨ ψ] (and vv)
Material Equivalence (Equiv)	From [φ ↔ ψ] deduce [[φ → ψ] & [ψ → φ]] (and vv) From [φ ↔ ψ] deduce [[φ & ψ] ∨ [¬φ & ¬ψ]] (and vv)
Exportation (Exp)	From [[φ & ψ] → ξ] deduce [φ → [ψ → ξ]] (and vv)
Tautology (Taut)	From φ deduce [φ ∨ φ] (and vv) From φ deduce [φ & φ] (and vv)
Conditional proof (CP)	From [φ/∴ψ] deduce [φ → ψ]
Reductio ad absurdum (RA)	From [φ/∴[ψ & ¬ψ]] deduce ¬φ

Examples of correct syntactic sequents: ¬A ⊢ ¬[A & B]; A, [A → B], [B → C] ⊢ C; ⊢ [A → [B → A]]; ⊢ [A ∨ ¬A]; [A → B] ⊢ [¬B → ¬A]; ⊢ [¬A → [A → B]]; ¬A ⊢ [A → B].

Soundness and Completeness of the system

We want a deductive system that is *sound* and *complete*.

Soundness. If Γ ⊢ φ then Γ ⊨ φ. That is: if the deductive system allows us to deduce the wff φ from the set of wffs Γ (possibly empty), then there is no interpretation in which all the members of Γ are true but φ is false.

Completeness. If Γ ⊨ φ then Γ ⊢ φ. That is: if φ is a wff and Γ is a (possibly empty) set of wffs, and there is no interpretation in which all the members of Γ are true but φ is false, then the deductive system allows us to deduce φ from Γ.

<u>Exercises</u>

1. Which of the following formulae are wffs?
   a. [→AB]
   b. A & B
   c. ¬[A ∨ B]
   d. A → [A ∨ B]
   e. [¬[A & B]]
   f. ¬[A → ¬A]
   g. [AB&]

2. Draw a tree to show the constituent structure of the following wffs:
   a. [[A & A] & ¬C]
   b. [[A & B] → ¬[A ∨ B]]
   c. ¬¬[A ∨ [A → [A & A]]]

3. Which of the following semantic sequents are correct? Provide counterexamples to those that are not:
   a. [A → B], [C & ¬B] ⊨ ¬A
   b. [¬A ∨ B], ¬A ⊨ B
   c. [A & B], ¬B ⊨ ¬A
   d. [A ∨ B], [A → C], [B → C] ⊨ C

e. $[A \to \neg A] \models \neg A$
f. $[A \to [B \vee C]] \models [A \to B]$

4. Prove the following by valid deduction:
   a. $[A \to B], [C \& \neg B] \vdash \neg A$
   b. $[A \vee B], [A \to C], [B \to C] \vdash C$
   c. $[A \to \neg B], \neg[C \& \neg A] \vdash [C \to \neg B]$
   d. $[[G \to \neg H] \to I], \neg[G \& H] \vdash [I \vee \neg H]$
   e. $[J \vee [\neg J \& K]], [J \to L] \vdash [[L \& J] \leftrightarrow J]$
   f. $[[R \vee S] \to [T \& U]], [\neg R \to [V \to \neg V]], \neg T \vdash \neg V$

5. Translate the following English language sentences into wffs of PC:
   a. If John and Mary go to the movies then John will go to the movies.
   b. I won't go unless you come with me.
   c. If it's raining it must be cloudy.

6. Translate and prove:[1]
   a. Either the manager didn't notice the change or else he approves of it. He noticed it all right. So he must approve of it. (N, A)
   b. If the first disjunct of a disjunction is true, then the disjunction as a whole is true. Therefore, if both the first and second disjuncts of the disjunction are true, then the disjunction as a whole is true. (F, W, S)
   c. Jones will come if she gets the message, provided that she is still interested. Although she didn't come, she is still interested. Therefore she didn't get the message. (C, M, I)
   d. Either the robber came in the door, or else the crime was an inside one and one of the servants is implicated. The robber could come in the door only if the latch had been raised from the inside; but one of the servants is surely implicated if the latch was raised from the inside. Therefore one of the servants is implicated. (D, I, S, L)
   e. If either Socrates was happily married or else he wasn't, then Socrates was a great philosopher. Therefore Socrates was a great philosopher. (H, G)

[1] Examples taken from Copi, I. M. and Cohen, C. (1998), *Introduction to Logic*, 10th edition (Upper Saddle River, NJ: Prentice-Hall), pp. 413-7.