Peter Smith, *Introduction to Formal Logic* (CUP, 2nd edition)

Exercises 10: PL semantics: evaluating wffs

(a) Suppose we are working in a PL language where 'P' means *Fred is a fool*; 'Q' means *Fred knows some logic*; 'R' means *Fred is a rocket scientist.* Translate the following sentences into this formal language as best you can. What do you think is lost in the translations, if you can only use the 'colourless' connectives '$\land$', '$\lor$' and '$\neg$'?

(1) Even Fred is a rocket scientist.

(2) Fred is a rocket scientist, but he knows no logic.

(3) Fred is a rocket scientist, moreover he knows some logic.

(4) Fred's a fool, even though he knows some logic.

(5) Although Fred's a rocket scientist, he's a fool and even knows no logic.

(6) Fred's a fool, yet he's a rocket scientist who knows some logic.

(7) Fred is a fool despite the fact that he knows some logic.

(8) Fred is not a rocket scientist who knows some logic.

(9) Fred knows some logic unless he is a fool.

(b) Confirm that the following strings are wffs by producing parse trees. Suppose that $\mathsf{P} \coloneqq$ T, $\mathsf{Q} \coloneqq$ F, $\mathsf{R} \coloneqq$ T. Evaluate the wffs first by chasing values up the trees. Then do the working again in the short form (i.e. as a mini-table, skipping redundant working when you can).

(1) $((\mathsf{R} \lor \neg\mathsf{Q}) \land (\mathsf{Q} \lor \mathsf{P}))$

(2) $\neg(\mathsf{P} \lor ((\mathsf{Q} \land \neg\mathsf{P}) \lor \mathsf{R}))$

(3) $\neg(\neg\mathsf{P} \lor \neg(\mathsf{Q} \land \neg\mathsf{R}))$

(4) $(\neg(\mathsf{P} \land \neg\mathsf{Q}) \land \neg\neg\mathsf{R})$

(5) $(((\mathsf{P} \lor \neg\mathsf{Q}) \land (\mathsf{Q} \lor \mathsf{R})) \lor \neg\neg(\mathsf{Q} \lor \neg\mathsf{R}))$

Work out, in short form, the truth values of the following wffs on the assignment of values $\mathsf{P} \coloneqq$ F, $\mathsf{Q} \coloneqq$ F, $\mathsf{R} \coloneqq$ T, $\mathsf{S} \coloneqq$ F

(6) $\neg((\mathsf{P} \lor \mathsf{Q}) \land \neg(\neg\mathsf{Q} \lor \mathsf{R}))$

(7) $\neg\neg((\mathsf{P} \land \mathsf{Q}) \lor (\neg\mathsf{S} \lor \neg\mathsf{R}))$

(8) $(((\mathsf{S} \land \mathsf{Q}) \land \neg\neg\mathsf{R}) \lor \neg\mathsf{Q})$

(9) $(((\mathsf{P} \land (\mathsf{Q} \land \neg\mathsf{R})) \lor \neg\neg\neg(\mathsf{R} \land \mathsf{Q})) \lor (\mathsf{P} \land \mathsf{R}))$

(10) $(\neg((\mathsf{P} \lor (\mathsf{R} \land \neg\mathsf{S})) \lor \neg(\mathsf{Q} \land \neg\mathsf{P})) \land \neg(\mathsf{P} \lor \neg(\neg\mathsf{Q} \lor \mathsf{R})))$

(c*) In this book we have taken a maximalist line about the use of brackets in PL wffs. What conventions could we have adopted (while still writing '$\land$' and '$\lor$' *between* the wffs they connect) in order to reduce the numbers of brackets in a typical wff while not reintroducing semantic ambiguities?

(d*) *Polish notation* for the propositional calculus – introduced by Jan Łukasiewicz in the 1920s – is a bracket-free notation in which connectives are written *before* the wffs they connect.

Traditionally, for the Negation of α we write $N\alpha$; for the Konjunction of α and β we write $K\alpha\beta$; for the disjunction of the Alternatives α and β we write $A\alpha\beta$. Since capital letters are used for connectives, it is customary in Polish notation to use lower case letters for propositional atoms. Hence '$(\neg\mathsf{P} \land \mathsf{Q})$' becomes '$KNpq$', '$\neg(\mathsf{P} \land \mathsf{Q})$' becomes '$NKpq$', '$\neg((\mathsf{P} \land \neg\mathsf{Q}) \lor \mathsf{R})$' becomes '$NAKpNqr$', etc.

(1) Rewrite the syntactic rules of §9.1(c) for a language using Polish notation.

(2) Render the Polish wffs '*KNNpq*', '*NKpNq*', '*AKpqr*', '*ApKqr*', '*AANpNqNr*', '*AKN-pqKpNq*', '*ANKKpqKqrNArs*' into our notation.

(3) Render the wffs (1) to (5) from (b) into Polish notation.

(4) (Difficult!) Show that Polish notation, although bracket-free, introduces no semantic ambiguities (every Polish wff can be parsed in only one way).