Inference Rules as Strategies

This is an articulation of the eight inference rules as strategies in the reverse method. The reverse method instills a habit of approaching problems with *goal-oriented* strategies.

The most important feature of this method is its basis in wants and needs. Strategies are driven by what is desired, that is, by what you *want*. Knowledge of what you *need* comes out of your understanding of what you want.

NOTE: "Introduction" means it builds a statement
"Elimination" means it breaks apart a statement

And Introduction: *Conjunction* (Conj)

If you want a conjunction on a line (☐•△), then introduce it.
You'll need ☐ on one line, and △ on a separate line.

Want ☐•△, then need ☐ as well as △.

And Elimination: *Simplification* (Simp)

If you want part of a conjunction on a line (i.e. one of the conjuncts ☐ , △), then you can *simp*ly take it.

Want ☐ or △, have ☐• △, take it!

Or Introduction: *Addition* (Add)

If you want a disjunction on a line (☐∨△), then introduce it.
You'll need ☐ on a line, or △ on a line.

Want ☐∨△, then need ☐ or △.

Or Elimination: *Disjunctive Syllogism* (DS)

If you want part of a disjunction on a line (i.e. one of the disjuncts ☐,△), then eliminate the disjunction and take the part you want. You'll need the negation of one of its disjuncts ~☐ or ~△ on a line.

Want ☐ or △, have ☐∨△, then need ~☐ (for △) or ~△ (for ☐)

Inference Rules as Strategies

Or Introduction: *Constructive Dilemma* (CD)

If you want a disjunction on a line (○∨☆), then introduce it the hard way. You'll need two conditionals whose antecedents (their □s) are the disjuncts of the disjunction (□∨△), and whose consequents (their △s)are the disjuncts of the disjunction desired (○∨☆), all on separate lines.

Want ○∨☆, then need □→○, as well as △→☆, and □∨△

Conditional Elimination: *Modus Ponens* (MP)

If you want the consequent (the △) of a conditional (□→△) on a line, then eliminate the conditional and cleanly get the consequent. You'll need the antecedent (the □) on a separate line to take the consequent.

Want △, have □→△, then need □.

Conditional Elimination "Backwards": *Modus Tollens* (MT)

If you want the antecedent (the □) of a conditional (□→△) on a line, then that's just too bad, you'll never get it! If, however, you want the *negation* of the antecedent (~□) on a line, then eliminate the conditional and get it. You'll need the negation of the consequent (~△) on a separate line to take the negation of the antecedent.

Want ~ □, have □→△, then need ~△.

Conditional Introduction: *Hypothetical Syllogism* (HS)

If you want the conditional (□→○) on a line, then introduce it by building it from two other conditionals. You'll need one conditional on a line that "starts" with the antecedent you're trying to build (i.e. □), and you need another conditional on a separate line that "ends" with the consequent of the conditional you're trying to build (i.e. ○). Also, *both* conditionals must "share" a common element △, as the consequent of first and as the antecedent of the second.

Want □→○, then need □→△ as well as △→○.