TO-DO LIST INFERENCE ALGORITHM

ROBERT SANDERS

Both the primary inference engine and the propogation of operator partial functions follow the same basic procedure. We begin with some database D_0, which can be taken to be a set of propositions that are regarded as true. We also assume that there is a set R of inference rulses, which are tuples of the form

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi.$$

The inference algorithm that Complexity Zoology employs is as follows:

(1) Populate a list L with the proopositions in D_0, and set $D = \emptyset$.
(2) While L is nonempty, carry out the steps (3) through (6).
(3) Remove the top proposition φ from L.
(4) If $\varphi \in D$, return to step (3).
(5) Add φ to the set D.
(6) For each inference rule $\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi$ and all $\varphi'_1, \varphi'_2, \ldots, \varphi'_{n-1} \in D$, check whether some permutation of $\varphi, \varphi'_1, \ldots, \varphi'_{n-1}$ matches $\varphi_1, \varphi_2, \ldots, \varphi_n$. If it does, append φ to L.

The resulting database D has D_0 as a subset and is closed under inference rules. Moreover, this algorith eventually terminates, because when a proposition has been removed from L once, it cannot again result in any additional proposition being appended to L. Thus, the algorithm deduces all logical consequences of the initial database D_0, and it does so faster than the naive approach of repeatedly applying all inference rules to all the propositions in D until D grows no further.