Constructing Possible Worlds Algorithmically

The story of linguistic ersatzism goes something like this: ‘possible worlds’ can be stepwise constructed (i) and (ii) evaluating a modal claim will typically involve construction only up to a point dependent on the logical structure and the constituent terms of the claim under consideration.

The whole construction concerns a specific language L and a corresponding consequence relation (logic). In the simplest case possible worlds are negation-complete, consistent and deductively closed. In all interesting cases L contains a negation symbol and (at least) one detachable conditional. As possible worlds are interesting only in modal semantics L should have modal operators, in the simplest case with universal accessibility between the possible worlds.

One can model this construction in terms of Turing Machines (TMs). What TMs compute is computed constructively and finitely (apart from the assumption of indefinite storage capacities).

The possible worlds are constructed by a complex TM TM_{pw}, which consists of several sub-machines. The complex TM_{pw} executes alternately chunks (or single steps) of the constituent machines. (Because of the alternating execution some copying, adding and replacing steps have to be repeated.)

The input of TM_{pw} consists of:

1. A list of general terms of the language
2. A list of singular terms of the language
3. A list of the axioms (including nominal definitions and meaning postulates) of the language

The output of TM_{pw} consists of:

1. An indexed list of state descriptions
2. An indexed list of possible worlds

In the second list a supposed possible world that within the construction process below turns out to be inconsistent (and so no possible world after all) is marked closed.

The sub-machines are the following TMs:

1. A TM_{sd} that sets up the state descriptions:
    a. MT_{sd} looks for a general term and a singular term not dealt with and applies the general term to the singular term (giving an atomic sentence); these terms are searched for starting from the list of terms: a general term and a singular term are new if in the list of state descriptions no corresponding atomic fact can be found.
    b. if the list of state descriptions is empty, two entries are created: one with the atomic fact, one without it; otherwise: the list of state descriptions is extended by a self-copy where the first half of the state descriptions are extended by the atomic fact;
    c. MT_{sd} proceeds to either another general term or another singular term (alternately) and goes back to step (a), it stops when all terms have been dealt with (i.e. the search for new terms terminates with failure).

2. A TM_t which enumerates the theorems of the language and adds them to all possible worlds. For all theorems α TM_t adds $\Box\alpha$ to all possible worlds. [TM_t exits by well-known computability theory theorems.]

3. A TM_c which copies from state descriptions to possible worlds:
    a. TM_c goes to the first not treated index of a state description;
    b. If there no possible world with that index a copy of the state description is added with that index to the list of possible worlds;
    c. If there is a possible world with that index and is marked closed TM_c moves to the next index; otherwise: TM_c copies sentences in the state description not present in the possible world to the possible world. [This can be required because of the mutual stepwise construction of state descriptions and possible worlds.]
    d. TM_c goes to the next not treated index or stops otherwise.

4. A TM_{cl} which computes the deductive closure of possible worlds:
    a. TM_{cl} goes to the first not treated index of a possible world;
    b. If the world at that index is marked closed TM_{cl} goes to the next index.
    c. If the world contains sentences with a general term δ and sentences with a singular term γ but not the atomic sentence $\delta(\gamma)$ TM_{cl} adds $\neg\delta(\gamma)$ to that world;
    d. If the world contains conditionals, then for each of them: If the world contains the antecedent of the conditional and the negation of the consequent, then TM_{cl} marks the world as closed, otherwise TM_{cl} adds the consequent to the world. [Introduction of conjunctions and disjunctions happen then by closure with respect to corresponding conditionals derived as theorems.]
    e. For each sentence α in the possible world: if $\Diamond\alpha$ is not contained in the world, $\Diamond\alpha$ is added to the world and to all possible worlds which do not contain $\Diamond\alpha$;
    f. For each sentence α in the possible world TM_{cl} checks whether the sentence is contained in all other possible worlds; if so, $\Box\alpha$ is added to the world, otherwise $\neg\Box\alpha$ is added to the world;
    g. For each sentence of the form $\Box\alpha$ in the world TM_{cl} checks whether α is contained in all other worlds; if not so, $\Box\alpha$ is replaced by $\neg\Box\alpha$. [By the stepwise construction of possible worlds non necessary sentences can transiently seem being necessary.]
    h. TM_{cl} goes to the next not treated index or stops otherwise.

If one was to develop the machine tables of the TMs involved in detail one would need to program immense amounts of copying and shifting of contents, because of the extension of state descriptions and possible worlds. The running time in steps of computation for any mildly complex language L will be astronomical.

The point of the outline of an algorithm of possible world construction is not, however, to proceed to program it in detail and use the output in an philosophical inquiry. The description of TM_{pw} is *a proof in principle* that such a construction is available.

MB, 2024.