Undecidability and Complexity for Super-Turing Models of Computation

Eugene Eberbach

Dept. of Eng. and Science, Rensselaer Polytechnic Institute, Hartford, CT, USA eeberbach@gmail.com

We introduce 3 new classes of TM undecidable problems, inspired by the NP-complete class definition.

Definition 1. (on U-complete languages)
We say a language Lis U-complete (Universal Turing Machine complete) iff

1. Any word w can be decided in a finite number of steps if $w \in L$, or it requires an infinite number ofsteps if$w \notin L$ (semi-decidability condition).
2. For any languagesatisfying (1) there is p-decidable, or e-decidable reduction of L' to L(completeness condition).

Examples of U-complete languages include L_u (a basic representative to call the whole class), PCP, L_{ne}, BBP, ECP, planning problem, optimization problem.

The **U-hard languages**, a superset of U-complete languages, satisfy only completeness condition from the above definition.

Definition 2. (on D-complete languages)
We say a language Lis D-complete (Diagonalization complete) iff

1. Any word wfrom L cannot be decided in a finite number of steps(undecidability condition).
2. For any language satisfying (1) there is p-decidable, or e-decidablereduction of L' to L (completeness condition).

Examples of D-complete languages include L_d (a basic representative to call the whole class), L_e, EIP, complement of L_d, complement of L_u, complement of BBP.

The **D-hard languages**, a superset of D-complete languages, satisfy only completeness condition from the above definition.

Definition 3. (Thehyper-diagonalization language)
The hyper-diagonalization language L_{hd} consists of all strings w such that TM M whose codeis wwill not accept even in an infinite number of steps when given w as input.

Definition 4. (on H-complete languages)
We say a language Lis H-complete (Hypercomputation complete) iff

1. Any word wfrom or outside of Lcannot be decided in a finite numberof steps (undecidability condition).
2. For any language L' satisfying (1) there is an a-decidable ori-decidable reduction of L' to L (completeness condition).

A canonical representative of this class is L_{hd} .

The **H-hard languages**, a superset of H-complete languages, satisfy only completeness condition from the above definition.

Definition 5. A word w is accepted in the *terminal mode* of the automaton E if given theword w as input to the automaton E, there is a number n such that the automaton A[n] from E comes to an accepting state.

Definition 6. The terminal language TL(E) of the automaton E is the set of all words acceptedin the terminal mode of the automaton E.

It has been proven that Evolutionary Automata (e.g., Evolutionary Turing Machines or Evolutionary Finite Automata) and Interaction Machines accept arbitrary languages over a given alphabet.

Theorem 1. Terminal languages of Evolutionary Automata and Interaction Machines coincide with the class of all languages in the alphabet X.

We believe that analogous proofs can be derived for $-calculus, π-calculus (pending that replication operator allows for infinity), cellular automata (extended to random automata networks, where each cell may represent a different finite state automaton), neural networks, Turing u-machines (pending that they allow for an infinite number of nodes), i.e., models where we can derive the sequence of components inheriting all needed information from their predecessors, i.e., we can repeat essentially the proofs for evolutionary automata and interaction machines. Thus we will write, skipping the proofs, the following.

.

Conjecture 2. Terminal languages for $-calculus, π-calculus, cellular automata generalized to random automata networks, neural networks and Turing u-machines coincide with the class of all languages in the alphabet X.

We can also safely assume that models based on Oracles, i.e., Turing o-machines and Site and Internet Machines can also accept arbitrary languages over a given alphabet.

Theorem 3. Terminal languages of o-machines and Site and Internet Machines coincide with the class of all languages in the alphabet X.

From Theorems 1 and 3 and Conjecture 2 we can derive immediately the conclusion.

Corollary 4. Expressiveness of $-calculus, π-calculus, cellular automata, neural networks, Turing o-machines and u-machines, Evolutionary Automata and Interaction Machines is the same and allow to accept all languages over a given finite alphabet.

It is not clear at this moment how to classify expressiveness of Infinite Time Turing Machines and Accelerating Turing Machines - simply, the conditions of an infinite number of steps or doubling the speed of each successive step alone seem not be sufficient to prove that those models can accept all languages over a given alphabet. Similarly, we do not have enough details on c-machines, because they were only briefly mentioned in the original paper on Turing machines. Also we cannot properly classify at this moment the expressiveness of Inductive Turing Machines and Persistent Turing Machines in the form of the stand-alone components. However, it is clear that they, as components of Evolutionary Automata or Interaction Machines, may achieve such enormous expressiveness of their hosts.

From Corollary 4 we conclude.

Corollary 5. Turingo-machines and u-machines, Site and Internet Machines, $-calculus, π-calculus, cellular automata, neural networks, Evolutionary Automata and Interaction Machines accept all U-complete, D-complete and H-complete languages.