

# WHAT'S CONNECTIONISM GOT TO DO WITH IT?!

**A. R. Kian Abolfazlian**  
Computer Science Department  
Aarhus University  
Ny Munkegade, Bldg. 540  
DK-8000 Århus C  
Denmark

## Abstract

*In this paper, I shall prove, why the old fashion Artificial Intelligence cannot be the right answer.*

**Key words.** *GOFAI, Fodor, Pylyshyn, Connectionism, Turing machines, Turing-undecidability, Universal Turing machines.*

## 1 MOTIVATION

The criticism of the old fashion Artificial Intelligence, or as John Haugeland has called it GOFAI<sup>1</sup>, has almost begun with the birth of this area of research. The very bold theories of people like Newell, Simon, Fodor, Pylyshyn, Winograd, Minsky, Schank, et. al. about Human understanding and cognition in writings like [11, Newell (1958)], [12, Newell (1980)], [13, Newell & Simon (1981)], [21, Simon (1979)], [22, Simon (1981)], [1, Fodor et al. (1974)], [2, Fodor (1975)], [3, Fodor (1980)], [4, Fodor (1981)], [5, Fodor (1983)], [6, Fodor (1987)], [7, Fodor & Pylyshyn (1988)], [14, Pylyshyn (1979)], [15, Pylyshyn (1989)], [16, Pylyshyn et al. (1989)], [17, Pylyshyn (1984)], [8, Minsky (1966)], [9, Minsky (1967)], [10, Minsky (1968)], [23,

---

<sup>1</sup>*Good Old Fashion AI.*

Winograd (1972)], [24, Winograd (1973)], [25, Winograd (1979)], [18, Schank & Childers (1984)], [19, Schank (1986)], [20, Schank (1990)] and so on, have awoken a lot of attention towards GOFAI. There has been a lot of discussions between researchers pro and against GOFAI, each party trying to prove that the other party is wrong. These discussions have been going on for years now, and it does not seem, that they are going to end in any near future.

In the beginning of 80's something very interesting happened. Connectionism began to come to the surface of the Earth again, and its development accelerated in a manner never seen before, and with it researchers in GOFAI started attacking Connectionism, as if it was their biggest opponent. Yes, the old arch-rival of GOFAI was reincarnated. They tried right from the start of rising period for Connectionism to show, that Connectionism does not have a complete account for Human cognition, and Cognitive Science should almost turn the face from Connectionism, for that road is doomed to fail. I wonder, whether there is any substance in these kind of ideas. As a matter of fact, I shall suppose, that GOFAI has spoken the truth, and I shall follow my way from that point towards the unknown future.

## 2 RESULTS

Let's now assume, that it is in fact true, that Human beings are Turing machines. What evil thing could happen as a consequence of this?! Well, let's find out.

**Assumption 1 (GOFAI's Fundamental Assumption (GFA))** *Human beings are Turing machines.*

There are actually very interesting question about Turing machines, which we will now look at<sup>2</sup>. Let's assume, that  $\mathcal{M} = (K, \Sigma, \delta, s)$  is a Turing machine.

**Lemma 1 ((A 1))** *The question, "Given a Turing machine  $\mathcal{M}$  and a fixed input string  $w_0$ , does  $\mathcal{M}$  halt on  $w_0$ ?" is Turing-undecidable.*

**Lemma 2 ((A 2))** *The question, "Given a Turing machine  $\mathcal{M}$ , is its output language Turing-decidable?", is Turing undecidable.*

---

<sup>2</sup>The Proofs for the following Lemma's will appear in an Appendix as the last section of this paper.

**Lemma 3 ((A 3))** *The question, “Given a Turing machine  $\mathcal{M}$ , is its output language Context-free?”, is Turing-undecidable.*

**Lemma 4 ((A 4))** *The question, “Given a Turing machine  $\mathcal{M}$  and a string  $w \in \text{OUT}(\mathcal{M})$ , is it correct for  $u \in \Sigma^*$ , that*

$$(s_{\mathcal{M}}, \#u\#) \vdash_{\mathcal{M}}^* (h_{\mathcal{M}}, \#w\#)$$

*is Turing-undecidable.*

**Lemma 5 ((A 5))** *The question, “Given a Turing machine  $\mathcal{M}$ , does  $\mathcal{M}$  decide  $\mathcal{L}(\mathcal{M})$ ?” is Turing-undecidable.*

**Lemma 6 ((A 6))** *The question, “Given a Turing machine  $\mathcal{M}$ , does  $\mathcal{M}$  decide  $\text{OUT}(\mathcal{M})$ ?” is Turing-undecidable.*

**Lemma 7 ((A 7))** *The question, “Given a Turing machine  $\mathcal{M}$  and a fixed string  $w_0 \in \Sigma^*$ , is  $w_0$  in the  $\text{OUT}(\mathcal{M})$ ?” is Turing-undecidable.*

**Lemma 8 ((A 8))** *The question, “Given a Turing machine  $\mathcal{M}$ , is it an Universal Turing machine?” is Turing-undecidable.*

But we have also,

**Remark 1 (Abolfazlian (R 1))** *(GFA) implies, that if a question about Turing machines is Turing-undecidable, then it cannot be answered by Human beings either.*

**Remark 2 (Abolfazlian (R 2))** *If a question about Turing machines is Turing-undecidable, then it will still be Turing-undecidable, if we substitute, in the question, the Turing machine by Universal Turing machine (UTM).*

**Remark 3 (Abolfazlian (R 3))** *We Human beings can simulate any Turing machine on any input string, given an encoded version of the Turing machine and the input string. Thus we must be UTMs.*

Now we have, that:

**Theorem 1 (Abolfazlian (K 1))** *Human beings cannot answer the question, “Given a Human being, is his/hers behaviour human?”.*

**Proof:** Lemma (A 2) & Remarks (R 1&2&3) do the job. It goes like this:

**Lemma 9 ((A 2.1))** *The question, “Given a Turing machine  $\mathcal{M}$ , is its output language Turing-decidable?”, cannot be answered by Human beings.*

**Proof:** Lemma (A 2) & Remark (R 1).

**Lemma 10 ((A 2.2))** *The question, “Given a Universal Turing machine  $\mathcal{M}$ , is its output language Turing-decidable?”, cannot be answered by Human beings.*

**Proof:** Lemma (A 2.1) & Remark (R 2).

Now we have the theorem by using Lemma (A 2.2) & Remark (R 3).

**Theorem 2 (Abolfazlian (K 2))** *Human beings cannot answer the question, “Given a Human being, is his/her behaviour Context-free?”.*

**Proof:** Lemma (A 3) & Remarks (R 1&2&3) do the job. It goes like this:

**Lemma 11 ((A 3.1))** *The question, “Given a Turing machine  $\mathcal{M}$ , is its output language Context-free?”, cannot be answered by Human beings.*

**Proof:** Lemma (A 3) & Remark (R 1).

**Lemma 12 ((A 3.2))** *The question, “Given a Universal Turing machine  $\mathcal{M}$ , is its output language Context-free?”, cannot be answered by Human beings.*

**Proof:** Lemma (A 3.1) & Remark (R 2).

Now we have the theorem by Lemma (A 3.2) & Remark (R 3).

**Theorem 3 (Abolfazlian (K 3))** *Human beings cannot answer the question, “Given a Human being and an example of his/hers behaviour, what has caused this particular behaviour?”.*

**Proof:** Lemma (A 4) & Remarks (R 1&2&3) do the job. It goes like this:

**Lemma 13 ((A 4.1))** *The question, “Given a Turing machine  $\mathcal{M}$  and a string  $w \in \text{OUT}(\mathcal{M})$ , is it correct for  $u \in \Sigma^*$ , that*

$$(s_{\mathcal{M}}, \#u\#) \vdash_{\mathcal{M}}^* (h_{\mathcal{M}}, \#w\#)$$

*cannot be answered by Human beings.*

**Proof:** Lemma (A 4) & Remark (R 1).

**Lemma 14 ((A 4.2))** *The question, “Given a Universal Turing machine  $\mathcal{M}$  and a string  $w \in \text{OUT}(\mathcal{M})$ , is it correct for  $u \in \Sigma^*$ , that*

$$(s_{\mathcal{M}}, \#u\#) \vdash_{\mathcal{M}}^* (h_{\mathcal{M}}, \#w\#)$$

*cannot be answered by Human beings.*

**Proof:** Lemma (A 4.1) & Remark (R 2).

Now we have the theorem by Lemma (A 4.2) & Remark (R 3).

**Theorem 4 (Abolfazlian (K 4))** *Human beings cannot answer the question, “Given a Human being and an arbitrary pattern of Human behaviour, can he/she decide (answer), whether he/she can produce that particular behaviour?”.*

**Proof<sup>3</sup>:** Lemma (A 6) & Remarks (R 1&2&3).

**Theorem 5 (Abolfazlian (K 5))** *Human beings cannot answer the question, “Given a Human being and a fixed pattern of Human behaviour, does he/she produce that particular behaviour?”.*

---

<sup>3</sup>I have not explained the proofs for this and following Theorems in so many details as for Theorem (K 1), because the proofs go in very similar manner.

**Proof:** Lemma (A 7) & Remarks (R 1&2&3).

**Theorem 6 (Abolfazlian (K 6))** *Human beings cannot answer the question, “Given a Human being, is he/she really a Human being?!!”.*

**Proof:** Lemma (A 8) & Remarks (R 1&2&3).

### 3 CONCLUSION

Now I wonder, if it would still be a “very” good idea to follow up the research program of GOFAI. May be people like Fodor and Pylyshyn should be more worried about consistency of their own ideas, instead of trying to show, that Connectionism has problems! Maybe after all Connectionism hasn’t got anything to do with GOFAI’s troubles, and GOFAI was unfortunately dead before it was ever born. But then again, it is a free World!!!

# References

- [1] Jerry A. Fodor, T. G. Bever & M. F. Garrett, “*The Psychology of Language*”, New York: McGraw Hill, 1974
- [2] Jerry A. Fodor, “*The Language of Thought*”, New York: Thomas Crowell, 1975
- [3] Jerry A. Fodor, “*Methodological solipsism considered as a research strategy in cognitive psychology*”, *Behavioural and Brain Sciences*, , pp. 63-109, 1980.
- [4] Jerry A. Fodor, “*The current status of the innateness controversy*”, *Representations*, Cambridge Mass.: M.I.T. Press, 1981.
- [5] Jerry A. Fodor, “*Modularity of Mind*”, M.I.T./Bradford, 1983.
- [6] Jerry A. Fodor, “*Psycho-semantics: The problem of meaning in the philosophy of mind*”, Cambridge, Mass.: M.I.T. Press, 1987.
- [7] Jerry A. Fodor & Zenon W. Pylyshyn, “*Connectionism and Cognitive Architecture: A Critical Analysis*”, *Cognition*, 28, pp. 3-71, 1988.
- [8] Marvin Minsky, “*Artificial Intelligence*”, *Scientific American*, pp. 251-252, September 1966.
- [9] Marvin Minsky, “*Computation: Finite and Infinite Machines*”, Prentice-Hall, 1967.
- [10] Marvin Minsky ed., “*Semantic Information Processing*”, M.I.T. Press, pp. 25-26, 1968.
- [11] Herbert A. Simon & Allan Newell, “*Heuristic Problem Solving: The Next Advance in Operations Research*”, *Operations Research*, 6, 1958.
- [12] Allan Newell, “*Physical Symbol Systems*”, *Cognitive Science*, vol. 4, nr. 2, 135-183, 1980.
- [13] Allan Newell & Herbert A. Simon, “*Computer Science as empirical inquiry: Symbols and search*”, *Communications of the Association for Computing Machinery*, 19, pp. 113-26, 1981.
- [14] Zenon W. Pylyshyn, “*Complexity and the study of artificial Intelligence*”, *Philosophical Perspectives in Artificial Intelligence*, M. Ringle ed., Humanities Press, 1979.

- [15] Zenon W. Pylyshyn, “Computers, Knowledge, and Human Mind”, *Perspectives on the Computer Revolution*, Zenon W. Pylyshyn & Liam J. Bannon (eds.), Ablex Publication Corporation, Norwood, New Jersey, 1989.
- [16] Zenon W. Pylyshyn et al., “What is Artificial Intelligence?”, *Perspectives on the Computer Revolution*, Zenon W. Pylyshyn & Liam J. Bannon (eds.), Ablex Publication Corporation, Norwood, New Jersey, 1989.
- [17] Zenon W. Pylyshyn, “Computation and Cognition: Towards a Foundation for Cognitive Science”, Cambridge Mass.: M.I.T. Press, 1984.
- [18] Roger C. Schank & Peter G. Childers, “The Cognitive Computer: On Language, Learning, and Artificial Intelligence”, Reading Mass.: Addison-Wesley Publication Company Inc., 1984.
- [19] Roger C. Schank, “Explanation Patterns: Understanding Mechanically and Creatively”, Hillsdale N.J.: Lawrence Erlbaum Associates Inc., 1986.
- [20] Roger C. Schank, “Tell me a Story: A New Look at Real and Artificial Memory”, New York: Scribners, 1990.
- [21] Herbert A. Simon, “Models of Thought”, Yale University Press, pp. 386-403, 1979.
- [22] Herbert A. Simon, “Studying Human Intelligence by Creating Artificial Intelligence”, *American Scientist*, pp. 308, 1981.
- [23] Terry Winograd, “Understanding Natural Language”, *Cognitive Psychology*, 3, pp. 8-11, 1972.
- [24] Terry Winograd, “A Procedural Model of Language Understanding”, *Computer Models of Thought and Language*, Roger Schank & Kenneth Colby eds., W. H. Freeman Press, 1973.
- [25] Terry Winograd, “Artificial Intelligence and Language Comprehension”, *Artificial Intelligence and Language Comprehension*, National Institute of Education, 1979.



# APPENDIX

The following definitions are taken from the book, “ELEMENTS OF THE THEORY OF COMPUTATION”, by Harry R. Lewis & Christos H. Papadimitriou, Prentice-Hall International Editions, 1981.

**Definition 1 (Turing machines)** A Turing machine is a quadruple  $(K, \Sigma, \delta, s)$ , where

$K$  is a finite set of states, not containing the halt state  $h$ ;

$\Sigma$  is an alphabet, containing the blank symbol  $\#$ , but not containing the symbols  $L$  and  $R$ ;

$s \in K$  is the initial state;

$\delta$  is a function from  $K \times K$  to  $(K \cup \{h\}) \times (\Sigma \cup \{L, R\})$ .

**Definition 2 (Configuration)** A configuration of a Turing machine  $\mathcal{M} = (K, \Sigma, \delta, s)$  is a member of

$$(K \cup \{h\}) \times \Sigma^* \times \Sigma \times (\Sigma^*(\Sigma - \{\#\}) \cup \{e\})$$

where  $e$  is the empty string. A configuration whose state component is  $h$  will be a halted configuration

**Definition 3** Let  $\mathcal{M} = (K, \Sigma, \delta, s)$  be a Turing machine and let  $(q_1, w_1, a_1, u_1)$  and  $(q_2, w_2, a_2, u_2)$  be configurations of  $\mathcal{M}$ . Then

$$(q_1, w_1, a_1, u_1) \vdash_{\mathcal{M}} (q_2, w_2, a_2, u_2)$$

if and only if, for some  $b \in \Sigma \cup \{L, R\}$ ,  $\delta(q_1, a_1) = (q_2, b)$  and either

1.  $b \in \Sigma, w_1 = w_2, u_1 = u_2$ , and  $a_2 = b$ ;

or

2.  $b = L, w_1 = w_2 a_2$ , and either
  - $u_2 = a_1 u_1$ , if  $a_1 \neq \#$  or  $u_1 \neq e$ , or
  - $u_2 = e$ , if  $a_1 \neq \#$  and  $u_1 = e$ ;

or

3.  $b = R, w_2 = w_1 a_1$ , and either

- $u_1 = a_2 u_2$ , or
- $u_1 = u_2 = e$  and  $a_2 = \#$ .

Note that if  $b = L$  and  $w_1 = e$ , then  $(q_1, w_1, a_1, u_1)$  yields no configuration, since there is no  $w_2 \in \Sigma^*$  and  $a_2 \in \Sigma$  such that  $w_1 = w_2 a_2$ . Such a configuration will be called a hanging configuration. On the other hand, every configuration, that is not a halted or a hanging configuration yields exactly one configuration in one step.

**Definition 4** For any Turing machine  $\mathcal{M}$ ,  $\vdash_{\mathcal{M}}^*$  is the reflexive, transitive closure of  $\vdash_{\mathcal{M}}$ ; we say that configuration  $C_1$  yields configuration  $C_2$  if  $C_1 \vdash_{\mathcal{M}}^* C_2$ . A computation  $y$   $\mathcal{M}$  is a sequence of configurations  $C_0, C_1, \dots, C_n$  for some  $n \geq 0$  such that

$$C_0 \vdash_{\mathcal{M}} C_1 \vdash_{\mathcal{M}} C_2 \vdash_{\mathcal{M}} \dots \vdash_{\mathcal{M}} C_n.$$

We say that the computation is of length  $n$  or has  $n$  steps.

**Definition 5 (Turing-computable function)** Let  $\Sigma_0$  and  $\Sigma_1$  be alphabets not containing the blank symbol  $\#$ . Let  $\mathcal{F}$  be a function from  $\Sigma_0^*$  to  $\Sigma_1^*$ . A Turing machine  $\mathcal{M} = (K, \Sigma, \delta, s)$  is said to compute  $\mathcal{F}$  if  $\Sigma_0, \Sigma_1 \subseteq \Sigma$  and for  $w \in \Sigma_0^*$ , if  $F(w) = u$ , then

$$(s, \#w\#) \vdash_{\mathcal{M}}^* (h, \#u\#)$$

If some such Turing machine  $\mathcal{M}$  exists, then  $\mathcal{F}$  is said to be a Turing-computable function.

**Definition 6 (Turing-decidable languages)** Let  $\Sigma_0$  be a fixed alphabet not containing blank symbol  $\#$ . Let  $\mathbf{Y}$  and  $\mathbf{N}$  be two fixed symbols not in  $\Sigma_0$ . Then the language  $\mathcal{L} \subseteq \Sigma_0$  is Turing decidable if and only if the function  $\chi_{\mathcal{L}} : \Sigma_0^* \rightarrow \{\mathbf{Y}, \mathbf{N}\}$  is Turing-computable, where for each  $w \in \Sigma_0^*$ ,

$$\chi_{\mathcal{L}}(w) = \begin{cases} \mathbf{Y} & \text{if } w \in \mathcal{L} \\ \mathbf{N} & \text{if } w \notin \mathcal{L} \end{cases}$$

If  $\chi_{\mathcal{L}}$  is computed by a Turing machine  $\mathcal{M}$ , then  $\mathcal{M}$  is said to decide  $\mathcal{L}$ , or to be a decision procedure for  $\mathcal{L}$ .

**Definition 7** If  $\mathcal{M} = (K, \Sigma, \delta, s)$  is a Turing machine and  $w \in \Sigma_0^*$ , then  $\mathcal{M}$  is said to halt on input  $w$  if and only if  $(s, \#w\#)$  yields some halted configuration. Similarly,  $\mathcal{M}$  is said to hang on input  $w$  if  $(s, \#w\#)$  yields some hanging configuration.

**Definition 8 (Turing acceptable languages)** We say that a Turing machine  $\mathcal{M}$  accepts a string  $w \in \Sigma_0^*$ , if  $\mathcal{M}$  halts on input  $w$ . Thus  $\mathcal{M}$  accepts a language  $\mathcal{L} \subseteq \Sigma_0^*$  if and only if

$$\mathcal{L} = \{w \in \Sigma_0^* : \mathcal{M} \text{ accepts } w\}$$

and a language is said to be Turing-acceptable if there is some Turing machine that accepts it.

And now to the Prooves of the Lemmas presented in this paper;

**Proof of Lemma (A 1):** Let's suppose, that we have a Turing machine  $\mathcal{M}'$ , which can decide, whether if an arbitrary Turing machine  $\mathcal{M}$  halts on our fixed input string  $w_0$ . Then we can decide, whether if an arbitrary Turing machine  $\mathcal{M}_0$  halts on the empty tape. Given  $\mathcal{M}_0$  we construct a new Turing machine  $\mathcal{M}_0^*$ , which acts on  $w_0$  as follows:

1. It erases  $w_0$  from the input tape.
2. It simulates  $\mathcal{M}_0$  on the empty tape.

It is now obvious, that  $\mathcal{M}_0^*$  halts on  $w_0$  if and only if  $\mathcal{M}_0$  halts on the empty tape, and  $\mathcal{M}'$  can decide, whether if  $\mathcal{M}_0^*$  halts on  $w_0$ . Thus  $\mathcal{M}'$  can decide, whether if  $\mathcal{M}_0$  halts on the empty tape. This is a contradiction (LP 6.3.1.3).

**Proof of Lemma (A 2):** Lets  $OUT(\mathcal{M})$  be the  $\mathcal{M}$ 's output language. Let's suppose, that  $\mathcal{M}'$  can decide  $OUT(\mathcal{M})$  for an arbitrary  $\mathcal{M}$ . We can now use  $\mathcal{M}'$  to decide the language, which an arbitrary Turing machine  $\mathcal{M}_0$  accepts ( $\mathcal{L}(\mathcal{M}_0)$ ). This is, because there is a Turing machine  $\tilde{\mathcal{M}}_0$ , for which we have

$$OUT(\tilde{\mathcal{M}}_0) = \mathcal{L}(\mathcal{M}_0) \quad (\text{cf. LP 6.2.2})$$

$\mathcal{M}'$  decides  $OUT(\tilde{\mathcal{M}}_0)$ , which means, that  $\mathcal{M}'$  decides  $\mathcal{L}(\mathcal{M}_0)$ . This is a contradiction (LP 6.3.2.1).

**Proof of Lemma (A 3):** *Analog to the proof of Lemma (A 2).*

**Proof of Lemma (A 4):** *Let us suppose, that there is a Turing machine  $\mathcal{M}'$  that can decide*

$$(s_{\mathcal{M}}, \#u\#) \vdash_{\mathcal{M}}^* (h_{\mathcal{M}}, \#w\#).$$

*Then we can decide whether a Turing machine  $\mathcal{M}_0$  halts on the empty tape. Consider the Turing machine  $\mathcal{M}_1$ .*

$$\mathcal{M}_1: \mathcal{M}_- \mathcal{M}_0 \mathcal{M}_+ \mathcal{M}$$

*where  $\mathcal{M}_-$  erase whatever is on the tape and  $\mathcal{M}_+$  writes whatever  $\mathcal{M}_-$  has erased. Now we have, that if  $\mathcal{M}_0$  halts on empty tape, then  $\mathcal{M}'$  can decide, whether if*

$$(s_{\mathcal{M}_1}, \#u\#) \vdash_{\mathcal{M}_1}^* (h_{\mathcal{M}_1}, \#w\#).$$

*This means, that  $\mathcal{M}'$  can decide whether if  $\mathcal{M}_0$  halts on the empty tape. This is a contradiction.*

**Proof of Lemma (A 5):** *Let's suppose, that there is a Turing machine  $\mathcal{M}^*$ , which does the job for us. This means, that given a Turing machine  $\mathcal{M}$ ,  $\mathcal{M}^*$  can decide  $\mathcal{L}(\mathcal{M})$ , a.e.  $\mathcal{M}^*$  can decide whether  $\mathcal{M}$  halts on an arbitrary input string  $w$ . Thus  $\mathcal{M}^*$  can also decide whether  $\mathcal{M}$  halts on empty input string (the halting problem). This is a contradiction.*

**Proof of Lemma (A 6):** *Analog to the proof of Lemma (A 5).*

**Proof of Lemma (A 7):** *We know, that  $W_0 \in \text{OUT}(\mathcal{M})$  if and only if some Turing machine  $\mathcal{M}^*$  accepts  $w_0$  (cf. LP 6.2.2). Thus, the question asked here, is equivalent to the question, "Given a Turing machine  $\mathcal{M}$  and a fixed input string  $w_0$ , does  $\mathcal{M}$  halt on  $w_0$ ?", which is Turing-undecidable (Lemma (A 1)).*

**Proof of Lemma (A 8):**  *$\mathcal{M}$  is an Universal Turing machine (UTM) if and only if*

$$\mathcal{L}(\mathcal{M}) = \{\rho(\mathcal{M}')\rho(w) \mid \mathcal{M}' \text{ accepts } w\}$$

*If we suppose, that  $\mathcal{M}^*$  can decide, whether if  $\mathcal{M}$  is an UTM, then it means, that it can decide  $\mathcal{L}(\mathcal{M})$ , which means, that it can decide, whether  $\rho(\mathcal{M}_0)\rho(\#) \in \mathcal{L}(\mathcal{M})$  for an arbitrary Turing machine  $\mathcal{M}_0$ . This means, that  $\mathcal{M}^*$  can decide, whether if  $\mathcal{M}_0$  halts on  $\#$ . This is a contradiction.*