

# Homework 5

Joe Baker, Brett Schreiber, Brian Knotten

January 24, 2018

## 7

### a

A valid computation history of the given Turing Machine  $M$  on the input  $I = 101_2$  is:

$H = \#q_0101_2\#1q_101_2\#11q_11_2\#110q_0\#110q_0\#110q_0\#110q_0\#...$

This computation history is infinite because  $M$  does not halt on an input with an even number of 1s.

### b

Let  $|\Gamma| = |\{q_0, q_1, q_h, \#, \sqcup, 0, 1\}| = 7$  be the alphabet of a computation history for TM  $M$  from part a. Let  $H$  be a base-7 number, where each member of  $\Gamma$  is mapped to a specific digit, and  $H$  represents a computation history of  $M$ . Let  $S$  be the following Godel sentence:

$$\exists_H \text{ such that } \text{init}(H) \wedge \text{transition}() \wedge \text{halt}()$$

where  $\text{init}(H)$ ,  $\text{transition}(H)$ , and  $\text{halt}(H)$  are macros defined by first order logic and other macros below:

#### **place(j)**

$$(H \bmod B^{j+1}) / B^j$$

Represents an arithmetic expression that returns the digit at location  $j$  in  $H$ .

#### **same(i,j,k,l)**

$$\forall_z \text{ where } i - j > z > 0 \wedge i > j \wedge k > l \wedge i - j = k - l, \text{ then } \text{place}(z + j) = \text{place}(z + l)$$

Represents a logical expression that will be true  $\Leftrightarrow$  digits  $i$  through  $j$  are identical to digits  $k$  through  $l$ .

#### **digits(i)**

$$\lceil \log_{|\Gamma|} i \rceil$$

Represents the number of digits in a given integer  $i$ .

#### **leftmost(i,j)**

$$\lfloor j / |\Gamma|^{(\text{DIGITS}(j) - i)} \rfloor$$

Represents the  $i$  most significant digits of the integer  $j$ .

### table(i,j)

$$\begin{aligned} &(i = 0 \wedge i + 1 = q_0 \wedge i + 2 = 0 \wedge j = 0 \wedge j + 1 = 0 \wedge j + 2 = q_0) \\ &\vee (i = 1 \wedge i + 1 = q_0 \wedge i + 2 = 0 \wedge j = 1 \wedge j + 1 = 0 \wedge j + 2 = q_0) \\ &\vee (i = \# \wedge i + 1 = q_0 \wedge i + 2 = 0 \wedge j = \# \wedge j + 1 = 0 \wedge j + 2 = q_0) \\ &\vee (i = 0 \wedge i + 1 = q_0 \wedge i + 2 = 1 \wedge j = 0 \wedge j + 1 = 1 \wedge j + 2 = q_1) \\ &\vee (i = 1 \wedge i + 1 = q_0 \wedge i + 2 = 1 \wedge j = 1 \wedge j + 1 = 1 \wedge j + 2 = q_1) \\ &\vee (i = \# \wedge i + 1 = q_0 \wedge i + 2 = 1 \wedge j = \# \wedge j + 1 = 1 \wedge j + 2 = q_1) \\ &\vee (i = 0 \wedge i + 1 = q_0 \wedge i + 2 = \sqcup \wedge j = 0 \wedge j + 1 = q_0 \wedge j + 2 = \sqcup) \\ &\vee (i = 1 \wedge i + 1 = q_0 \wedge i + 2 = \sqcup \wedge j = 1 \wedge j + 1 = q_0 \wedge j + 2 = \sqcup) \\ &\vee (i = \# \wedge i + 1 = q_0 \wedge i + 2 = \sqcup \wedge j = 0 \wedge j + 1 = q_0 \wedge j + 2 = \sqcup) \\ &\vee (i = 0 \wedge i + 1 = q_1 \wedge i + 2 = 0 \wedge j = 0 \wedge j + 1 = 1 \wedge j + 2 = q_1) \\ &\vee (i = 1 \wedge i + 1 = q_1 \wedge i + 2 = 0 \wedge j = 1 \wedge j + 1 = 1 \wedge j + 2 = q_1) \\ &\vee (i = 0 \wedge i + 1 = q_1 \wedge i + 2 = 1 \wedge j = 0 \wedge j + 1 = 0 \wedge j + 2 = q_0) \\ &\vee (i = 1 \wedge i + 1 = q_1 \wedge i + 2 = 1 \wedge j = 1 \wedge j + 1 = 0 \wedge j + 2 = q_0) \\ &\vee (i = 0 \wedge i + 1 = q_1 \wedge i + 2 = \sqcup \wedge j = 0 \wedge j + 1 = q_h \wedge j + 2 = \sqcup) \\ &\vee (i = 1 \wedge i + 1 = q_1 \wedge i + 2 = \sqcup \wedge j = 1 \wedge j + 1 = q_h \wedge j + 2 = \sqcup) \end{aligned}$$

Represents a local expression that will be true iff digits  $i$ ,  $i + 1$  and  $i + 2$  in  $H$  represents a tape symbol, state in  $Q$  and a tape symbol respectively, and digits  $j$ ,  $j + 1$  and  $j + 2$  in  $H$  evolve properly from digits  $i$ ,  $i + 1$ , and  $i + 2$  according to  $M$ .

### init(H)

$$leftmost(5, H) = \#q_0101$$

Represents the valid initial state for  $M$  on input  $I$ .

### transition()

$$\forall_x place(x) = \# \wedge x \neq 0 \wedge x \neq leftmost(1, x) \wedge validTransition(x) \wedge validCopy(x)$$

Represents a valid sequence of digits on both "sides" of any internal  $\#$ . Each "side" is a configuration. The transition between configurations is validated with  $validTransition()$  and the rest of the tape is validated with  $validCopy()$ .

### validTransition(w)

$$\exists_{x,y} \text{ such that } table(x-1, y-1) \wedge place(x) \in \{q_0, q_1\} \wedge place(y) \in \{q_0, q_1, q_h\} \wedge x > w > y \wedge \forall_z \text{ where } x > z > w \text{ or } w > z > y, place(z) \notin \{q_0, q_1, q_h, \#\}$$

Checks that the transition from the state left of a  $\#$  to the right of a  $\#$  is valid based on the current read head value and state and written value and state.

### validCopy(c)

$$\exists_{a,b,d,e} \text{ such that } same(a, b-2, c, d-2) \wedge same(b+2, c, d+2, e) \wedge place(b) \in \{q_0, q_1\} \wedge place(d) \in \{q_0, q_1, q_h\} \wedge place(a) = place(e) = \# \wedge \forall_z \text{ where } a > z > b \vee b > z > c \vee c > z > d \vee d > z > e, \text{ then } place(z) \notin \{q_0, q_1, q_h, \#\}$$

Checks that the tape before the previous state matches the tape before the next state. Also checks that the tape after the previous state matches the tape after the next state. Botch checks exclude the digit immediately before and after a state.

### halt()

$$\exists_j \text{ such that } place(j) = q_h \wedge \forall_k \text{ where } 0 < k < j, place(k) = \sqcup \wedge place(0) = \#$$

Represents valid halting states from the state symbol to the final  $\#$ .  $q_h$  should be the rightmost state in the

integer, and the integer must end with #

The Godel sentence is valid if and only if a number represents a valid computation history. The computation history is valid if:

1. the first configuration is the initial configuration
2. each configuration follows from the previous
3. the final configuration is a halting configuration

Each of these conditions is met by the macros *init()*, *transition()*, and *halt()*.