CS_275 Automata and Formal Language Theory

Course Notes
Additional Material
(This material is no longer taught and not exam relevant)
Part III: Limits of Computation
Chapt. III.3: Turing Machines

Anton Setzer

http://www.cs.swan.ac.uk/∼csetzer/lectures/automataFormalLanguage/current/index.html

May 5, 2017

No Additional Material

For this subsection no additional material has been added yet.

Formal Lemma URM-computable $\Rightarrow$ TM-computable

Lemma (3.4)

If $f : \mathbb{N}^n \xrightarrow{\sim} \mathbb{N}$ is URM-computable then it is Turing-computable by a TM with alphabet $\{0, 1, \sqcup\}$.

Remark

The proof that every Turing computable function is URM computable will not be given in this Section.
(It could be done directly. A much nicer argument which makes use of the notion of partial recursive functions can be found in the notes of "Computability Theory").

Notation: $\widetilde{\mathrm{bin}}$

- In this proof we will represent a configuration of a URM by a sequence of possibly non-normalised strings on the tape representing the registers.
- So we want to get a short notation for
  "The tape contains $s_0 \sqcup s_1 \sqcup s_2 \sqcup \cdots \sqcup s_k$ where s_i is a binary representation of n_i"
  (where n_i is the simulated content of register R_i.
- We define $\widetilde{\mathrm{bin}}(n)$ as one of the binary representations of s.
- Then we can write for the above:
  "The tape contains $\widetilde{\mathrm{bin}}(n_0) \sqcup \widetilde{\mathrm{bin}}(n_1) \sqcup \cdots \sqcup \widetilde{\mathrm{bin}}(n_k)$".
- So $\widetilde{\mathrm{bin}}(n)$ denotes one of the possible choices for strings s

  s.t. $(s)_2 = n$.
  - So $\widetilde{\mathrm{bin}}(1)$ can be "1", "01", "001", etc.
  - In the special case 0 we treat the empty string as one of the possible representations, so $\widetilde{\mathrm{bin}}(0)$ can be "", "0", "00", "000", etc.

Notation: $\widetilde{\mathrm{bin}}$

- When carrying out intermediate calculations, it is easier to refer to $\widetilde{\mathrm{bin}}(n)$ rather than $\mathrm{bin}(n)$
  - E.g. we can set a number on the tape easily to an element of $\widetilde{\mathrm{bin}}(0)$ by overwriting it with 0s.
  - In order to set it to $\mathrm{bin}(0)$ one would need to make sure that exactly one 0 remains. Then one usually has to shift left the content of the tape to the right of the original number.

Proof of Lemma 3.4

Notation
The tape of a TM contains $a_0, \ldots, a_l$ means:

- Starting from the head position, the cells of the tape contain $a_0, \ldots, a_l$.
- All other cells contain $\sqcup$.

Proof of Lemma 3.4

Assume

- $f = \mathrm{U}^{(n)}$,
- U refers only to $\mathrm{R}_0, \ldots, \mathrm{R}_{l-1}$ and $l > n$,

We define a TM T, which simulates U. Done as follows:

- That the registers $\mathrm{R}_0, \ldots, \mathrm{R}_{l-1}$ contain $a_0, \ldots, a_{l-1}$ is simulated by the tape containing $\widetilde{\mathrm{bin}}(a_0)\sqcup \cdots \sqcup\widetilde{\mathrm{bin}}(a_{l-1})$.
- An instruction I_j will be simulated by states $s_{j,0}, \ldots, s_{j,i}$ with instructions for those states.

Conditions on the Simulation

- Assume the URM U is in a state s.t.
  - $\mathrm{R}_0, \ldots, \mathrm{R}_{l-1}$ contain $a_0, \ldots, a_{l-1}$,
  - the URM is about to execute I_j.
- Assume after executing I_j, the URM is in a state where
  - $\mathrm{R}_0, \ldots, \mathrm{R}_{l-1}$ contain $b_0, \ldots, b_{l-1}$,
  - the PC contains k.
- Then we want that, if configuration of the TM T is, s.t.
  - the tape contains $\widetilde{\mathrm{bin}}(a_0)\sqcup\widetilde{\mathrm{bin}}(a_1)\sqcup \cdots \sqcup\widetilde{\mathrm{bin}}(a_{l-1})$,
  - and the TM is in state $s_{j,0}$,
- then the TM reaches a configuration s.t.
  - the tape contains $\widetilde{\mathrm{bin}}(b_0)\sqcup\widetilde{\mathrm{bin}}(b_1)\sqcup \cdots \sqcup\widetilde{\mathrm{bin}}(b_{l-1})$,
  - the TM is in state $s_{k,0}$.

Example

- Assume the URM is about to execute instruction
  - $4 : \mathrm{R}_2 := \mathrm{R}_2 \dot{-} 1$ (i.e. PC $= 4$),
  - with register contents

R_0	R_1	R_2
2	1	3

- Then the URM will end with
  - PC $= 5$
  - and register contents

R_0	R_1	R_2
2	1	2

Example

- Then we want that, if the simulating TM is
  - in state $s_{4,0}$,
  - with tape content $\widetilde{\mathrm{bin}}(2)\sqcup\widetilde{\mathrm{bin}}(1)\sqcup\widetilde{\mathrm{bin}}(3)$
- it should reach
  - state $s_{5,0}$
  - with tape content $\widetilde{\mathrm{bin}}(2)\sqcup\widetilde{\mathrm{bin}}(1)\sqcup\widetilde{\mathrm{bin}}(2)$

Proof of Lemma 3.4

- Furthermore, we need initial states $s_{\mathrm{init},0}, \ldots, s_{\mathrm{init},j}$ and corresponding instructions, s.t.
  - if the TM initially contains

$$\widetilde{\mathrm{bin}}(b_0)\sqcup\widetilde{\mathrm{bin}}(b_1)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(b_{n-1})$$

  - it will reach state $s_{0,0}$ with the tape containing

$$\widetilde{\mathrm{bin}}(b_0)\sqcup\widetilde{\mathrm{bin}}(b_1)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(b_{n-1})\sqcup\underbrace{0\sqcup 0\sqcup\cdots\sqcup 0}_{l-n \text{ times}}\sqcup$$

Proof of Lemma 3.4

Assume the run of the URM, starting with R_i containing $a_{0,i} = a_i$ $i = 0, \ldots, n-1$, and $a_{0,i} = 0$ for $i = n, \ldots, l-1$ is as follows:

Instruction	R_0	R_1	$\cdots$	R_{n-1}	R_n	$\cdots$	R_{l-1}
0	a_0	a_1	$\cdots$	a_{n-1}	0	$\cdots$	0
$=$	$=$	$=$		$=$	$=$		$=$
k_0	$a_{0,0}$	$a_{0,1}$	$\cdots$	$a_{0,n-1}$	$a_{0,n}$	$\cdots$	$a_{0,l-1}$
k_1	$a_{1,0}$	$a_{1,1}$	$\cdots$	$a_{1,n-1}$	$a_{1,n}$	$\cdots$	$a_{1,l-1}$
k_2	$a_{2,0}$	$a_{2,1}$	$\cdots$	$a_{2,n-1}$	$a_{2,n}$	$\cdots$	$a_{2,l-1}$
		$\cdots$					

Proof of Lemma 3.4

Then the corresponding TM will successively reach the following configurations:

State	Tape contains
$s_{\mathrm{init},0}$	$\widetilde{\mathrm{bin}}(a_0)\sqcup\widetilde{\mathrm{bin}}(a_1)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(a_{n-1})\sqcup$
$s_{0,0}$	$\widetilde{\mathrm{bin}}(a_0)\sqcup\widetilde{\mathrm{bin}}(a_1)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(a_{n-1})\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(0)\sqcup$
$=$	$=$
$s_{k_0,0}$	$\widetilde{\mathrm{bin}}(a_{0,0})\sqcup\widetilde{\mathrm{bin}}(a_{0,1})\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(a_{0,l-1})\sqcup$
$s_{k_1,0}$	$\widetilde{\mathrm{bin}}(a_{1,0})\sqcup\widetilde{\mathrm{bin}}(a_{1,1})\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(a_{1,l-1})\sqcup$
$s_{k_2,0}$	$\widetilde{\mathrm{bin}}(a_{2,0})\sqcup\widetilde{\mathrm{bin}}(a_{2,1})\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(a_{2,l-1})\sqcup$
	$\cdots$

Example

Consider the URM program U (which was discussed already in the section on URMs):

$0 :$ `if` $\mathrm{R}_0 = 0$ `then goto` 3

$1 : \mathrm{R}_0 := \mathrm{R}_0 \dot{-} 1$

$2 :$ `if` $\mathrm{R}_1 = 0$ `then goto` 0

$\mathrm{U}^{(1)}(a) \simeq 0$.

Example

$0 : \texttt{if } \mathrm{R}_0 = 0 \texttt{ then goto } 3$
$1 : \mathrm{R}_0 := \mathrm{R}_0 \dot{-} 1$
$2 : \texttt{if } \mathrm{R}_1 = 0 \texttt{ then goto } 0$

We saw in the last section that a run of $\mathrm{U}^{(1)}(2)$ is as follows:

Instruction	R_0	R_1
0	2	0
1	2	0
2	1	0
0	1	0
1	1	0
2	0	0
0	0	0
3	0	0

URM Stops

Corresponding TM Simulation

$0 : \texttt{if } \mathrm{R}_0 = 0 \texttt{ then goto } 3$
$1 : \mathrm{R}_0 := \mathrm{R}_0 \dot{-} 1$
$2 : \texttt{if } \mathrm{R}_1 = 0 \texttt{ then goto } 0$

Instruction	R_0	R_1	State of TM	Content of Tape
			$s_{\mathrm{init},0}$	$\widetilde{\mathrm{bin}}(2)\sqcup\sqcup$
0	2	0	$s_{0,0}$	$\widetilde{\mathrm{bin}}(2)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
1	2	0	$s_{1,0}$	$\widetilde{\mathrm{bin}}(2)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
2	1	0	$s_{2,0}$	$\widetilde{\mathrm{bin}}(1)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
0	1	0	$s_{0,0}$	$\widetilde{\mathrm{bin}}(1)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
1	1	0	$s_{1,0}$	$\widetilde{\mathrm{bin}}(1)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
2	0	0	$s_{2,0}$	$\widetilde{\mathrm{bin}}(0)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
0	0	0	$s_{0,0}$	$\widetilde{\mathrm{bin}}(0)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$
3	0	0	$s_{3,0}$	$\widetilde{\mathrm{bin}}(0)\sqcup\sqcup\widetilde{\mathrm{bin}}(0)\sqcup\sqcup$

URM Stops TM Stops

Proof of Lemma 3.4

If we have defined this we have

- If
$$\mathrm{U}^{(n)}(a_0, \ldots, a_{n-1})\downarrow \ ,$$
$$\mathrm{U}^{(n)}(a_0, \ldots, a_{n-1}) \simeq c \ ,$$
then U eventually stops with R_i containing some values b_i, where $b_0 = c$.
Then, the TM T starting with
$$\mathrm{bin}(a_0)\sqcup\sqcup \cdots \sqcup\sqcup\mathrm{bin}(a_{n-1})$$
will eventually terminate in a configuration
$$\widetilde{\mathrm{bin}}(b_0)\sqcup\sqcup \cdots \sqcup\sqcup\widetilde{\mathrm{bin}}(b_{k-1}) \ .$$
for some $k \geq n$.
Therefore $\mathrm{T}^{(n)}(a_0, \ldots, a_{n-1}) \simeq b_0 = c$.

Proof of Lemma 3.4

- If
$$\mathrm{U}^{(n)}(a_0, \ldots, a_{n-1})\uparrow \ ,$$
the URM U will loop and the TM T will carry out the same steps as the URM and loop as well.
Therefore
$$\mathrm{T}^{(n)}(a_0, \ldots, a_{n-1})\uparrow \ ,$$
again
$$\mathrm{U}^{(n)}(a_0, \ldots, a_{n-1}) \simeq \mathrm{T}^{(n)}(a_0, \ldots, a_{n-1}) \ .$$

Proof of Lemma 3.4

- It follows
$$\mathrm{U}^{(n)} = \mathrm{T}^{(n)} \ ,$$
and the proof is complete, if the simulation has been introduced.
- The following slides contain a detailed proof, which will not be presented in the lecture this year.
Jump over remaining proof.

Proof of Lemma 3.4

Informal description of the simulation of URM instructions.

- **Initialisation.**
Initially, the tape contains $\mathrm{bin}(a_0)\sqcup\cdots\sqcup\mathrm{bin}(a_{n-1})$.
We need to obtain configuration:
$$\widetilde{\mathrm{bin}}(a_0)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(a_{n-1})\sqcup\underbrace{\widetilde{\mathrm{bin}}(0)\sqcup\cdots\sqcup\widetilde{\mathrm{bin}}(0)}_{l-n \text{ times}}.$$
Achieved by
  - moving head to the end of the initial configuration
  - inserting, starting from the next blank, $l-n$-times $0\sqcup$,
  - then moving back to the beginning.

Proof of Lemma 3.4

- **Simulation of URM instructions.**
  - **Simulation of instruction $k : \mathrm{R}_j := \mathrm{R}_j + 1$.**
  Need to increase $(j+1)$st binary number by 1 Initial configuration:

$$\widetilde{\mathrm{bin}}(c_0) \ \sqcup \ \widetilde{\mathrm{bin}}(c_1) \ \sqcup \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_j) \ \sqcup \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_l) \ \sqcup$$
$\uparrow$
$s_{k,0}$

  - First move to the $(j+1)$st blank to the right. Then we are at the end of the $(j+1)$st binary number.

$$\widetilde{\mathrm{bin}}(c_0) \ \sqcup \ \widetilde{\mathrm{bin}}(c_1) \ \sqcup \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_j) \ \underset{\uparrow}{\sqcup} \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_l) \ \sqcup$$

Proof of Lemma 3.4

$$\widetilde{\mathrm{bin}}(c_0) \ \sqcup \ \widetilde{\mathrm{bin}}(c_1) \ \sqcup \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_j) \ \underset{\uparrow}{\sqcup} \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_l) \ \sqcup$$

- Now perform the operation for increasing by 1 as above.
At the end we obtain:

$$\widetilde{\mathrm{bin}}(c_0) \ \sqcup \ \widetilde{\mathrm{bin}}(c_1) \ \sqcup \ \cdots \ \sqcup \ \underset{\uparrow}{\widetilde{\mathrm{bin}}}(c_j+1) \ \sqcup \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_l) \ \sqcup$$

- It might be that we needed to write over the separating blank a 1, in which case we have:

$$\widetilde{\mathrm{bin}}(c_0) \ \sqcup \ \widetilde{\mathrm{bin}}(c_1) \ \cdots \ \widetilde{\mathrm{bin}}(c_{j-1}) \ \underset{\uparrow}{\widetilde{\mathrm{bin}}}(c_j+1) \ \sqcup \ \cdots \ \sqcup \ \widetilde{\mathrm{bin}}(c_l)$$

Proof of Lemma 3.4

- In the latter case, shift all symbols to the left once left, in order to obtain a separating $\sqcup$ between the lth and $l-1$st entry.
  We obtain

$$\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \widetilde{\mathrm{bin}}(c_1) \;\sqcup\; \cdots \; \widetilde{\mathrm{bin}}(c_{j-1}) \;\sqcup\; \widetilde{\mathrm{bin}}(c_j+1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c$$
$\uparrow$

- Otherwise, move the head to the left, until we reach the $(j+1)$st blank to the left, and then move it once to the right.
  We obtain

$$\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \widetilde{\mathrm{bin}}(c_1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_j+1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$$
$\uparrow$

Proof of Lemma 3.4

- **Simulation of instruction $k : \mathrm{R}_j := \mathrm{R}_j \dot{-} 1$.**
  - Assume the configuration at the beginning is :

$$\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \widetilde{\mathrm{bin}}(c_1) \;\sqcup\; \cdots \; \widetilde{\mathrm{bin}}(c_j) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$$
$\uparrow$

We want to achieve

$$\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \widetilde{\mathrm{bin}}(c_1) \;\sqcup\; \cdots \; \widetilde{\mathrm{bin}}(c_j \dot{-} 1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$$
$\uparrow$

Done as follows:

Proof of Lemma 3.4

Initially: $\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_j) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$
$\uparrow$
Finally: $\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_j \dot{-} 1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$
$\uparrow$

- Move to end of the $(j+1)$st number.
- Check, if the number consists only of zeros or not.
  - If it consists only of zeros, $\mathrm{R}_j := \mathrm{R}_j \dot{-} 1$ doesn't change anything.
  - Otherwise, number is of the form $b_0 \cdots b_k 1 \underbrace{00\cdots0}_{l' \text{ times}}$.

    Replace it by $b_0 \cdots b_k 0 \underbrace{11\cdots1}_{l' \text{ times}}$.
    Done as for $\mathrm{R}_j := \mathrm{R}_j + 1$.

Proof of Lemma 3.4

Initially: $\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_j) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$
$\uparrow$
Finally: $\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_j \dot{-} 1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$
$\uparrow$

- We have achieved

$$\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \widetilde{\mathrm{bin}}(c_1) \;\sqcup\; \cdots \; \widetilde{\mathrm{bin}}(c_j \dot{-} 1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$$
$\uparrow$

- Move back to the beginning:

$$\widetilde{\mathrm{bin}}(c_0) \;\sqcup\; \widetilde{\mathrm{bin}}(c_1) \;\sqcup\; \cdots \; \widetilde{\mathrm{bin}}(c_j \dot{-} 1) \;\sqcup\; \cdots \;\sqcup\; \widetilde{\mathrm{bin}}(c_l) \;\sqcup$$
$\uparrow$

Proof of Lemma 3.4

- **Simulation of instruction** $k : \texttt{if } \mathrm{R}_j = 0 \texttt{ then goto } k'$.
  - Move to $j+1$st binary number on the tape.
  - Check whether it contains only zeros.
    - If yes, switch to state $s_{k',0}$.
    - Otherwise switch to state $s_{k+1,0}$.

This completes the simulation of the URM U.

Halting Problem with no Inputs

Theorem (3.8)

It is undecidable, whether a Turing machine started with a blank tape terminates.

- **Proof:**
  - Let

$$\begin{aligned}\mathrm{Halt}'(e) :\Leftrightarrow\ & e \text{ is a code for a Turing machine} \mathrm{T} \\ & \text{and } \mathrm{T} \text{ started with a blank tape} \\ & \text{terminates}\end{aligned}$$

  - Assume Halt' were decidable.

Halting Problem with no Inputs

- Then we can decide $\mathrm{Halt}(e, n)$ as follows:
  - Assume inputs e, n.
  - If e is not a code for a Turing machine, we return 0.
  - Otherwise, let $\mathrm{encode}(\mathrm{T}) = e$.
  - Define a Turing machine V as follows:
    - V first writes $\mathrm{bin}(n)$ on the tape and moves head to the left most bit of $\mathrm{bin}(n)$.
    - Then it executes the Turing machine T.
  - We have
    - V, run with blank tape, terminates

    iff T run with tape containing $\mathrm{bin}(n)$ terminates

    iff $\mathrm{T}^{(1)}(n)\downarrow$

    iff $\{e\}(n)\downarrow$.

Halting Problem with no Inputs

V, run with blank tape, terminates iff $\{e\}(n)\downarrow$.

  - Let $\mathrm{encode}(\mathrm{V}) = e'$. Then

$$\mathsf{Halt}'(e') \Leftrightarrow \mathsf{Halt}(e, n)$$

  - Therefore using the decidability of Halt′ we can decide $\mathsf{Halt}(e, n)$.
- So we have decided Halt, a contradiction.

No Additional Material

For this subsection no additional material has been added yet.