

Towards a Formalisation of Cook's Theorem in Coq

Lennard Gäher

Advisor: Fabian Kunze

Saarland University

10 January 2020

Second Bachelor Seminar Talk

Reminder: Complexity Theory in L

- L: call-by-value λ -calculus
- reasonable model for computational complexity theory¹
- most definitions carry over, i.e.

$$\text{NP-hard } A := \forall B, \text{ inNP } B \rightarrow B \preceq_p A$$

¹ [Forster et al., 2019]

Cook's Theorem³

The satisfiability problem on CNFs **SAT** is NP-hard.

SAT

Given a Boolean formula in conjunctive normal form, does there exist a satisfying assignment?

No *formal* proof available²

²At least none that I am aware of

³*The complexity of theorem-proving procedures* [Cook, 1971]

Generic NP-hard Problem

Idea: encode computation as Boolean formula

L: non-local computations, too high-level ☹️

Generic NP-hard Problem for Turing Machines

Idea: encode computation as Boolean formula

GenNP

GenNP $(M, input, t) := M$ is a nondet. 1-tape TM
 $\wedge M$ accepts on $input$ in $\leq t$ steps

Generic NP-hard Problem for Turing Machines

Idea: encode computation as Boolean formula

GenNP

GenNP $(M, k, t) := M$ is a det. 1-tape TM

$\wedge \exists \textit{input}, |\textit{input}| \leq k$

$\wedge M$ accepts on \textit{input} in $\leq t$ steps

Generic NP-hard Problem for Turing Machines

Idea: encode computation as Boolean formula

GenNP

GenNP $(M, k, t) := M$ is a det. 1-tape TM

$\wedge \exists \textit{input}, |\textit{input}| \leq k$

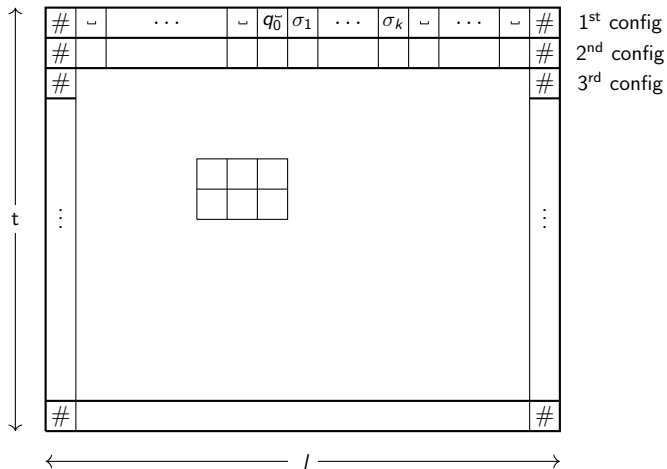
$\wedge M$ accepts on \textit{input} in $\leq t$ steps

$(M, k, t) \in \mathbf{GenNP} \leftrightarrow f(M, k, t) \in \mathbf{SAT}$

Boundedness

SAT formula has a fixed size, but:

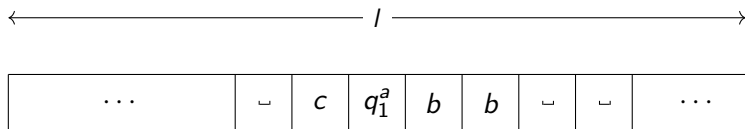
- TM may have different space usage depending on input
- TM may take a different number of steps until it halts

Tableau⁴

⁴based on [Sipser, 1997], similar to [Cook, 1971]

String-based Configurations

$$\Sigma_{\text{TM}} = \{a, b, c\}$$



special blanks □ for unused regions of the string

String-based Configurations

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

$$\delta(q_1, a) = (q_2, \circ b, L)$$

←────────────────── | ───────────────────→

...	␣	c	q_1^a	b	b	␣	␣	...
...	␣	q_2^c	b	b	b	␣	␣	...

special blanks ␣ for unused regions of the string

String-based Configurations

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

$$\delta(q_1, a) = (q_2, \circ b, L)$$

←----- | -----→

...	␣	c	q_1^a	b	b	␣	␣	...
...	␣	q_2^c	b	b	b	␣	␣	...

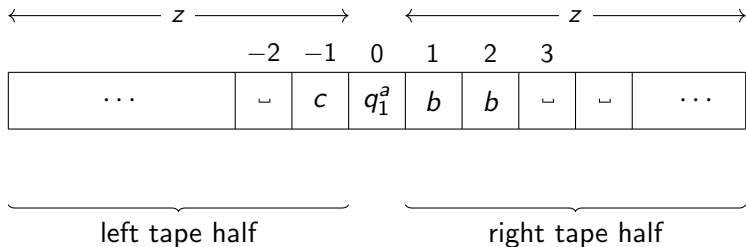
Non-unique representation:

...	␣	␣	c	q_1^a	b	b	␣	...
...	␣	␣	q_2^c	b	b	b	␣	...

special blanks ␣ for unused regions of the string

Fixed State Position

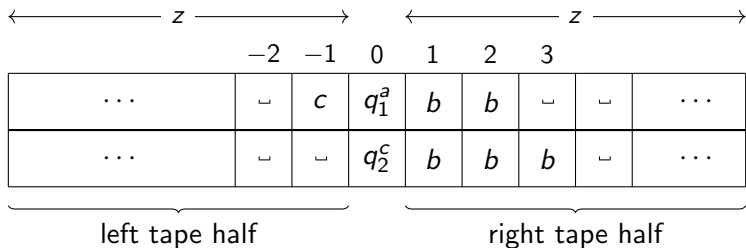
$$\Sigma_{\text{TM}} = \{a, b, c\}$$



Fixed State Position

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

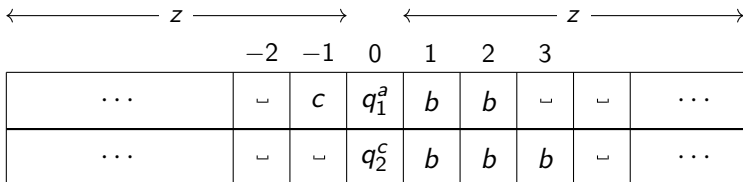
$$\delta(q_1, a) = (q_2, \circ b, L)$$



Rewrite Windows: Force Valid Configuration Changes

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

$$\delta(q_1, a) = (q_2, \circ b, L)$$



Rewrite Windows: Force Valid Configuration Changes

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

$$\delta(q_1, a) = (q_2, \circ b, L)$$

$\leftarrow z \text{ ————— } \rightarrow$ $\leftarrow z \text{ ————— } \rightarrow$

-2 -1 0 1 2 3

...	⊔	c	q_1^a	b	b	⊔	⊔	...
...	⊔	⊔	q_2^c	b	b	b	⊔	...

c	q_1^a	b
⊔	q_2^c	b

Rewrite Windows: Force Valid Configuration Changes

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

$$\delta(q_1, a) = (q_2, \circ b, L)$$

$\longleftarrow z \quad \longrightarrow$ $\longleftarrow z \quad \longrightarrow$

-2 -1 0 1 2 3

...	␣	c	q_1^a	b	b	␣	␣	...
...	␣	␣	q_2^c	b	b	b	␣	...

c	q_1^a	b	q_1^a	b	b
␣	q_2^c	b	q_2^c	b	b

Rewrite Windows: Force Valid Configuration Changes

$$\Sigma_{\text{TM}} = \{a, b, c\}$$

$$\delta(q_1, a) = (q_2, \circ b, L)$$

\leftarrow ————— z ————— \rightarrow \leftarrow ————— z ————— \rightarrow

-2 -1 0 1 2 3

...	␣	c	q_1^a	b	b	␣	␣	...
...	␣	␣	q_2^c	b	b	b	␣	...

c	q_1^a	b
␣	q_2^c	b

q_1^a	b	b
q_2^c	b	b

b	b	␣
b	b	b

Rewrite Rules

Add one symbol to the right half of the tape:

<i>b</i>	<i>b</i>	⊔	⊔	...
<i>b</i>	<i>b</i>	<i>b</i>	⊔	...

<i>b</i>		<i>b</i>		⊔
<hr/>				
<i>b</i>		<i>b</i>		<i>b</i>

Rewrite Rules

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

$$\sigma_i \in \Sigma_{\text{TM}}$$

σ_1	σ_2	\sqcup
σ_3	σ_1	σ_2

Rewrite Rules

Add one symbol to the right half of the tape:

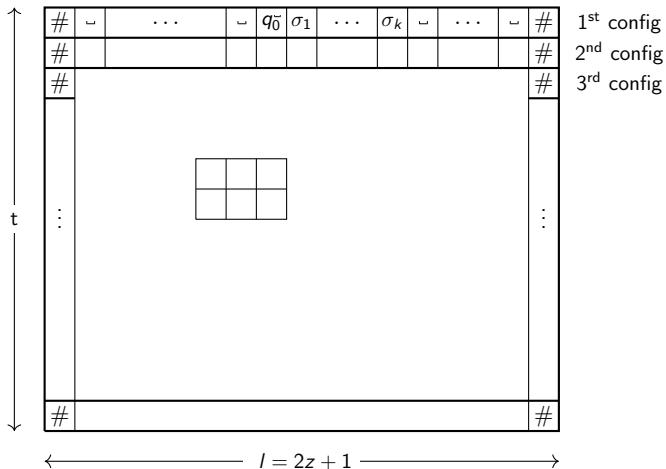
σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

$$\sigma_i \in \Sigma_{\text{TM}}$$

σ_1	σ_2	\sqcup	σ_2	\sqcup	\sqcup	\sqcup	\sqcup
σ_3	σ_1	σ_2	σ_1	σ_2	\sqcup	σ_2	\sqcup

\dots

Tableau: Deterministic Simulation



Parallel Rewriting (**PR**)

Given:

- an alphabet Σ and a string length l
- an initial string $x_0 \in \Sigma^l$ and a step count t
- a width w of rewrite windows
- a set of rewrite windows R
- a set of final substring constraints R_{final}

Determine: $\exists x_1, \dots, x_{t-1} \in \Sigma^l$ s.t.

- $x_i \rightsquigarrow x_{i+1}$: “for all offsets, there exists a rewrite window”
- there exists an element $x \in R_{final}$ which is a substring of x_{t-1}

Nondeterminism

- “Guess” input string of length $\leq k$ with a single rewrite step
- Add symbols $\{\underline{\#}, \underline{=}, \underline{*}, \underline{q^-}\}$ for initial state q

Initial string:

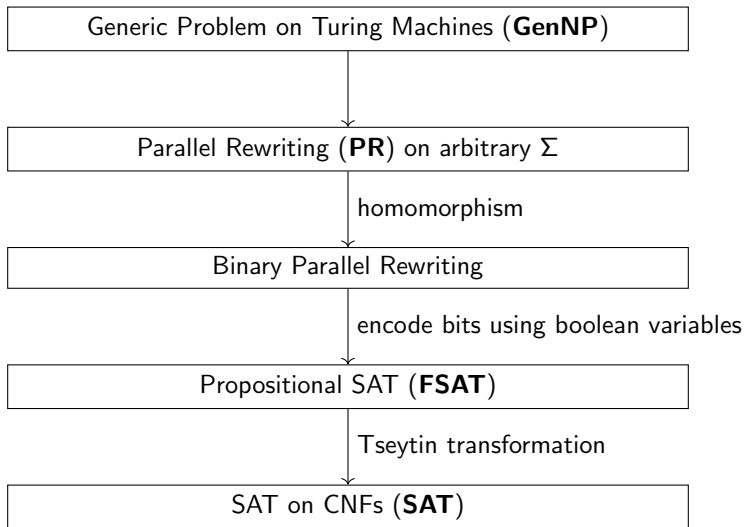
←————— z —————→ ←———— k —————×———— $z - k$ —————→

<u>#</u>	<u>=</u>	...	<u>=</u>	<u>q⁻</u>	<u>*</u>	...	<u>*</u>	<u>=</u>	...	<u>=</u>	<u>#</u>
#	⌊	...	⌊	q ⁻	σ ₁	σ ₂	⌊	...	⌊	#	

Mechanisation: Challenges

- massive number of cases (100): proof *heavily* relies on automation
 - rewrite rules formalised as inductive predicates
- proofs that rewrites are unique require *a lot* of inversions

Reduction of **GenNP** to **SAT**



Conclusion

Contributions:

- factorisation of proof⁵ into tractable parts
- changes to the original construction...
 - ... to fit our notion of Turing machines
 - ... to make inductive proofs work nicely
- Coq: verified reduction of **GenNP** to **PR**

Roadmap:

- reduction of **PR** to binary **PR**
- reduction of binary **PR** to formula satisfiability
- (reduction of formula satisfiability to CNF satisfiability)
- extraction to L

⁵ [Sipser, 1997]

LOC

Component	Spec	Proof
preliminaries	86	196
definition of PR	136	232
single-tape TMs	35	78
nondeterminism: Preludes	47	169
reduction of TM to PR	930	1501
encoding of finite types	9	70
list-based rules (wip)	581	672
total	1824	2918

Tape Shifts

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_3	σ_1	σ_2

Tape Shifts

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

$$\frac{\sigma_1 \mid \sigma_2 \mid \sqcup}{\sigma_3 \mid \sigma_1 \mid \sigma_2} \quad \sigma_i \in \Sigma_{\text{TM}}$$

Tape Shifts

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_3	σ_1	σ_2

$$\sigma_i \in \Sigma_{\text{TM}}$$

Leave the tape unchanged:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_1	σ_2	\sqcup	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_1	σ_2	\sqcup

Tape Shifts

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_3	σ_1	σ_2

$$\sigma_i \in \Sigma_{\text{TM}}$$

Leave the tape unchanged:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_1	σ_2	\sqcup	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_1	σ_2	\sqcup

\dots	\sqcup	c	q_1^a	b	b	\sqcup	\sqcup	\dots
\dots	\sqcup	\sqcup	q_2^c	b	b	b	\sqcup	\dots

Tape Shifts

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_3	σ_1	σ_2	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_3	σ_1	σ_2

$$\sigma_i \in \Sigma_{\text{TM}}$$

Leave the tape unchanged:

σ_1	σ_2	\sqcup	\sqcup	\dots
σ_1	σ_2	\sqcup	\sqcup	\dots

σ_1	σ_2	\sqcup
σ_1	σ_2	\sqcup

\dots	\sqcup	c	q_1^a	b	b	\sqcup	\sqcup	\dots
\dots	\sqcup	\sqcup	q_2^c	b	b	\sqcup	\sqcup	\dots

Tape Shifts

Add one symbol to the right half of the tape:

σ_1	σ_2	␣	␣	...
σ_3	σ_1	σ_2	␣	...

σ_1	σ_2	␣
σ_3	σ_1	σ_2

$$\sigma_i \in \Sigma_{\text{TM}}$$

Leave the tape unchanged:

σ_1	σ_2	␣	␣	...
σ_1	σ_2	␣	␣	...

σ_1	σ_2	␣
σ_1	σ_2	␣

...	␣	c	q_1^a	b	b	␣	␣	...
...	␣	␣	q_2^c	b	b	␣	␣	...

Polarities $\{\overleftarrow{\cdot}, \overline{\cdot}, \overrightarrow{\cdot}\}$

Add one symbol to the right half of the tape:

σ_1	σ_2	$_$	$_$	\dots
$\overrightarrow{\sigma_3}$	$\overrightarrow{\sigma_1}$	$\overrightarrow{\sigma_2}$	$_$	\dots

σ_1	σ_2	$_$
$\overrightarrow{\sigma_3}$	$\overrightarrow{\sigma_1}$	$\overrightarrow{\sigma_2}$

Leave the tape unchanged:

σ_1	σ_2	$_$	$_$	\dots
$\overline{\sigma_1}$	$\overline{\sigma_2}$	$_$	$_$	\dots

σ_1	σ_2	$_$
$\overline{\sigma_1}$	$\overline{\sigma_2}$	$_$

Polarities $\{\overleftarrow{\cdot}, \overline{\cdot}, \overrightarrow{\cdot}\}$

Add one symbol to the right half of the tape:

σ_1	σ_2	\sqcup	\sqcup	\dots
$\overrightarrow{\sigma_3}$	$\overrightarrow{\sigma_1}$	$\overrightarrow{\sigma_2}$	\sqcup	\dots

σ_1	σ_2	\sqcup
$\overrightarrow{\sigma_3}$	$\overrightarrow{\sigma_1}$	$\overrightarrow{\sigma_2}$

Leave the tape unchanged:

σ_1	σ_2	\sqcup	\sqcup	\dots
$\overline{\sigma_1}$	$\overline{\sigma_2}$	\sqcup	\sqcup	\dots

σ_1	σ_2	\sqcup
$\overline{\sigma_1}$	$\overline{\sigma_2}$	\sqcup

\dots	\sqcup	c	q_1^a	b	b	\sqcup	\sqcup	\dots
\dots	\sqcup	\sqcup	q_2^c	\overrightarrow{b}	\overrightarrow{b}	\overrightarrow{b}	\sqcup	\dots

Transition Rules – Example

$$m \in \Sigma_{\text{TM}} \cup \{\sqcup\}, \sigma \in \Sigma_{\text{TM}}$$

$$\delta(q, a) = (p, \circ b, L):$$

$$\begin{array}{c}
 \begin{array}{c|c|c} \sqcup & q^a & m_1 \\ \hline \sqcup & p^\sqcup & \vec{b} \end{array} \quad \begin{array}{c|c|c} \sigma_1 & q^a & m_1 \\ \hline \vec{m}_2 & p^{\sigma_1} & \vec{b} \end{array} \\
 \\
 \begin{array}{c|c|c|c|c|c} \sqcup & \sqcup & q^a & \sqcup & \sigma_1 & q^a \\ \hline \sqcup & \sqcup & p^\sqcup & \sqcup & \sqcup & p^{\sigma_1} \end{array} \quad \begin{array}{c|c|c|c|c|c} \sigma_1 & \sigma_2 & q^a & \vec{m}_1 & \vec{\sigma}_1 & p^{\sigma_2} \\ \hline \sigma_1 & \sigma_2 & q^a & \vec{m}_1 & \vec{\sigma}_1 & p^{\sigma_2} \end{array} \\
 \\
 \begin{array}{c|c|c} q^a & \sqcup & \sqcup \\ \hline p^{m_1} & \vec{b} & \sqcup \end{array} \quad \begin{array}{c|c|c} q^a & \sigma_1 & m_1 \\ \hline p^{m_2} & \vec{b} & \vec{\sigma}_1 \end{array}
 \end{array}$$

Transition Rules – Example

$$m \in \Sigma_{\text{TM}} \cup \{\perp\}, \sigma \in \Sigma_{\text{TM}}$$

In Coq mechanisation:

$$\delta(q, a) = (p, \circ b, L):$$

$$\frac{m_1 \mid q^a \mid m_2}{\vec{m}_3 \mid p^{m_1} \mid \vec{b}} \quad \frac{m_1 \mid m_2 \mid q^a}{\vec{m}_3 \mid \vec{m}_1 \mid p^{m_2}} \quad \frac{q^a \mid m_1 \mid m_2}{p^{m_3} \mid \vec{b} \mid \vec{m}_1}$$

Contains garbage, i.e.

$$\frac{\perp \mid q^a \mid \perp}{\vec{\sigma} \mid p^{\perp} \mid \vec{b}}$$

Representation Relations

Representation of tape halves:

$$u \sim_t^n \underbrace{\boxed{u \quad _ \quad _ \quad \dots \quad _ \quad _ \quad \#}}_n$$

Representation of configurations:

$$q; (ls, \sigma, rs) \sim_c \boxed{\text{rev left} \quad q^\sigma \quad \text{right}}, \text{ where:}$$

- $ls \sim_t^{z'} \text{ left}$
- $rs \sim_t^{z'} \text{ right}$

Deterministic Simulation

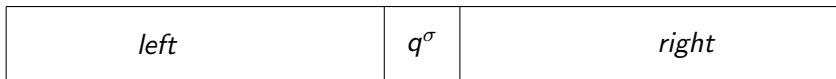
$$\begin{array}{ccc} q; (ls, \sigma, rs) \sim_c & \boxed{\text{rev left} \mid q^\sigma \mid \text{right}} \\ \Upsilon & \downarrow \\ q'; (ls', \sigma', rs') \sim_c & \boxed{\text{rev left}' \mid q'^{\sigma'} \mid \text{right}'} \end{array}$$

where $ls \sim_t^{z'} \text{left}$, $rs \sim_t^{z'} \text{right}$ and $ls' \sim_t^{z'} \text{left}'$, $rs' \sim_t^{z'} \text{right}'$

Deterministic Simulation

$$\begin{array}{ccc}
 q; (ls, \sigma, rs) \sim_c & \boxed{\text{rev left} \mid q^\sigma \mid \text{right}} \\
 \Upsilon & \downarrow \\
 q'; (ls', \sigma', rs') \sim_c & \boxed{\text{rev left}' \mid q'^{\sigma'} \mid \text{right}'}
 \end{array}$$

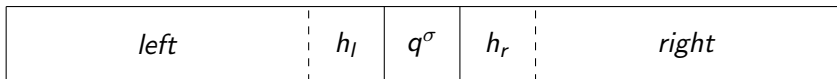
where $ls \sim_t^{z'} \text{left}$, $rs \sim_t^{z'} \text{right}$ and $ls' \sim_t^{z'} \text{left}'$, $rs' \sim_t^{z'} \text{right}'$



Deterministic Simulation

$$\begin{array}{ccc}
 q; (ls, \sigma, rs) \sim_c & \boxed{\text{rev left} \mid q^\sigma \mid \text{right}} \\
 \Upsilon & \downarrow \zeta \\
 q'; (ls', \sigma', rs') \sim_c & \boxed{\text{rev left}' \mid q'^{\sigma'} \mid \text{right}'}
 \end{array}$$

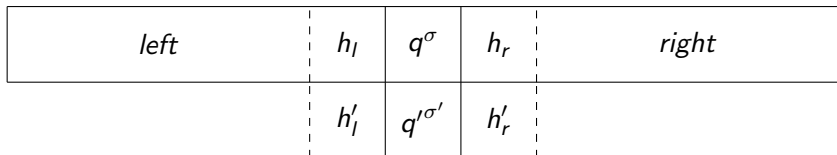
where $ls \sim_t^{z'} \text{left}$, $rs \sim_t^{z'} \text{right}$ and $ls' \sim_t^{z'} \text{left}'$, $rs' \sim_t^{z'} \text{right}'$



Deterministic Simulation

$$\begin{array}{ccc}
 q; (ls, \sigma, rs) \sim_c & \boxed{\text{rev left} \mid q^\sigma \mid \text{right}} \\
 \Upsilon & \downarrow \\
 q'; (ls', \sigma', rs') \sim_c & \boxed{\text{rev left}' \mid q'^{\sigma'} \mid \text{right}'}
 \end{array}$$

where $ls \sim_t^{z'} \text{left}$, $rs \sim_t^{z'} \text{right}$ and $ls' \sim_t^{z'} \text{left}'$, $rs' \sim_t^{z'} \text{right}'$



Deterministic Simulation

$$\begin{array}{ccc}
 q; (ls, \sigma, rs) \sim_c & \boxed{\text{rev left} \mid q^\sigma \mid \text{right}} \\
 \Upsilon & \downarrow \\
 q'; (ls', \sigma', rs') \sim_c & \boxed{\text{rev left}' \mid q'^{\sigma'} \mid \text{right}'}
 \end{array}$$

where $ls \sim_t^{z'} \text{left}$, $rs \sim_t^{z'} \text{right}$ and $ls' \sim_t^{z'} \text{left}'$, $rs' \sim_t^{z'} \text{right}'$

<i>left</i>	<i>h_l</i>	<i>q^σ</i>	<i>h_r</i>	<i>right</i>
$\exists! \text{left}'$	<i>h'_l</i>	<i>q'^{σ'}</i>	<i>h'_r</i>	$\exists! \text{right}'$

Tape Transformations

Add symbol:

$$rs \sim_t h \wedge |rs| < z' \rightarrow \exists! h', (h \rightsquigarrow \vec{a} :: h') \wedge a :: rs \sim_t^+ \vec{a} :: h'$$

$$ls \sim_t h \wedge |ls| < z' \rightarrow \exists! h', (\text{rev}h \rightsquigarrow \text{rev}\overleftarrow{a} :: h') \wedge a :: ls \sim_t^- \overleftarrow{a} :: h'$$

Remove symbol:

$$a :: b :: rs \sim_t a :: b :: h \rightarrow \exists! h', (a :: b :: h \rightsquigarrow \overleftarrow{b} :: h') \wedge b :: rs \sim_t^- \overleftarrow{b} :: h'$$

Leave unchanged:

$$a :: rs \sim_t a :: h \rightarrow \exists! h', (a :: h \rightsquigarrow \bar{a} :: h) \wedge a :: rs \sim_t^{\circ} \bar{a} :: h'$$

Main Simulation Results

Completeness

Let (q, tape) be a configuration with $|\text{tape}| \leq k$. There exists s with $(q, \text{tape}) \sim_c s$.

If $(q, \text{tape}) \triangleright^{\leq t} (q', \text{tape}')$, then there exists s' with $s \rightsquigarrow^t s'$, $(q', \text{tape}') \sim_c s'$ and $s' \models R_{\text{final}}$.

Soundness

Let s be given such that $(q, \text{tape}) \sim_c s$ and $|\text{tape}| \leq k$ for some q, tape .

If $s \rightsquigarrow^t s'$ and $s' \models R_{\text{final}}$, then there exists (q', tape') with $(q', \text{tape}') \sim_c s'$ such that $(q, \text{tape}) \triangleright^{\leq t} (q', \text{tape}')$ and $|\text{tape}'| \leq z'$.

Challenges due to Turing Machine Formalisation

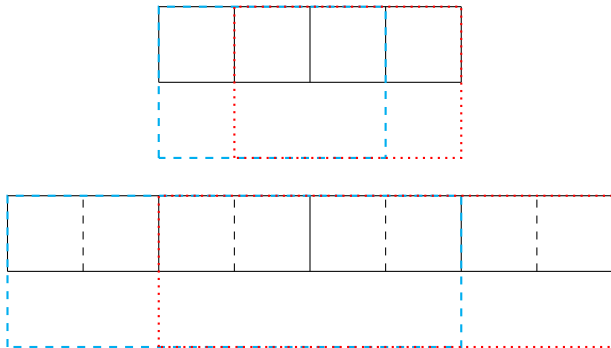
- left half of the tape is reversed wrt the Turing machine formalisation
 - use symmetry of rewrite rules for tapes
- Turing machine formalisation does not have notion of blanks
 - mechanisation: blanks *also* have polarities

Reduction to Binary Alphabet

$$\Sigma = \{\sigma_1, \dots, \sigma_n\}$$

$$\text{Homomorphism: } f : \Sigma \rightarrow \{0, 1\}^n, \sigma_i \mapsto 0^{i-1}10^{n-i}$$

Example ($|\Sigma| = 2$):



Parallel Rewriting (**PR**)

Given:

- an alphabet Σ and a string length l
- an initial string $x_0 \in \Sigma^l$ and a step count t
- a width w of rewrite windows and a rewriting offset o
- a set of rewrite windows R
- a set of final substring constraints R_{final}

Determine: $\exists x_1, \dots, x_{t-1} \in \Sigma^l$ s.t.

- $x_i \rightsquigarrow x_{i+1}$: “for all offsets, there exists a rewrite window”
- there exists an element $x \in R_{final}$ which is a substring of x_{t-1}




String Rewriting (SR) (and why it does not work for us)

- rules u/v where $u, v \in \Sigma^*$
- string rewriting system R over Σ : finite set of rules
- rewrite relation \Rightarrow_R ; given x, y , determine whether $x \Rightarrow_R^* y$

Problems:

- essentially unbounded, would require a modified restricted version for **SAT**; *this is the hard part*
- only a single final string
- only a single rewrite in each step, does not allow tape shifting

References

-  Bläser, M.
Theoretical computer science: An introduction.
-  Cook, S. A. (1971).
The complexity of theorem-proving procedures.
In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA. ACM.
-  Forster, Y., Kunze, F., and Roth, M. (2019).
The weak call-by-value lambda-calculus is reasonable for both time and space.
Technical report.
Full version appeared as arXiv:1902.07515 To appear.

References



Sipser, M. (1997).

Introduction to Theory of Computation.

PWS Publishing Company, 1 edition.