Joint Estonian-Latvian Theory Days in Riga

Probabilistic verification of all languages

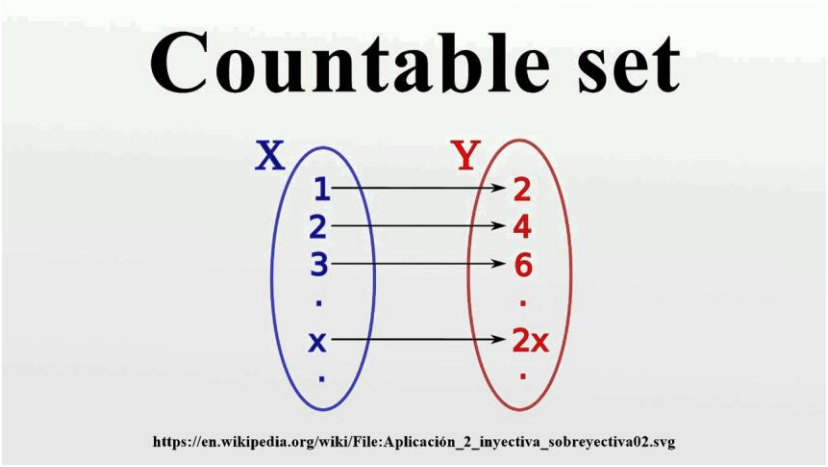
University of Latvia
Faculty of computing
PhD candidate

Maksims Dimitrijevs,
Abuzer Yakaryılmaz

Deterministic TM

Deterministic Turing machines can recognize only countably many languages. We can write the program of a TM in binary, and then enumerate all possible programs in ascending lexicographical order.

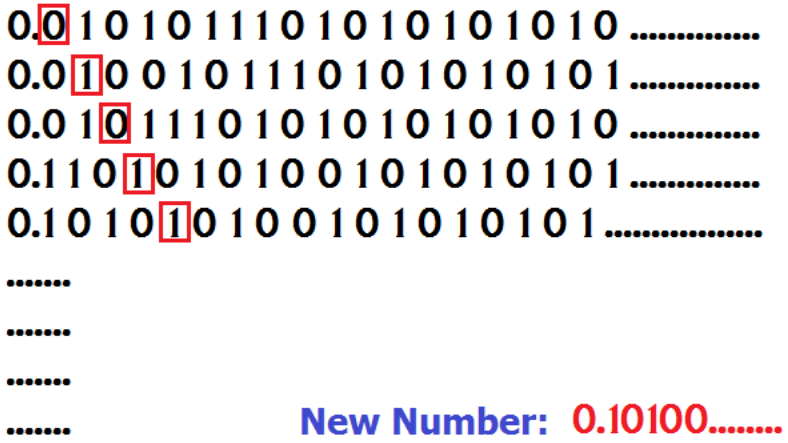
0
1
00
01
10
11
000



$\aleph_0$

Uncountable transitions

Probabilistic or quantum models can be defined with real transition values, therefore, their cardinalities are uncountably many.



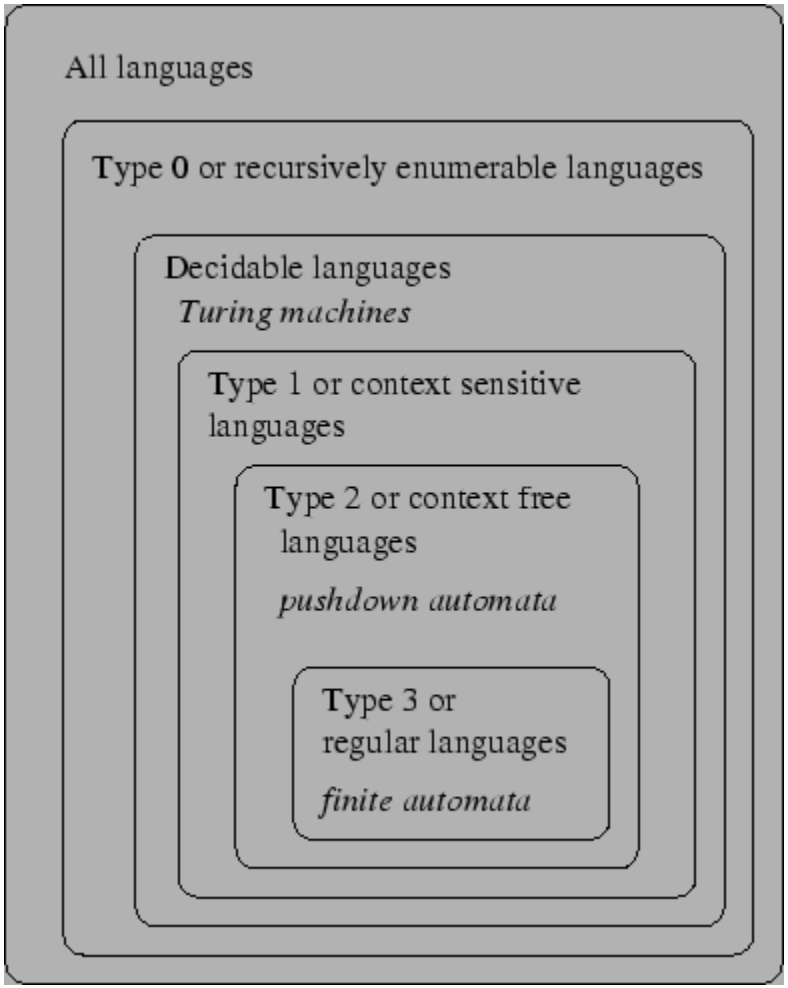
$\aleph_1$

Our scope

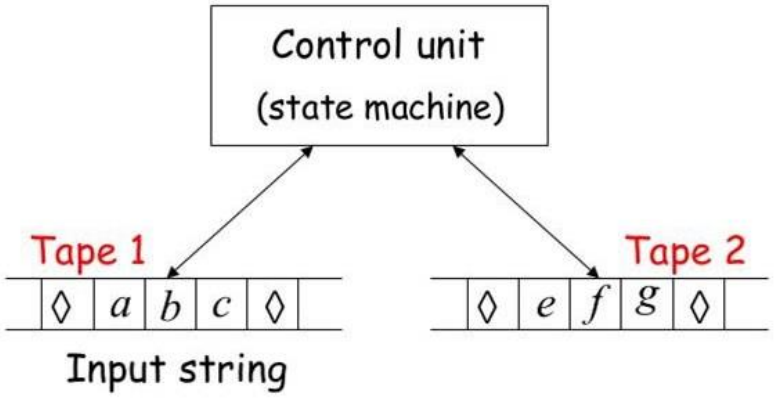
How many resources is enough for probabilistic models to define uncountably many languages?

We investigate different bounded-error probabilistic models.

All formal languages



Probabilistic TM

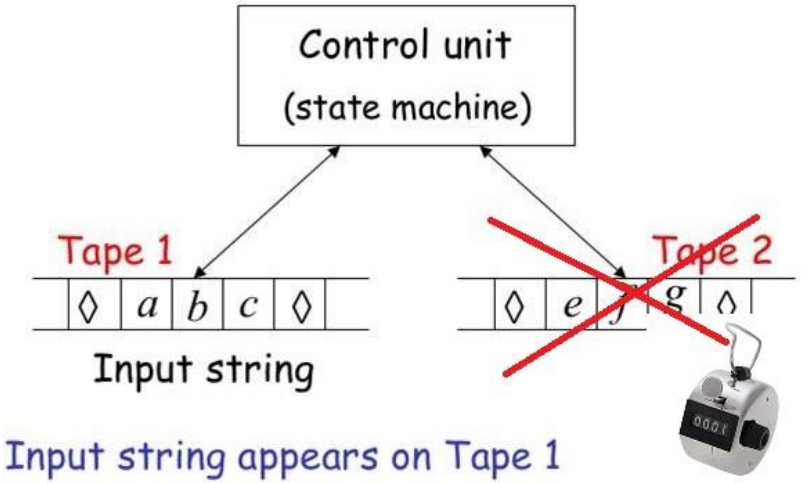


Input string appears on Tape 1

$$\delta: S \times \tilde{\Sigma} \times \tilde{\Gamma} \rightarrow S \times \tilde{\Gamma} \times \{\leftarrow, \downarrow, \rightarrow\} \times \{\leftarrow, \downarrow, \rightarrow\}$$

$$\delta: S \times \tilde{\Sigma} \times \tilde{\Gamma} \times S \times \tilde{\Gamma} \times \{\leftarrow, \downarrow, \rightarrow\} \times \{\leftarrow, \downarrow, \rightarrow\} \rightarrow [0,1]$$

Probabilistic counter automaton

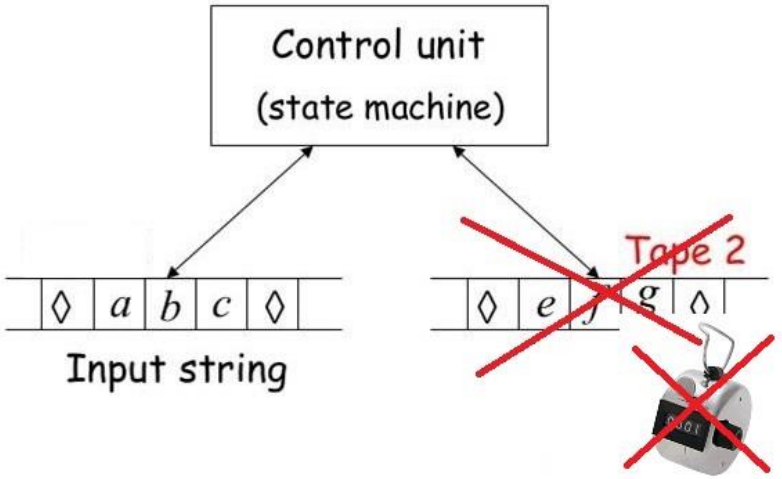


$$\delta: S \times \tilde{\Sigma} \times \{0,1\}^k \times S \times \{\leftarrow, \downarrow, \rightarrow\} \times \{-1,0,1\}^k \rightarrow [0,1]$$

Operations with the counter:

- check, whether the value is zero (?=0),
- update the value of the counter +{-1,0,1}.

Probabilistic finite automaton



$\delta: S \times \tilde{\Sigma} \times S \times \{\leftarrow, \downarrow, \rightarrow\} \rightarrow [0,1]$

Restricted input head

- 2-way model.
- Sweeping model.
- One-way model.

Recognition

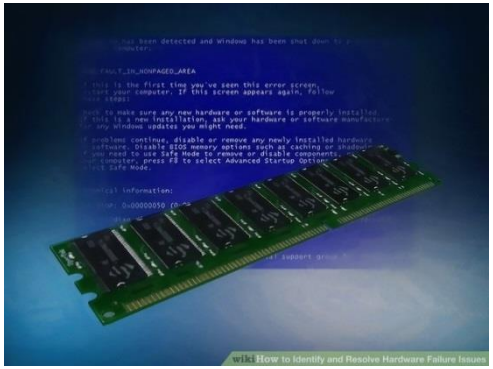
Language $L \subseteq \Sigma^*$ is said to be recognized by a machine M with error bound ϵ if:

- any member is accepted by M with probability at least $1 - \epsilon$,
- any non-member is rejected by M with probability at least $1 - \epsilon$.

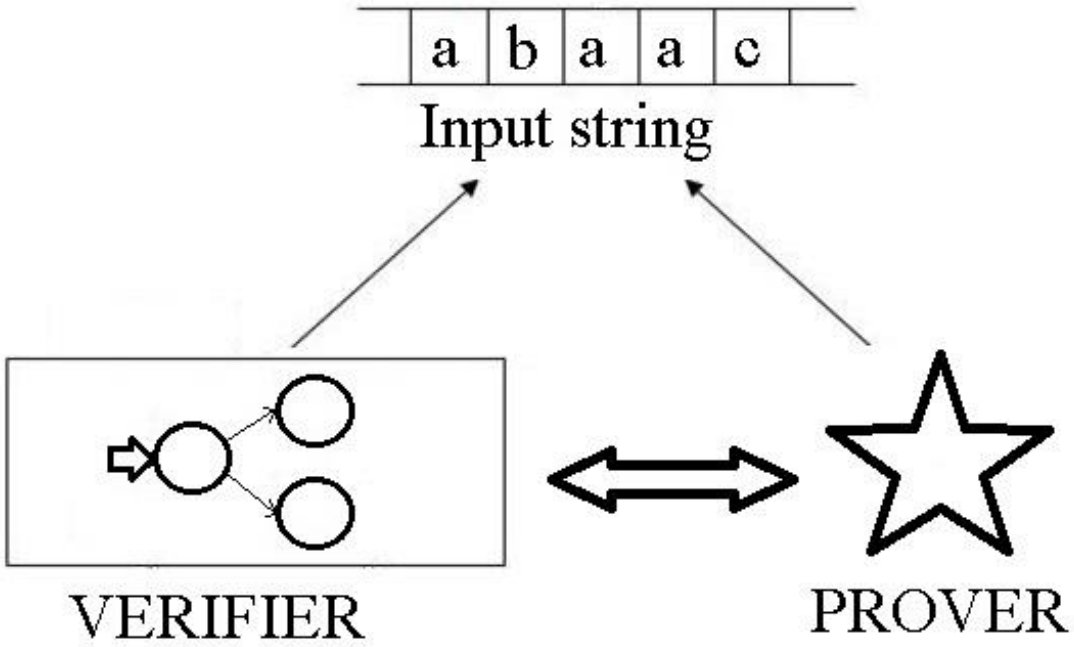
Bounded error

- The probability of correct answer is higher than the probability of error.
- We can repeat the calculation and choose the most frequent answer as a result.

Even if the probability of correct answer is a bit higher than $\frac{1}{2}$, with probability amplification we can obtain arbitrarily small probability of error.



Interactive proof system



Verification

The language $L \subseteq \Sigma^*$ is verifiable by V with error bound $\epsilon < \frac{1}{2}$ if:

- there exists a honest prover P such that any $x \in L$ is accepted by V with probability at least $1 - \epsilon$ by communicating with P, and,
- any $x \notin L$ is always rejected by V with probability at least $1 - \epsilon$ when communicating with any possible prover P^*.

Verifier for EQUAL

$$EQUAL = \{0^m 1 0^n | m = n\}$$

- $w = 0001000$
- $y = 000$

- $w = 0001000$
- $y = 000$

Interaction with two provers

- Two provers (P_1, P_2) and a probabilistic verifier (V).
- Different communication channel with each prover.
- One prover does not see the communication with the other prover.
- Multi-Prover model – provers collaborate.
- Noisy-Oracle model – both provers oppose each other.

Simulation of a work tape

Uriel Feige, Adi Shamir proposed the following solution:

- P_1 and P_2.
- Contents of the work tape: $m = m_1 m_2 m_3 m_4$ ....
- V secretly picks random values a, b, r_0, each between 0 and $p-1$, where p is a prime number.
- $s_i = (m_i * a + r_i * b + r_{i-1}) \, mod \, p$, where r_i is picked randomly and $0 \leq r_i \leq p-1$.

Simulation of a work tape

- P_1 and P_2.
- $s_i = (m_i * a + r_i * b + r_{i-1}) \, mod \, p$.
- $(m_1, r_1, s_1), (m_2, r_2, s_2), (m_3, r_3, s_3), (m_4, r_4, s_4), \ldots$
- To read the content, V asks all data and checks the correctness of signatures.
- To update the content, V picks new r_0 and scans the input. For each triple (m_i, r_i, s_i), V generates new r_i, recalculates s_i, and asks the provers to replace the values.

Simulation of a work tape

- $s_i = (m_i * a + r_i * b + r_{i-1}) \, mod \, p$.
- The provers cannot learn the values of secretly picked a and b from the information provided by the verifier.
- If $s'_i = (m'_i * a + r'_i * b + r_{i-1}) \, mod \, p$:

$$(s'_i - s_i) = ((m'_i - m_i) * a + (r'_i - r_i) * b + r_{i-1}) \, mod \, p$$

Exactly p pairs of (a, b)'s satisfy this equation, and there are total p^2 different pairs of (a, b)'s. The probability to cheat successfully is $\frac{1}{p}$.

Lemma for 64^k coin flips

Let $x = x_1x_2x_3$ ... be an infinite binary sequence. If a biased coin lands on head with binary probability value $p = 0.x_101x_201x_301$ ..., then the value x_k can be determined with probability $\frac{3}{4}$ after 64^k coin tosses.

Recognition of any language

- Encode the language into $p = 0.x_1 01 x_2 01 x_3 01 \ldots$, $x_k = 1 \leftrightarrow \Sigma^*(k) \in L$. So, we order all elements of Σ^* lexicographically, $\Sigma^*(1) = \varepsilon$.
- We have $\Sigma^*(k)$ on the input tape, our task is to compute k.
- After computing k, write on the work tape $1(000000)^k$, which is 64^k.
- Perform 64^k coin tosses, get the value x_k.
- Exponential space complexity. Double exponential time complexity.

1PFA verifier

- Use the algorithm for the recognition of any language with bounded error.
- Interact with two provers to simulate the work tape.
- Read the input once and store it on the “work tape”.

Other results

Recognition of any language

Alphabet	Machine	Space	Time
unary	PTM	$O(n)$	$O(2^n)$
binary	PTM	$O(2^n)$	$O(2^{2^n})$

Verification of any language

Alphabet	Machine	Space	Time
unary	PTM	$O(logn)$	$O(2^n)$
binary	PTM	$O(n)$	$O(2^{2^n})$

Verification of any language

Condon and Lipton – the prover provides the computational steps for the verifier.

- Constant-space verifier interacts with one prover, but non-members may not be rejected with high probability.
- We first show how to obtain a 1P4CA for every language, and then how to simulate the computation – the prover provides the values of the counters.

Uncountable set

$\mathrm{I} = \{I | I \subseteq Z^{+}\}$, cardinality of I is $\aleph_1$

We can map this set to the set of real numbers in interval $[0; 1)$:

$0.x_1x_2x_3 \ldots, x_i = 1 \leftrightarrow i \in I$

Uncountably many languages

- $\left\{a^{64^k} | k \in I\right\}$ - the potential members of the recognizable language, say L_I.
- The set I is encoded into $p_I = 0.x_1 01 x_2 01 x_3 01 \ldots$, $x_k = 1 \leftrightarrow k \in I$.
- $\mathrm{I} = \{I | I \subseteq Z^+\}$ is uncountable set, bijection between I and L_I.

Results

Verification of uncountably many languages

Alphabet	Machine	Space	Time
unary	2PFA	$O(1)$	$O(n^2)$
binary	sweeping PFA	$O(1)$	$O(n)$

Open question

Is it possible to verify any language with constant space by interacting with a single prover and by guaranteeing the rejecting of any non-member with high probability?

Thank you for your attention!

Aitäh!

Paldies!