One-counter verifiers for decidable languages

ABUZER YAKARYILMAZ

Faculty of Computing

University of Latvia

arXiv:1207.3880

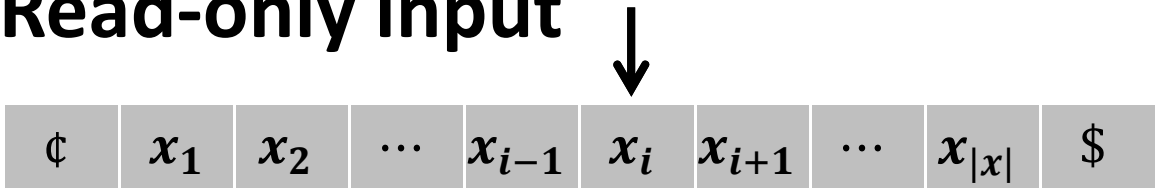
September 28, 2012

Medzābaki

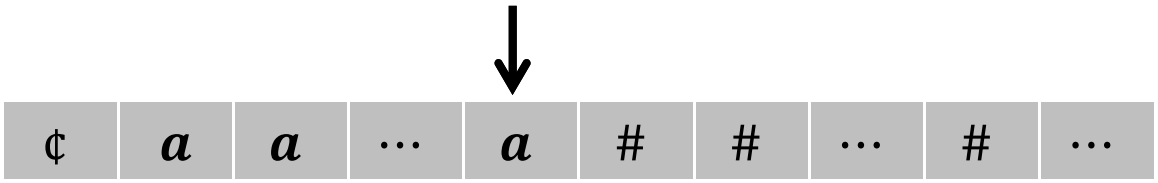
Overview of models – 1

Two-way finite automata with one counter

- **Read-only input**



- **Memory**
  - **A finite set of states**
  - **A counter holds an integer (a unary stack)**



Overview of models – 2

Two-way probabilistic one-counter automaton (2pca)

- **Source of randomness**
  - **A random number generator, a coin, etc.**

Overview of models – 3

Two-way one-counter automaton with quantum and classical states (2qcca)

- **Source of quantumness**
  - **A fixed size of quantum register (in addition to the classical states)**
    - **The computation is governed classically.**
    - **The classical part determines the operators applied to the quantum register, and also reads the measurement outcomes.**
- **The counter is still classical!**

- **If we replace the classical counter with a quantum one, then we obtain a two-way quantum one-counter automaton (2qca).**
- **If we remove the counter, then we obtain a two-way automaton with quantum and classical states (2qcfa).**

LANGUAGE RECOGNIZER

&

VERIFIER

(as a part of interactive proof systems)

LANGUAGE RECOGNIZER

A 2qc**c**a algorithm

$$\text{TWIN} = \{u\#u \mid u \in \{a,b\}^*\}$$

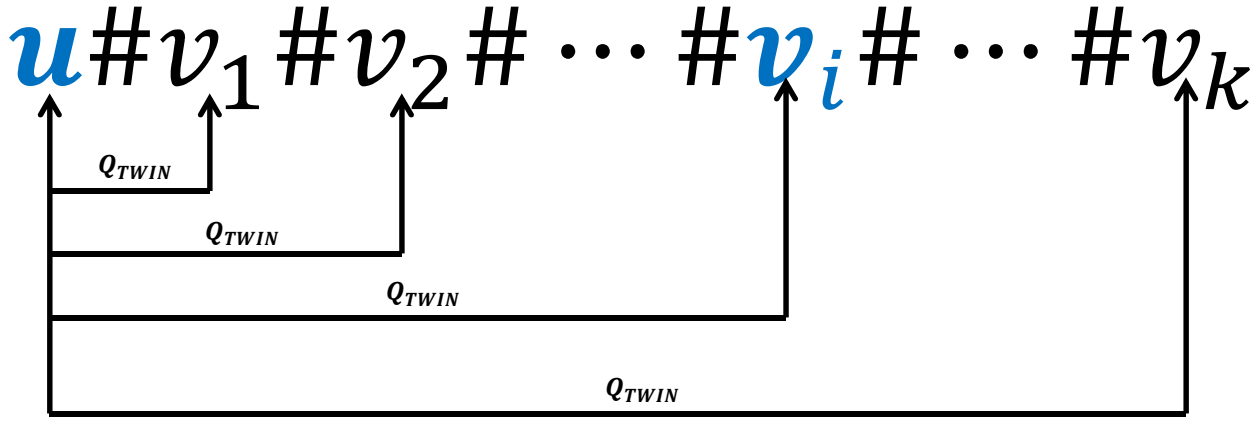
$$\text{EXISTTWIN} = \{\boldsymbol{u}\#v_1\#v_2\#\cdots\#\boldsymbol{v_i}\#\cdots\#v_k \mid k > 0,$$
$$u, v_j \in \{a,b\}^* \ (1 \le j \le k), and \ \exists i \in \{1,\ldots,k\}(u = v_i)\}$$

Let Q_{TWIN} be a 2qc**f**a for language TWIN such that

- It accepts the members exactly, and
- It rejects the non-members with a probability at least $\frac{4}{5}$.

A 2qcca algorithm for EXISTTWIN – 1

$$\text{EXISTTWIN} = \{u\#v_1\#v_2\#\cdots\#v_i\#\cdots\#v_k\}$$



FOR $i = 1$ TO k (v_i is selected)
RUN $\mathcal{Q}_{\text{TWIN}}$ on $x' = u\#v_i$
IF $\mathcal{Q}_{\text{TWIN}}$ accepts x' THEN **TERMINATE** FOR-LOOP
IF $\mathcal{Q}_{\text{TWIN}}$ rejects x' AND $i = k$ THEN **REJECT** the input
END FOR
ACCEPT x with probability $\left(\frac{1}{5}\right)^k$
RESTART the algorithm

A 2qcca algorithm for EXISTTWIN – 2

FOR $i = 1$ TO k (v_i is selected)
  RUN $\mathcal{Q}_{\text{TWIN}}$ on $x' = u\#v_i$
    IF $\mathcal{Q}_{\text{TWIN}}$ accepts x' THEN **TERMINATE** FOR-LOOP
    IF $\mathcal{Q}_{\text{TWIN}}$ rejects x' AND $i = k$ THEN **REJECT** the input
END FOR
ACCEPT x with probability $\left(\frac{1}{5}\right)^k$
RESTART the algorithm

The analysis of the algorithm

- If the input is a member of the language, then
  - the input is **never rejected**,
  - but accepted with some nonzero probability after each execution of the algorithm.

  ⇒ The members are accepted exactly.
- If the input is not a member of the language, then, *in a single execution of the algorithm,*
  - The input is rejected with probability $\left(\frac{4}{5}\right)^k$, and
  - accepted with a probability no more than $\left(\frac{1}{5}\right)^k$.

  ⇒ The nonmembers are accepted with a probability at most $\frac{1}{5}$.

Other results regarding recognizers

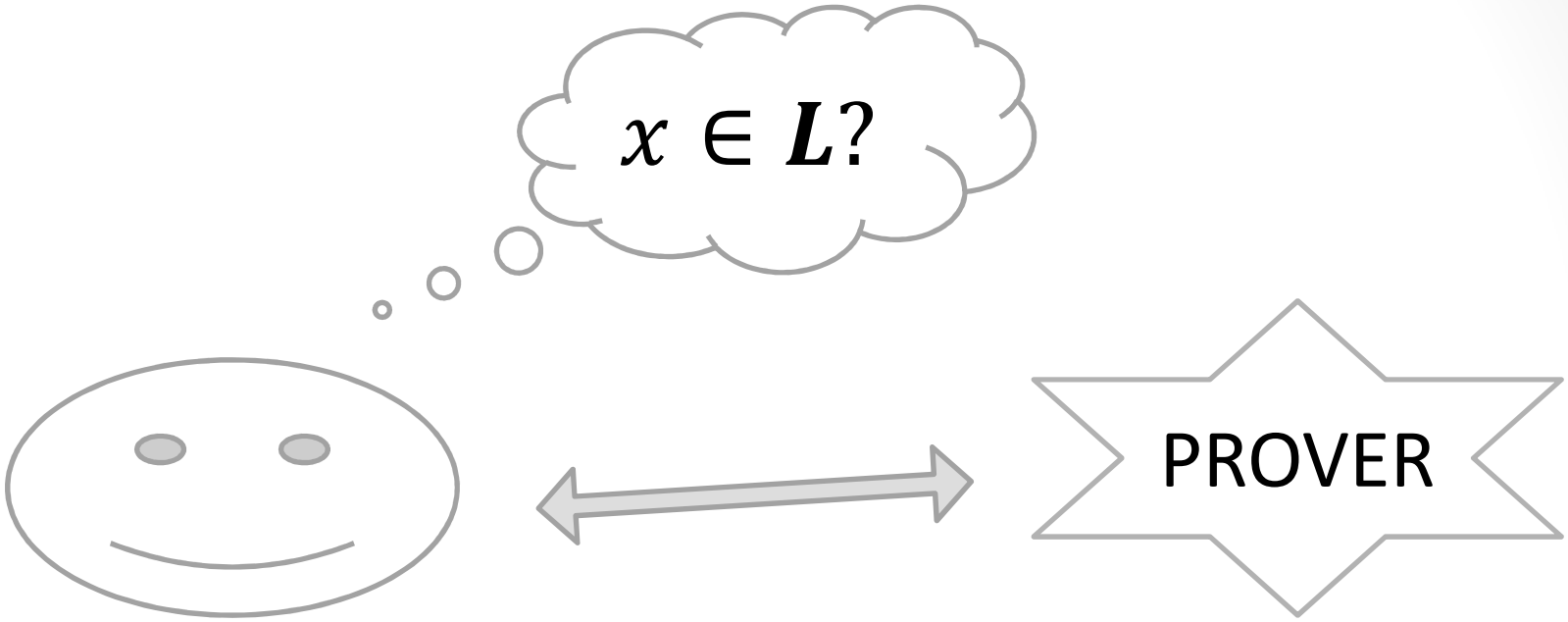
- 2qc**c**a's can recognize $\mathrm{USQAURE}=\{b^{n^2}|n \geq 1\}$ with bounded error by using a 2qc**f**a recognizing $\mathrm{SQUARE} = \{a^n b^{n^2}|n \geq 1\}$ with bounded error.

$$bbbbb \cdots bbbb \cdots bbbb$$
$$\uparrow \qquad\qquad\quad \uparrow$$
$$aaaaa \cdots aaaaaa$$

- We also present a bounded-error simulation of a given two-way nondeterministic one-counter automaton by 2pca's.

VERIFIER
(as a part of interactive proof systems)

An interactive proof system for a language



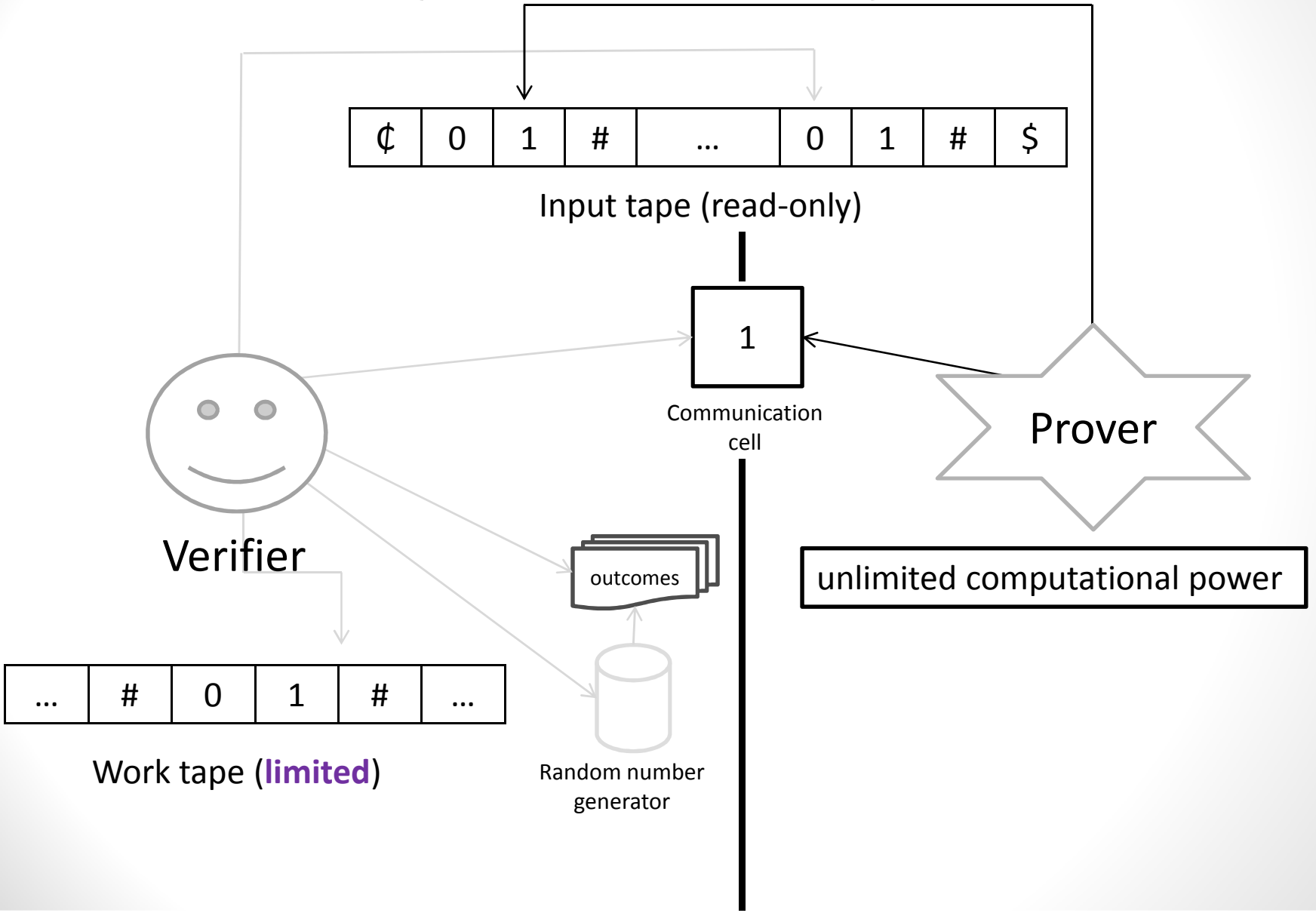
VERIFIER

limited computational power

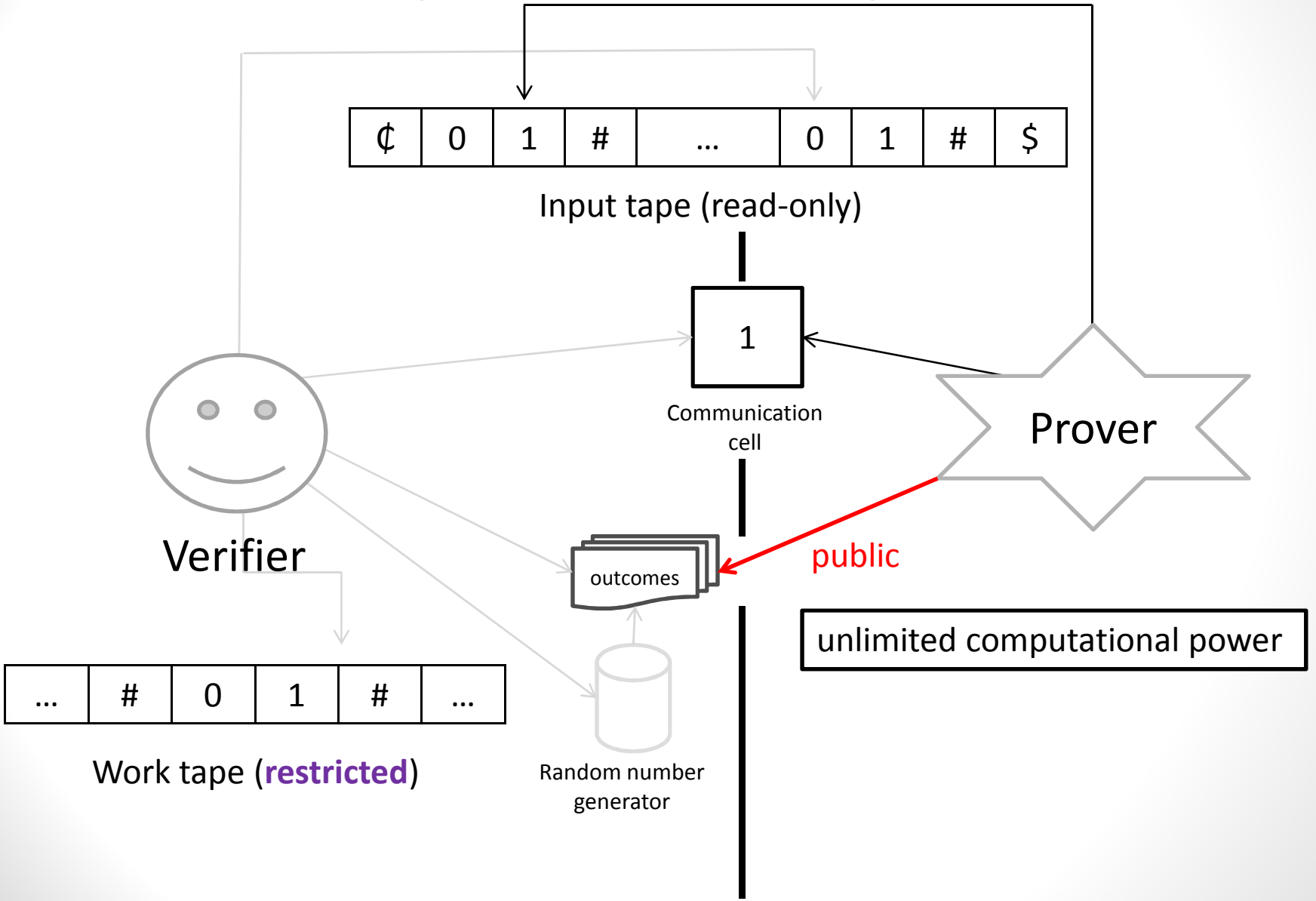
unlimited computational power

The prover can ***cheat***!

Interactive proof systems (IPS) (space-bounded)



Public IPS (Arthur-Merlin games) (space-bounded)



Two criteria for IPS:

Language L has a proof system (P, V) if

COMPLETENESS

For every w $\in L$, the verifier V always accepts with high probability after interacting with the prover P.

SOUNDNESS

For every $w \notin L$ and every P^*, the verifier V rejects with high probability after interacting with P^*.

Motivation

- Condon and Lipton (FOCS 1989) showed that the class of languages having a space-bounded IPS is a proper subset of decidable languages, where the verifier is a probabilistic Turing machine.
  - It is bounded by double-exponential alternating time

- What happens if we use architecturally restricted verifiers instead of restricting working memory: Replacing the work tape with a single counter?

The class names

IP(2pca) – the class of languages having an IP system with a 2pca verifier.

IP(2qcca) – the class of languages having an IP system with a 2qcca verifier

AM(2qca) – the class of languages having an AM system with a 2qca verifier, where the measurement outcomes are shared with the prover.

Universality of two-counter

- Any Turing-recognizable language can be recognized by a two-way deterministic two-counter automaton (2d2ca).

- The head divides a tape into two parts and each part can be represented by a binary number.
- Two counters can keep these two numbers. By using a few more auxiliary counter, the updates on the tape can be implemented on the counters.
- Any number of counter can be simulated by two counters.

- If the simulated TM is a decider, then the 2d2ca is a decider, too.

Each decidable language is in IP(2pca) – 1

Let L be a decidable language recognized by a 2d2ca (decider), say D.

A configuration of a D can be represented by a 4-tuple:

(head-position,state,value-of-1st counter,value-of-2nd counter)

We present a protocol simulating the computation of D on a given input x.

- If the prover provides the contents of the counters, then the verifier can simulate D on x.
  - The contents of the counters: $a^{u_1}b^{v_1}\#a^{u_2}b^{v_2}\#\cdots\#a^{u_t}b^{v_t}\#$

What happens if the prover is cheating?

- The verifier can compare one of four choices with its counter:
  1. (u_1 and u_2), (u_3 and u_4), (u_5 and u_6), …
  2. (u_2 and u_3), (u_4 and u_5), (u_6 and u_7), …
  3. (v_1 and v_2), (v_3 and v_4), (v_5 and v_6), …
  4. (v_2 and v_3), (v_4 and v_5), (v_6 and v_7), …

  So the verifier picks one of them privately with probability $\frac{1}{4}$ at the beginning of the computation.

Each decidable language is in IP(2pca) – 2

- If the prover is honest, the verifier accepts the input exactly.
- If the prover is cheating, the verifier rejects the input with a probability at least $\frac{1}{4}$.
  - The prover can mislead the verifier in some branches, and so the verifier enters an infinite loop.

- We modify the protocol as follows:
  - After each step, the verifier flips a coin:
    - The protocol continues with probability $\frac{1}{2}$, and
    - the current protocol is terminated, and a new protocol is restarted with probability $\frac{1}{2}$.
  - Any infinite loop can be terminated with probability 1, and by using probability amplification techniques, the error bound can be arbitrary small.

Quantum results

- Any decidable language is in IP(2qcca),
  - where the protocol simulates the computation of a TM (decider),
  - and so, the space used on the counter is dramatically less when compared to the previous protocol.

The private part of this protocol is only the operations on the (classical) counter.

- In fact, all the other parts can be public (seen by the prover).

The verifier seems to be in a superposition of different configurations if the random choices are hidden from the prover. On the other hand, quantum machines can really be in a superposition of different configurations.

- By replacing the classical counter with a quantum one, the operation on the counter can be implemented in a public manner.
- Any decidable language is in AM(2qca).

THANK YOU