Regular Sensing

Shaull Almagor\textsuperscript{1}, Denis Kuperberg\textsuperscript{2}, Orna Kupferman\textsuperscript{1}

\textsuperscript{1}Hebrew University of Jerusalem

\textsuperscript{2}Onera/DTIM - IRIT.

Journées FAC
22-04-2015
Toulouse
Deterministic automata scanning the environment and checking a specification.
• **Deterministic** automata scanning the environment and checking a specification.

• **Input:** $S$ set of signals, $\Sigma = 2^S$ alphabet of the automaton.
- **Deterministic** automata scanning the environment and checking a specification.

- **Input:** $S$ set of signals, $\Sigma = 2^S$ alphabet of the automaton.

- **New approach:** Reading signals via sensors costs energy.
- **Deterministic** automata scanning the environment and checking a specification.

- **Input:** $S$ set of signals, $\Sigma = 2^S$ alphabet of the automaton.

- **New approach:** Reading signals via sensors costs energy.

- **Goal:** Minimize the energy consumption in an average run.
Deterministic automaton $\mathcal{A}$ on \{00, 01, 10, 11\}.

$q$ state: $scost(q) =$ number of relevant signals in $q$. 
Deterministic automaton $\mathcal{A}$ on \{00, 01, 10, 11\}.

$q$ state: $scost(q) =$ number of relevant signals in $q$. 
Deterministic automaton $A$ on $\{00, 01, 10, 11\}$.

$q$ state: $\text{scost}(q) =$ number of relevant signals in $q$.

$w$ word: $\text{scost}(w) =$ average cost of states in the run of $A$ on $w$. 

Deterministic automaton $\mathcal{A}$ on \{00, 01, 10, 11\}.

$q$ state: $scost(q) = \text{number of relevant signals in } q$.

$w$ word: $scost(w) = \text{average cost of states in the run of } \mathcal{A} \text{ on } w$.

$$scost(\mathcal{A}) = \lim_{m \to \infty} |\Sigma|^m \sum_{w:|w|=m} scost(w)$$
Remarks on the definition of sensing cost:

- Initial state plays a role but not acceptance condition.
- Works on finite or infinite words.
- Cost is deduced from the transition structure.
- Signals can be weighted with different probabilities or sensing cost.
Remarks on the definition of sensing cost:

- Initial state plays a role but not acceptance condition.
- Works on finite or infinite words.
- Cost is deduced from the transition structure.
- Signals can be weighted with different probabilities or sensing cost.

**Theorem**

*Sensing cost of an automaton is computable in polynomial time.*

By computing the *stationary distribution* of the induced Markov chain.
Stationary distribution: \( \frac{1}{2}, \frac{1}{2} \)

Sensing cost: \( \frac{3}{2} \).
Stationary distribution: $\frac{2}{5}, \frac{3}{5}$

Sensing cost: $\frac{7}{5}$. 
Stationary distribution: $\frac{2}{5}, \frac{3}{5}$

Sensing cost: $\frac{7}{5}$.

Limitation of the probabilistic model: Safety or Reachability automata always have cost 0. Only ergodic components matter in the long run.
Sensing cost as a measure of complexity of regular languages.

\[ scost(L) := \inf \{ scost(A) | L(A) = L \} \]

Can we compute the sensing cost of a language? How hard is it?
Sensing cost as a measure of complexity of regular languages.

\[ scost(L) := \inf\{ scost(A) | L(A) = L \} \].

Can we compute the sensing cost of a language? How hard is it?

**Theorem**

*On finite words, the optimal sensing cost of a language is always reached by its minimal automaton.*

→ Sensing as a complexity measure is not interesting on finite words, coincides with size.
On infinite words: deterministic parity automata.
On infinite words: deterministic \textit{parity} automata.

Computing the minimal number of states is \textbf{NP}-complete [Schewe ’10].
On infinite words: deterministic parity automata.

Computing the minimal number of states is \textbf{NP}-complete \cite{Schewe'10}.

Third complexity measure of \(\omega\)-languages: \textit{parity rank}.
Sensing cost of $\omega$-regular languages

- On infinite words: deterministic parity automata.
- Computing the minimal number of states is $\text{NP}$-complete [Schewe ’10].
- Third complexity measure of $\omega$-languages: parity rank.

**Theorem**

*The sensing cost of an $\omega$-regular language is the one of its residual automaton.*

**Corollary**

*Computing the sensing cost of an $\omega$-regular language is in $\text{PTime}$.***
Remarks on the result:

- Optimal sensing cost might be reached only in the limit, not by a particular automaton.
Remarks on the result:

- Optimal sensing cost might be reached only in the limit, not by a particular automaton.

Remarks on the result:

- Optimal sensing cost might be reached only in the limit, not by a particular automaton.
- Proof uses lemma of [Niwinski, Walukiewicz '98] on the structure of automata of optimal parity index.
- Trade-off between sensing cost and size.
Remarks on the result:

- Optimal sensing cost might be reached only in the limit, not by a particular automaton.
- Proof uses lemma of [Niwinski, Walukiewicz '98] on the structure of automata of optimal parity index.
- Trade-off between sensing cost and size.
- No trade-off between sensing cost and parity rank.
Remarks on the result:

- Optimal sensing cost might be reached only in the limit, not by a particular automaton.
- Trade-off between sensing cost and size.
- No trade-off between sensing cost and parity rank.
- Idea of the proof of general interest: one can “ignore” the input for arbitrary long periods and still recognize the language.
On-going work

- Minimally-sensing transducer for safety specifications (exponential)
- Alternative definitions for
  - Safety languages
  - Transient components

Future work:

- Cost of realizing for parity specifications
- Precise study of the trade-off between different complexity measures