Minimizing the sensing cost in monitoring and synthesis

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

\textsuperscript{1}Hebrew University of Jerusalem

\textsuperscript{2}TUM.

IAS kick-off meeting
24-11-2015
IAS, Garching
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 $\mathcal{A}$ on $\{00, 01, 10, 11\}$.

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

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

![Diagram of deterministic automaton $\mathcal{A}$ on $\{00, 01, 10, 11\}$.

The diagram shows two states, $q_0$ and $q_1$, with transitions labeled by $00, 11, 10, 01$.

- From state $q_0$ to $q_1$ labeled with $00, 11$.
- From state $q_1$ to $q_0$ labeled with $10, 01$.
- From state $q_0$ to $q_0$ labeled with $00, 01$.
- From state $q_1$ to $q_1$ labeled with $10, 11$.](image)
Deterministic automaton $\mathcal{A}$ on \{00, 01, 10, 11\}.

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

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

$$\text{scost}(\mathcal{A}) = \lim_{m \to \infty} |\Sigma|^{-m} \sum_{|w|=m} \text{scost}(w)$$
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.
Computing the 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.
Back to the example

Stationary distribution: $\frac{1}{2}, \frac{1}{2}$

Sensing cost: $\frac{3}{2}$. 
Back to the example

Stationary distribution: $\frac{2}{5}, \frac{3}{5}$

Sensing cost: $\frac{7}{5}$. 
Sensing cost as a measure of complexity of regular languages.

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

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.*
On infinite words: deterministic parity automata.
On infinite words: deterministic parity automata.

Computing the minimal number of states is $\text{NP}$-complete [Schewe ’10].
Sensing cost of $\omega$-regular languages

- On infinite words: deterministic parity automata.
- Computing the minimal number of states is $\mathbf{NP}$-complete [Schewe ’10].
- Third complexity measure of $\omega$-languages: 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 $\text{P}$. 
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.
- 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.
Limitation of the probabilistic model: Safety automata always have cost 0. Only ergodic components matter in the long run.

Solution: Average only on accepted words.
Safety Setting

Limitation of the probabilistic model: Safety automata always have cost 0. Only ergodic components matter in the long run.

Solution: Average only on accepted words.

Two options:
Word average:

\[
\text{wcost}(A) = \lim_{m \to \infty} \frac{1}{|L \cap \Sigma^m|} \sum_{w \in L \cap \Sigma^m} \text{scost}(w)
\]

Letter-by-letter: \( \text{lcost}(A) \) via Markov chain induced by \( A \).
Two variants of safety cost

\[ lcost(\mathcal{A}) = 1, \ wcost(\mathcal{A}) = 2. \]

Remark: \( wcost \) takes into account the transient components: if left self-loop has 2 labels, then \( wcost(\mathcal{A}) = \frac{3}{2} \).
Two variants of safety cost

\[ lcost(A) = 1, \ wcost(A) = 2. \]

**Remark:** \( wcost \) takes into account the transient components: if left self-loop has 2 labels, then \( wcost(A) = 3/2 \).

**Theorem**

\( lcost(A) \) and \( wcost(A) \) are computable in polynomial time. Their minimal is reached on the minimal automaton.

For \( wcost \): generating series, algorithms on algebraic numbers.
Cost of \textit{synthesis} of a I/O specification $L$: Infimum of costs of \textit{transducers} realizing $L$.

\textbf{Computational problem:}

\textbf{Input}: Deterministic automaton for $L \subseteq (I \cup O)\omega$.

\textbf{Output}: Cost of synthesis of $L$.
Cost of synthesis of a I/O specification $L$: Infimum of costs of transducers realizing $L$.

**Computational problem:**
**Input:** Deterministic automaton for $L \subseteq (I \cup O)^\omega$.
**Output:** Cost of synthesis of $L$.

**Theorem**
*For a safety specification, the problem is EXPTIME-complete, and an optimal transducer always exists.*

**Remarks**
- Optimal transducer can be exponential in the input deterministic automaton.
- Membership in EXPTIME by a game argument.
- Hardness by reduction from Tree Automata Intersection.
- For general languages as inputs, decidability open.
Conclusion

Results

- General definition of sensing cost for finite and infinite words.
- Optimal cost computable in $\text{P}$. 
- Refined definitions for safety languages, well-behaved.
- EXPTIME-completeness of optimal safety synthesis.
- Minimally-sensing transducer for safety specifications (exponential)

Future work:

- Decidability of cost of synthesis for parity specifications
- Precise study of the trade-off between different complexity measures
- Refining the model: cost of switching, ...