# Regular Sensing

Shaull Almagor[1], **Denis Kuperberg**[2], Orna Kupferman[1]

[1]Hebrew University of Jerusalem

[2]Onera/DTIM - IRIT.

Journées FAC
22-04-2015
Toulouse

# Framework

- Deterministic automata scanning the environment and checking a specification.

# Framework

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

# Framework

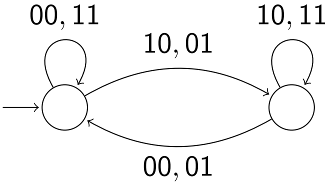
- Deterministic automata scanning the environment and checking a specification.
- **Input:** $S$ set of signals, $\Sigma = 2^S$ alphabet of the automaton.
- **New approach:** Reading signals via sensors costs energy.

# Framework

- 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.

# Sensing cost of a deterministic automaton

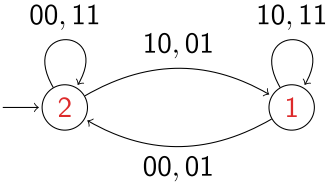
Deterministic automaton $\mathcal{A}$ on $\{00, 01, 10, 11\}$.




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

# Sensing cost of a deterministic automaton

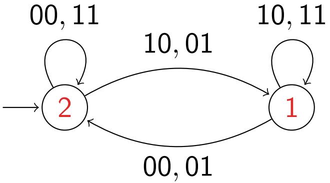
Deterministic automaton $\mathcal{A}$ on $\{00, 01, 10, 11\}$.




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

# Sensing cost of a deterministic automaton

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

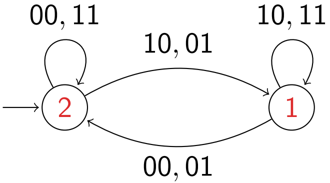



$q$ **state** : $\mathit{scost}(q)$ = number of relevant signals in $q$.

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

# Sensing cost of a deterministic automaton

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




$q$ **state** : $scost(q)$ = number of relevant signals in $q$.

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

$$scost(\mathcal{A}) = \lim_{m\to\infty} |\Sigma|^{-m} \sum_{w:|w|=m} scost(w)$$

# 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.

# Computing the cost

Remarks on the definition of sensing cost:

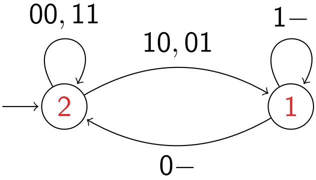
- Initial state plays a role but not acceptance condition.
- Works on finite or infinite words.
- Cost is deduced from the transition structure.
- Signals can be weighted with different probabilities or sensing cost.

**Theorem**

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

By computing the stationary distribution of the induced Markov chain.

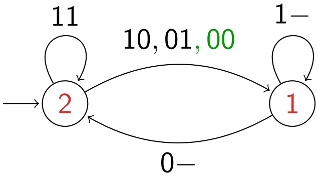
# Back to the example




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

Sensing cost: $\frac{3}{2}$.

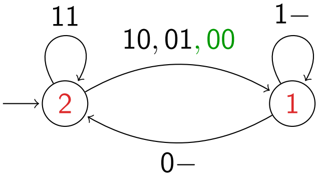
# Back to the example




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

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

# Back to the example




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 of a regular language

Sensing cost as a measure of complexity of regular languages.

$$scost(L) := \inf\{scost(\mathcal{A}) | L(\mathcal{A}) = L\}.$$

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

# Sensing cost of a regular language

Sensing cost as a measure of complexity of regular languages.

$$scost(L) := \inf\{scost(\mathcal{A}) | L(\mathcal{A}) = L\}.$$

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.*

$\rightarrow$ Sensing as a complexity measure is not interesting on finite words, coincides with size.

# Sensing cost of $\omega$-regular languages

- On infinite words: deterministic parity automata.

# Sensing cost of $\omega$-regular languages

- On infinite words: deterministic parity automata.
- Computing the minimal number of states is **NP**-complete [Schewe '10].

# Sensing cost of $\omega$-regular languages

- On infinite words: deterministic parity automata.
- Computing the minimal number of states is **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 **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* ***PTime****.*

# Remarks

Remarks on the result:

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

# Remarks

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.

# Remarks

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

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

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.
- Idea of the proof of general interest: one can "ignore" the input for arbitrary long periods and still recognize the language.

# Conclusion

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