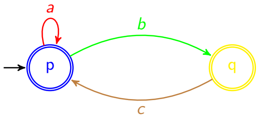
Génération aléatoire uniforme pour les réseaux d'automates

Nicolas Basset
(Travail commun avec Michèle Soria et Jean Mairesse)

Université libre de Bruxelles

Journées Aléa 2017

Motivations (1/2)



Automata are omni-present in computer science.

Given a regular language, it is natural to ask

- what does a typical word of a fixed length n look like ?
- what does an infinite typical word look like ?

The literature provides answer based on

- Uniform sampling (from combinatorics);
- Maximal entropy measure (from information & ergodic theory)

when a deterministic finite state automaton (DFA) recognising the language is provided.

These methods are polynomial in the size of the given DFA.

Motivations (2/2)

Automata in verification of concurrent systems

- Computational systems (software or hardware) are often composed of several components that interact together;
- Networks of automata are an elegant and useful framework to model concurrent systems;
- The associated product automaton $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_K$ is of exponential size $|\mathcal{A}| = |\mathcal{A}_1| \times \cdots \times |\mathcal{A}_K|$.

In this talk we will see how to do

- uniform sampling of words of a given length;
- sampling according to the maximal entropy measure;

for a network of DFAs in a compositional fashion.

A previous work on the subject by [Denise et al., STTT 2012] gives applications to model based testing.

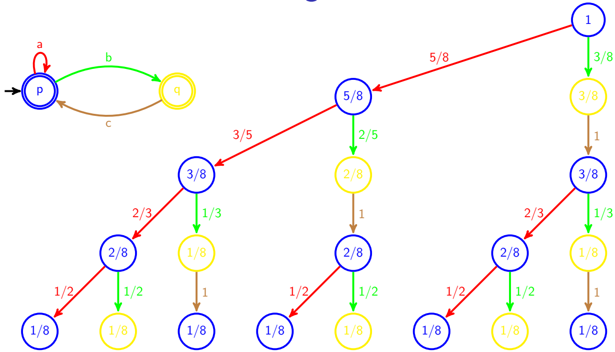
Monolithic methods of sampling for a single DFA (a recap)

Compositional methods of sampling for Network of DFAs

Conclusion and perspective

Uniform sampling of words of an automaton (1/3).

Fixed length. Recursive Method.



Languages $\mathcal{L}_{p,k}$ → Cardinalities $|\mathcal{L}_{p,k}|$ → Probabilities $p_k(p \xrightarrow{b} q) = \frac{|\mathcal{L}_{q,n-k}|}{|\mathcal{L}_{p,n-k+1}|}$

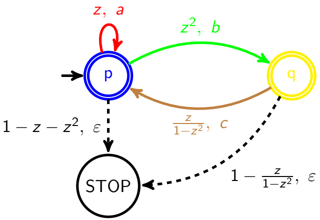
$$\mathcal{L}_{p,k} = a\mathcal{L}_{p,k-1} \cup b\mathcal{L}_{q,k-1}; \quad \mathcal{L}_{q,k} = c\mathcal{L}_{p,k-1}.$$

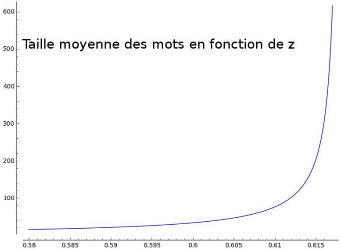
$$|\mathcal{L}_{p,k}| = |\mathcal{L}_{p,k-1}| + |\mathcal{L}_{q,k-1}|; \quad |\mathcal{L}_{q,k}| = |\mathcal{L}_{p,k-1}|.$$

$$\begin{pmatrix} |\mathcal{L}_{p,k}| \\ |\mathcal{L}_{q,k}| \end{pmatrix} = M \begin{pmatrix} |\mathcal{L}_{p,k-1}| \\ |\mathcal{L}_{q,k-1}| \end{pmatrix} = M^k \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ with } M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Uniform sampling of words of an automaton (2/3).

Random length. Boltzmann Sampling [Duchon, Flajolet, Louchard, Schaeffer, ICALP'02].





- Generating function : $L_p(z) = \sum_{w \in \mathcal{L}_p} z^{|w|} = \frac{1}{1-z-z^2}$ with $z < \frac{1}{\phi}$.
- Proba of a word w : $\texttt{Prob}(w) = \frac{z^{|w|}}{L_p(z)}$.

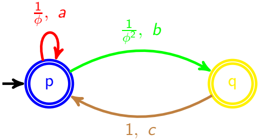
Languages $\mathcal{L}_p \rightarrow$ Generating functions $L_p(z) \rightarrow$ Probabilities $p_z(b) = z\frac{L_q(z)}{L_p(z)}$

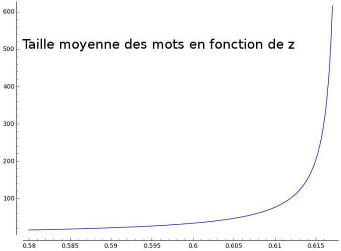
$$\mathcal{L}_p = a\mathcal{L}_p \cup b\mathcal{L}_q \cup \{\varepsilon\}; \quad \mathcal{L}_q = c\mathcal{L}_p \cup \{\varepsilon\}.$$

$$L_p(z) = zL_p(z) + zL_q(z) + 1; \quad L_q(z) = zL_p(z) + 1.$$

$$\mathbf{L}(z) = zM\mathbf{L}(z) + \mathbf{1}_F; \quad \mathbf{L}(z) = (I - zM)^{-1}\mathbf{1}_F.$$

Uniform sampling of words of an automaton (3/3). Infinite length. Parry sampling.





- For a strongly connected automaton.
- Defined by Shannon, known as Parry measure in ergodic theory. Here, we call it Boltzmann critic.

ω-regular Languages $\mathcal{L}_{p,\omega}$ $\rightarrow$ Perron eigenvector $\mathbf{v}$ $\rightarrow$ Probabilities $p_{\frac{1}{\rho}}(b) = \frac{v_q}{\rho v_p}$

$$\mathcal{L}_{p,\omega} = a\mathcal{L}_{p,\omega} \cup b\mathcal{L}_{q,\omega}; \quad \mathcal{L}_{q,\omega} = c\mathcal{L}_{p,\omega}.$$

$$\rho v_p = v_p + v_q; \quad \rho v_q = v_p \text{ avec } \rho \text{ v.p. maximale.}$$

$$\rho \mathbf{v} = M\mathbf{v}.$$

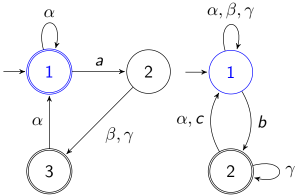
Monolithic methods of sampling for a single DFA (a recap)

Compositional methods of sampling for Network of DFAs

Conclusion and perspective

Network of DFAs

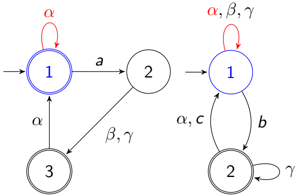
A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha ba\gamma d$

Network of DFAs

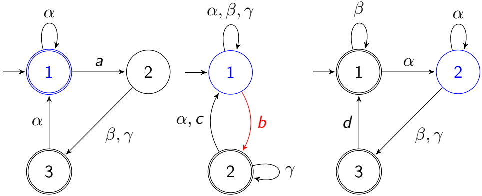
A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha ba\gamma d$

Network of DFAs

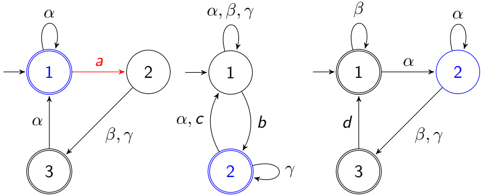
A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha b a \gamma d$

Network of DFAs

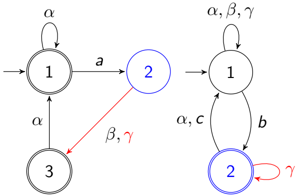
A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha b a \gamma d$

Network of DFAs

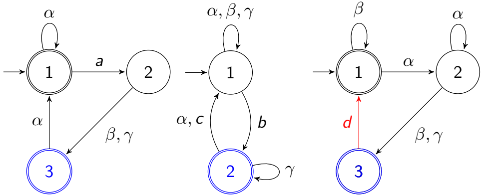
A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha b a \gamma d$

Network of DFAs

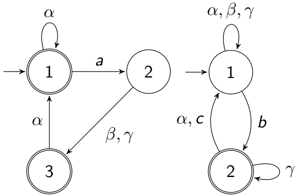
A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha ba\gamma d$

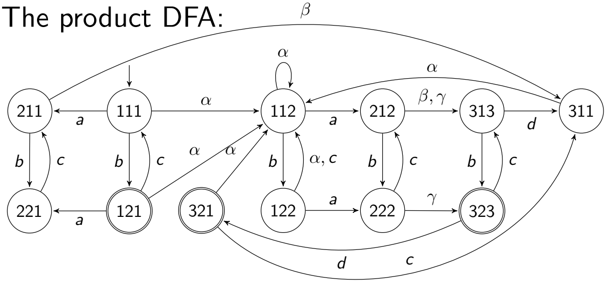
Network of DFAs

A network of three DFAs with shared actions $\{\alpha, \beta, \gamma\}$



Example of words recognised: $\alpha b a \gamma d$

The product DFA:



The easy case: no shared action

Language of the product = shuffle of languages.

$$\mathcal{L}(\mathcal{A}^{(1)} \times \cdots \times \mathcal{A}^{(K)}) = \mathcal{L}(\mathcal{A}^{(1)}) \sqcup\!\sqcup \cdots \sqcup\!\sqcup \mathcal{L}(\mathcal{A}^{(K)})$$

Shuffle of languages

- Shuffle of words $ab \sqcup\!\sqcup cd = \{abcd, acbd, acdb, cabd, cdab\}$
- Shuffle of two languages:

$$\mathcal{L}^{(1)} \sqcup\!\sqcup \mathcal{L}^{(2)} = \bigcup_{(w^{(1)}, w^{(2)}) \in \mathcal{L}^{(1)} \times \mathcal{L}^{(2)}} w^{(1)} \sqcup\!\sqcup w^{(2)}$$

- Naturally extends to K languages.

Computing the cardinalities of shuffle of languages

For the shuffle of two languages

$$|(\mathcal{L} \sqcup\!\sqcup \mathcal{L}')_n| = \sum_{k=0}^{n} \binom{n}{k} |\mathcal{L}_k| \cdot |\mathcal{L}'_{n-k}|. \quad (1)$$

For the shuffle of K languages $\mathcal{L} = \mathcal{L}^{(1)} \sqcup\!\sqcup \cdots \sqcup\!\sqcup \mathcal{L}^{(K)}$

- Do not use

$$|\mathcal{L}_n| = \sum_{n^{(1)}+\cdots+n^{(K)}=n} \binom{n}{n^{(1)}, \ldots, n^{(K)}} |\mathcal{L}^{(1)}_{n^{(1)}}| \cdots |\mathcal{L}^{(K)}_{n^{(K)}}|$$

  There are exponentially many coefficients!

- Instead apply equation (1) $K-1$ times
  $\mathcal{L} = (\ldots((\mathcal{L}^{(1)} \sqcup\!\sqcup \mathcal{L}^{(2)}) \sqcup\!\sqcup \mathcal{L}^{(3)}) \sqcup\!\sqcup \cdots) \sqcup\!\sqcup \mathcal{L}^{(K)}$.

This can be transformed into a recursive method of sampling for $\mathcal{L} = \mathcal{L}^{(1)} \sqcup\!\sqcup \cdots \sqcup\!\sqcup \mathcal{L}^{(K)}$.

Generating functions for shuffle of languages

Exponential generating functions $\hat{L}(z) = \sum_{n\in\mathbb{N}} |\mathcal{L}_n| z^n/n!$

Exponential Boltzmann measure $\hat{\mu}_z(w) = \frac{z^{|w|}}{|w|!\hat{L}(z)}$

- Given $\mathcal{L} = \mathcal{L}^{(1)} \sqcup\!\sqcup \cdots \sqcup\!\sqcup \mathcal{L}^{(K)}$,

$$\hat{L}(z) = \hat{L}^{(1)}(z) \times \cdots \times \hat{L}^{(K)}(z)$$

- $L(z) = \int_0^{+\infty} e^{-u}\hat{L}(zu)du$

Boltzmann sampler of parameter z for $\mathcal{L}$

- Choose u according to weight function: $u \mapsto e^{-u}\hat{L}(zu) = e^{-u}\prod_{i=1}^{K}\hat{L}^{(i)}(zu)$;
- For $i = 1$ to K, let $w^{(i)}$ be chosen using an exponential Boltzmann sampler of parameter zu for $\mathcal{L}^{(i)}$.
- Return a word uniformly at random in $w^{(1)} \sqcup\!\sqcup \cdots \sqcup\!\sqcup w^{(K)}$

Shannon Parry-Markov chain for the shuffle of languages

Recap of the definition

$P(p \xrightarrow{a} q) = v_q/(\rho v_p)$ with $Mv = \rho v$

Lemma

Let $\mathcal{A} = \mathcal{A}^{(1)} \times \cdots \times \mathcal{A}^{(K)}$ be the product of K strongly connected DFAs without synchronisation.
Then $\rho = \sum_{i=1}^{n} \rho^{(i)}$, $v_s = \prod_{i=1}^{K} v^{(i)}_{s^{(i)}}$.

The sampling according to the Shannon-Parry Markov chain

Repeat forever the following:
With probability $\rho^{(i)}/\rho$ make one step $(s^{(i)}, a, t^{(i)})$ of the Shannon-Parry Markov chain number i, write a on the output tape;

Difficulties come from synchronisation

Recap no shared actions=shuffle of languages=everything is easy;

All letters shared

- Language of the product = intersection of languages :

$$\mathcal{L}(\mathcal{A}^{(1)} \times \cdots \times \mathcal{A}^{(K)}) = \mathcal{L}(\mathcal{A}^{(1)}) \cap \cdots \cap \mathcal{L}(\mathcal{A}^{(K)})$$

- $\mathcal{L}(\mathcal{A}^{(1)}) \cap \cdots \cap \mathcal{L}(\mathcal{A}^{(K)}) \stackrel{?}{=} \emptyset$ is a PSPACE-complete problem.

In our framework

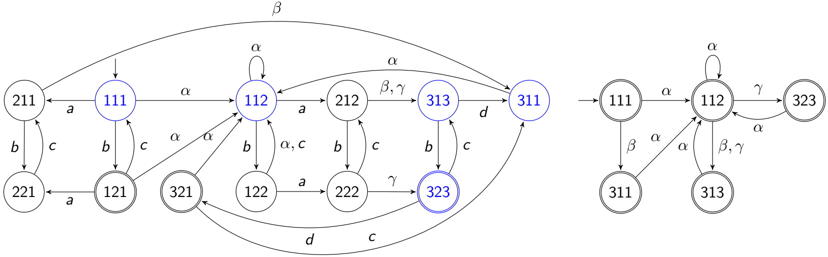
We introduce the **reduced automaton**:

- It keeps only the synchronised part of the product automaton (the true difficulty that needs sequential reasoning).
- The non-synchronised part is projected out (easy to treat by combining independent local works).

The reduced automaton

The *reduced automaton* of a DFA $\mathcal{A} = (Q, \Sigma, \iota, F, \delta)$ is a finite automaton $\mathcal{A}_{\texttt{red}} = (Q_{\texttt{red}}, \Sigma_{\texttt{red}}, \iota_{\texttt{red}}, F_{\texttt{red}}, \Delta_{\texttt{red}})$ such that

- $Q_{\texttt{red}} \subseteq Q$ are states occurring just after a shared action + initial state ι;
- $\Sigma_{\texttt{red}}$ set of shared action;
- $\iota_{\texttt{red}} = \iota$ (same initial state);
- Final states $F_{\texttt{red}}$ irrelevant
- $\Delta_{\texttt{red}} = \{(s, \alpha, t) \mid s \xrightarrow{u\alpha} t \text{ for some } u \in (\Sigma \setminus \Sigma_{\texttt{red}})^*\}$



Do not compute $\mathcal{A}_{\texttt{red}}$ from the product DFA $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^K$ but use $\mathcal{A}_{\texttt{red}} = \mathcal{A}^1_{\texttt{red}} \times \cdots \times \mathcal{A}^K_{\texttt{red}}$.

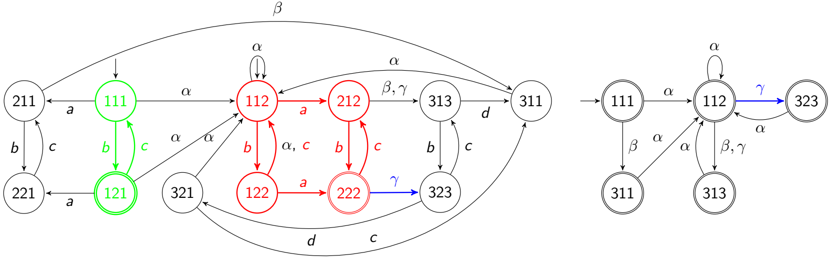
Languages associated to the reduced automaton

Given a DFA $\mathcal{A}$ and its reduced automaton $\mathcal{A}_{\texttt{red}}$.

- $\tilde{\mathcal{L}}_s$: language from state s without shared action.
- $\mathcal{L}_\delta = \{u \in (\Sigma \setminus \Sigma_{\texttt{red}})^* \mid s \xrightarrow{u\alpha} t\}$, for $\delta = (s, \alpha, t) \in \Delta_{\texttt{red}}$

These language are obtained by modifying slightly the automaton.

Example $\tilde{\mathcal{L}}_{111}$ and $\mathcal{L}_{(112,\gamma,323)}$



In fact, compute everything locally and use shuffle of languages:

$\mathcal{L}_{(112,\gamma,323)} = \mathcal{L}^{(1)}_{(1,\gamma,3)} \sqcup\!\sqcup \mathcal{L}^{(2)}_{(1,\gamma,2)} \sqcup\!\sqcup \mathcal{L}^{(3)}_{(3,\gamma,3)} = a \sqcup\!\sqcup (bc)^* b \sqcup\!\sqcup \varepsilon.$

Equations on languages related to the reduced automaton

Theorem: Equations on languages

$$\mathcal{L}_s = \tilde{\mathcal{L}}_s \cup \bigcup_{\delta=(s,\alpha,t)\in\Delta_{\mathtt{red}}} \mathcal{L}_\delta \cdot \alpha \cdot \mathcal{L}_t$$

$$\tilde{\mathcal{L}}_s = \sqcup_{i=1}^{K} \tilde{\mathcal{L}}^{(i)}_{s^{(i)}}; \quad \mathcal{L}_\delta = \sqcup_{i=1}^{K} \mathcal{L}^{(i)}_{\delta^{(i)}}$$

Our generic recipe to randomly generate a word $w \in \mathcal{L}_s$

- Choose whether a synchronisation will occur or not; if not choose $w \in \tilde{\mathcal{L}}_s = \sqcup_{i=1}^{K} \tilde{\mathcal{L}}^{(i)}_{s^{(i)}}$; otherwise
- choose $\delta = (s, \alpha, t) \in \Delta_{\mathtt{red}}$;
- choose $u \in \mathcal{L}_\delta = \sqcup_{i=1}^{K} \mathcal{L}^{(i)}_{\delta^{(i)}}$;
- write $u\alpha$ and repeat from t to generate the rest of the word.

Our generic recipe to randomly generate a word $w \in \mathcal{L}_{s,n}$ (1/3)
Fixed length uniform sampling

1. Choose whether a synchronisation will occur or not;
   - No synchronisation with probability $|\tilde{\mathcal{L}}_{s,n}|/|\mathcal{L}_{s,n}|$.

   if not choose $w \in \tilde{\mathcal{L}}_s = \sqcup_{i=1}^{K} \tilde{\mathcal{L}}_{s^{(i)}}^{(i)}$; otherwise
2. choose $\delta = (s, \alpha, t) \in \Delta_{\mathtt{red}}$;
   - choose the length m with weight

$$\frac{\sum_{\delta=(s,\alpha,t)\in\Delta_{\mathtt{red}}} |\mathcal{L}_{\delta,m-1}|}{\sum_{m=1}^{n} \sum_{\delta=(s,\alpha,t)\in\Delta_{\mathtt{red}}} |\mathcal{L}_{\delta,m-1}|};$$

   - choose $\delta = (s, \alpha, t) \in \Delta_{\mathtt{red}}$ with weight $\frac{|\mathcal{L}_{\delta,m-1}|}{\sum_{\delta'=(s,\alpha',t')} |\mathcal{L}_{\delta',m-1}|}$;
3. choose $u \in \mathcal{L}_{\delta,m-1} = \sqcup_{i=1}^{K} \mathcal{L}_{\delta^{(i)},m-1}^{(i)}$;
4. write $u\alpha$ and repeat from t to generate the rest of the word of length $n - m$.

Our generic recipe to randomly generate a word $w \in \mathcal{L}_s$ (2/3)
Boltzmann sampling

$$\text{Recap: } L_s(z) = \tilde{L}_s(z) + z \sum_{\delta=(s,\alpha,t)\in\Delta_{\mathtt{red}}} L_\delta(z) L_t(z). \qquad (2)$$

1. Choose whether a synchronisation will occur or not;
   - No synchronisation with probability $\tilde{L}_s(z)/L_s(z)$.

   if not choose $w \in \tilde{\mathcal{L}}_s = \sqcup_{i=1}^{K} \tilde{\mathcal{L}}_{s^{(i)}}^{(i)}$ using Boltzmann sampling with parameter z; otherwise
2. choose $\delta = (s, \alpha, t) \in \Delta_{\mathtt{red}}$ with probability

$$\frac{L_\delta(z) L_t(z)}{\sum_{\delta'=(s,\alpha',t')\in\Delta_{\mathtt{red}}} L_{\delta'}(z) L_{t'}(z)}$$

3. choose $u \in \mathcal{L}_\delta = \sqcup_{i=1}^{K} \mathcal{L}_{\delta^{(i)}}^{(i)}$ with probability $z^{|u|}/L_\delta(z)$ using Boltzmann sampling with parameter z;
4. write $u\alpha$ and repeat from t to generate the rest of the word.

Our generic recipe to randomly generate a word $w \in \mathcal{L}_{s,\omega}$ (3/3)
Parry sampling

Assume the product automaton is strongly connected and let $v \geq 0$ and ρ such that $Mv = \rho v$.

1. A synchronisation occurs in the future with probability 1;
2. choose $\delta = (s, \alpha, t) \in \Delta_{\texttt{red}}$ with probability

$$L_\delta(1/\rho)\frac{v_t}{\rho v_s}$$

3. choose $u \in \mathcal{L}_\delta = \sqcup_{i=1}^{K} \mathcal{L}^{(i)}_{\delta^{(i)}}$ with probability

$$\frac{1}{\rho^{|u|} L_\delta(1/\rho)}$$

   using Boltzmann sampling with parameter $1/\rho$;
4. write $u\alpha$ and repeat from t to generate the rest of the word.

Characterisation of the generating functions in the reduced automaton

Recap equations on languages:

$$\mathcal{L}_s = \tilde{\mathcal{L}}_s \cup \bigcup_{\delta=(s,\alpha,t)\in\Delta_{\texttt{red}}} \mathcal{L}_\delta \cdot \alpha \cdot \mathcal{L}_t \qquad (3)$$

Theorem: Equations on generating functions

$$L_s(z) = \tilde{L}_s(z) + z \sum_{\delta=(s,\alpha,t)\in\Delta_{\texttt{red}}} L_\delta(z) L_t(z)$$

In matrix form

Let $\mathfrak{M}(z)$ be the $Q_{red} \times Q_{red}$ matrix defined by

$$\mathfrak{M}_{s,t}(z) = \sum_{\delta=(s,\alpha,t)\in\Delta_{\texttt{red}}} L_\delta(z) \qquad (4)$$

$$\mathbf{L}(z) = \tilde{\mathbf{L}}(z) + z\mathfrak{M}(z)\mathbf{L}(z); \text{ then } \mathbf{L}(z) = (I - z\mathfrak{M}(z))^{-1}\tilde{\mathbf{L}}(z) \qquad (5)$$

Computing cardinalities for all languages

Let n be the length of words to sample.

Languages without synchronisation
$(|\tilde{\mathcal{L}}_{s,m}|)_{m\leq n, s\in Q_{\mathtt{red}}}$ and $(|\mathcal{L}_{\delta,m}|)_{m\leq n,\delta\in\Delta_{\mathtt{red}}}$

See before, shuffle of languages.
Polynomial in n and K.

Languages with synchronisations $(|\mathcal{L}_{s,m}|)_{m\leq n, s\in Q_{\mathtt{red}}}$

- Write $\tilde{\mathbf{L}}_s(z) \mod z^{n+1} = \sum_{m=0}^{n} |\tilde{\mathcal{L}}_{s,m}| z^m$
  and $\mathfrak{M}_{s,t}(z) \mod z^{n+1} = \sum_{m=0}^{n} \sum_{\delta=(s,\alpha,t)\in\Delta_{\mathtt{red}}} |\mathcal{L}_{\delta,m}| z^m$
- Find $\mathbf{L}(z) \mod z^{n+1}$ by taking all operations modulo z^{n+1} in

$$\mathbf{L}(z) = (I - z\mathfrak{M}(z))^{-1}\tilde{\mathbf{L}}(z).$$

Polynomial in n and $|\mathcal{A}_{\mathtt{red}}|$.

A Perron Frobenius Theorem for the reduced automaton

Let $\mathcal{A}$ be a product automaton that is strongly connected and $\mathcal{A}_{\texttt{red}}$ its reduced automaton.

Spectral attributes of the matrix $\mathfrak{M}(z)$

Given $\lambda \in \mathbb{C}$ and $\mathbf{v} \neq \mathbf{0}$. If $\mathfrak{M}(1/\lambda)\mathbf{v} = \lambda\mathbf{v}$ then λ is called a reduced eigenvalue and v a reduced eigenvector.

Theorem

- Existence of ρ and $\mathbf{v}_{\texttt{red}}$:
  - There exists a reduced eigenvalue $\rho > 0$ such that $|\lambda| \leq \rho$ for every reduced eigenvalue λ.
  - There exists a unique $\mathbf{v}_{\texttt{red}} \geq 0$ (up to a multiplicative constant) which is a reduced eigenvector. It satisfies $\mathfrak{M}(1/\rho)\mathbf{v}_{\texttt{red}} = \rho\mathbf{v}_{\texttt{red}}$.
- Link with $\mathcal{A}$ and its adjacency matrix M
  - ρ is the spectral radius of M
  - $\mathbf{v}_{\texttt{red}}$ is the restriction to $Q_{\texttt{red}}$ of the unique eigenvector $\mathbf{v} \geq 0$ (it satisfies $M\mathbf{v} = \rho\mathbf{v}$)

Monolithic methods of sampling for a single DFA (a recap)

Compositional methods of sampling for Network of DFAs

Conclusion and perspective

What we have seen

- A recap in the monolithic case of
  - Uniform sampling
  - Boltzmann sampling
  - Sampling according to Shannon-Parry Markov chain

  and their link to entropy
- **Compositional methods** for these sampling for **network of DFAs** based on the notion of **reduced automata**.

Possible further works

- Precise study of numerical computations (e.g. for finding reduced spectral radius).
- Design of algorithms with better bit complexity.
- Implementations and applications to
  - statistical model checking;
  - model based testing.
- Extension of the theory to weighted automata.
- Extension of the theory to timed automata.