Approximation of the mean speedup in trace monoids and random generation of traces

Alberto Bertoni, Roberto Radicioni

Dip. di Scienze dell'Informazione
Università degli Studi di Milano

Developments and New Tracks in Trace Theory

ESF AutoMathA project 8 - 10, Cremona (Italy)

Brief history

Sequential Processes $\Longrightarrow$ Free Monoids

$\Sigma = \{a, b, c, d, e\}$
$w = acabde \in \Sigma^*$

Concurrent Processes $\Longrightarrow$ Trace Monoids

$(\Sigma, C) =$

[Mazurkiewicz, 77]

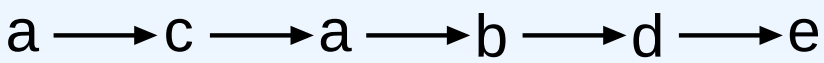
$t = [acabde] \in \Sigma^* / \equiv,$
$\equiv \leftarrow \{xy = yx \mid xCy\}$

Originally, they were introduced in [Cartier/Foata, 69]

Combinatorial Context $\Longrightarrow$ Part. Comm. Monoids

Brief history

Sequential Processes $\Longrightarrow$ Free Monoids



$$\Sigma = \{a, b, c, d, e\}$$
$$w = acabde \in \Sigma^*$$

Concurrent Processes $\Longrightarrow$ Trace Monoids

[Mazurkiewicz, 77]

$(\Sigma, C) =$

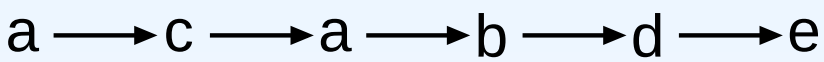
$$t = [acabde] \in \Sigma^* / \equiv,$$
$$\equiv \leftarrow \{xy = yx \mid xCy\}$$

Originally, they were introduced in [Cartier/Foata, 69]

Combinatorial Context $\Longrightarrow$ Part. Comm. Monoids

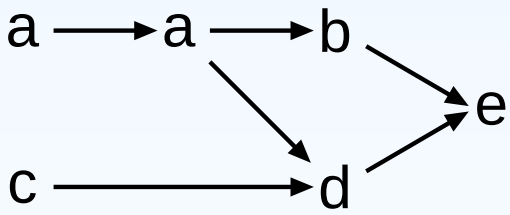
Brief history

Sequential Processes $\Longrightarrow$ Free Monoids



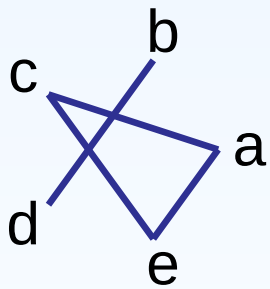
$\Sigma = \{a, b, c, d, e\}$
$w = acabde \in \Sigma^*$

Concurrent Processes $\Longrightarrow$ Trace Monoids



[Mazurkiewicz, 77]

$(\Sigma, C) =$



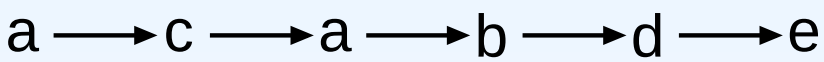
$t = [acabde] \in \Sigma^* / \equiv,$
$\equiv \leftarrow \{xy = yx \mid xCy\}$

Originally, they were introduced in [Cartier/Foata, 69]

Combinatorial Context $\Longrightarrow$ Part. Comm. Monoids

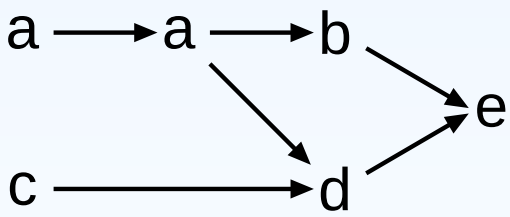
Brief history

Sequential Processes $\Longrightarrow$ Free Monoids



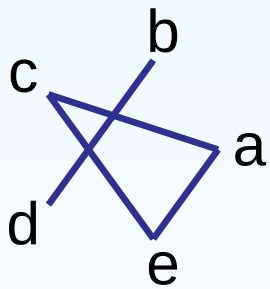
$\Sigma = \{a, b, c, d, e\}$
$w = acabde \in \Sigma^*$

Concurrent Processes $\Longrightarrow$ Trace Monoids



[Mazurkiewicz, 77]

$(\Sigma, C) =$



$t = [acabde] \in \Sigma^* / \equiv,$
$\equiv \leftarrow \{xy = yx \mid xCy\}$

Originally, they were introduced in [Cartier/Foata, 69]

Combinatorial Context $\Longrightarrow$ Part. Comm. Monoids

Traces and Foata Normal Form

A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.

$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

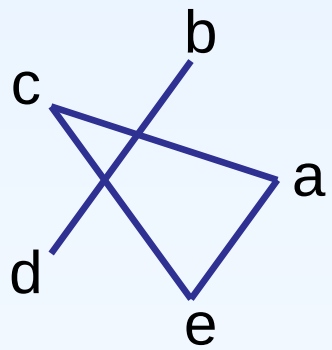
The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.

$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

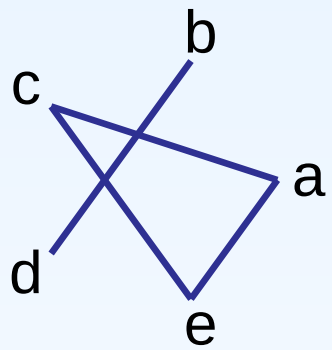
The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.

$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

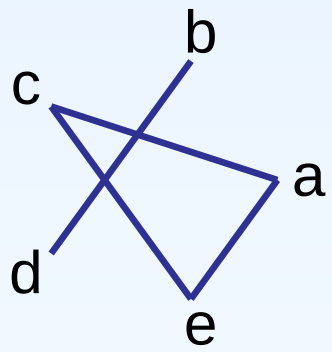
The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.

$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

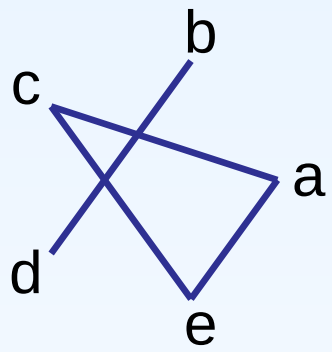
The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.

$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

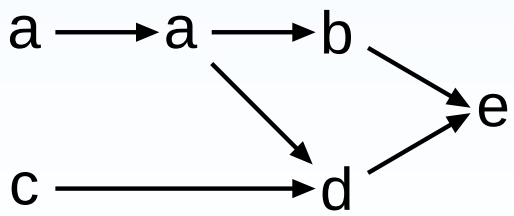
A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

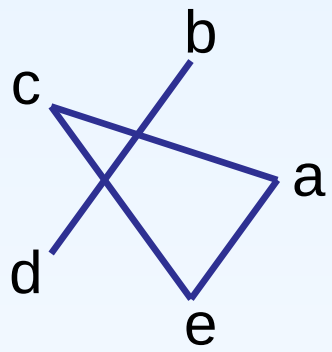
- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.



$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

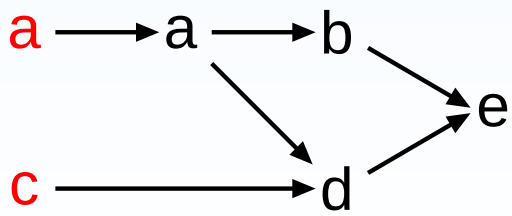
A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

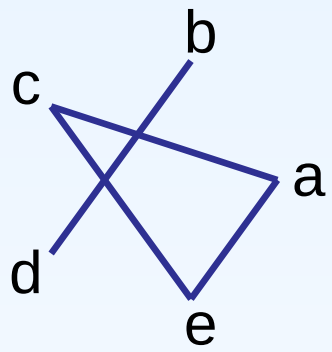
- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.



$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

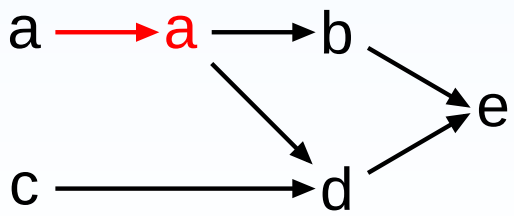
A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

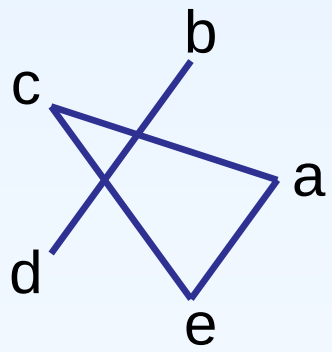
- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.



$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

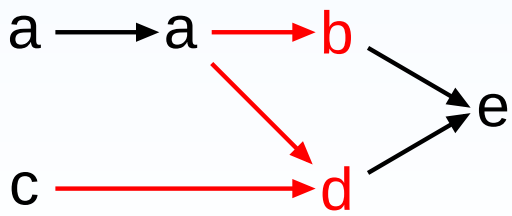
A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

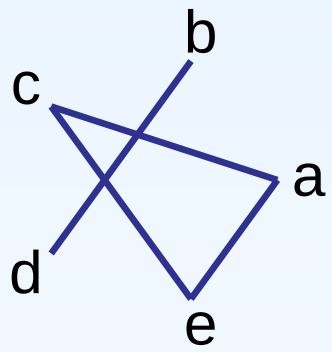
- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.



$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Traces and Foata Normal Form

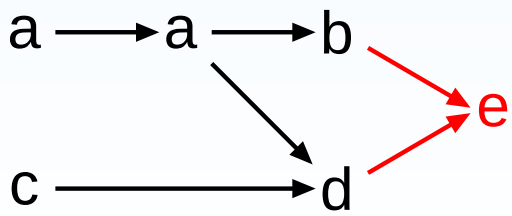
A trace t is an equivalence class in Σ^*, $t = [w]$ where $w \in \Sigma^*$.



$$[acabde] = \{aacbde, aacdbe, acabde, acadbe, caabde, caadbe\}$$

The Foata Normal Form of a trace t is a univocal representation of t by means of cliques in (Σ, C):

- A pair of cliques (c, c') is *admissible* ($\mathrm{adm}(c, c')$) if $\forall \sigma' \in c', \exists \sigma \in c \mid (\sigma, \sigma') \notin C$.
- $t = (c_1, \ldots, c_m)$, where (c_i, c_{i+1}) is adm. for $1 \leq i < m$.



$$t = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$$

Unbounded Parallelism

Assume to have a number $\geq |\Sigma|$ of processors. Let $t \in \Sigma^*/\equiv$:

- The length $l(t)$ is the length of w, where $t = [w]$.
- The height $h(t)$ is the length of its Foata Normal Form.

Look at $t = [acabde] = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$ as a process:

$l(t) =$ Sequential time

$h(t) =$ Parallel time

$$\text{speedup of } t = \frac{l(t)}{h(t)} = \frac{\text{sequential time}}{\text{parallel time}}$$

Unbounded Parallelism

Assume to have a number $\geq |\Sigma|$ of processors. Let $t \in \Sigma^* / \equiv$:

- The length $l(t)$ is the length of w, where $t = [w]$.
- The height $h(t)$ is the length of its Foata Normal Form.

Look at $t = [acabde] = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$ as a process:

$l(t) =$ Sequential time

$h(t) =$ Parallel time

$$\text{speedup of } t = \frac{l(t)}{h(t)} = \frac{\text{sequential time}}{\text{parallel time}}$$

Unbounded Parallelism

Assume to have a number $\geq |\Sigma|$ of processors. Let $t \in \Sigma^*/\equiv$:

- The length $l(t)$ is the length of w, where $t = [w]$.
- The height $h(t)$ is the length of its Foata Normal Form.

Look at $t = [acabde] = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$ as a process:

a ⟶ c ⟶ a ⟶ b ⟶ d ⟶ e $l(t) =$ Sequential time

$h(t) =$ Parallel time

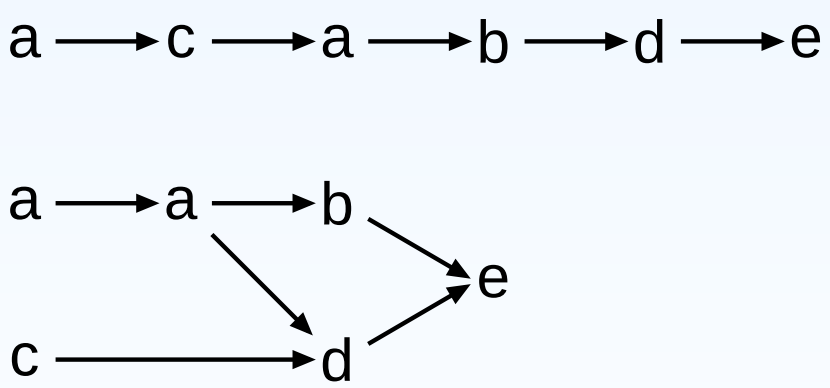
$$\text{speedup of } t = \frac{l(t)}{h(t)} = \frac{\text{sequential time}}{\text{parallel time}}$$

Unbounded Parallelism

Assume to have a number $\geq |\Sigma|$ of processors. Let $t \in \Sigma^* / \equiv$:

- The length $l(t)$ is the length of w, where $t = [w]$.
- The height $h(t)$ is the length of its Foata Normal Form.

Look at $t = [acabde] = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$ as a process:



$l(t) =$ Sequential time

$h(t) =$ Parallel time

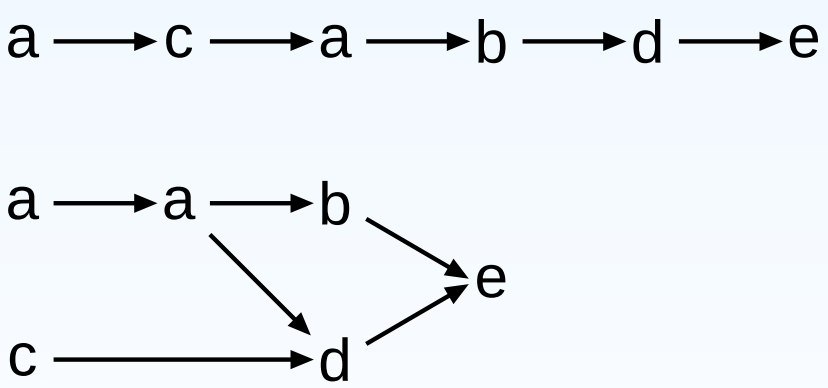
$$\text{speedup of } t = \frac{l(t)}{h(t)} = \frac{\text{sequential time}}{\text{parallel time}}$$

Unbounded Parallelism

Assume to have a number $\geq |\Sigma|$ of processors. Let $t \in \Sigma^* / \equiv$:

- The length $l(t)$ is the length of w, where $t = [w]$.
- The height $h(t)$ is the length of its Foata Normal Form.

Look at $t = [acabde] = (\{a, c\}, \{a\}, \{b, d\}, \{e\})$ as a process:



$l(t) =$ Sequential time

$h(t) =$ Parallel time

$$\text{speedup of } t = \frac{l(t)}{h(t)} = \frac{\text{sequential time}}{\text{parallel time}}$$

Mean (Average) Speedup

Mean Speedup over the words of equal length

Uniform Distribution over Σ^n:

$$\lambda_*(\Sigma, C) = \lim_{n\to\infty} \frac{1}{|\Sigma|^n} \sum_{w\in\Sigma^n} \frac{n}{h([w])}$$

Same process execution time (sequential framework).

Mean Speedup over the traces of equal height

Uniform Distribution over $h^{-1}(n)$:

$$\eta_{\mathcal{M}}(\Sigma, C) = \lim_{n\to\infty} \frac{1}{|h^{-1}(n)|} \sum_{t\in h^{-1}(n)} \frac{l(t)}{n}$$

Same process execution time (parallel framework).

Mean (Average) Speedup

Mean Speedup over the words of equal length

Uniform Distribution over Σ^n:

$$\lambda_*(\Sigma, C) = \lim_{n\to\infty} \frac{1}{|\Sigma|^n} \sum_{w\in\Sigma^n} \frac{n}{h([w])}$$

Same process execution time (sequential framework).

Mean Speedup over the traces of equal height

Uniform Distribution over $h^{-1}(n)$:

$$\eta_{\mathcal{M}}(\Sigma, C) = \lim_{n\to\infty} \frac{1}{|h^{-1}(n)|} \sum_{t\in h^{-1}(n)} \frac{l(t)}{n}$$

Same process execution time (parallel framework).

The MSWL and MSTH Problems

Name: MSWL (Mean Speedup on Words of equal Length)
Input: a parallel alphabet (Σ, C).
Output: $\lambda_*(\Sigma, C)$.

- $\lambda_*(\Sigma, C)$ exists [Sayeb, 89]; and is not algebraic in general [Krob et al., 03];
- MSWL is a restriction of a NP-hard approximable problem (max-algebra spectral radius problem) [Blondel et al., 00];

Name: MSTH (Mean Speedup on Traces of equal Height)
Input: a parallel alphabet (Σ, C).
Output: $\eta_{\mathcal{M}}(\Sigma, C)$.

- $\eta_{\mathcal{M}}(\Sigma, C)$ is algebraic [Krob et al., 03];

The MSWL and MSTH Problems

Name: MSWL (Mean Speedup on Words of equal Length)
Input: a parallel alphabet (Σ, C).
Output: $\lambda_*(\Sigma, C)$.

- $\lambda_*(\Sigma, C)$ exists [Sayeb, 89]; and is not algebraic in general [Krob et al., 03];
- MSWL is a restriction of a NP-hard approximable problem (max-algebra spectral radius problem) [Blondel et al., 00];

Name: MSTH (Mean Speedup on Traces of equal Height)
Input: a parallel alphabet (Σ, C).
Output: $\eta_{\mathcal{M}}(\Sigma, C)$.

- $\eta_{\mathcal{M}}(\Sigma, C)$ is algebraic [Krob et al., 03];

The MSWL and MSTH Problems

Name: MSWL (Mean Speedup on Words of equal Length)
Input: a parallel alphabet (Σ, C).
Output: $\lambda_*(\Sigma, C)$.

- $\lambda_*(\Sigma, C)$ exists [Sayeb, 89]; and is not algebraic in general [Krob et al., 03];
- MSWL is a restriction of a NP-hard approximable problem (max-algebra spectral radius problem) [Blondel et al., 00];

Name: MSTH (Mean Speedup on Traces of equal Height)
Input: a parallel alphabet (Σ, C).
Output: $\eta_{\mathcal{M}}(\Sigma, C)$.

- $\eta_{\mathcal{M}}(\Sigma, C)$ is algebraic [Krob et al., 03];

The MSWL and MSTH Problems

Name: MSWL (Mean Speedup on Words of equal Length)
Input: a parallel alphabet (Σ, C).
Output: $\lambda_*(\Sigma, C)$.

- $\lambda_*(\Sigma, C)$ exists [Sayeb, 89]; and is not algebraic in general [Krob et al., 03];
- MSWL is a restriction of a NP-hard approximable problem (max-algebra spectral radius problem) [Blondel et al., 00];

Name: MSTH (Mean Speedup on Traces of equal Height)
Input: a parallel alphabet (Σ, C).
Output: $\eta_{\mathcal{M}}(\Sigma, C)$.

- $\eta_{\mathcal{M}}(\Sigma, C)$ is algebraic [Krob et al., 03];

The MSWL and MSTH Problems

Name: MSWL (Mean Speedup on Words of equal Length)
Input: a parallel alphabet (Σ, C).
Output: $\lambda_*(\Sigma, C)$.

- $\lambda_*(\Sigma, C)$ exists [Sayeb, 89]; and is not algebraic in general [Krob et al., 03];
- MSWL is a restriction of a NP-hard approximable problem (max-algebra spectral radius problem) [Blondel et al., 00];

Name: MSTH (Mean Speedup on Traces of equal Height)
Input: a parallel alphabet (Σ, C).
Output: $\eta_{\mathcal{M}}(\Sigma, C)$.

- $\eta_{\mathcal{M}}(\Sigma, C)$ is algebraic [Krob et al., 03];

Approximability and non-approximability results

MSWL admits a *probabilistic* fully polynomial time approximation scheme (pFPTAS), i.e.

Theorem ([B./R. 2007])

There exists a randomized algorithm A, taking as input (Σ, C) and an error $0 < \epsilon < 1$, such that

- *A is polynomial in $|\Sigma|$ and $1/\epsilon$.*
- *with probability at least $1/2$,*

$$(1+\epsilon)^{-1} A(\Sigma, C, \epsilon) \leq \lambda_*(\Sigma, C) \leq (1+\epsilon) A(\Sigma, C, \epsilon).$$

MSTH is not efficiently approximable:

Theorem ([B./R. 2007])

Unless NP $=$ coR, MSTH problem is hard to approximate within a factor $|\Sigma|^{1-\epsilon}$.

Approximability and non-approximability results

MSWL admits a *probabilistic* fully polynomial time approximation scheme (pFPTAS), i.e.

Theorem ([B./R. 2007])

There exists a randomized algorithm A, taking as input (Σ, C) and an error $0 < \epsilon < 1$, such that

- *A is polynomial in $|\Sigma|$ and $1/\epsilon$.*
- *with probability at least $1/2$,*

$$(1+\epsilon)^{-1} A(\Sigma, C, \epsilon) \leq \lambda_*(\Sigma, C) \leq (1+\epsilon) A(\Sigma, C, \epsilon).$$

MSTH is not efficiently approximable:

Theorem ([B./R. 2007])

Unless NP $=$ coR, MSTH problem is hard to approximate within a factor $|\Sigma|^{1-\epsilon}$.

Approximability and non-approximability results

MSWL admits a *probabilistic* fully polynomial time approximation scheme (pFPTAS), i.e.

Theorem ([B./R. 2007])

There exists a randomized algorithm A, taking as input (Σ, C) and an error $0 < \epsilon < 1$, such that

- *A is polynomial in $|\Sigma|$ and $1/\epsilon$.*
- *with probability at least $1/2$,*

$$(1+\epsilon)^{-1} A(\Sigma, C, \epsilon) \leq \lambda_*(\Sigma, C) \leq (1+\epsilon) A(\Sigma, C, \epsilon).$$

MSTH is not efficiently approximable:

Theorem ([B./R. 2007])

Unless NP $=$ coR, MSTH problem is hard to approximate within a factor $|\Sigma|^{1-\epsilon}$.

Approximability and non-approximability results

MSWL admits a *probabilistic* fully polynomial time approximation scheme (pFPTAS), i.e.

Theorem ([B./R. 2007])

There exists a randomized algorithm A, taking as input (Σ, C) and an error $0 < \epsilon < 1$, such that

- *A is polynomial in $|\Sigma|$ and $1/\epsilon$.*
- *with probability at least $1/2$,*

$$(1+\epsilon)^{-1} A(\Sigma, C, \epsilon) \leq \lambda_*(\Sigma, C) \leq (1+\epsilon) A(\Sigma, C, \epsilon).$$

MSTH is not efficiently approximable:

Theorem ([B./R. 2007])

Unless NP $=$ coR, MSTH problem is hard to approximate within a factor $|\Sigma|^{1-\epsilon}$.

Approximation and random generation

MSWL is approximable, MSTH is not. Why?

- Approximating MSWL: random generation of

$$t \in \mathcal{M}_G \cap I^{-1}(k) \qquad \text{with} \qquad \text{Prob}(t) = \frac{\left|\{w \in \Sigma^k \mid [w] = t\}\right|}{|\Sigma|^k}.$$

- It is equal to sample uniformly at random words in Σ^k.

Question: What about uniform random generation of traces of given height?

Approximation and random generation

MSWL is approximable, MSTH is not. Why?

- Approximating MSWL: random generation of

$$t \in \mathcal{M}_G \cap l^{-1}(k) \qquad \text{with} \qquad \text{Prob}(t) = \frac{\left|\{w \in \Sigma^k \mid [w] = t\}\right|}{|\Sigma|^k}.$$

- It is equal to sample uniformly at random words in Σ^k.

Question: What about uniform random generation of traces of given height?

Approximation and random generation

MSWL is approximable, MSTH is not. Why?

- Approximating MSWL: random generation of

$$t \in \mathcal{M}_G \cap l^{-1}(k) \qquad \text{with} \qquad \text{Prob}(t) = \frac{\left|\{w \in \Sigma^k \mid [w] = t\}\right|}{|\Sigma|^k}.$$

- It is equal to sample uniformly at random words in Σ^k.

Question: What about uniform random generation of traces of given height?

Random generation of traces

Name: URGT (Uniform Random Generation of Traces)
Input: a graph (Σ, C), an integer in unary repr. 1^k.
Output: $t \in \mathcal{M}_{(\Sigma,C)}$, $h(t) = k$, with probability $1/\left|h^{-1}(k)\right|$.

Our Contribution:

Theorem
Let A be a polynomial time random generator for URGT.

- p_G = *distrib. produced by A over* $\{t \in \mathcal{M}_G \mid h(t) = k\}$;
- u_G = *uniform distribution over* $\{t \in \mathcal{M}_G \mid h(t) = k\}$.

Then, for an infinite set of graphs,

$$\|p_G - u_G\| \geq 2 - o(1).$$

Random generation of traces

Name: URGT (Uniform Random Generation of Traces)
Input: a graph (Σ, C), an integer in unary repr. 1^k.
Output: $t \in \mathcal{M}_{(\Sigma,C)}$, $h(t) = k$, with probability $1/\left|h^{-1}(k)\right|$.

Our Contribution:

Theorem

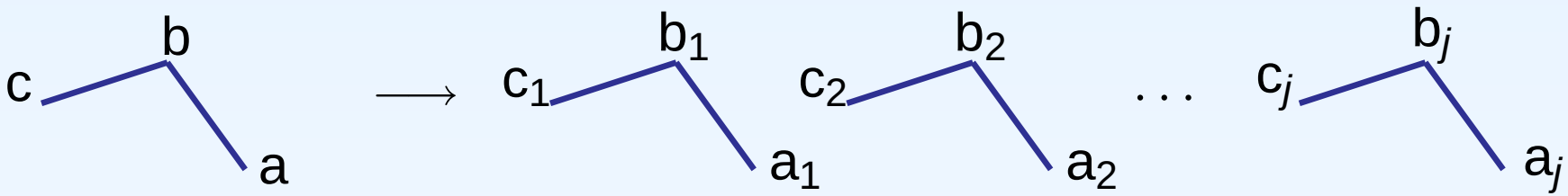
Let A be a polynomial time random generator for URGT.

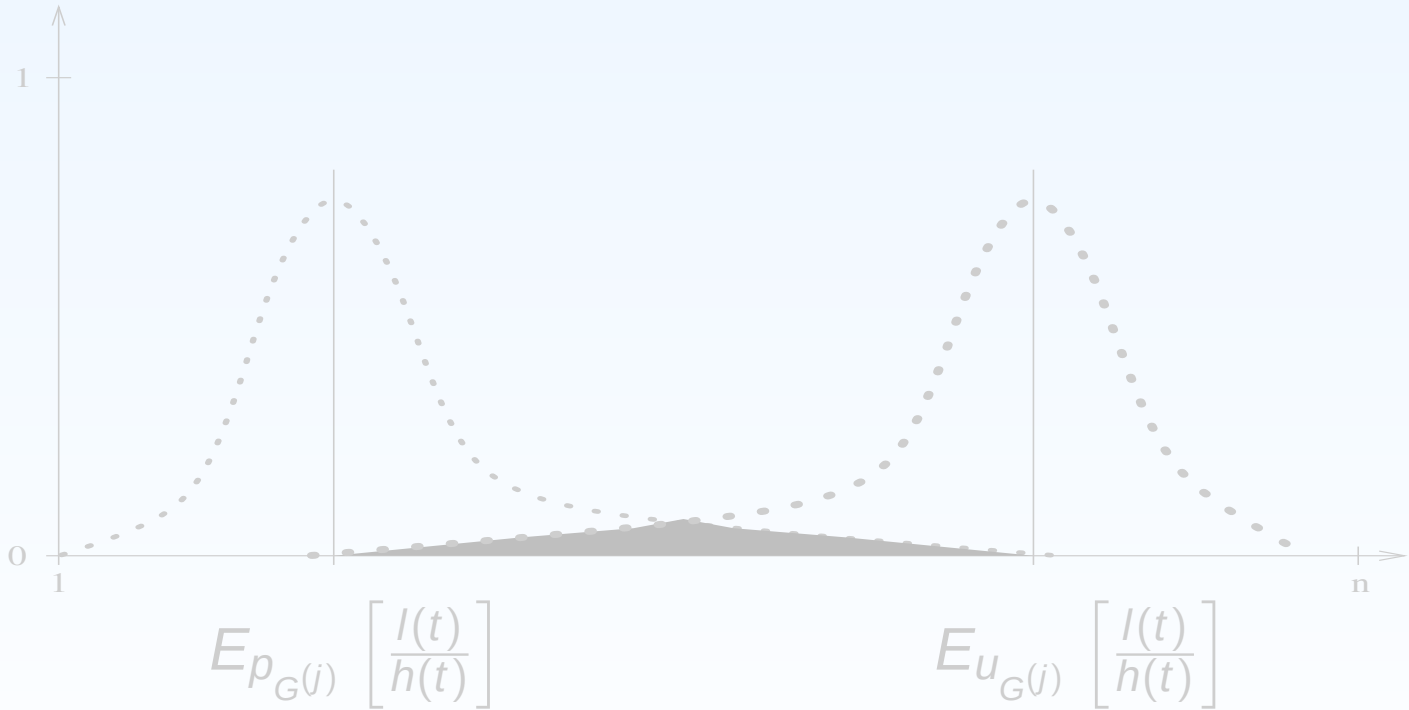
- p_G = *distrib. produced by A over* $\{t \in \mathcal{M}_G \mid h(t) = k\}$;
- u_G = *uniform distribution over* $\{t \in \mathcal{M}_G \mid h(t) = k\}$.

Then, for an infinite set of graphs,

$$\|p_G - u_G\| \geq 2 - o(1).$$

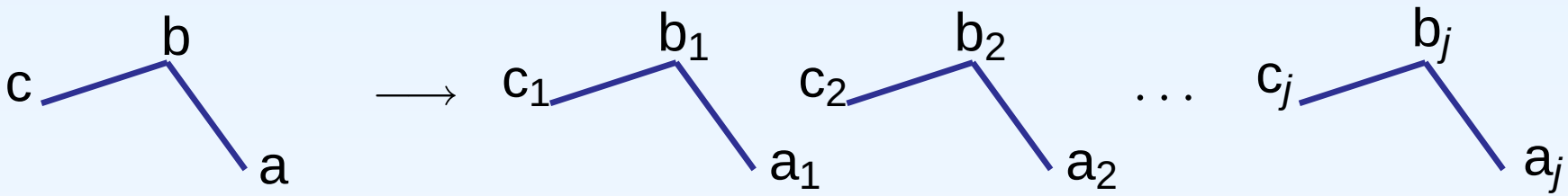
Let $G^{(j)} = (\Sigma^{(j)}, C^{(j)})$ be the disjoint union of j copies of G.

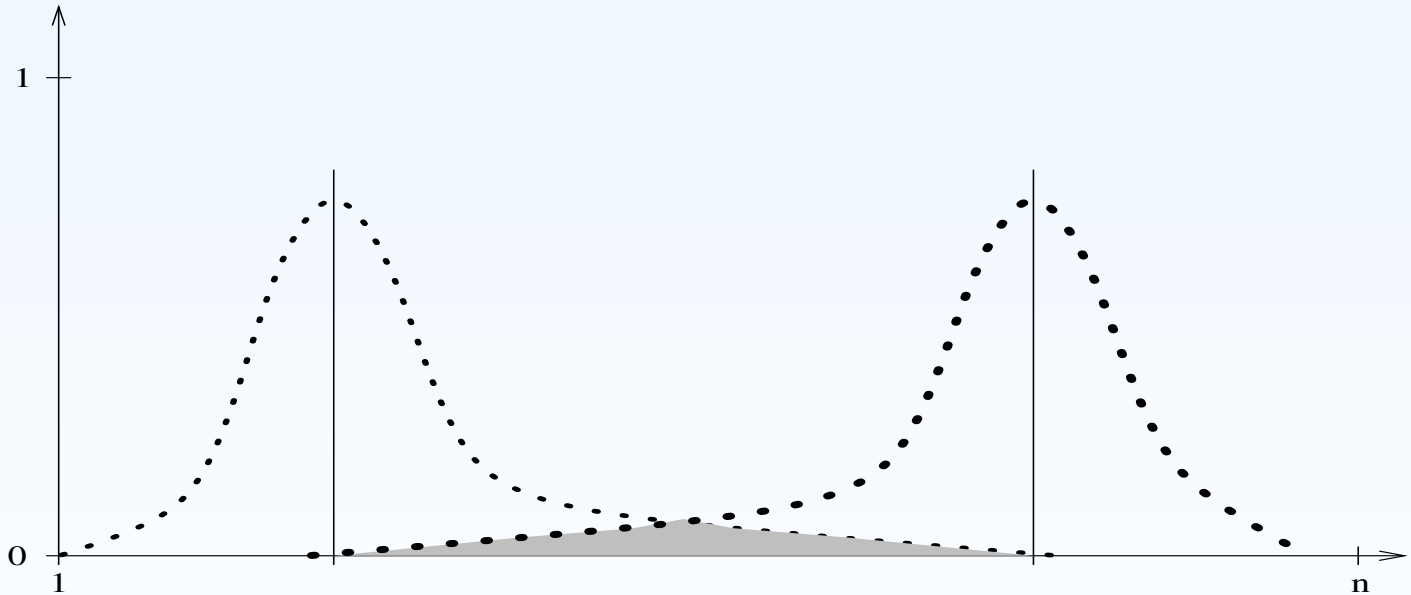




The grey area reduces to 0 for an infinite number of graphs.

Let $G^{(j)} = (\Sigma^{(j)}, C^{(j)})$ be the disjoint union of j copies of G.





$$E_{p_{G^{(j)}}}\left[\frac{l(t)}{h(t)}\right] \qquad E_{u_{G^{(j)}}}\left[\frac{l(t)}{h(t)}\right]$$

The grey area reduces to 0 for an infinite number of graphs.

Series-Parallel Graphs

Question: is there a subclass of graphs for which URGT admits a polynomial solution?

Answer: Yes. Series-parallel graphs.

Consider $G_1 = (\Sigma_1, C_1)$ and $G_2 = (\Sigma_2, C_2)$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$:

- *Series composition* $G_1 \cdot G_2 = (\Sigma_1 \sqcup \Sigma_2, C_1 \sqcup C_2)$.

- *Parallel composition* $G_1 || G_2 = (\Sigma_1 \sqcup \Sigma_2, \overline{\overline{C_1} \sqcup \overline{C_2}})$;

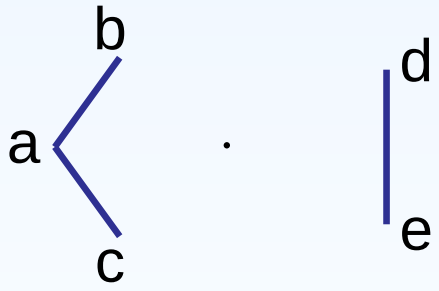
Series-Parallel Graphs

Question: is there a subclass of graphs for which URGT admits a polynomial solution?

Answer: Yes. Series-parallel graphs.

Consider $G_1 = (\Sigma_1, C_1)$ and $G_2 = (\Sigma_2, C_2)$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$:

- *Series composition* $G_1 \cdot G_2 = (\Sigma_1 \sqcup \Sigma_2, C_1 \sqcup C_2)$.



- *Parallel composition* $G_1 || G_2 = (\Sigma_1 \sqcup \Sigma_2, \overline{\overline{C_1} \sqcup \overline{C_2}})$;

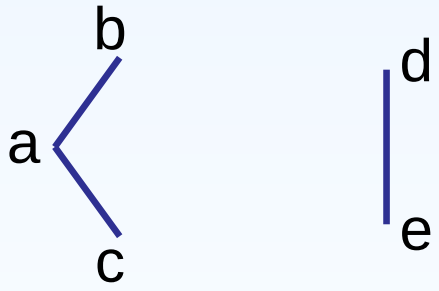
Series-Parallel Graphs

Question: is there a subclass of graphs for which URGT admits a polynomial solution?

Answer: Yes. Series-parallel graphs.

Consider $G_1 = (\Sigma_1, C_1)$ and $G_2 = (\Sigma_2, C_2)$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$:

- *Series composition* $G_1 \cdot G_2 = (\Sigma_1 \sqcup \Sigma_2, C_1 \sqcup C_2)$.



- *Parallel composition* $G_1 || G_2 = (\Sigma_1 \sqcup \Sigma_2, \overline{\overline{C_1} \sqcup \overline{C_2}})$;

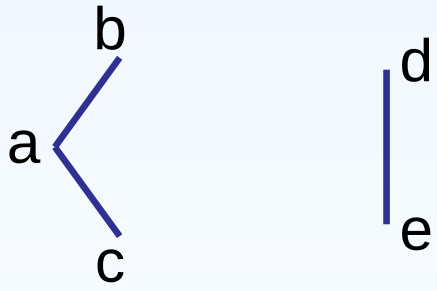
Series-Parallel Graphs

Question: is there a subclass of graphs for which URGT admits a polynomial solution?

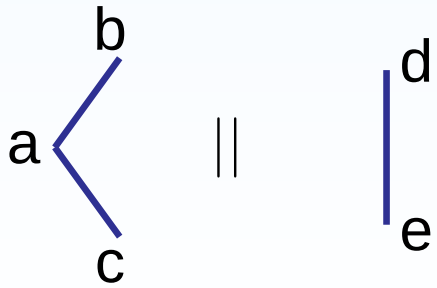
Answer: Yes. Series-parallel graphs.

Consider $G_1 = (\Sigma_1, C_1)$ and $G_2 = (\Sigma_2, C_2)$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$:

- *Series composition* $G_1 \cdot G_2 = (\Sigma_1 \sqcup \Sigma_2, C_1 \sqcup C_2)$.



- *Parallel composition* $G_1 || G_2 = (\Sigma_1 \sqcup \Sigma_2, \overline{\overline{C_1} \sqcup \overline{C_2}})$;



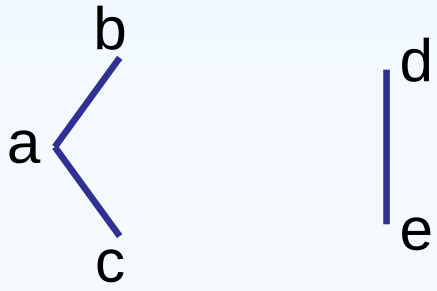
Series-Parallel Graphs

Question: is there a subclass of graphs for which URGT admits a polynomial solution?

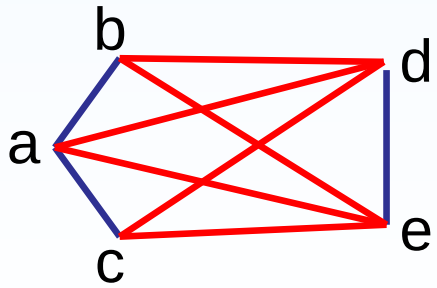
Answer: Yes. Series-parallel graphs.

Consider $G_1 = (\Sigma_1, C_1)$ and $G_2 = (\Sigma_2, C_2)$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$:

- *Series composition* $G_1 \cdot G_2 = (\Sigma_1 \sqcup \Sigma_2, C_1 \sqcup C_2)$.



- *Parallel composition* $G_1 || G_2 = (\Sigma_1 \sqcup \Sigma_2, \overline{\overline{C_1} \sqcup \overline{C_2}})$;



The SPG class

Series-Parallel Graphs (SPG) can be defined recursively:

- $(\{a_1\}, \{\}) \in$ SPG, $\forall i \in \mathbf{N}$;
- $(\Sigma_1, C_1)||(\Sigma_2, C_2)$, $(\Sigma_1, C_1) \cdot (\Sigma_2, C_2) \in$ SGP if $\Sigma_1 \cap \Sigma_2 = \emptyset$.

Properties:

1. $t \in \mathcal{M}_{G_1||G_2}$ implies $t = [w_1 w_2] = [w_2 w_1]$, where

   $[w_1] \in \mathcal{M}_{G_1}$, $[w_2] \in \mathcal{M}_{G_2}$ and $h(t) = \max\{h(t_1), h(t_2)\}$;

2. $t \in \mathcal{M}_{G_1 \cdot G_2}$ implies $t = t_1 t_2 \cdots t_j$ where

   $t_i \neq \epsilon$, $t_i \in \mathcal{M}_{G_1} \Leftrightarrow t_{i+1} \in \mathcal{M}_{G_2}$, $1 \leq i \leq j$ and

   $$h(t) = \sum_{1 \leq i \leq j} h(t_i).$$

The SPG class

Series-Parallel Graphs (SPG) can be defined recursively:

- $(\{a_1\}, \{\}) \in \mathrm{SPG}$, $\forall i \in \mathbf{N}$;
- $(\Sigma_1, C_1)||(\Sigma_2, C_2)$, $(\Sigma_1, C_1) \cdot (\Sigma_2, C_2) \in \mathrm{SGP}$ if $\Sigma_1 \cap \Sigma_2 = \emptyset$.

Properties:

1. $t \in \mathcal{M}_{G_1||G_2}$ implies $t = [w_1 w_2] = [w_2 w_1]$, where

$$[w_1] \in \mathcal{M}_{G_1},\ [w_2] \in \mathcal{M}_{G_2} \quad \text{and} \quad h(t) = \max\{h(t_1), h(t_2)\};$$

2. $t \in \mathcal{M}_{G_1 \cdot G_2}$ implies $t = t_1 t_2 \cdots t_j$ where

$$t_i \neq \epsilon, \qquad t_i \in \mathcal{M}_{G_1} \Leftrightarrow t_{i+1} \in \mathcal{M}_{G_2},\ 1 \leq i \leq j \quad \text{and}$$

$$h(t) = \sum_{1 \leq i \leq j} h(t_i).$$

The SPG class

Series-Parallel Graphs (SPG) can be defined recursively:

- $(\{a_1\}, \{\}) \in \text{SPG}$, $\forall i \in \mathbf{N}$;
- $(\Sigma_1, C_1)||(\Sigma_2, C_2)$, $(\Sigma_1, C_1) \cdot (\Sigma_2, C_2) \in \text{SGP}$ if $\Sigma_1 \cap \Sigma_2 = \emptyset$.

Properties:

1. $t \in \mathcal{M}_{G_1||G_2}$ implies $t = [w_1 w_2] = [w_2 w_1]$, where

$$[w_1] \in \mathcal{M}_{G_1},\ [w_2] \in \mathcal{M}_{G_2} \quad \text{and} \quad h(t) = \max\{h(t_1), h(t_2)\};$$

2. $t \in \mathcal{M}_{G_1 \cdot G_2}$ implies $t = t_1 t_2 \cdots t_j$ where

$$t_i \neq \epsilon, \qquad t_i \in \mathcal{M}_{G_1} \Leftrightarrow t_{i+1} \in \mathcal{M}_{G_2},\ 1 \leq i \leq j \quad \text{and}$$

$$h(t) = \sum_{1 \leq i \leq j} h(t_i).$$

Counting properties of S-P trace monoids

Property 1: $t \in \mathcal{M}_{G_1||G_2}$ implies $t = [w_1 w_2] = [w_2 w_1]$, where

$[w_1] \in \mathcal{M}_{G_1}$, $[w_2] \in \mathcal{M}_{G_2}$ and $h(t) = \max\{h(t_1), h(t_2)\}$;

Lemma 1: Let $H_k(G) = |\{t \in \mathcal{M}_G \mid h(t) = k\}|$. Then:

$$H_k(G_1||G_2) = H_k(G_1)\sum_{j=1}^{k-1} H_j(G_2) + H_k(G_2)\sum_{j=1}^{k-1} H_j(G_1) + H_k(G_1)H_k(G_2).$$

Counting properties of S-P trace monoids

Property 1: $t \in \mathcal{M}_{G_1||G_2}$ implies $t = [w_1 w_2] = [w_2 w_1]$, where

$$[w_1] \in \mathcal{M}_{G_1},\ [w_2] \in \mathcal{M}_{G_2} \quad \text{and} \quad h(t) = \max\{h(t_1), h(t_2)\};$$

Lemma 1: Let $H_k(G) = |\{t \in \mathcal{M}_G \mid h(t) = k\}|$. Then:

$$H_k(G_1||G_2) = H_k(G_1)\sum_{j=1}^{k-1} H_j(G_2) + H_k(G_2)\sum_{j=1}^{k-1} H_j(G_1) + H_k(G_1)H_k(G_2).$$

Counting properties of S-P trace monoids (cont.)

Property 2: $t \in \mathcal{M}_{G_1 \cdot G_2}$ implies $t = t_1 t_2 \cdots t_j$ where

$$t_i \neq \epsilon, \qquad t_i \in \mathcal{M}_{G_1} \Leftrightarrow t_{i+1} \in \mathcal{M}_{G_2},\ 1 \leq i \leq j \quad \text{and}$$

$$h(t) = \sum_{1 \leq i \leq j} h(t_i).$$

Lemma 2: Let

$$\varphi(G_1 \cdot G_2)(x) = \sum_{t \in \mathcal{M}_{G_1 \cdot G_2}} x^{h(t)} \qquad \varphi^{(c)}(G_1 \cdot G_2)(x) = \sum_{\substack{t \in \mathcal{M}_{G_1 \cdot G_2} \\ t = t_1 \cdots t_j, \\ t_1 \in \mathcal{M}_{G_c}}} x^{h(t)}$$

with $c = 1, 2$. Then:

$$\begin{aligned} \varphi^{(1)}(G_1 \cdot G_2) &= \varphi(G_1)[\varphi(G_2)\varphi(G_1)]^*(\varphi(G_2) + 1); \\ \varphi^{(2)}(G_1 \cdot G_2) &= \varphi(G_2)[\varphi(G_1)\varphi(G_2)]^*(\varphi(G_1) + 1); \\ \varphi(G_1 \cdot G_2) &= \varphi^{(1)}(G_1 \cdot G_2) + \varphi^{(2)}(G_1 \cdot G_2). \end{aligned}$$

Counting properties of S-P trace monoids (cont.)

Property 2: $t \in \mathcal{M}_{G_1 \cdot G_2}$ implies $t = t_1 t_2 \cdots t_j$ where

$$t_i \neq \epsilon, \qquad t_i \in \mathcal{M}_{G_1} \Leftrightarrow t_{i+1} \in \mathcal{M}_{G_2},\ 1 \leq i \leq j \quad \text{and}$$

$$h(t) = \sum_{1 \leq i \leq j} h(t_i).$$

Lemma 2: Let

$$\varphi(G_1 \cdot G_2)(x) = \sum_{t \in \mathcal{M}_{G_1 \cdot G_2}} x^{h(t)} \qquad \varphi^{(c)}(G_1 \cdot G_2)(x) = \sum_{\substack{t \in \mathcal{M}_{G_1 \cdot G_2} \\ t = t_1 \cdots t_j, \\ t_1 \in \mathcal{M}_{G_c}}} x^{h(t)}$$

with $c = 1, 2$. Then:

$$\begin{aligned} \varphi^{(1)}(G_1 \cdot G_2) &= \varphi(G_1)[\varphi(G_2)\varphi(G_1)]^*(\varphi(G_2) + 1); \\ \varphi^{(2)}(G_1 \cdot G_2) &= \varphi(G_2)[\varphi(G_1)\varphi(G_2)]^*(\varphi(G_1) + 1); \\ \varphi(G_1 \cdot G_2) &= \varphi^{(1)}(G_1 \cdot G_2) + \varphi^{(2)}(G_1 \cdot G_2). \end{aligned}$$

Random Generation of traces for *SPG*

Recursive Method [Wilf, Nijenhuis 78] by using Lemmas 1 & 2.

Uniform random generation of $t \in \mathcal{M}_G$ of height k:

$G = G_1 || G_2$

If $G = (\{a_i\}, \{\})$, then $t = a_i^k$.

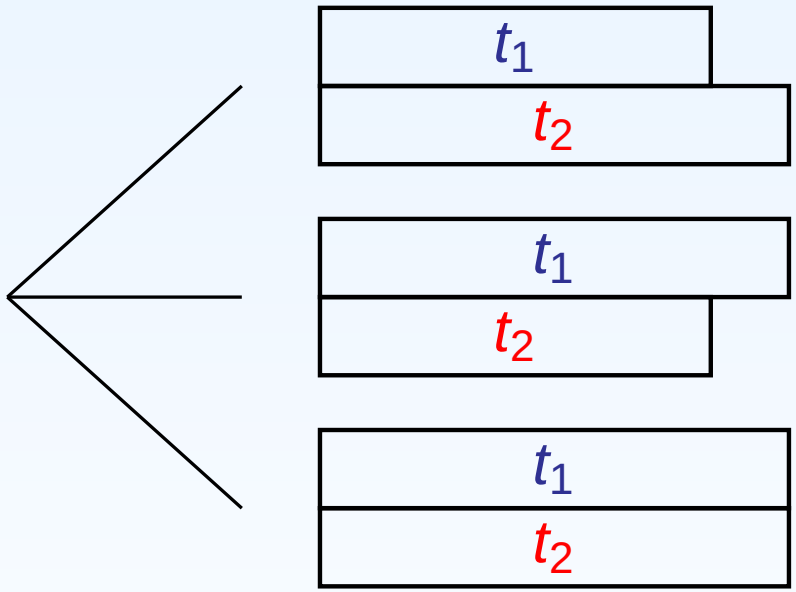
$G = G_1 \cdot G_2$

Random Generation of traces for *SPG*

Recursive Method [Wilf, Nijenhuis 78] by using Lemmas 1 & 2.

Uniform random generation of $t \in \mathcal{M}_G$ of height k:

$G = G_1 || G_2$



If $G = (\{a_i\}, \{\})$, then $t = a_i^k$.

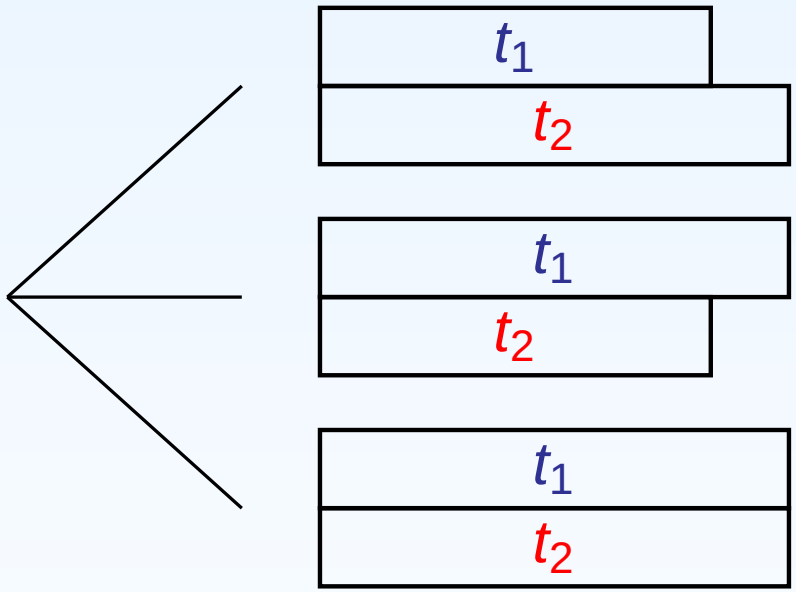
$G = G_1 \cdot G_2$

Random Generation of traces for *SPG*

Recursive Method [Wilf, Nijenhuis 78] by using Lemmas 1 & 2.

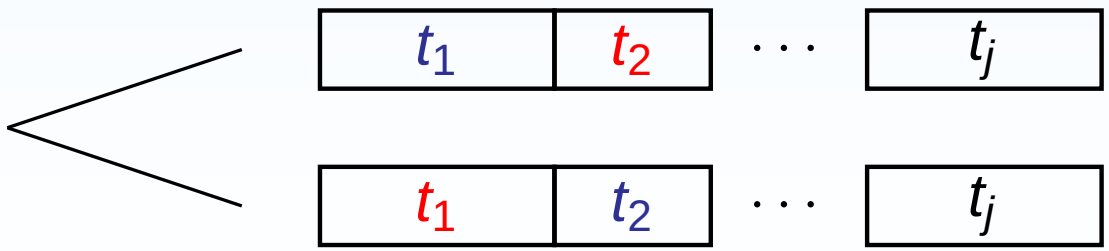
Uniform random generation of $t \in \mathcal{M}_G$ of height k:

$G = G_1 || G_2$



If $G = (\{a_i\}, \{\})$, then $t = a_i^k$.

$G = G_1 \cdot G_2$

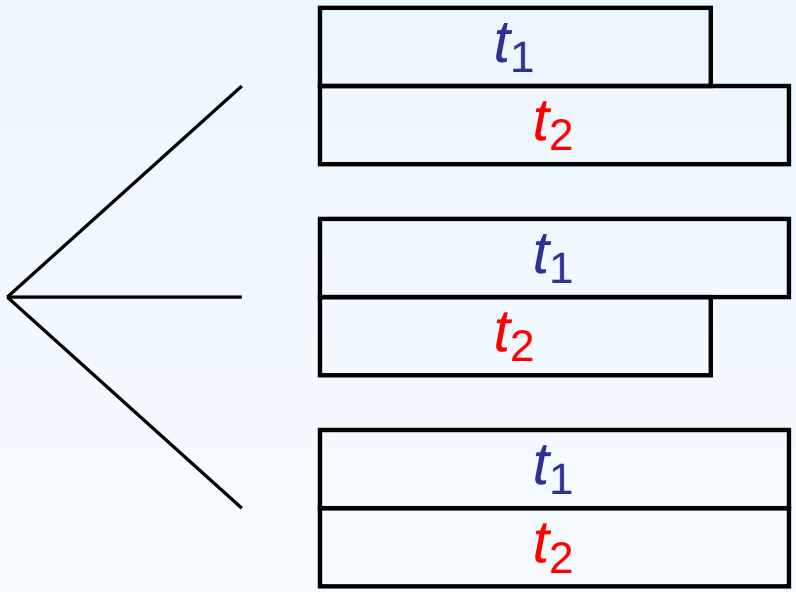


Random Generation of traces for *SPG*

Recursive Method [Wilf, Nijenhuis 78] by using Lemmas 1 & 2.

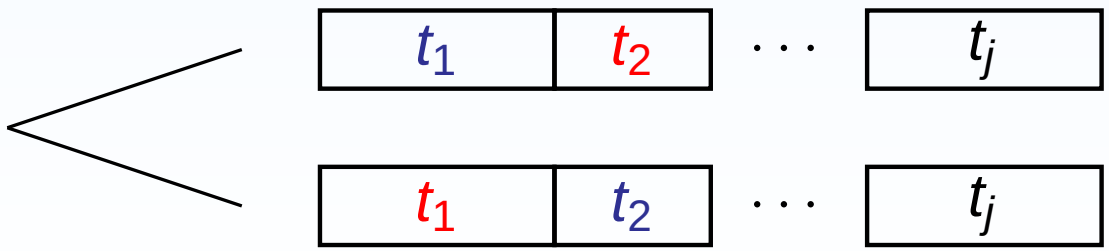
Uniform random generation of $t \in \mathcal{M}_G$ of height k:

$G = G_1 || G_2$



If $G = (\{a_i\}, \{\})$, then $t = a_i^k$.

$G = G_1 \cdot G_2$



Thank you