Heuristic universality detection over regular expressions specified by systems

Carine Pivoteau, Florent Koechlin and Pablo Rotondo

CNRS, LIPN, Université Sorbonne Paris Nord

August 20th

Motivation

Regular expressions are very popular in computer science

- pattern searches, `grep`, etc.
- several methods to convert them into automata
- need for quick and efficient algorithms

Motivation

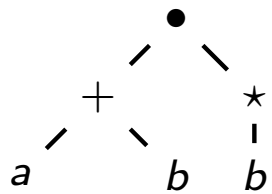
Regular expressions are very popular in computer science

- pattern searches, `grep`, etc.
- several methods to convert them into automata
- need for quick and efficient algorithms

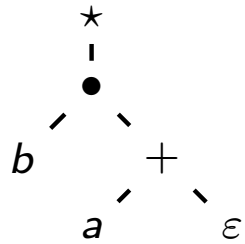
Random regular expressions are a natural way to test algorithms

- theoretically (average case complexity)
- or to produce random benchmarks

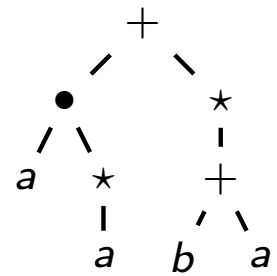
Uniform random regular expressions



$(a + b) \cdot b^\star$



$(b \cdot (a + \varepsilon))^\star$



$(a \cdot a^\star) + (b + a)^\star$

Expression trees $R = a + b + \varepsilon + \overset{\star}{|}_{R} + \overset{\bullet}{/\backslash}_{R\ R} + \overset{+}{/\backslash}_{R\ R}$

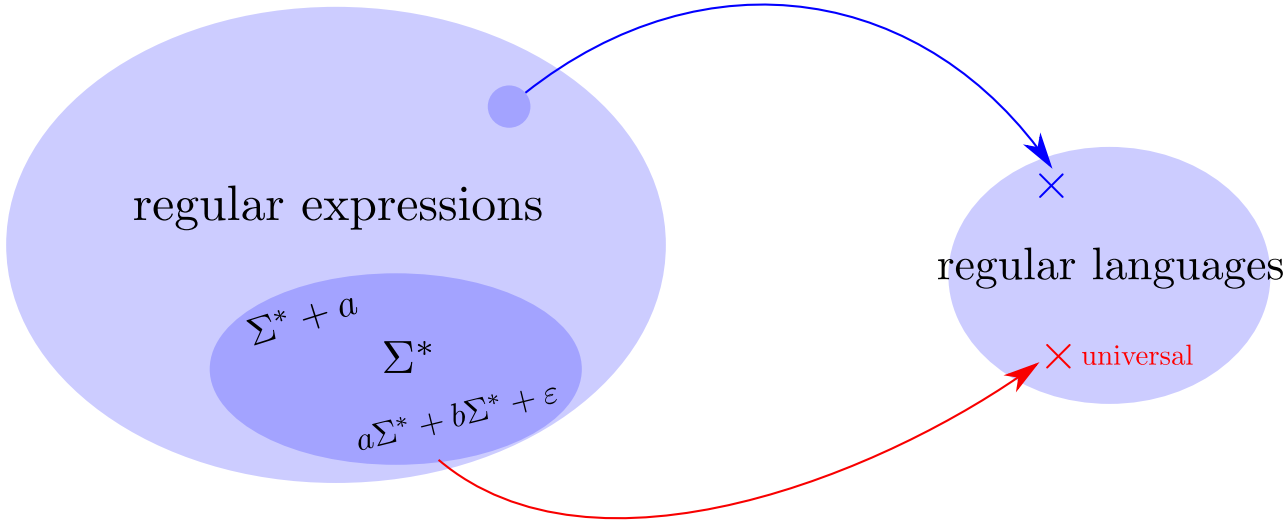
Size: $|T|$ = number of nodes

Uniform distribution: Fix a size n, generate with same probability any expression tree of size n.
$\rightarrow$ natural a priori distribution for analysis
$\rightarrow$ efficient algorithms for random sampling

Problem: redundancies

Redundancies: $T^* \equiv (T^*)^*$, $(a+b)^* + b^*.a \equiv (a+b)^*, \ldots$

Problem: Many algorithms dealing with regular expressions are more interested in the language described



Good distribution over expressions may give bad distribution over languages, that is irrelevant for analysis for such algorithms

Expressions specified by systems

All syntactic regular expression trees:

$$R = a + b + \varepsilon + \overset{\star}{|}_{R} + \overset{\bullet}{/\backslash}_{R\ R} + \overset{+}{/\backslash}_{R\ R}$$

Expressions specified by systems

All syntactic regular expression trees:

$$R = a + b + \varepsilon + \overset{\star}{\underset{R}{|}} + \overset{\bullet}{\underset{R\ R}{\wedge}} + \overset{+}{\underset{R\ R}{\wedge}}$$

Avoiding double stars $(T^*)^* \equiv T^*$:

$$S = R + \overset{\star}{\underset{R}{|}}, \qquad R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}}.$$

Expressions specified by systems

All syntactic regular expression trees:

$$R = a + b + \varepsilon + \overset{\star}{\underset{R}{|}} + \overset{\bullet}{\underset{R\ R}{\wedge}} + \overset{+}{\underset{R\ R}{\wedge}}$$

Avoiding double stars $(T^*)^* \equiv T^*$:

$$S = R + \overset{\star}{\underset{R}{|}}, \qquad R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}}.$$

Double stars and associativity of $+$:

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S\ S_p}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \\ S_s = a + b + \varepsilon + \overset{+}{\underset{S\ S_p}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \\ S_p = a + b + \varepsilon + \overset{\bullet}{\underset{S\ S}{\wedge}} + \overset{\star}{\underset{S_s}{|}} \end{cases}$$

Expressions specified by systems

All syntactic regular expression trees:

$$R = a + b + \varepsilon + \overset{\star}{|}_{R} + \overset{\bullet}{/\backslash}_{R\ R} + \overset{+}{/\backslash}_{R\ R}$$

Avoiding double stars $(T^*)^* \equiv T^*$:

$$S = R + \overset{\star}{|}_{R}, \qquad R = a + b + \varepsilon + \overset{+}{/\backslash}_{S\ S} + \overset{\bullet}{/\backslash}_{S\ S}.$$

Double stars and associativity of $+$:

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{|} + \overset{+}{/\backslash} + \overset{\bullet}{/\backslash} \\ S_s = \\ S_p = a + b + \varepsilon + \overset{\bullet}{/\backslash}_{S\ S} + \overset{\star}{|}_{S_s} \end{cases}$$

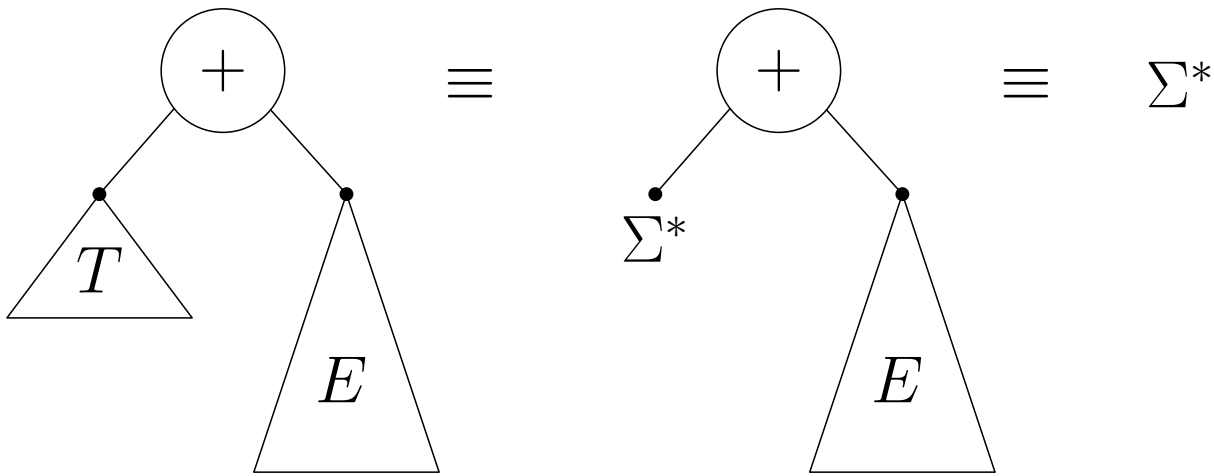
→ Natural way to avoid redundancies
→ Adapted for uniform distribution
→ Amenable to analytic combinatorics

But...

One main issue: absorbing patterns

Worrying result: Under natural conditions, if the system fails to avoid one single universal expression under a union node, then the system is degenerated [K., Nicaud, Rotondo 2020]
$\rightarrow$ after a simplification in linear time, the expected size is $O(1)$

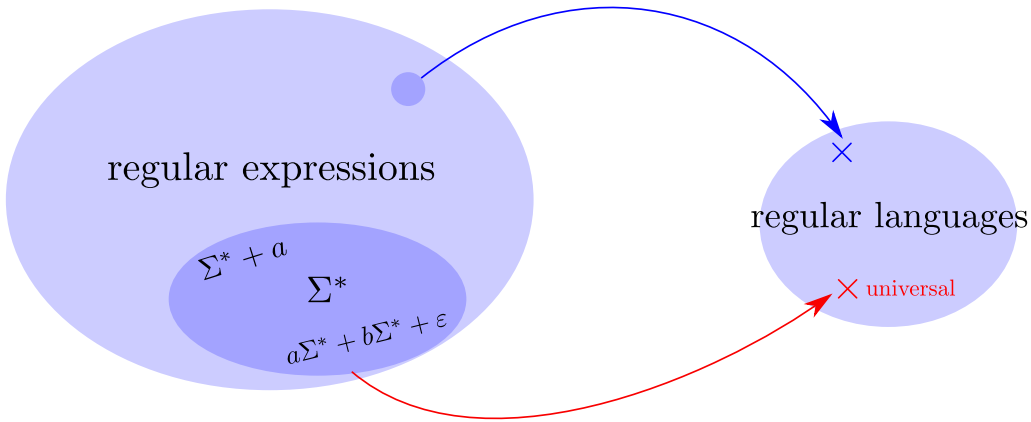
Intuition: expressions equivalent to Σ^* can "semantically" cut arbitrarily long branches



How to check if a system is degenerated?

One main issue: testing universality is PSPACE-complete
$\rightarrow$ no easy way to detect absorbing patterns :(

In practice: heuristically detect an over-abundance of universal expressions described by a system



Simple heuristic detection of universals proposed in [K., Rotondo '21] for all syntactic regular expressions

$$R = a + b + \varepsilon + \overset{\star}{|}_{R} + \overset{\bullet}{/\backslash}_{R\ R} + \overset{+}{/\backslash}_{R\ R}$$

Heuristic detection of universality

Heuristic universal detection: if the language of T contains Σ, then T^* is universal.

Propagation rules: if T is universal, then so are

$$T^*, \qquad T + T_1, \quad \text{and} \quad T \cdot T_1 \quad \text{if } \varepsilon \in \mathcal{L}(T_1)$$

Heuristic detection of universality

Heuristic universal detection: if the language of T contains Σ, then T^* is universal.

Propagation rules: if T is universal, then so are

$$T^*, \qquad T + T_1, \quad \text{and} \quad T \cdot T_1 \quad \text{if } \varepsilon \in \mathcal{L}(T_1)$$

$\rightarrow$ linear time detection, as as detecting if ε or $a \in \Sigma$ is in the language of T is easy (by induction)

$$T = a, \qquad T = \overset{\star}{|}_{T_1}, \qquad T = \overset{+}{\wedge}_{T_1\ T_2}, \qquad T = \overset{\bullet}{\wedge}_{T_1\ T_2}$$

$\rightarrow$ we miss many universals! $\Sigma \cdot \Sigma^* + \varepsilon$.

Heuristic estimation of universals

- $\mathcal{U}$: expressions detected as universal by our heuristic
- $T_{\Sigma,\varepsilon}$: expressions with ε and Σ in their language
- For any class $\mathcal{Y}$ of regular expression trees, we have:

$$\mathcal{Y} \cap \mathcal{U} \subseteq \text{universals of } \mathcal{Y} \subseteq \mathcal{Y} \cap T_{\Sigma,\varepsilon}$$

Heuristic estimation of universals

- $\mathcal{U}$: expressions detected as universal by our heuristic
- $T_{\Sigma,\varepsilon}$: expressions with ε and Σ in their language
- For any class $\mathcal{Y}$ of regular expression trees, we have:

$$\mathcal{Y} \cap \mathcal{U} \subseteq \text{universals of } \mathcal{Y} \subseteq \mathcal{Y} \cap T_{\Sigma,\varepsilon}$$

[K., Rotondo '21] In the case of all syntactic regular expressions

$$R = a + b + \varepsilon + \overset{\star}{|}_{R} + \overset{\bullet}{/\backslash}_{R\ R} + \overset{+}{/\backslash}_{R\ R}$$

then for n large enough, $31\% \leq Pr_n(\mathit{univ.}) \leq 46\%$

Main question

What about regular expressions described by **systems** avoiding redundancies?

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{|}_{S_s} + \overset{+}{/\backslash}_{S\ S_p} + \overset{\bullet}{/\backslash}_{S\ S} \\ S_s = a + b + \varepsilon + \overset{+}{/\backslash}_{S\ S_p} + \overset{\bullet}{/\backslash}_{S\ S} \\ S_p = a + b + \varepsilon + \overset{\bullet}{/\backslash}_{S\ S} + \overset{\star}{|}_{S_s} \end{cases}$$

$\rightarrow$ we expect better results on complex systems avoiding useless redundancies.

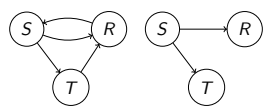
Our contribution: a method to test systems against an overabundance of universal expressions

Hypothesis on systems: D-analysable

Observation: All systems we have encountered were "nice"

D-analysable system: A system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ is D-analyzable if

- The system is not linear 

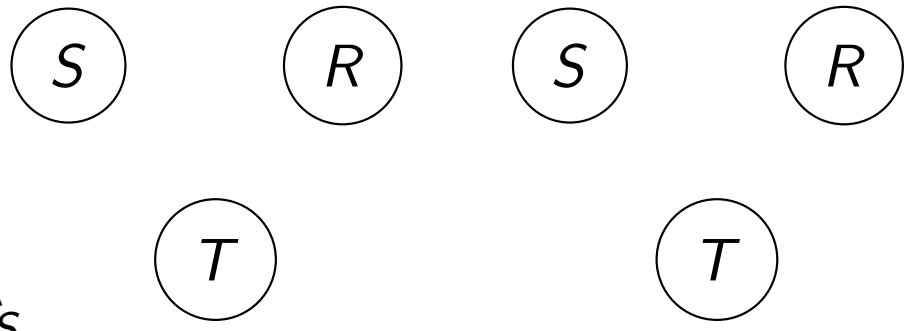
- The system is unambiguous
- The system is aperiodic: for n_0 large enough, for all $n \geq n_0$, every class $\mathcal{Y}_i$ contains an expression of size n.
- $\mathcal{G}(\vec{\Phi})$ is strongly connected, and $\mathcal{G}_{unit}(\vec{\Phi})$ is acyclic. 

Basically they satisfy the hypotheses of **Drmota's theorem**
$\rightarrow$ the combinatorics is well-known

Hypothesis on systems: D-analysable

Dependency digraphs:

$$\begin{cases} S = R + T\,, \\ T = \overset{\star}{\underset{R}{\mid}}\,, \\ R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \end{cases}$$

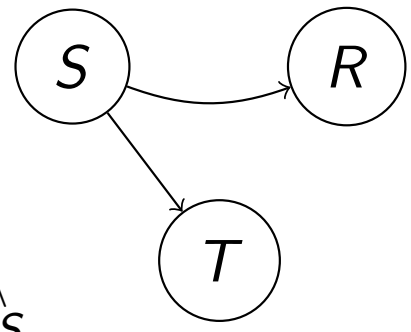


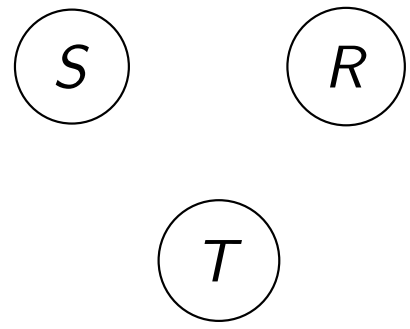
$$\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}}) \qquad \mathcal{G}(\vec{\Phi}) \qquad \mathcal{G}_{unit}(\vec{\Phi})$$

Hypothesis on systems: D-analysable

Dependency digraphs:

$$\begin{cases} S = R + T\,, \\ T = \overset{\star}{\underset{R}{|}}\,, \\ R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \end{cases}$$



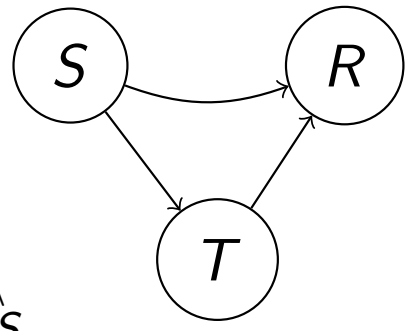


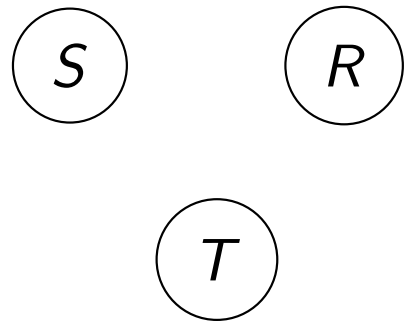
$$\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}}) \qquad \mathcal{G}(\vec{\Phi}) \qquad \mathcal{G}_{unit}(\vec{\Phi})$$

Hypothesis on systems: D-analysable

Dependency digraphs:

$$\begin{cases} S = R + T\,, \\ T = \overset{\star}{\underset{R}{|}}\,, \\ R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \end{cases}$$



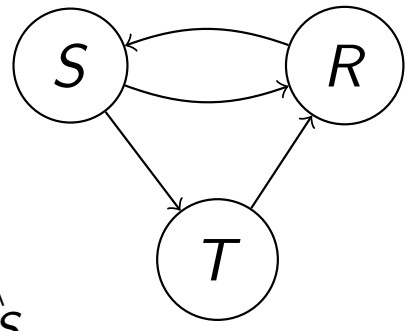


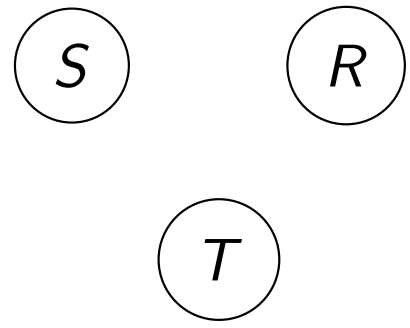
$\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ $\quad \mathcal{G}(\vec{\Phi})$ $\quad \mathcal{G}_{unit}(\vec{\Phi})$

Hypothesis on systems: D-analysable

Dependency digraphs:

$$\begin{cases} S = R + T\,, \\ T = \overset{\star}{\underset{R}{|}}\,, \\ R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{/\backslash}} + \overset{\bullet}{\underset{S\ S}{/\backslash}} \end{cases}$$



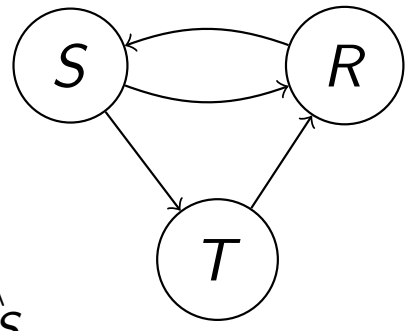


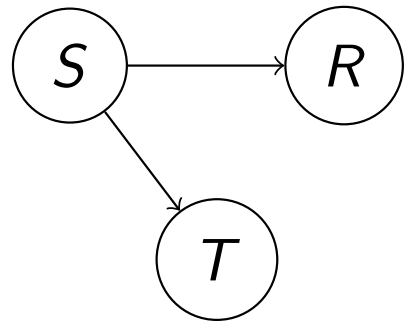
$\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ $\qquad$ $\mathcal{G}(\vec{\Phi})$ $\qquad$ $\mathcal{G}_{unit}(\vec{\Phi})$

Hypothesis on systems: D-analysable

Dependency digraphs:

$$\begin{cases} S = R + T\,, \\ T = \overset{\star}{\underset{R}{\mid}}\,, \\ R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \end{cases}$$





$\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ $\quad$ $\mathcal{G}(\vec{\Phi})$ $\quad$ $\mathcal{G}_{unit}(\vec{\Phi})$

Observation: In most examples from the litterature, $\mathcal{G}_{unit}(\vec{\Phi})$ is acyclic and $\mathcal{G}(\vec{\Phi})$ is strongly connected (or almost)

Generating functions

To each class $\mathcal{Y}$ of regular expressions, associate its generating function

$$y(z) = \sum_{n \geq 0} y_n z^n$$

y_n = number of expressions in $\mathcal{Y}$ of size n

Generating functions

To each class $\mathcal{Y}$ of regular expressions, associate its generating function

$$y(z) = \sum_{n \geq 0} y_n z^n$$

y_n = number of expressions in $\mathcal{Y}$ of size n

$$\begin{cases} S = R + T\,, \\ T = \overset{\star}{\underset{R}{|}}\,, \\ R = a + b + \varepsilon + \overset{+}{\underset{S\ S}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \end{cases} \rightarrow \begin{cases} S(z) = R(z) + T(z)\,, \\ T(z) = \ zR(z)\,, \\ R(z) = 3z + 2zS(z)^2 \end{cases}$$

Analytic combinatorics principle

Asymptotic behaviour of $y(z)$ around its dominant singularities gives asymptotic expansion for y_n

Hypothesis on systems: D-analysable

Drmota's theorem. If $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ is D-analyzable then

- the $y_i(z)$ share a unique common dominant singularity $\rho > 0$
- near $z = \rho$, $y_i(z) = a_i(z) - b_i(z) \cdot \sqrt{1 - z/\rho}$
- $y_{i,n} \sim C_i n^{-3/2} \rho^{-n}$

$\rightarrow$ In all systems we have encountered, all classes of expressions shared a common asymptotic behaviour $\propto n^{-3/2}\rho^{-n}$

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

- For each subset $S \subseteq \Sigma \cup \{\varepsilon\}$, divide $\mathcal{Y}_i$ into the subclasses

$$\mathcal{Y}_i^{\langle S \rangle} = \mathcal{Y}_i \cap \text{described language contains } S \text{ but not } \Sigma \cup \varepsilon \setminus S$$

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

Examples

$$\mathcal{Y}_i^{\langle a \rangle} = \mathcal{Y}_i \cap \text{ language contains } a \text{ but not } b, \text{ nor } \varepsilon$$

$$\mathcal{Y}_i^{\langle a,b,\varepsilon \rangle} = \mathcal{Y}_i \cap \text{ language contains every letter and } \varepsilon$$

$$\mathcal{Y}_i^{\langle \emptyset \rangle} = \mathcal{Y}_i \cap \text{ language contains no letter, nor } \varepsilon$$

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

- For each subset $S \subseteq \Sigma \cup \{\varepsilon\}$, divide $\mathcal{Y}_i$ into the subclasses

  $$\mathcal{Y}_i^{\langle S \rangle} = \mathcal{Y}_i \cap \text{described language contains } S \text{ but not } \Sigma \cup \varepsilon \setminus S$$

- Furthermore divide $\mathcal{Y}_i^{\langle \Sigma, \varepsilon \rangle} = \mathcal{Y}_i^{\langle \mathcal{G} \rangle} \uplus \mathcal{Y}_i^{\langle \mathcal{U} \rangle}$
- Remark: $\mathcal{Y}_i = \mathcal{Y}_i^{\langle \mathcal{G} \rangle} \uplus \mathcal{Y}_i^{\langle \mathcal{U} \rangle} \uplus \biguplus_{S \subsetneq \Sigma \cup \varepsilon} \mathcal{Y}_i^{\langle S \rangle}$

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

- For each subset $S \subseteq \Sigma \cup \{\varepsilon\}$, divide $\mathcal{Y}_i$ into the subclasses

  $\mathcal{Y}_i^{\langle S \rangle} = \mathcal{Y}_i \cap$ described language contains S but not $\Sigma \cup \varepsilon \setminus S$

- Furthermore divide $\mathcal{Y}_i^{\langle \Sigma, \varepsilon \rangle} = \mathcal{Y}_i^{\langle \mathcal{G} \rangle} \uplus \mathcal{Y}_i^{\langle \mathcal{U} \rangle}$
- Remark: $\mathcal{Y}_i = \mathcal{Y}_i^{\langle \mathcal{G} \rangle} \uplus \mathcal{Y}_i^{\langle \mathcal{U} \rangle} \uplus \biguplus_{S \subsetneq \Sigma \cup \varepsilon} \mathcal{Y}_i^{\langle S \rangle}$

Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S \rangle}$.

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

$$R = \ldots + \overset{\bullet}{\underset{S}{/}\!\!\underset{T}{\backslash}} + \ldots$$

$$R^{\langle a,\varepsilon\rangle} = \ldots +$$

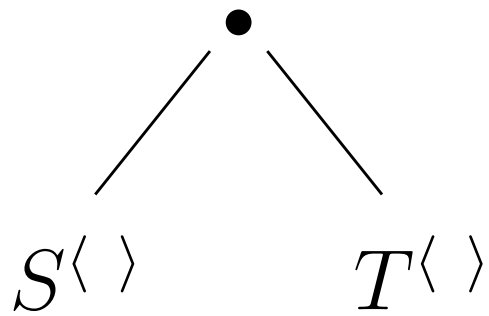
Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S\rangle}$.

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

$$R = \ldots + \underset{S\ T}{\wedge} + \ldots$$



$$R^{\langle a,\varepsilon\rangle} = \ldots +$$

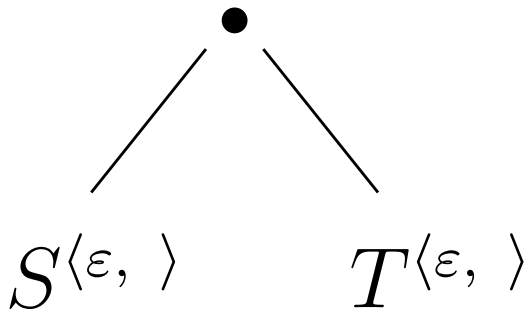
Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S\rangle}$.

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

$$R = \ldots + \underset{S\ T}{\wedge} + \ldots$$



$$R^{\langle a,\varepsilon\rangle} = \ldots +$$

Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S\rangle}$.

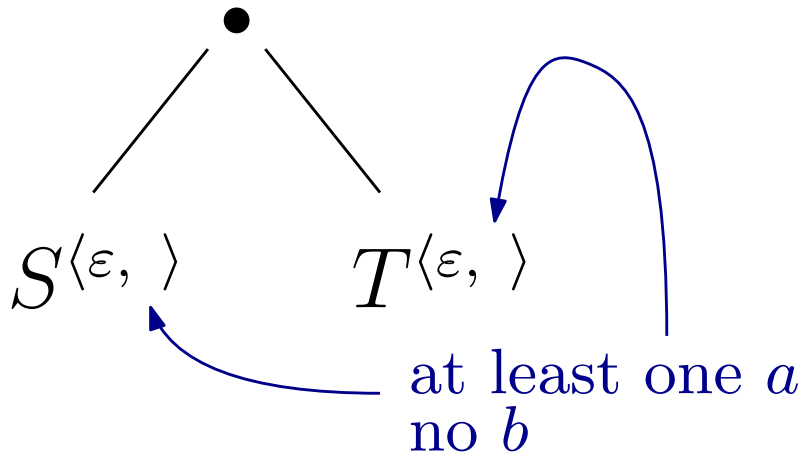
Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

$$R = \ldots + \underset{S\ T}{\wedge} + \ldots$$

$$R^{\langle a,\varepsilon\rangle} = \ldots +$$



Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S\rangle}$.

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

$$R = \ldots + \overset{\bullet}{/\ \backslash}\ {}_{S}\ {}_{T} + \ldots$$

$$R^{\langle a,\varepsilon\rangle} = \ldots + \overset{\bullet}{/\ \backslash}\ {}_{S^{\langle a,\varepsilon\rangle}}\ {}_{T^{\langle a,\varepsilon\rangle}} + \overset{\bullet}{/\ \backslash}\ {}_{S^{\langle a,\varepsilon\rangle}}\ {}_{T^{\langle\varepsilon\rangle}} + \overset{\bullet}{/\ \backslash}\ {}_{S^{\langle\varepsilon\rangle}}\ {}_{T^{\langle a,\varepsilon\rangle}} + \ldots$$

Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S\rangle}$.

Refinement of systems

Input: a nice D-analysable system $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ of regular expressions, where each class is counted by $\propto n^{-3/2}\rho^{-n}$

Output: overabundance of universals?

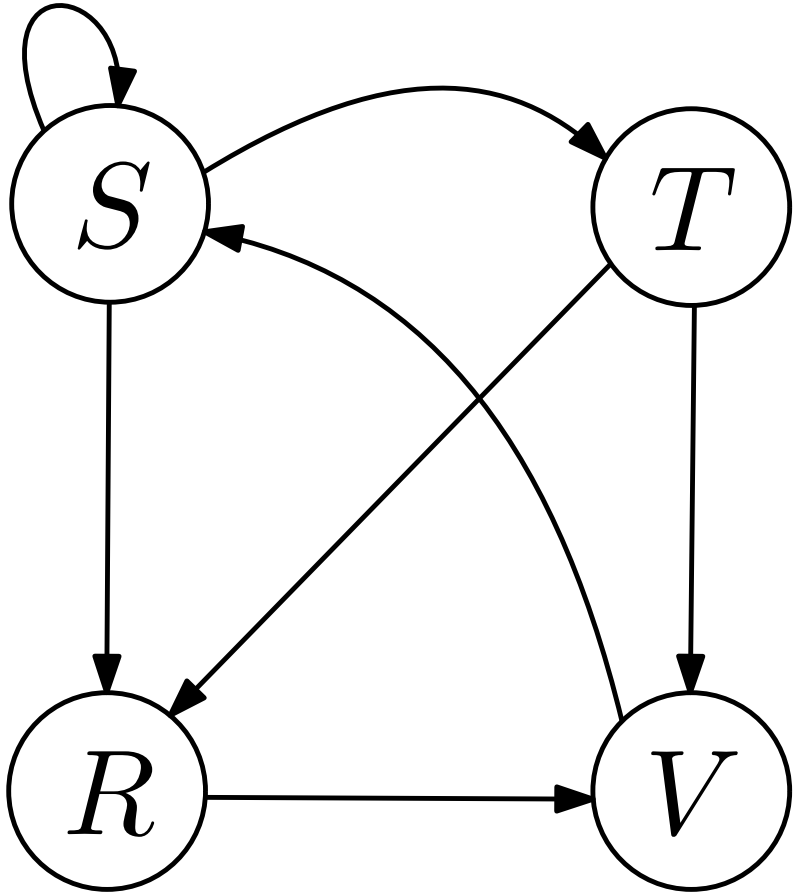
- For each subset $S \subseteq \Sigma \cup \{\varepsilon\}$, divide $\mathcal{Y}_i$ into the subclasses

  $\mathcal{Y}_i^{\langle S \rangle} = \mathcal{Y}_i \cap$ described language contains S but not $\Sigma \cup \varepsilon \setminus S$

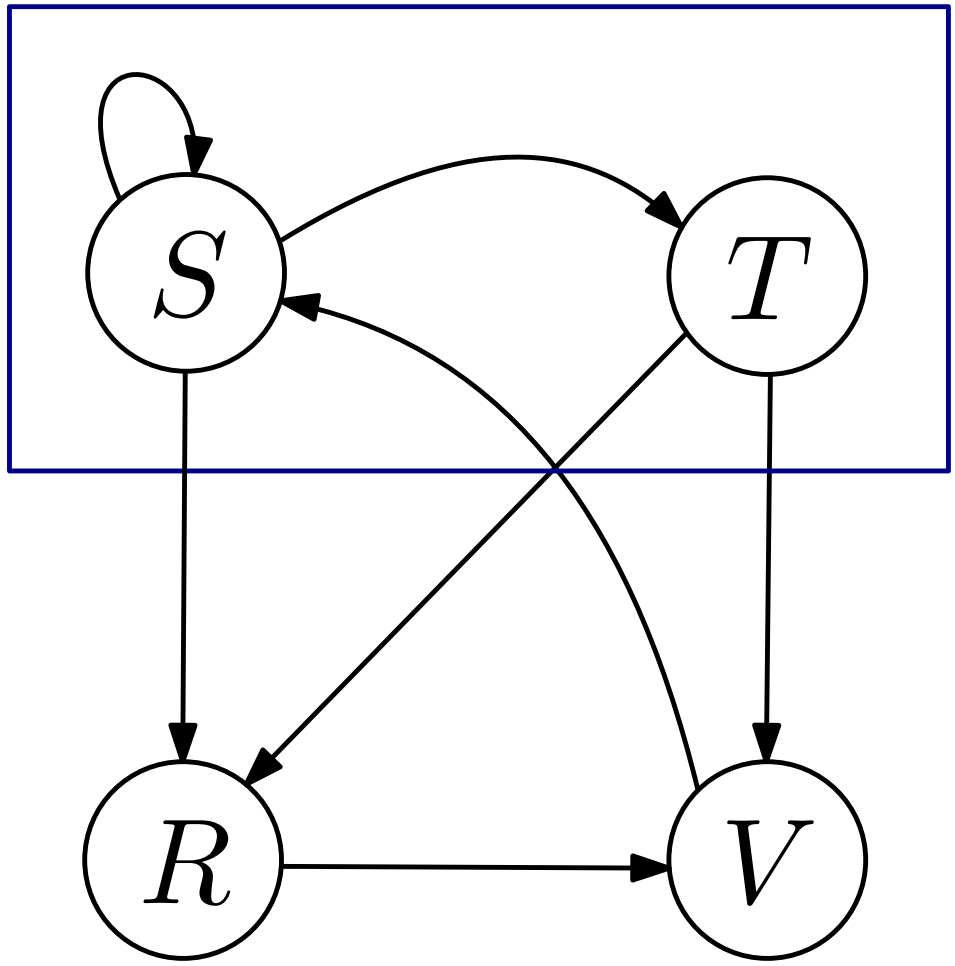
- Furthermore divide $\mathcal{Y}_i^{\langle \Sigma, \varepsilon \rangle} = \mathcal{Y}_i^{\langle \mathcal{G} \rangle} \uplus \mathcal{Y}_i^{\langle \mathcal{U} \rangle}$
- Remark: $\mathcal{Y}_i = \mathcal{Y}_i^{\langle \mathcal{G} \rangle} \uplus \mathcal{Y}_i^{\langle \mathcal{U} \rangle} \uplus \biguplus_{S \subsetneq \Sigma \cup \varepsilon} \mathcal{Y}_i^{\langle S \rangle}$

Theorem: We can convert $\vec{\mathcal{Y}} = \vec{\Phi}(\vec{\mathcal{Y}})$ into a new extended system for the classes $\mathcal{Y}_i^{\langle S \rangle}$.

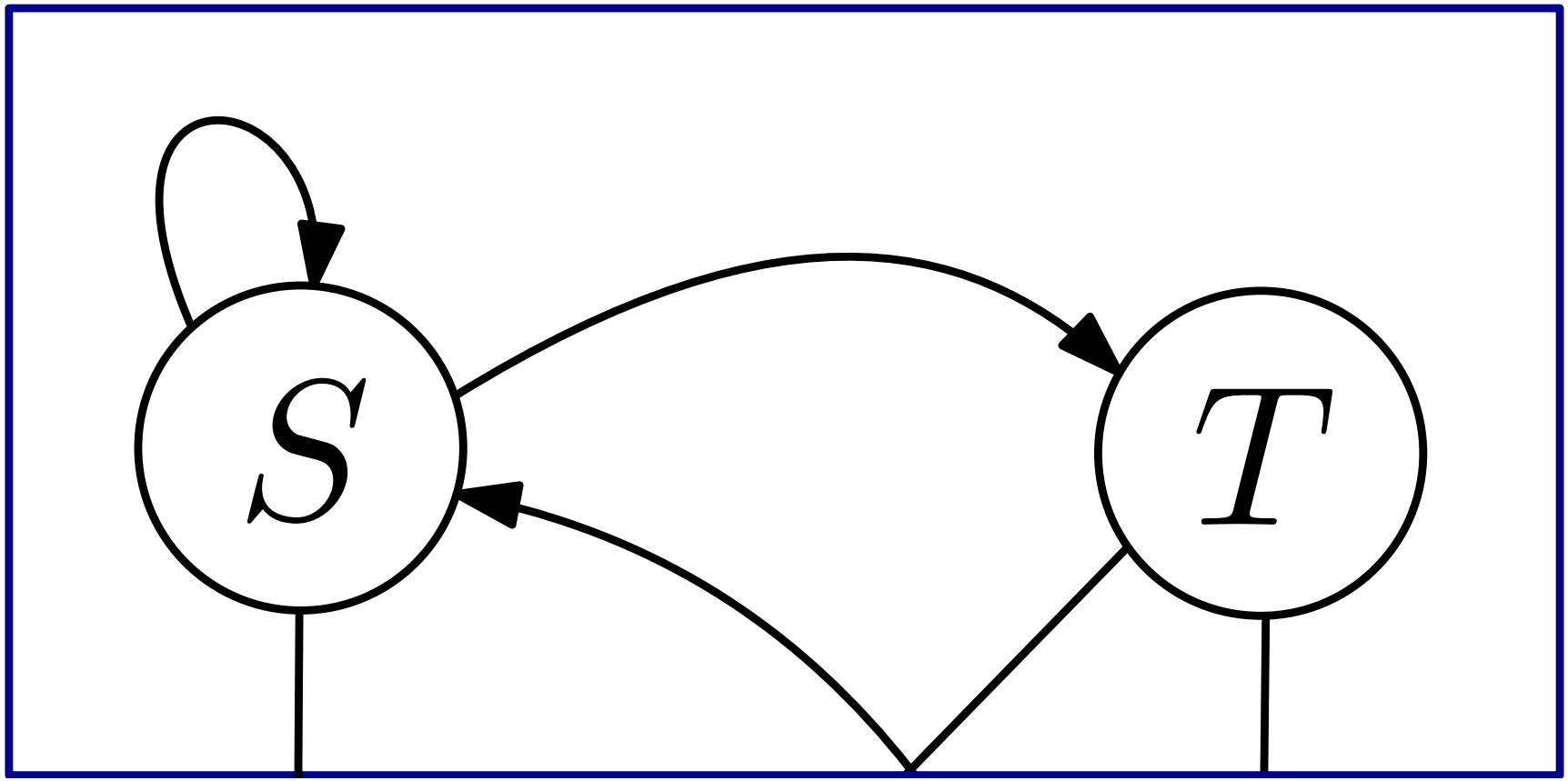
Dependency graph for the refined system



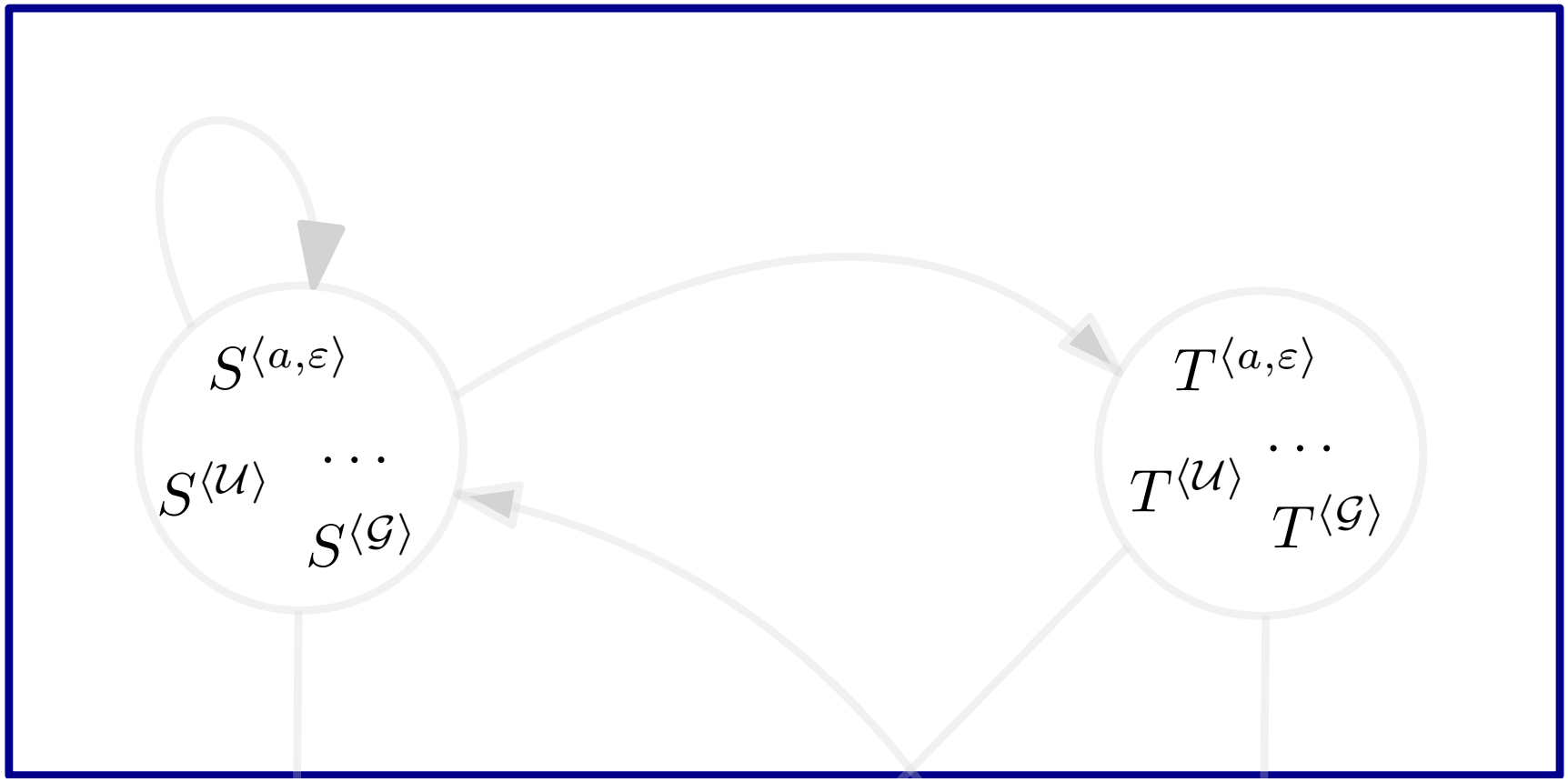
Dependency graph for the refined system



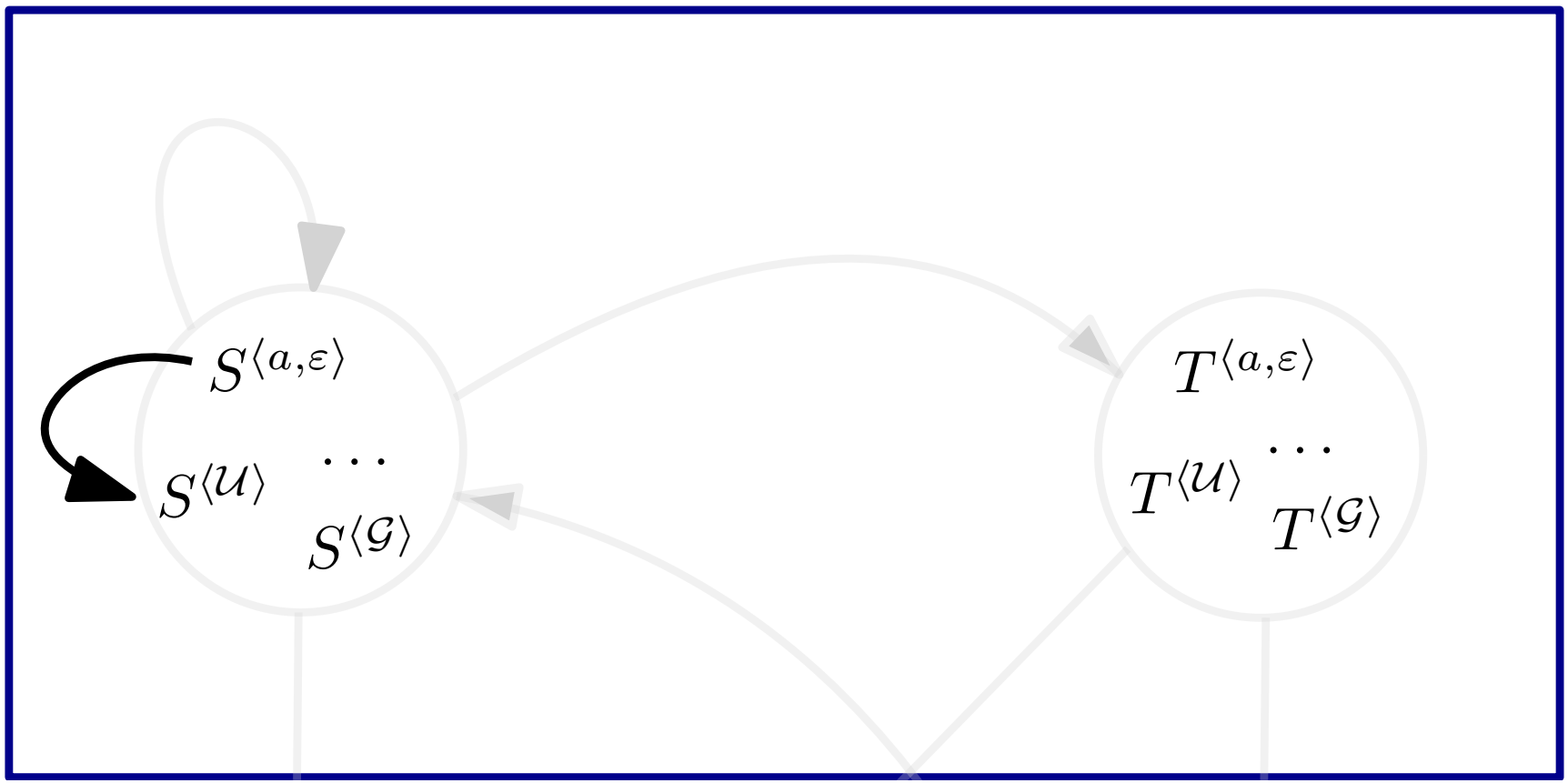
Dependency graph for the refined system



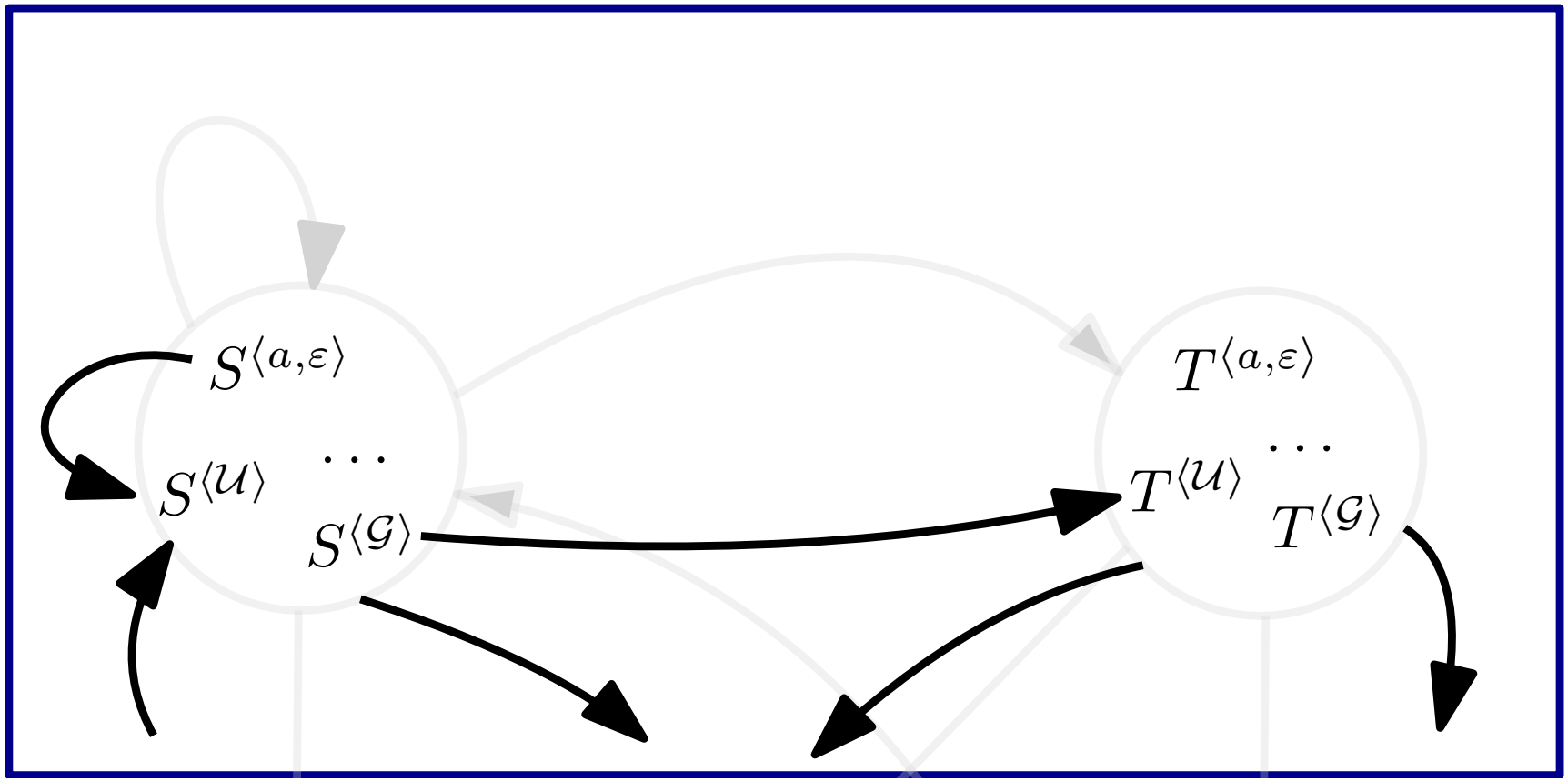
Dependency graph for the refined system



Dependency graph for the refined system

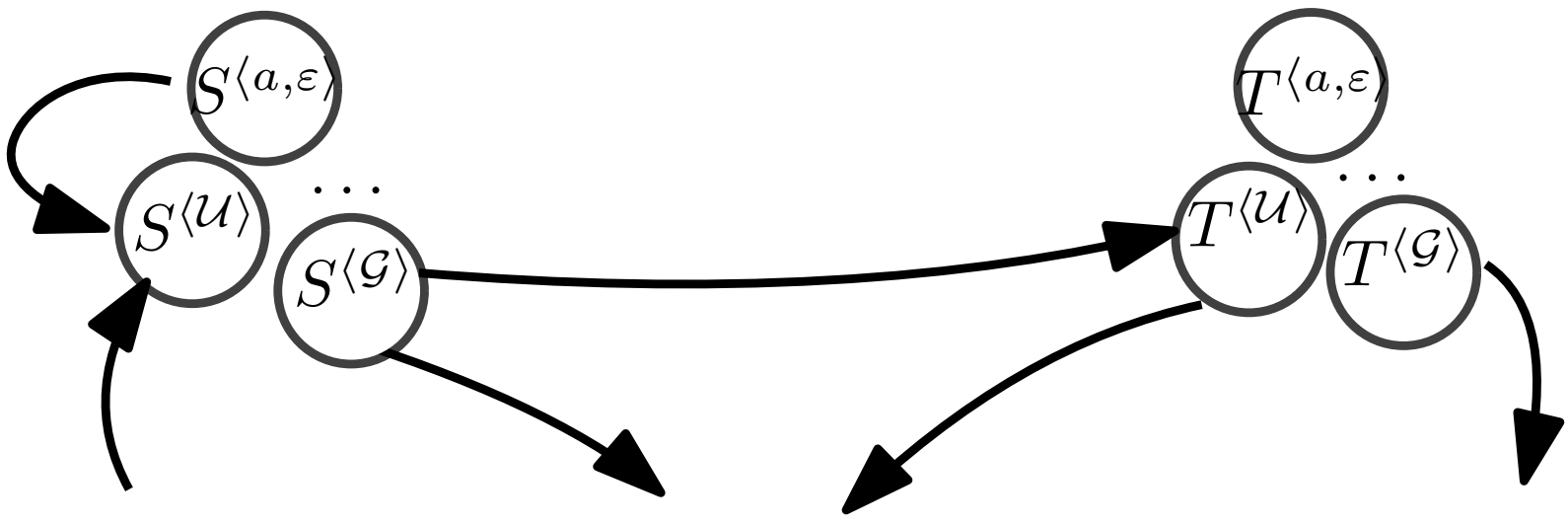


Dependency graph for the refined system

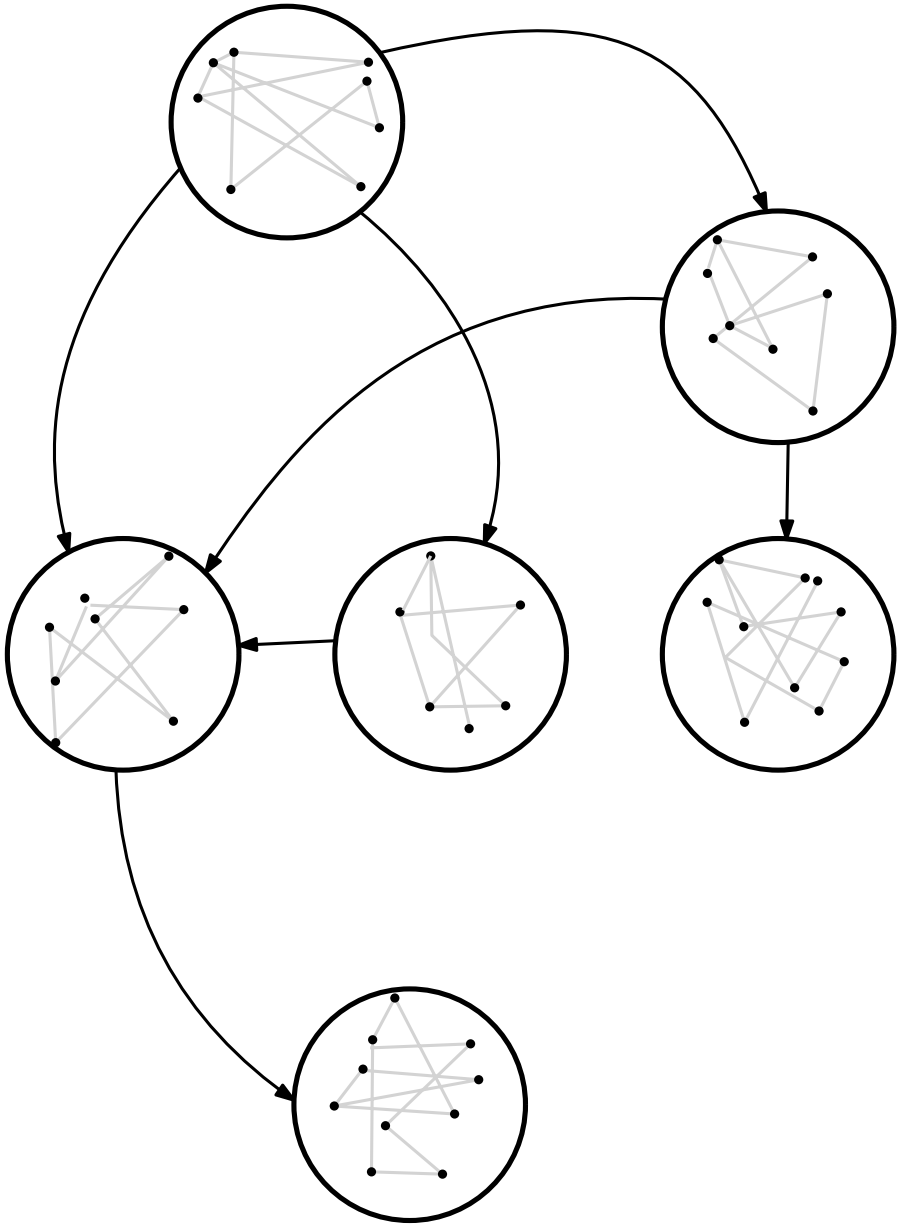


Dependency graph for the refined system

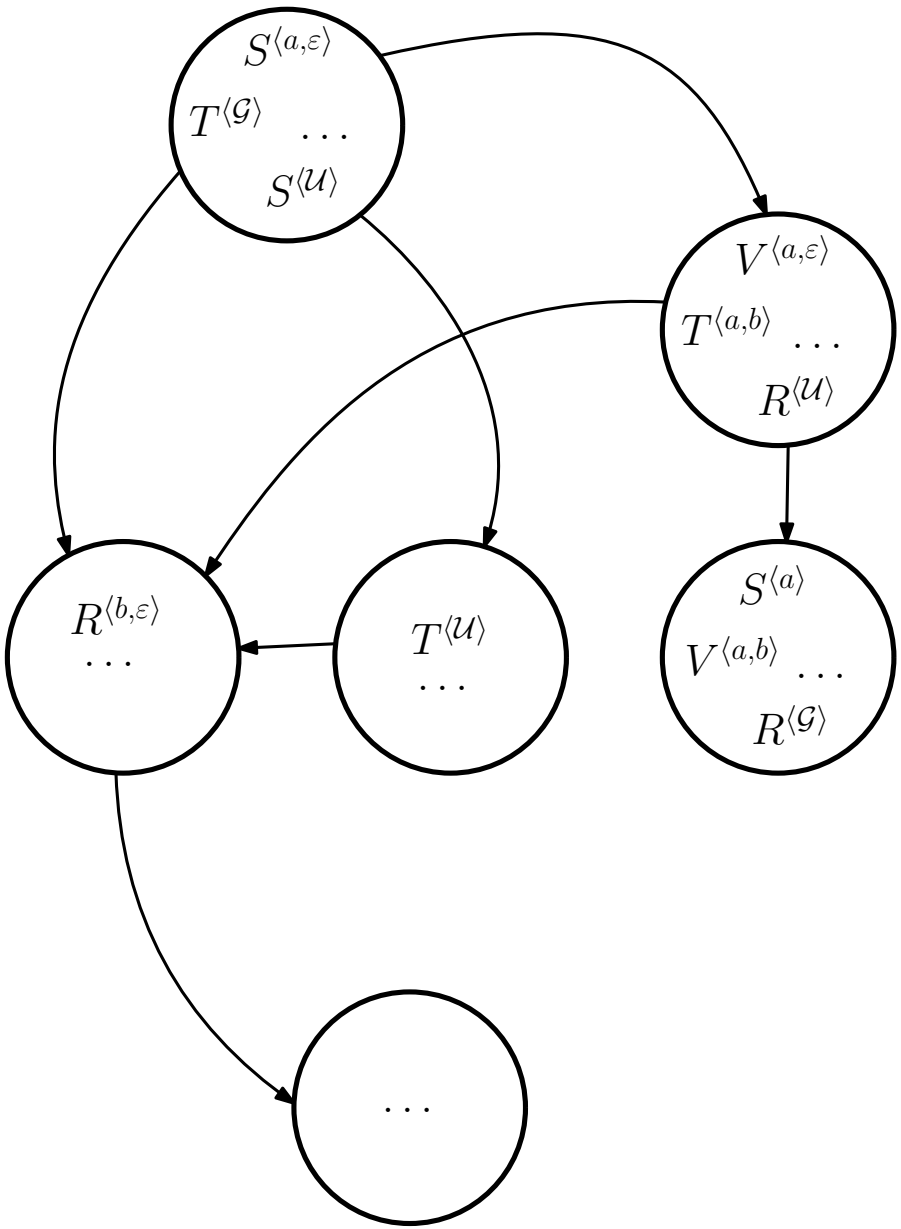
New system may not be strongly connected anymore!

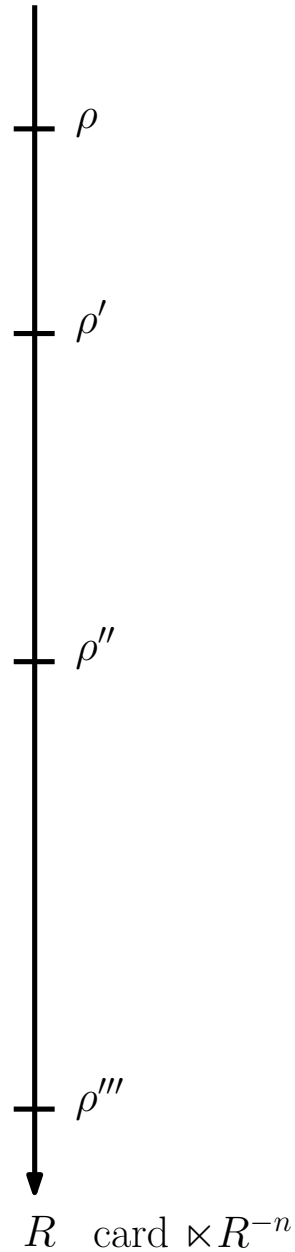


Decomposition into DAG of scc's

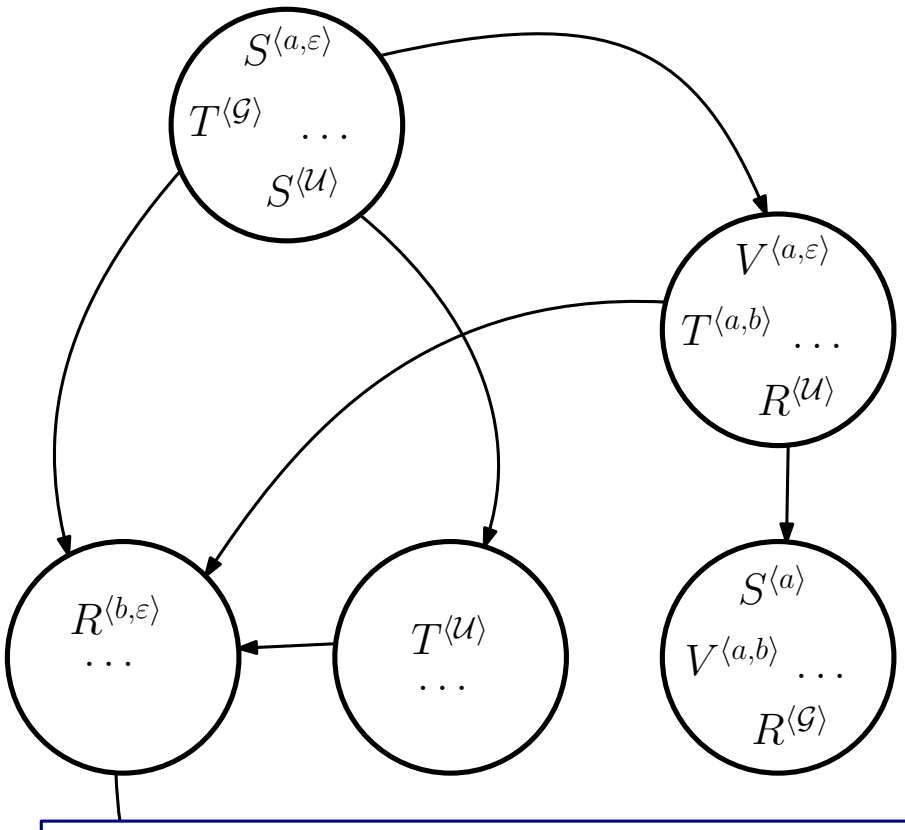


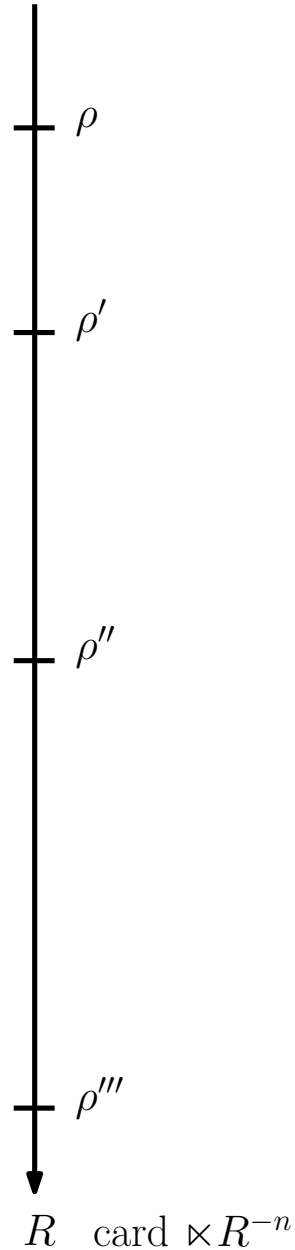
Decomposition into DAG of scc's





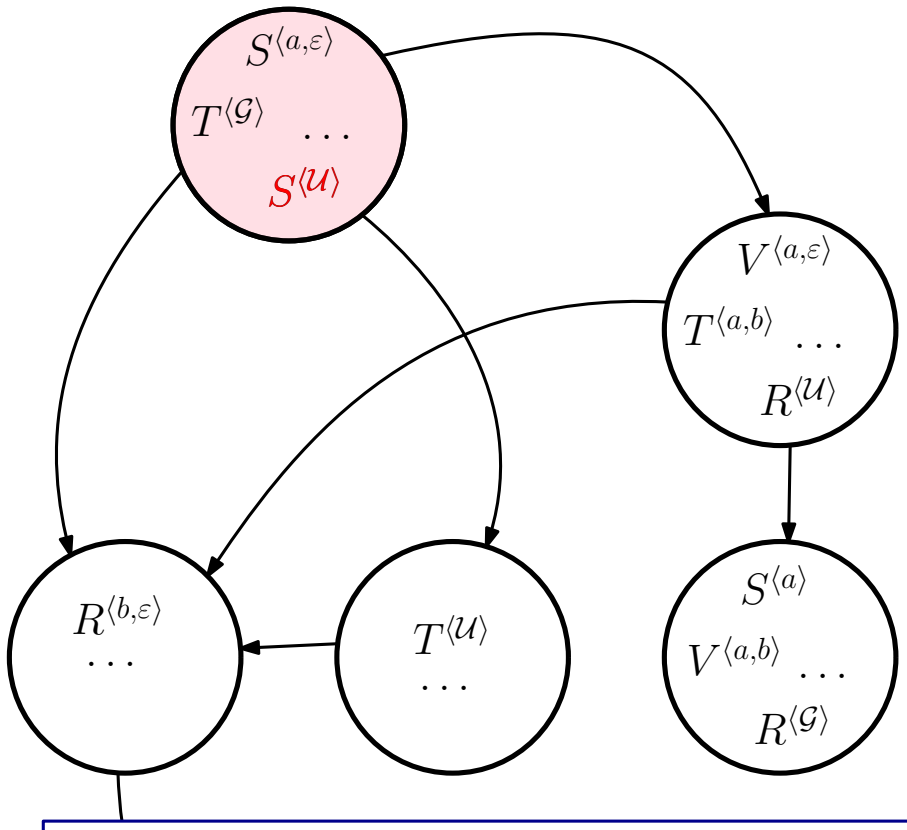
Decomposition into DAG of scc's

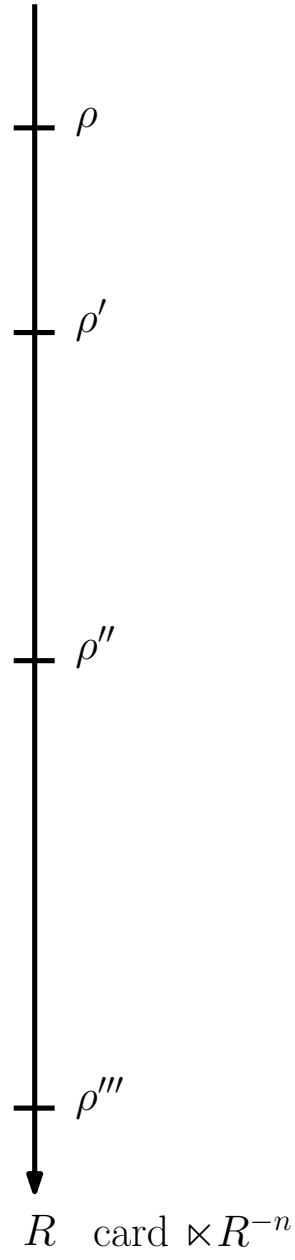




$\mathcal{Y}^{\langle\mathcal{U}\rangle}$ is in a scc with $R = \rho$
$\Rightarrow$ overabundance of universal expressions!

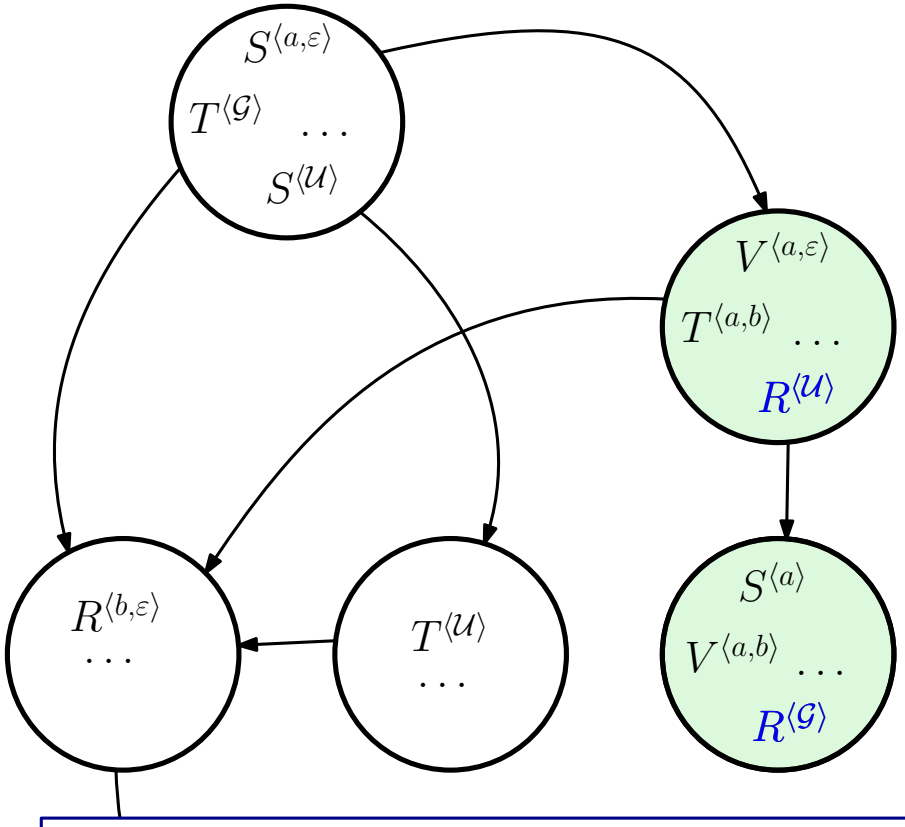
Decomposition into DAG of scc's

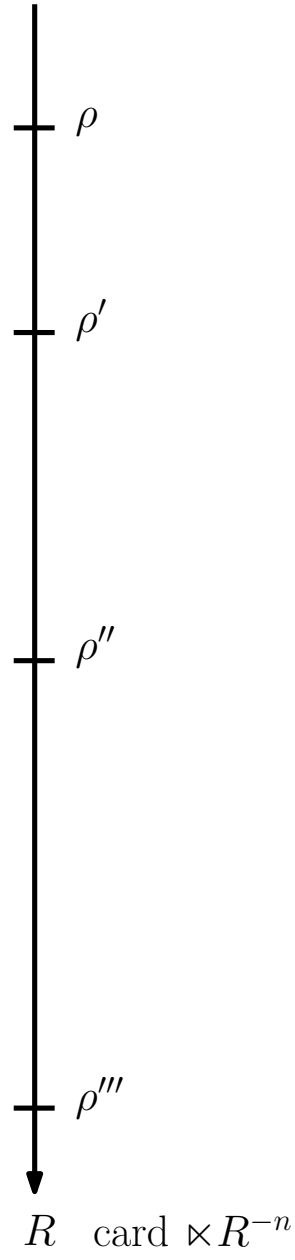




$\mathcal{Y}^{\langle\mathcal{U}\rangle}$ is in a scc with $R = \rho$
$\Rightarrow$ overabundance of universal expressions!

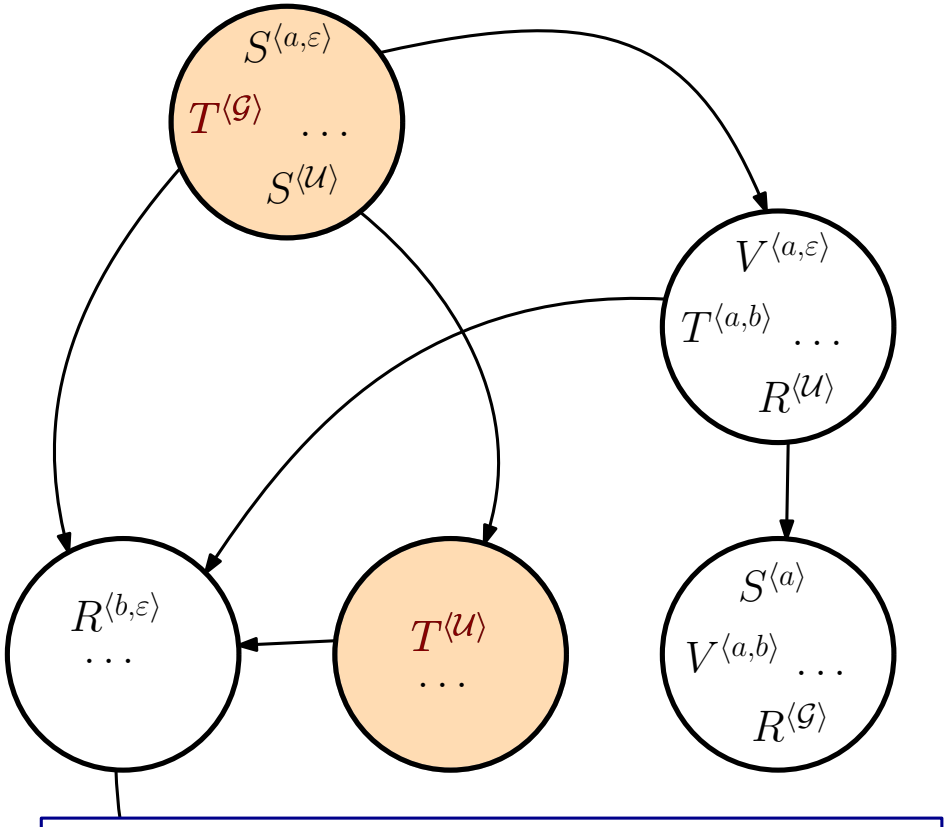
Decomposition into DAG of scc's



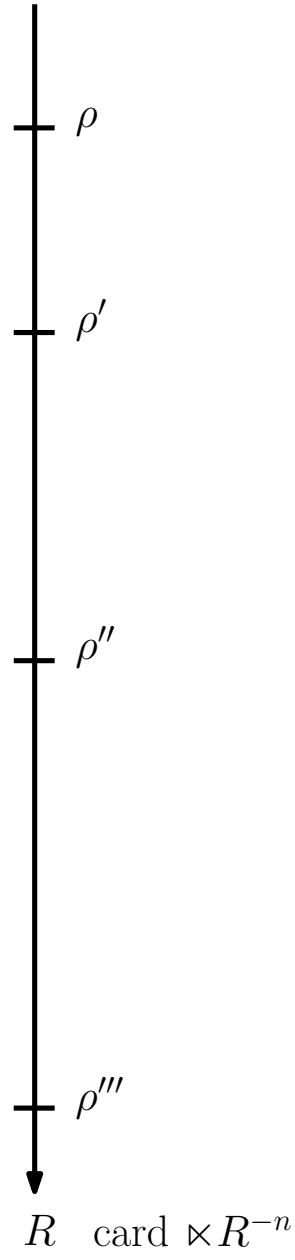


$\mathcal{Y}^{\langle\mathcal{U}\rangle}$, $\mathcal{Y}^{\langle\mathcal{G}\rangle}$ is scc's with $R > \rho$
$\Rightarrow$ no overabundance of universal expressions!

Decomposition into DAG of scc's



Otherwise: our heuristic is not sufficient to conclude



Partial Implementation

- We can determine the scc's C with radius $R = \rho$ using the characteristic system $\det(\mathtt{Id} - \mathtt{Jac}_{\mathbf{w}_C}[\psi_C](\rho; \mathbf{w}(\rho))) = 0$
- Further computations make it possible to determine also the C_i's in

$$y_{i,n} \sim C_i n^{-3/2} \rho^{-n}$$

- We can then compute the proportion of universals

Some implementation issues

- Numerical approximation
- Not all scenarios are implemented yet (but for the moment all systems we have encountered were "nice")

Some results on concrete systems

$$R = a + b + \varepsilon + \overset{\star}{|}_{R} + \overset{\bullet}{/\backslash}_{R\ R} + \overset{+}{/\backslash}_{R\ R} \qquad \text{(all regular expression trees)}$$

$$S = R + \overset{\star}{|}_{R}, \quad R = a + b + \varepsilon + \overset{+}{/\backslash}_{S\ S} + \overset{\bullet}{/\backslash}_{S\ S}. \qquad \text{(no two stars)}$$

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{|}_{S_s} + \overset{+}{/\backslash}_{S\ S_p} + \overset{\bullet}{/\backslash}_{S\ S} \\ S_s = a + b + \varepsilon + \overset{+}{/\backslash}_{S\ S_p} + \overset{\bullet}{/\backslash}_{S\ S} \\ S_p = a + b + \varepsilon + \overset{\bullet}{/\backslash}_{S\ S} + \overset{\star}{|}_{S_s} \end{cases} \qquad \text{(and associativity of } +\text{)}$$

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{|}_{S_s} + \overset{+}{/\backslash}_{S\ S_p} + \overset{\bullet}{/\backslash}_{S\ S_c} \\ S_s = a + b + \varepsilon + \overset{+}{/\backslash}_{S\ S_p} + \overset{\bullet}{/\backslash}_{S\ S_c} \\ S_p = a + b + \varepsilon + \overset{\star}{|}_{S_s} + \overset{\bullet}{/\backslash}_{S\ S_c} \\ S_c = a + b + \varepsilon + \overset{\star}{|}_{S_s} + \overset{+}{/\backslash}_{S\ S_p} \end{cases} \qquad \text{(and associativity of } \bullet\text{)}$$

Some results on concrete systems

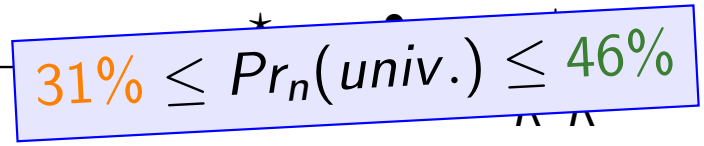
$R = a + \overset{\star}{\bullet}\ \overset{\bullet}{\bullet}\ \overset{+}{\bullet}\ R\ R$ $\quad 31\% \leq Pr_n(univ.) \leq 46\%$ (all regular expression trees)

$S = R + \overset{\star}{\bullet}\ S\ S + \overset{\bullet}{\underset{S}{\wedge}}\ S$. $\quad 27\% \leq Pr_n(univ.) \leq 42\%$ (no two stars)

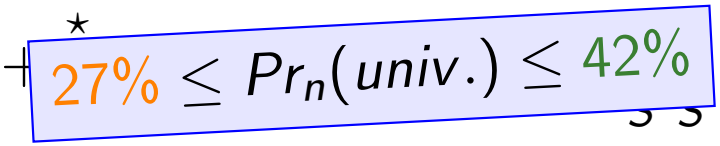
$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S}{\wedge}}\ S_p + \overset{\bullet}{\underset{S}{\wedge}}\ S \\ S_s = a + b + \varepsilon + \overset{+}{\underset{S}{\wedge}}\ S_p + \overset{\bullet}{\underset{S}{\wedge}}\ S \\ S_p = a + b + \varepsilon + \overset{\bullet}{\underset{S}{\wedge}}\ S + \overset{\star}{\underset{S_s}{|}} \end{cases}$$ $\quad 16\% \leq Pr_n(univ.) \leq 27\%$ (and associativity of $+$)

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S}{\wedge}}\ S_p + \overset{\bullet}{\underset{S}{\wedge}}\ S_c \\ S_s = a + b + \varepsilon + \overset{+}{\underset{S}{\wedge}}\ S_p + \overset{\bullet}{\underset{S}{\wedge}}\ S_c \\ S_p = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{\bullet}{\underset{S}{\wedge}}\ S_c \\ S_c = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S}{\wedge}}\ S_p \end{cases}$$ (and associativity of $\bullet$)

Some results on concrete systems

$R = a +$ 

(all regular expression trees)

$S = R +$ 

$+$  . (no two stars)

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S\ S_p}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \\ S_s = a + b + \varepsilon + \overset{+}{\underset{S\ S_s}{\wedge}} + \overset{\bullet}{\underset{S\ S}{\wedge}} \\ S_p = a + b + \varepsilon + \overset{\bullet}{\underset{S\ S}{\wedge}} + \overset{\star}{\underset{S_s}{|}} \end{cases}$$

(and associativity of $+$)

$$\begin{cases} S = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S\ S_p}{\wedge}} + \overset{\bullet}{\underset{S\ S_c}{\wedge}} \\ S_s = a + b + \varepsilon + \overset{+}{\underset{S\ S_s}{\wedge}} + \overset{\bullet}{\underset{S\ S_c}{\wedge}} \\ S_p = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{\bullet}{\underset{S\ S_c}{\wedge}} \\ S_c = a + b + \varepsilon + \overset{\star}{\underset{S_s}{|}} + \overset{+}{\underset{S\ S_p}{\wedge}} \end{cases}$$

(and associativity of $\bullet$)

A more complex example: Lee and Shallit 2005

Elaborate system taking many redudancies into account:

$$A = \overset{+}{\underset{\varepsilon \quad A_B}{/\backslash}} + \overset{+}{\underset{C \quad A_C}{/\backslash}} + \overset{+}{\underset{B \quad A_B}{/\backslash}} + \overset{+}{\underset{D \quad A_D}{/\backslash}} + \varepsilon \qquad A' = \overset{+}{\underset{B \quad A_B}{/\backslash}} + \overset{+}{\underset{D \quad A_D}{/\backslash}} \qquad A_C = C + \overset{+}{\underset{C \quad A_C}{/\backslash}} + A_B$$

$$A_B = B + \overset{+}{\underset{B \quad A_D}{/\backslash}} + A_D \qquad A_D = D + \overset{+}{\underset{D \quad A_D}{/\backslash}} \qquad B = \overset{\bullet}{\underset{B_0 \quad B_0}{/\backslash}} + \overset{\bullet}{\underset{B_0 \quad B}{/\backslash}}$$

$$B_0 = A + C + D \qquad C = \overset{\star}{\underset{A'}{|}} + \overset{\star}{\underset{B}{|}} + \overset{\star}{\underset{D}{|}} \qquad D = a + b + \varepsilon\,.$$

A more complex example: Lee and Shallit 2005

Elaborate system taking many redudancies into account:

$$A = \overset{+}{\underset{\varepsilon \quad A_B}{/\backslash}} + \overset{+}{\underset{C \quad A_C}{/\backslash}} + \overset{+}{\underset{B \quad A_B}{/\backslash}} + \overset{+}{\underset{D \quad A_D}{/\backslash}} + \varepsilon \qquad A' = \overset{+}{\underset{B \quad A_B}{/\backslash}} + \overset{+}{\underset{D \quad A_D}{/\backslash}} \qquad A_C = C + \overset{+}{\underset{C \quad A_C}{/\backslash}} + A_B$$

$$A_B = B + \overset{+}{\underset{B \quad A_D}{/\backslash}} + A_D \qquad A_D = D + \overset{+}{\underset{D \quad A_D}{/\backslash}} \qquad B = \overset{\bullet}{\underset{B_0 \quad B_0}{/\backslash}} + \overset{\bullet}{\underset{B_0 \quad B}{/\backslash}}$$

$$B_0 = A + C + D \qquad C = \overset{\star}{\underset{A'}{|}} + \overset{\star}{\underset{B}{|}} + \overset{\star}{\underset{D}{|}} \qquad D = a + b + \varepsilon\,.$$

$$30\% \leq Pr_n(univ.) \leq 40\%$$

Conclusion

- Generating relevant random regular expressions is hard, and counter-intuitive!
- Difficult to test and analyse algorithms working with them
- Surely the uniform distribution is not adapted

Further work:

- Finish the implementation to allow fully automated check of systems

Conclusion

- Generating relevant random regular expressions is hard, and counter-intuitive!
- Difficult to test and analyse algorithms working with them
- Surely the uniform distribution is not adapted

Further work:

- Finish the implementation to allow fully automated check of systems

Thank you!