Matrices over a Kleene algebra

Jules Desharnais
Université Laval
Canada

Plan

1. Definition of Kleene algebra
2. Matrices over a KA
3. Operations on matrices
4. Modal formulae
5. Matrices of types
    - Simulations, bisimulations
    - Projections and products of matrices
6. Conclusion: controller synthesis

Definition of Kleene algebra

Definition. A *Kleene algebra* (KA) is a sixtuple $(K, \leq, \top, \cdot, 0, 1)$ satisfying the following properties:

1. $(K, \leq)$ is a complete lattice with least element 0 and greatest element $\top$. The supremum of a subset $L \subseteq K$ is denoted by $\sqcup L$.
2. $(K, \cdot, 1)$ is a monoid.
3. The operation $\cdot$ is universally disjunctive (i.e., distributes through arbitrary suprema) in both arguments.

The supremum of two elements $x, y \in K$ is given by $x + y \stackrel{\Delta}{=} \sqcup \{x, y\}$.

Definition. A KA is called *Boolean* if its underlying lattice $(K, \leq)$ is a Boolean algebra. This is occasionally needed in the sequel.

Other definitions are possible. For instance, Kozen does not require a KA to be a lattice.

Matrices over a KA

Definition. A matrix over a KA $(K, \leq, 0, \top, \cdot, 1)$ is a function

$$M : \{1, \ldots, m\} \times \{1, \ldots, n\} \to K,$$

where $m, n \in \mathbb{N}$. One can have $m = 0$ or $n = 0$.

Notation.

$\mathbf{A}$	matrix $\mathbf{A}$ with no indication of size
$\mathbf{A}_{ij}$	entry i, j of matrix $\mathbf{A}$
$\mathbf{0}$	matrice whose entries are all 0
$\mathbf{1}$	identity matrix (square),
T	matrix whose entries are all $\top$
$[\![a]\!]$	matrix whose entries are all a

The size of a matrix may be explicitly added in bold font: $\mathbf{A_{mn}}$.

Operations on matrices

$$
\begin{aligned}
\mathbf{0}_{ij} &= 0 \\
\mathbf{1}_{ij} &= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \\
\mathsf{T}_{ij} &= \top \\
(\overline{\mathbf{A}})_{ij} &= \overline{\mathbf{A}_{ij}} \\
(\mathbf{A} + \mathbf{B})_{ij} &= \mathbf{A}_{ij} + \mathbf{B}_{ij} \\
(\mathbf{A} \sqcap \mathbf{B})_{ij} &= \mathbf{A}_{ij} \sqcap \mathbf{B}_{ij} \\
(\mathbf{A} \cdot \mathbf{B})_{ij} &= \bigsqcup(k :: \mathbf{A}_{ik} \cdot \mathbf{B}_{kj}) \\
(\mathbf{A}^{\mathsf{T}})_{ij} &= \mathbf{A}_{ji} \\
\mathbf{A} \leq \mathbf{B} \Leftrightarrow & \ \forall(i, j :: \mathbf{A}_{ij} \leq \mathbf{B}_{ij})
\end{aligned}
$$

Note: $+, \sqcap, \cdot, \leq$ defined only for compatible size matrices.

Lemma. Let $\mathcal{M}_{\mathbf{mn}}$ be the set of matrices of size $\mathbf{m}$ by $\mathbf{n}$ over K. For all $\mathbf{n} \in \mathbb{N}$,

$$(\mathcal{M}_{\mathbf{nn}}, \leq, \mathbf{0}_{\mathbf{nn}}, \mathsf{T}_{\mathbf{nn}}, \cdot, \mathbf{1}_{\mathbf{nn}}) \text{ is a KA.}$$

To accomodate matrices with different sizes, a definition of heterogeneous KA can be given and the above lemma extends in the appropriate way to such KAs.

This is well known. See, e.g.,

D. Kozen. *The design and analysis of algorithms.* Springer-Verlag, New York, 1992.

Definition. A *type* is an element $t \leq 1$. The *negation* of a type $t \leq 1$ in a KA is $\neg t \triangleq \bar{t} \sqcap 1$.

A (square) matrix $\mathbf{T}$ is a type if $\mathbf{T} \leq \mathbf{1}$. E.g., if t_1, t_2, t_3 are types,

$$\begin{pmatrix} t_1 & 0 & 0 \\ 0 & t_2 & 0 \\ 0 & 0 & t_3 \end{pmatrix} \text{ is a type and } \quad \neg \begin{pmatrix} t_1 & 0 & 0 \\ 0 & t_2 & 0 \\ 0 & 0 & t_3 \end{pmatrix} = \begin{pmatrix} \neg t_1 & 0 & 0 \\ 0 & \neg t_2 & 0 \\ 0 & 0 & \neg t_3 \end{pmatrix}.$$

Lemma.

1. Composition of types is idempotent, i.e. $t \leq 1 \;\Rightarrow\; t \cdot t = t$.
2. The infimum of two types is their product: $s, t \leq 1 \;\Rightarrow\; s \sqcap t \;=\; s \cdot t$.

Other operations

Domain and codomain

Definition. The *domain* operation is defined by a Galois connection:

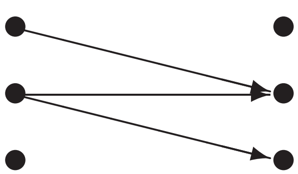
$$\forall (y : y \leq 1 : \ \ulcorner a \leq y \ \stackrel{\text{def}}{\Leftrightarrow} \ a \leq y \cdot \top)$$

(this is a well-defined operation).

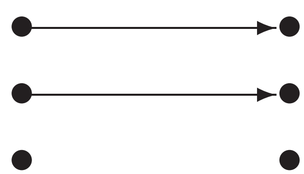
The *co-domain* $a^{\urcorner}$ is defined symmetrically.

Example in REL

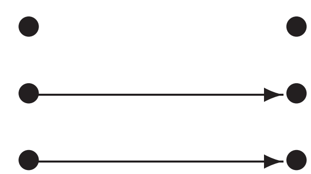
$a :$



$\ulcorner a :$



$a^{\urcorner} :$



Laws about domain and codomain

Lemma.

1. $\ulcorner a \cdot a \;=\; a$
2. $\ulcorner(a \cdot b) \leq \ulcorner a$
3. $x \leq 1 \;\;\Rightarrow\;\; \ulcorner x \;=\; x$
4. $\ulcorner a \;=\; 0 \Leftrightarrow a \;=\; 0$

Domain and codomain of a matrix

$$(\ulcorner\mathbf{A})_{ii} = \bigsqcup(j :: \ulcorner(\mathbf{A}_{ij})) \qquad\qquad i \neq j \Rightarrow (\ulcorner\mathbf{A})_{ij} = 0$$

$$(\mathbf{A}\urcorner)_{ii} = \bigsqcup(j :: (\mathbf{A}_{ji})\urcorner) \qquad\qquad i \neq j \Rightarrow (\mathbf{A}\urcorner)_{ij} = 0$$

This can be shown from the definition of $\ulcorner$ and $\urcorner$.

$$\ulcorner\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \ulcorner a + \ulcorner b & 0 \\ 0 & \ulcorner c + \ulcorner d \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\urcorner = \begin{pmatrix} a\urcorner + c\urcorner & 0 \\ 0 & b\urcorner + d\urcorner \end{pmatrix}$$

Residuals (factors)

$$\text{Left residual:} \quad a \cdot b \leq c \;\Leftrightarrow\; a \leq c/b$$

$$\text{Right residual:} \quad a \cdot b \leq c \;\Leftrightarrow\; b \leq a\backslash b$$

For matrices:

$$\text{Left residual:} \quad (\mathbf{A}/\mathbf{B})_{ij} \;=\; \textstyle\prod(k :: \mathbf{A}_{ik}/\mathbf{B}_{jk})$$

$$\text{Right residual:} \quad (\mathbf{A}\backslash\mathbf{B})_{ij} \;=\; \textstyle\prod(k :: \mathbf{A}_{ki}\backslash\mathbf{B}_{kj})$$

For instance,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} / \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a/e \sqcap b/f & a/g \sqcap b/h \\ c/e \sqcap d/f & c/g \sqcap d/h \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \backslash \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a\backslash e \sqcap c\backslash g & a\backslash f \sqcap c\backslash h \\ b\backslash e \sqcap d\backslash g & b\backslash f \sqcap d\backslash h \end{pmatrix}$$

Proof of $(\mathbf{A}/\mathbf{B})_{ij} = \sqcap(k :: \mathbf{A}_{ik}/\mathbf{B}_{jk})$

$$\forall(i,j :: \mathbf{X}_{ij} \le (\mathbf{A}/\mathbf{B})_{ij})$$

$\Leftrightarrow$ $\langle$ Definition of $\le$ for matrices $\rangle$

$$\mathbf{X} \le \mathbf{A}/\mathbf{B}$$

$\Leftrightarrow$ $\langle$ Definition of $/$ $\rangle$

$$\mathbf{X} \cdot \mathbf{B} \le \mathbf{A}$$

$\Leftrightarrow$ $\langle$ Definition of $\le$ for matrices $\rangle$

$$\forall(i,k :: (\mathbf{X} \cdot \mathbf{B})_{ik} \le \mathbf{A}_{ik})$$

$\Leftrightarrow$ $\langle$ Definition of $\cdot$ for matrices $\rangle$

$$\forall(i,k :: \bigsqcup(j :: \mathbf{X}_{ij} \cdot \mathbf{B}_{jk}) \le \mathbf{A}_{ik})$$

$\Leftrightarrow$ $\langle$ Definition of $\sqcup$ $\rangle$

$$\forall(i,j,k :: \mathbf{X}_{ij} \cdot \mathbf{B}_{jk} \le \mathbf{A}_{ik})$$

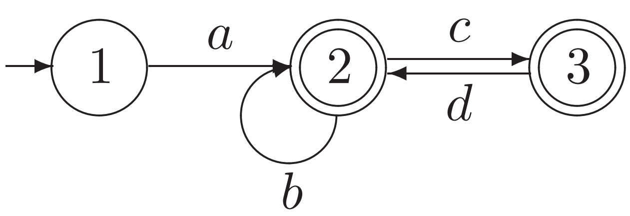
$\Leftrightarrow$ $\langle$ Definition of $/$ $\rangle$

$$\forall(i,j,k :: \mathbf{X}_{ij} \le \mathbf{A}_{ik}/\mathbf{B}_{jk})$$

$\Leftrightarrow$ $\langle$ Definition of $\sqcap$ $\rangle$

$$\forall(i,j :: \mathbf{X}_{ij} \le \sqcap(k :: \mathbf{A}_{ik}/\mathbf{B}_{jk}))$$

Representing automata or transition systems



$$M = (K, \mathbf{I}, \mathbf{A}, \mathbf{F})$$

where

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 0 & a & 0 \\ 0 & b & c \\ 0 & d & 0 \end{pmatrix} \qquad \mathbf{F} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

The element of K given by

$$\mathbf{I} \cdot \mathbf{A}^* \cdot \mathbf{F}$$

is the language of M if K is an algebra of languages and the angelic "input-output" relation of the graph if K is an algebra of relations.

Oege's problem

Given two automata $G \triangleq (K, \mathbf{I}_G, \mathbf{G}, [\![1]\!])$ and $P \triangleq (K, \mathbf{I}_P, \mathbf{P}, \mathbf{F}_P)$, find the *largest* (column) relation $\mathbf{S}$ such that

$$\mathbf{I}_G \cdot \mathbf{G}^* \cdot \mathbf{S} \;\leq\; \mathbf{I}_P \cdot \mathbf{P}^* \cdot \mathbf{F}_P \; .$$

We assume that the entries of $\mathbf{G}$ and $\mathbf{P}$ are joins of atoms that are prime elements (i.e., elements a such that $a \neq 1$ and $a = b \cdot c \;\Rightarrow\; b = 1 \vee c = 1$). Let n_G and n_P be the number of states of G and P, respectively.

$$\mathbf{I}_G \cdot \mathbf{G}^* \cdot \mathbf{S} \;\leq\; \mathbf{I}_P \cdot \mathbf{P}^* \cdot \mathbf{F}_P$$

$\Leftrightarrow$ $\langle$ Entries of matrices are joins of atoms that are prime elements (both automata move in step, reading one symbol at a time) $\rangle$

$$\forall(n : n \in \mathbb{N} : \mathbf{I}_G \cdot \mathbf{G}^n \cdot \mathbf{S} \;\leq\; \mathbf{I}_P \cdot \mathbf{P}^n \cdot \mathbf{F}_P)$$

$\Leftrightarrow$ $\langle$ Properties of finite automata: examining sequences longer than $n_G \times n_P$ brings no new constraints & Definition of residual $\rangle$

$$\forall(n : n \leq n_G \times n_P : \mathbf{S} \;\leq\; (\mathbf{I}_G \cdot \mathbf{G}^n)\backslash(\mathbf{I}_P \cdot \mathbf{P}^n \cdot \mathbf{F}_P))$$

The largest solution is $\mathbf{S} \;\triangleq\; \sqcap(n : n \leq n_G \times n_P : (\mathbf{I}_G \cdot \mathbf{G}^n)\backslash(\mathbf{I}_P \cdot \mathbf{P}^n \cdot \mathbf{F}_P)) \;\sqcap\; [\![1]\!]$.

Aside: the large, intuitive, steps in the proof have to be formalized.

An algorithm

A possible algorithm for computing $\mathbf{S}$ proceeds by computing $\mathbf{I}_G \cdot \mathbf{G}^n)\backslash(\mathbf{I}_P \cdot \mathbf{P}^n \cdot \mathbf{F}_P)$ for increasing values of n and then taking the meet.

At first sight, this seems reasonably efficient:

- No need to construct the deterministic automaton corresponding to P.
- Possibility to stop before $n_G \times n_P$ if one keeps track of visited states of (G, P) when increasing n.
- No need to calculate $\mathbf{G}^n$ (a square matrix), but only $\mathbf{I}_G \cdot \mathbf{G}^n$ (a linear matrix), and similarly for $\mathbf{P}$.

However, a more careful investigation reveals bad news. Suppose $\mathbf{I}_G \triangleq \begin{pmatrix} 1 & 1 \end{pmatrix}$ and $\mathbf{G} \triangleq \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Then,

$$\begin{array}{rcl} \mathbf{I}_G \cdot \mathbf{G}^0 & = & \begin{pmatrix} 1 & 1 \end{pmatrix} \\ \mathbf{I}_G \cdot \mathbf{G}^1 & = & \begin{pmatrix} a+c & b+d \end{pmatrix} \\ \mathbf{I}_G \cdot \mathbf{G}^2 & = & \begin{pmatrix} (a+c)\cdot a + (b+d)\cdot c & (a+c)\cdot b + (b+d)\cdot d \end{pmatrix} \end{array}$$

Note how the number of symbols in the result more than doubles at each iteration. This means that the computation of $\mathbf{I}_G \cdot \mathbf{G}^n)\backslash(\mathbf{I}_P \cdot \mathbf{P}^n \cdot \mathbf{F}_P)$ is *exponential in the size of G and also in the size of P.*

Conjecture

If P is deterministic, then the expression for **S** can be put under a form that can be evaluated in time polynomial in the size of P.

Even if this conjecture holds, the algorithm would still be exponential in the size of an arbitrary (nondeterministic) P. These is little hope to do better. Having a polynomial solution to the above problem would lead to a polynomial solution to the problem of determining the equivalence of two automata (this requires only a slight modification to Oege's problem). But there is no known such polynomial algorithm.

I thank Michel Sintzoff for pointing the relationship between Oege's problem and the problem of showing the equivalence of two automata.

Modal formulae

Next slides : two examples of modal operators.

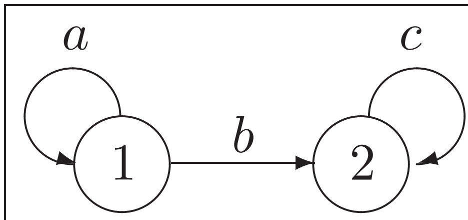
Other modal operators are treated similarly.

Modal formula $\langle b \rangle \phi$

Assume this is read as "there is a b transition leading to a state satisfying ϕ". Suppose the interpretation of ϕ is $t \leq 1$.

The interpretation of $\langle b \rangle \phi$ on $\mathbf{A}$ is the type $\ulcorner\big((\mathbf{A} \sqcap [\![b]\!]) \cdot (\mathbf{1} \sqcap [\![t]\!])\big)$.

$$
\begin{aligned}
& \langle b \rangle \phi \\
= & \qquad \langle \text{ Definition above \& Example in the box } \rangle \\
& \ulcorner\left(\left(\begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \sqcap \begin{pmatrix} b & b \\ b & b \end{pmatrix}\right) \cdot \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \sqcap \begin{pmatrix} t & t \\ t & t \end{pmatrix}\right)\right) \\
= & \qquad \langle \text{ Assuming } a \sqcap b = c \sqcap b = 0 \ \rangle \\
& \ulcorner\left(\begin{pmatrix} 0 & b \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} t & 0 \\ 0 & t \end{pmatrix}\right) \\
= & \\
& \ulcorner\begin{pmatrix} 0 & b \cdot t \\ 0 & 0 \end{pmatrix} \\
= & \\
& \begin{pmatrix} \ulcorner(b \cdot t) & 0 \\ 0 & 0 \end{pmatrix}
\end{aligned}
$$



$$\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}$$

Modal formula $\Diamond\phi$

Assume this is read as "every trace from the current state eventually leads to a state satisfying ϕ". Suppose the interpretation of ϕ is $t \leq 1$.

The interpretation of $\Diamond\phi$ on $\mathbf{A}$ is the type

$$\mu(x :: ([\![t]\!] \sqcap \mathbf{1}) \ \vee \ (\mathbf{A} \to x)) \ .$$

Matrices of types

Every matrix $\mathbf{R} \leq [\![1]\!]$ is a (fuzzy???) relation, with converse

$$\mathbf{R}^{\cup} \triangleq \mathbf{R}^{\mathsf{T}}$$

and complement

$$\widetilde{\mathbf{R}} \triangleq \overline{\mathbf{R}} \sqcap [\![1]\!].$$

If $\mathbf{P}, \mathbf{Q}, \mathbf{R} \leq [\![1]\!]$, then

$$\mathbf{P} \cdot \mathbf{Q} \leq \mathbf{R} \Leftrightarrow \mathbf{P}^{\cup} \cdot \widetilde{\mathbf{R}} \leq \widetilde{\mathbf{Q}} \Leftrightarrow \widetilde{\mathbf{R}} \cdot \mathbf{Q}^{\cup} \leq \widetilde{\mathbf{P}} \qquad \text{(Schröder equivalences)}$$

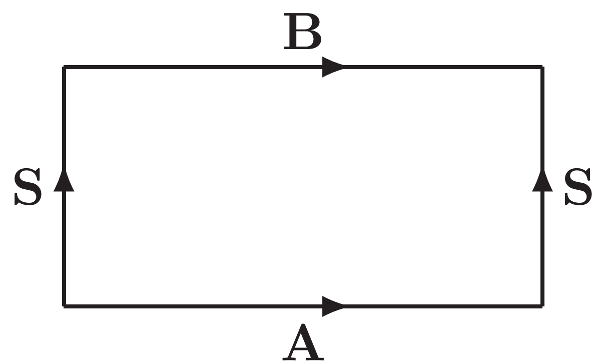
Simulations, bisimulations

We say that $\mathbf{B}$ *simulates* $\mathbf{A}$ if there is a relation $\mathbf{S}$ such that

$$\mathbf{S} \cdot \mathbf{B} \;\leq\; \mathbf{A} \cdot \mathbf{S} \;.$$

We say that $\mathbf{A}$ *bisimulates* $\mathbf{B}$ if there is a relation $\mathbf{S}$ such that

$$\mathbf{S}^{\cup} \cdot \mathbf{A} \;\leq\; \mathbf{B} \cdot \mathbf{S}^{\cup} \qquad \text{and} \qquad \mathbf{S} \cdot \mathbf{B} \;\leq\; \mathbf{A} \cdot \mathbf{S} \;.$$



The join of simulations (bisimulations) is again a simulation (bisimulation). Hence, there is a largest simulation (bisimulation).

Calculating largest bisimulations (for finite structures)

$$
\begin{array}{ll}
& \mathbf{A} \text{ bisimulates } \mathbf{B} \\
\Leftrightarrow & \qquad \langle \text{ Definition of bisimulation } \rangle \\
& \mathbf{S}^{\cup} \cdot \mathbf{A} \ \leq \ \mathbf{B} \cdot \mathbf{S}^{\cup} \quad \text{and} \quad \mathbf{S} \cdot \mathbf{B} \ \leq \ \mathbf{A} \cdot \mathbf{S} \\
\Leftrightarrow & \qquad \langle \text{ Definition of residuals } \rangle \\
& \mathbf{S}^{\cup} \ \leq \ (\mathbf{B} \cdot \mathbf{S}^{\cup})/\mathbf{A} \quad \text{and} \quad \mathbf{S} \ \leq \ (\mathbf{A} \cdot \mathbf{S})/\mathbf{B}
\end{array}
$$

Let $f(\mathbf{X}) \ \triangleq \ (\mathbf{B} \cdot \mathbf{X}^{\cup})/\mathbf{A} \ \sqcap \ \mathbf{R}$ and $g(\mathbf{X}) \ \triangleq \ (\mathbf{A} \cdot \mathbf{X})/\mathbf{B} \ \sqcap \ \mathbf{R}$.

1. Set $\mathbf{R} \ \triangleq \ [\![1]\!]$. Calculate $g(\mathbf{R}), g^2(\mathbf{R}), \ldots, g^m(\mathbf{R}) = g^{m+1}(\mathbf{R})$.

   $g^m(\mathbf{R})$ is the greatest fixed point of g (largest simulation) below $\mathbf{R}$.

2. Set $\mathbf{R} \ \triangleq \ (g^m(\mathbf{R}))^{\cup}$. Calculate the greatest fixed point $\mathbf{X}$ of f .

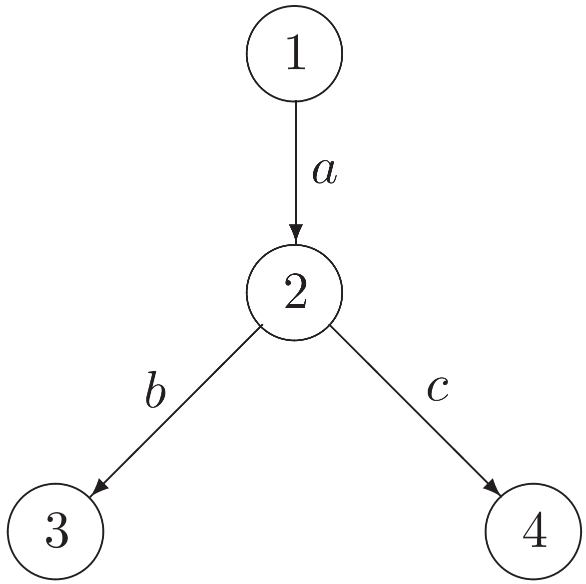
3. Set $\mathbf{R} \ \triangleq \ \mathbf{X}^{\cup}$. Calculate the greatest fixed point $\mathbf{X}$ of g .

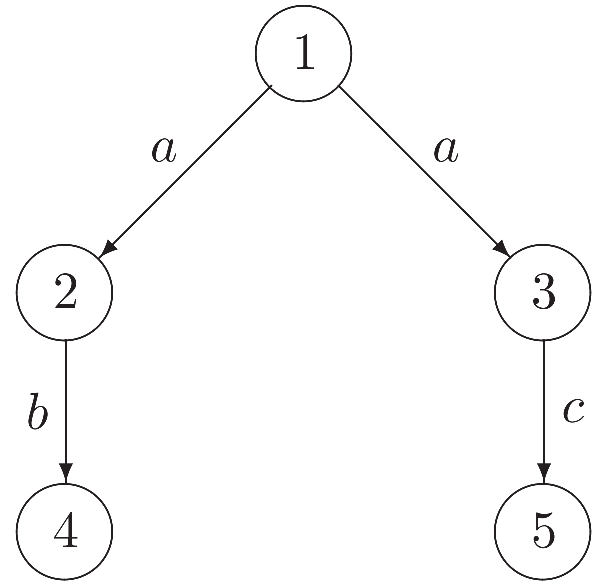
4. Set $\mathbf{R} \ \triangleq \ \mathbf{X}^{\cup}$. Etc., until obtaining a relation $\mathbf{S}$ such that $\mathbf{S}$ is a fixed point of g (with $\mathbf{R} \triangleq \mathbf{S}$) and $\mathbf{S}^{\cup}$ is a fixed point of g (with $\mathbf{R} \triangleq \mathbf{S}^{\cup}$).

The relation $\mathbf{S}$ thus found is the largest bisimulation.

Largest bisimulations (example 1)

Assume a, b, c mutually disjoint and $\ulcorner a = \ulcorner b = \ulcorner c = 1$ (e.g., in LAN).



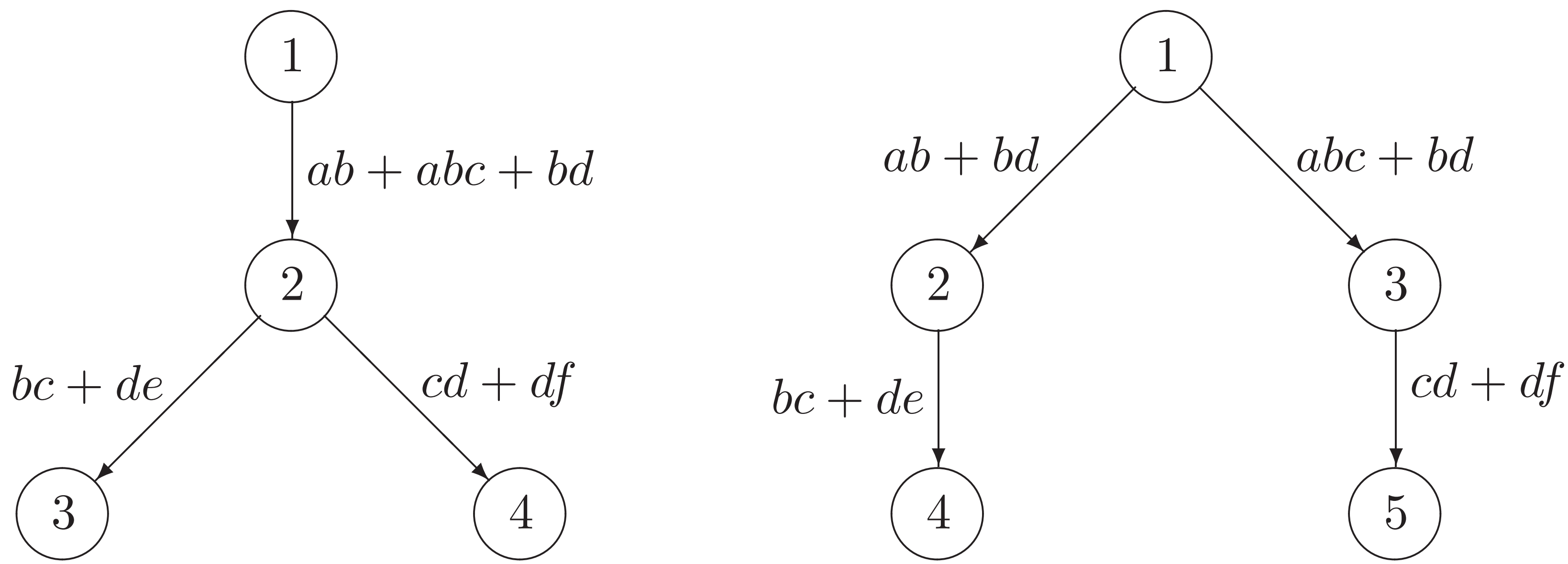


$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Largest bisimulations (example 2)

Let $ab, abc, bd, be, cd, de, df$ be elements of an algebra of paths (here, we denote composition by juxtaposition) and suppose that a, b, c, d, e, f are mutually disjoint and that

$$\ulcorner(ab) = \ulcorner(abc) = a, \quad \ulcorner(bc) = \ulcorner(bd) = b, \quad \ulcorner(cd) = c, \quad \ulcorner(de) = \ulcorner(df) = d \ .$$



$$\mathbf{S} = \begin{pmatrix} \epsilon + c + d + e + f & 0 & 0 & 0 & 0 \\ 0 & \epsilon + a + b + e + f & \epsilon + a + c + e + f & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Projections

The relations $\mathbf{P}_1, \mathbf{P}_2$ are called *conjugated projections* iff

$$\mathbf{P}_1^{\cup} \cdot \mathbf{P}_1 = \mathbf{1} \ , \qquad \mathbf{P}_2^{\cup} \cdot \mathbf{P}_2 = \mathbf{1} \ , \qquad \mathbf{P}_1 \cdot \mathbf{P}_1^{\cup} \ \sqcap \ \mathbf{P}_1 \cdot \mathbf{P}_1^{\cup} = \mathbf{1} \ , \qquad \mathbf{P}_1^{\cup} \cdot \mathbf{P}_2 = [\![1]\!]$$

(note: $\mathbf{P}_1^{\cup} \cdot \mathbf{P}_2 \neq \mathsf{T}$.) The *product* of $\mathbf{A}_1$ and $\mathbf{A}_2$ is

$$\mathbf{A}_1 \times \mathbf{A}_2 \ \ \triangleq \ \ \mathbf{P}_1 \cdot \mathbf{A}_1 \cdot \mathbf{P}_1^{\cup} \ \ \sqcap \ \ \mathbf{P}_2 \cdot \mathbf{A}_2 \cdot \mathbf{P}_2^{\cup} \ .$$

Projections (example)

$$\mathbf{P}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{P}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{P}_1^{\cup} \cdot \mathbf{P}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{P}_2^{\cup} \cdot \mathbf{P}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{P}_1 \cdot \mathbf{P}_1^{\cup} \ \sqcap \ \mathbf{P}_1 \cdot \mathbf{P}_1^{\cup} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{P}_1^{\cup} \cdot \mathbf{P}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \end{pmatrix} \qquad \mathbf{A}_2 = \begin{pmatrix} \mathbf{e} & \mathbf{f} & \mathbf{g} \\ \mathbf{h} & \mathbf{i} & \mathbf{j} \\ \mathbf{k} & \mathbf{l} & \mathbf{n} \end{pmatrix}$$

$$\mathbf{A}_1 \times \mathbf{A}_2 = \begin{pmatrix} \mathbf{a} \sqcap \mathbf{e} & \mathbf{a} \sqcap \mathbf{f} & \mathbf{a} \sqcap \mathbf{g} & \mathbf{b} \sqcap \mathbf{e} & \mathbf{b} \sqcap \mathbf{f} & \mathbf{b} \sqcap \mathbf{g} \\ \mathbf{a} \sqcap \mathbf{h} & \mathbf{a} \sqcap \mathbf{i} & \mathbf{a} \sqcap \mathbf{j} & \mathbf{b} \sqcap \mathbf{h} & \mathbf{b} \sqcap \mathbf{i} & \mathbf{b} \sqcap \mathbf{j} \\ \mathbf{a} \sqcap \mathbf{k} & \mathbf{a} \sqcap \mathbf{l} & \mathbf{a} \sqcap \mathbf{n} & \mathbf{b} \sqcap \mathbf{k} & \mathbf{b} \sqcap \mathbf{l} & \mathbf{b} \sqcap \mathbf{n} \\ \mathbf{c} \sqcap \mathbf{e} & \mathbf{c} \sqcap \mathbf{f} & \mathbf{c} \sqcap \mathbf{g} & \mathbf{d} \sqcap \mathbf{e} & \mathbf{d} \sqcap \mathbf{f} & \mathbf{d} \sqcap \mathbf{g} \\ \mathbf{c} \sqcap \mathbf{h} & \mathbf{c} \sqcap \mathbf{i} & \mathbf{c} \sqcap \mathbf{j} & \mathbf{d} \sqcap \mathbf{h} & \mathbf{d} \sqcap \mathbf{i} & \mathbf{d} \sqcap \mathbf{j} \\ \mathbf{c} \sqcap \mathbf{k} & \mathbf{c} \sqcap \mathbf{l} & \mathbf{c} \sqcap \mathbf{n} & \mathbf{d} \sqcap \mathbf{k} & \mathbf{d} \sqcap \mathbf{l} & \mathbf{d} \sqcap \mathbf{n} \end{pmatrix}$$

Conclusion

Potential application: controller synthesis

Various formulations of the problem (nonexhaustive list):

1. Given: an automaton G
   a language L such that $L \subseteq \mathcal{L}(G)$
   Find: a controller C (an automaton) such that $\mathcal{L}(G \times C) = L$

2. Given: an automaton G
   an automaton H such that $\mathcal{L}(H) \subseteq \mathcal{L}(G)$
   Find: a controller C such that $\mathcal{L}(G \times C) = \mathcal{L}(H)$

3. Given: an automaton G
   a modal logic formula ϕ
   Find: a controller C such that $G \times C$ satisfies ϕ

The solution may be trivial. E.g., for formulation 2, the solution is $C \triangleq H$.

Controllability and observability

The problem becomes interesting (and difficult) if some events (labels of G) are

- noncontrollable: C cannot prevent them, but may adjust its behavior according to their occurrence;
- nonobservable: C may prevent them, but cannot detect when they occur.

In this case, exact solutions need not exist. One then looks for extremal solutions to

$$\mathcal{L}(G \times C) \subseteq L \ .$$

Many variations of this problem are solved. However . . . (next slide).

Problems to solve

Many variations of the previous problem are solved. However:

1. combinatorial explosion is still a problem;
2. it is not always easy to understand the existing solutions, due to
   - heterogeneous objects: automata and modal formulae;
   - low-level algorithms;
3. the problem of decentralized control (having many cooperating controllers) is far from solved;
4. the problem of finding the least constraining controller C such that $G \times C$ simulates H is possibly not solved.