DM AAA : Automata, Algebra, Applications

Uli Fahrenberg Sven Dziadek Philipp Schlehuber Etienne Renault
Adrien Pommellet

July 5, 2022

1 Weighted automata

We are placing ourselves in the max-min semiring $S = (\mathbb{N} \cup \{\infty\}, \max, \min, 0, \infty)$.

1.1 Exercise

Give a detailed proof that S forms a semiring.

Solution

Trivial.

1.2 Exercise

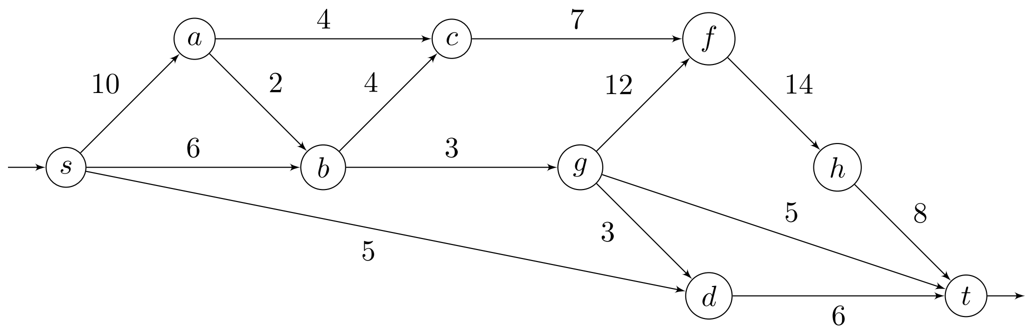
Describe automata weighted over S: What is the value of a path? What is the value of an automaton?

Solution

The value of a path is the minimum of its weights/capacities. The value of an automaton is the size of the thickest pipe.

1.3 Exercise

Let A be the following S-automaton:



What is $|A|$?

Solution

5, along the path $s \to d \to t$

1.4 Exercise

S-weighted automata are almost the same as maximum-flow problems, but not quite.

1. Seeing the automaton A from above as a maximum-flow problem, what is the maximum flow?
2. What precisely is the difference between maximum-flow problems and S-weighted automata? Is there a semiring S' such that maximum-flow problems can be posed as S'-weighted automata?

Solution

1. Use Ford-Fulkerson.
2. $S' = (\mathbb{N} \cup \{\infty\}, +, \min, 0, \infty)$, which is not a semiring because min does not distribute over $+$: for example, $\min(1, 1+1) \neq \min(1,1) + \min(1,1)$

1.5 Exercise

Prove that S is star-continuous and compute a^* for all $a \in S$.

Solution

$$\begin{aligned} a^* &= 1 + a + aa + aaa + \cdots \\ &= \max(\infty, a, \min(a,a), \min(a,a,a), \dots) \\ &= \infty \end{aligned}$$

for all $a \in S$, and infinite distributivity holds because ∞ is annihilating.

1.6 Exercise

Develop the matrix-star formula for a 2-by-2 matrix $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right] \in S^2$.

What is the $(2,3)$-component of the star of the matrix

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}?$$

Solution

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^* = \begin{bmatrix} (a+bd^*c)^* & (a+bd^*c)^*bd^* \\ (d+ca^*b)^*ca^* & (d+ca^*b)^* \end{bmatrix} = \begin{bmatrix} \infty & b \\ c & \infty \end{bmatrix}$$

Self-loops have infinite capacity.

In $S^{3\times 3}$ we have

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^* = \begin{bmatrix} \infty & b+ch & bf+c \\ d+fg & \infty & f+dc \\ hd+g & h+gb & \infty \end{bmatrix}.$$

That is, looking for example at the $(1,3)$-component, to get from state 1 to state 3 you go either directly (c) or over state 2 (bf).

In general, for the (i,j)-component of a matrix in $S^{n\times n}$, we add up all simple paths from i to j without ever visiting a state twice. This means that for the 4×4 matrix above, we have

$$M^*_{2,3} = g + ec + ho + edo + hmc.$$

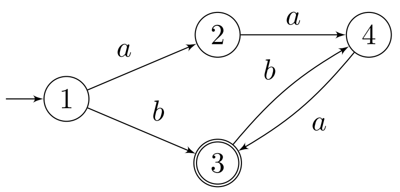
1.7 Exercise

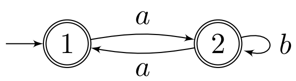
Write a program which implements the recursive matrix-star algorithm to compute M^* for an arbitrary square matrix over S. (Hint: Python, for example, has `math.inf` which should be useful here.) You can test your program on the automaton A from exercise 1.3.

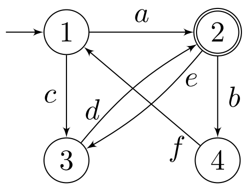
2 ω-Automata

2.1 Exercise

What are the languages of the following Büchi automata?

1. 

2. 

3. 

Solution

1. $(aa + bb)(ab)^\omega = (aaa + b)(ba)^\omega$
2. $a(aa + b)^\omega$
3. $\big((a + cd)(ed)^*bf\big)^\omega + \big((a + cd)(ed)^*bf\big)^*(a + cd)(ed)^\omega$

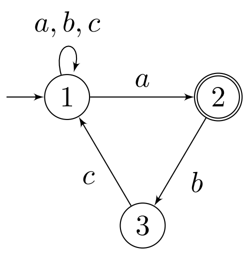
2.2 Exercise

Let

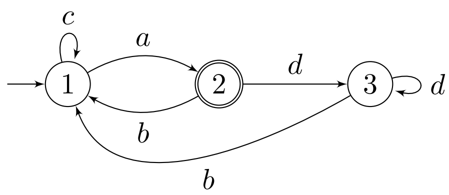
$$\begin{aligned} L_1 &= \{w \in \{a,b,c\}^\omega \mid w \text{ contains infinitely often the sequence } abc\} \\ &= \{w \in \{a,b,c\}^\omega \mid \{i \in \mathbb{N} \mid w_i = a, w_{i+1} = b, w_{i+2} = c\} \text{ is infinite}\} \end{aligned}$$

and $L_2 = (c^*ad^*b)^\omega$. Find Büchi automata A_1 and A_2 so that $L_1 = L(A_1)$ and $L_2 = L(A_2)$.

Solution

1. 

2.

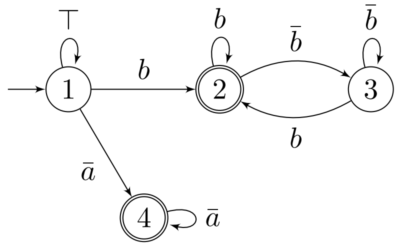


2.3 Exercise

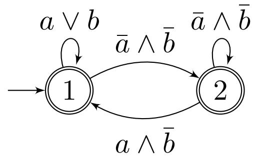
Let $AP = \{a, b\}$, and $\Sigma = 2^{AP}$ (the power set of AP).

For each Büchi automaton $\mathcal{A}$ below, give an LTL formula with $\mathcal{L}(\phi) = \{w \in \Sigma^\omega \mid w \models \phi\} = \mathcal{L}(\mathcal{A})$.

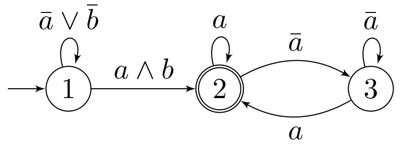
1.



2.



3.



Solution

1. $(GFa) \rightarrow (GFb)$
2. $G(Xb \rightarrow (a \vee b))$
3. $(GFa) \wedge (F(a \wedge b))$

2.4 Exercise

A Büchi automaton $\mathcal{A} = (Q, I, T, F)$ over Σ is called *deterministic* if $|I| \leq 1$, and for each state $q \in Q$ and symbol $a \in \Sigma$, we have $|\{(q, a, q') \in T \mid q' \in Q\}| \leq 1$.

Show that the class of languages recognizable by *deterministic* Büchi automata is closed under

1. intersection,
2. union.

Solution

1. The standard construction (not on the slides) works here:
For the deterministic Büchi automata $\mathcal{A}_1 = (Q_1, I_1, T_1, F_1)$ and $\mathcal{A}_2 = (Q_2, I_2, T_2, F_2)$, we construct
$$\mathcal{B} = (Q_1 \times Q_2 \times \{1, 2\}, I_1 \times I_2 \times \{1\}, T_\mathcal{B}, Q_1 \times F_2 \times \{2\})$$
with
$$\begin{aligned} T_\mathcal{B} = \{((q_1, q_2, i), \sigma, (q_1', q_2', j)) \mid \\ &(q_1, \sigma, q_1') \in T_1, (q_2, \sigma, q_2') \in T_2, i, j \in \{1, 2\}, \\ &q_i \notin F_i \to i = j \wedge q_i \in F_i \to i \notin j\}\,. \end{aligned}$$

2. Standard union does not work because it has multiple initial states in general. Try some product construction instead:
Let $\mathcal{A}_1 = (Q_1, I_1, T_1, F_1)$ and $\mathcal{A}_2 = (Q_2, I_2, T_2, F_2)$ be two deterministic Büchi automata. Assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are complete (just add a non-accepting dummy state). We construct
$$\mathcal{B} = (Q_1 \times Q_2, I_1 \times I_2, T_\mathcal{B}, F_\mathcal{B})$$
with
$$\begin{aligned} T_\mathcal{B} &= \{((q_1, q_2), \sigma, (q_1', q_2')) \mid \\ &\quad (q_1, \sigma, q_1') \in T_1, (q_2, \sigma, q_2') \in T_2\}\,, \\ F_\mathcal{B} &= \ (F_1 \times Q_2) \cup (Q_1 \times F_2)\,. \end{aligned}$$

Note: Interestingly, I believe this does not work in the (weighted) non-idempotent case because some runs that exist in both $\mathcal{A}_1$ and $\mathcal{A}_2$ can end up existing only once in $\mathcal{B}$.

3 Active Learning

3.1 Exercise

Apply the L^* algorithm table to the language $\mathcal{L}(\mathcal{A})$ accepted by the automaton $\mathcal{A}$ of Figure 1. Write the table, the various intermediary models, and the counter-examples to these models you have chosen.

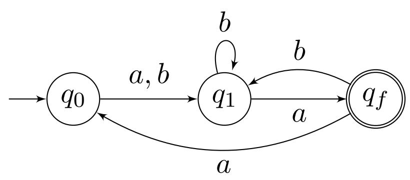
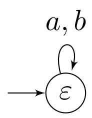


Figure 1: The automaton $\mathcal{A}$.

Solution

$\boldsymbol{P \cdot S}$	ε
ε	0
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε
$\varepsilon \cdot a$	0
$\varepsilon \cdot b$	0



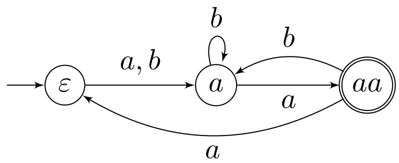
Not equivalent to $\mathcal{A}$, a counter-example is aa. We add a and aa to P.

$\boldsymbol{P \cdot S}$	ε
ε	0
a	0
aa	1
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε
$\varepsilon \cdot a$	0
$\varepsilon \cdot b$	0
$a \cdot a$	1
$a \cdot b$	0
$aa \cdot a$	0
$aa \cdot b$	0

The table is not consistent: ε and a are equivalent, but $\varepsilon \cdot a$ and $a \cdot a$ are not.

Thus we add a to S.

$\boldsymbol{P \cdot S}$	ε	a
ε	0	0
a	0	1
aa	1	0
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε	a
$\varepsilon \cdot a$	0	1
$\varepsilon \cdot b$	0	1
$a \cdot a$	1	0
$a \cdot b$	0	1
$aa \cdot a$	0	0
$aa \cdot b$	0	1



The last model is indeed equivalent to $\mathcal{A}$.

3.2 Exercise

Using active learning, find the minimal DFA accepting the language $\mathcal{L}_1$ of all words on $\Sigma = \{a, b\}$ with an odd number of a and an even number of b. Write the table, the various intermediary models, and the counter-examples to these models you have chosen.

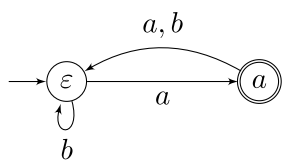
Solution

$\boldsymbol{P \cdot S}$	ε
ε	0
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε
$\varepsilon \cdot a$	1
$\varepsilon \cdot b$	0

The table is not closed.

We add a to P.

$\boldsymbol{P \cdot S}$	ε
ε	0
a	1
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε
$\varepsilon \cdot a$	1
$\varepsilon \cdot b$	0
$a \cdot a$	0
$a \cdot b$	0



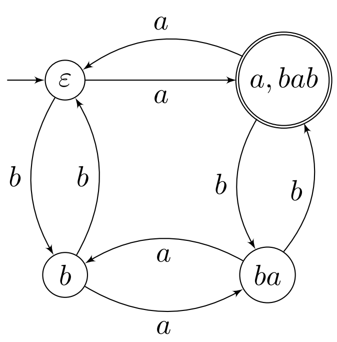
Does not accept $\mathcal{L}_1$, a counter-example is bab. We add b, ba and bab to P.

$\boldsymbol{P \cdot S}$	ε
ε	0
a	1
b	0
ba	0
bab	1
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε
$\varepsilon \cdot a$	1
$\varepsilon \cdot b$	0
$a \cdot a$	0
$a \cdot b$	0
$b \cdot a$	0
$b \cdot b$	0
$ba \cdot a$	0
$ba \cdot b$	1
$bab \cdot a$	0
$bab \cdot b$	0

The table is not consistent: ε and ba are equivalent but $\varepsilon \cdot a$ (resp. $\varepsilon \cdot b$) and $ba \cdot a$ (resp. $ba \cdot b$) are not.

Thus we add a and b to S.

$\boldsymbol{P \cdot S}$	ε	a	b
ε	0	1	0
a	1	0	0
b	0	0	0
ba	0	0	1
bab	1	0	0
$\boldsymbol{P \cdot \Sigma \cdot S}$	ε	a	b
$\varepsilon \cdot a$	1	0	0
$\varepsilon \cdot b$	0	0	0
$a \cdot a$	0	1	0
$a \cdot b$	0	0	1
$b \cdot a$	0	0	1
$b \cdot b$	0	1	0
$ba \cdot a$	0	0	0
$ba \cdot b$	1	0	0
$bab \cdot a$	0	1	0
$bab \cdot b$	0	0	1



The last model accepts $\mathcal{L}_2$.

3.3 Exercise

Using active learning, try to find the minimal DFA accepting the language $\mathcal{L}_2$ of all words on $\Sigma = \{a, b\}$ with the same number of a and b. Is it possible? How and why does the algorithm fail?

Solution

The L^* algorithm will never end, as there is an infinite number of distinguishable prefixes of the form $a^i b^j$ in $\mathcal{L}_2$. This is not a surprising result: $\mathcal{L}_2$ is context-free but not rational, and L^* was only designed to learn rational languages in the first place.