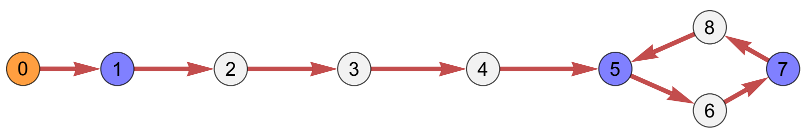
15-354: CDM	**K. Sutner**
Assignment 5	Due: **Oct. 4, 2024, 24:00**.

1. Minimization Algorithms (40)

Background

We have seen several minimization algorithms for DFAs. Their behavior is relatively easy to understand for tally languages $L \subseteq \{a\}^\star$. Here is a typical example of a lasso automaton $\mathcal{A}$ accepting a tally language.



Task

A. Trace the execution of Moore's algorithm on $\mathcal{A}$.

B. Trace the execution of Hopcroft's algorithm on $\mathcal{A}$.

C. Trace the execution of Brzozowski's algorithm on $\mathcal{A}$.

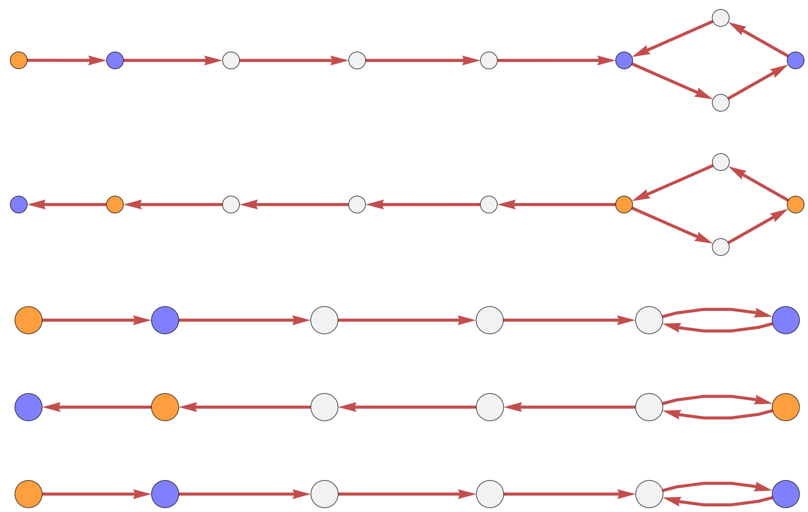
D. What can you say about the running time of these algorithms on a general lasso DFA with transient t and period p?

Solution: Minimization Algorithms

Part A: Moore

	0	1	2	3	4	5	6	7	8
0	0	1	0	0	0	1	0	1	0
1	0	1	2	2	0	1	0	1	0
2	0	1	2	3	0	5	0	5	0
3	0	1	2	3	4	5	4	5	4

Part B: Brzozowski



Part C: Hopcroft

blocks						C	$a^{-1}C$
$0, 2, 3, 4, 6, 8$	$1, 5, 7$					2	$0, 4, 6, 8$
$0, 4, 6, 8$	$1, 5, 7$	$2, 3$				3	$1, 2$
$0, 4, 6, 8$	$5, 7$	2	1	3		4	0
$4, 6, 8$	$5, 7$	2	1	3	0	3	2
$4, 6, 8$	$5, 7$	2	1	3	0	6	$-$

The critical block is given as an index into the current partition.

2. The UnEqual Language (40)

Background

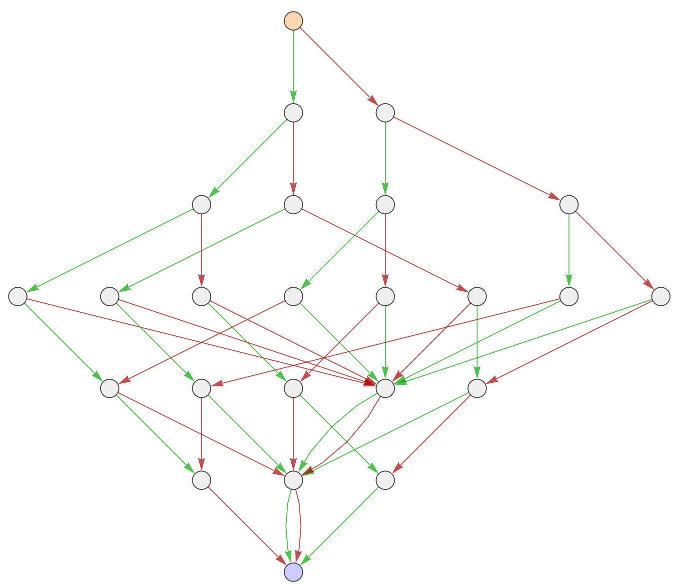
Consider the language of all strings of length $2k$ that are not of the form uu:

$$L_k = \{\, uv \in \{a,b\}^\star \mid |u| = |v| = k, u \neq v \,\}.$$

These languages are finite, hence trivially regular. The following table shows the state complexity of L_k up to $k = 6$.

k	1	2	3	4	5	6
sc	5	12	25	50	99	196

The minimal DFA for L_3 looks like so (the layout algorithm is not too great):



This is the PDFA without sink, the top state is initial and the bottom state final.

Task

A. What happens when you run Moore's algorithm on this DFA? How many rounds are there?

B. Determine all quotients for L_2.

C. Generalize. In particular explain the diagram for L_3.

D. Determine the state complexity of L_k.

E. Determine the state complexity of $K_k = \{\, uu \mid u \in \{a,b\}^k \,\}$.

Comment

From the diagram and the table it is not hard to conjecture a reasonable closed form for the state complexity. For a proof one can exploit the description of the minimal DFA in terms of quotients.

Solution: The Un-Equal Language

Part A: Moore

Initially, there are just two classes: the final state and everybody else (including the sink). In round 1, the 3 states above the final state split off, then the ones at the next higher level, and so on: in each round the next higher level in the graph becomes separated. In the last round, the sink also separates from everybody else.

Hence, after 6 rounds, all blocks are just singletons; the machine is already minimal.

Part B: Quotients L_2

The quotients of L_2 organized by length of the quotient string:

0	L_2
1	$(aab, aba, abb, baa, bba, bbb), (aaa, aab, abb, baa, bab, bba)$
2	$(ab, ba, bb), (aa, ba, bb), (aa, ab, bb), (aa, ab, ba)$
3	$(a), (b), (a, b)$
4	(ϵ)
5	$\emptyset$

Part C: Quotients

Ignoring the sink, states are organized in layers, at layer ℓ we have $x^{-1}L_k = P \subseteq \{a, b\}^{2k-\ell}$ where $|x| = \ell$. In particular for $\ell = k$ we have $u^{-1}L_k = \{a, b\}^k - \{u\}$ and the structure of the quotient automaton down to level k is a complete binary tree; the number of states in this part is $2^{k+1} - 1$.

The remainder of the machine has two kinds of states: those where a witness for inequality of u and v has already been found, and those where we are still waiting for such a witness.

Fix some $u \in \{a, b\}^k$. The first type of state is of the form $\delta(q_0, ux)$ where x is not a prefix of u and has behavior $\{a, b\}^{k-|x|}$, where $|x| \leq k$. The second type is of the form $\delta(q_0, ux)$ where x is a prefix of u and has behavior $\{a, b\}^{k-|x|} - \{y\}$ where $u = xy$. But then there are k states of the first type, and $2^k - 1$ states of the second type (these form another tree which is upside-down compared to the first, and has as root the sink of the DFA).

Part D: State Complexity

It follows from the analysis in part (B) that the state complexity of L_k is $3 \cdot 2^k + k - 2$.

Part E: Repetitions

The state complexity of K_k is $3 \cdot 2^k - 1$.

3. Syntactic Semigroups (40)

Background

The standard measure of complexity of a regular language L is the number of states of its minimal DFA M_L.

Recall the syntactic congruence $\equiv_L$ of a language L:

$$u \equiv_L v \iff \forall\, x, y \in \Sigma^\star \big(L(xuy) = L(xvy)\big)$$

The syntactic semigroup of L is the quotient $\Sigma^+/\equiv_L$ and is finite iff L is recognizable.

Computing the equivalence classes directly from the definition is quite cumbersome, but we can exploit the minimal DFA for L: $u \equiv_L v$ iff $\delta_u = \delta_v$.

Task

Fix the alphabet $\Sigma = \{a, b\}^\star$.

A. Compute the syntactic semigroup of $L = \{aab\}$ over alphabet $\{a, b\}$.

B. Compute the syntactic semigroup of $L = \{ab\}^\star$ over alphabet $\{a, b\}$.

C. Prove that $u \equiv_L v$ iff $\delta_u = \delta_v$.

Comment

Part (A) is pretty straightforward. For part (B), use whatever argument/trick you can come up with, the result is rather messy (there are 11 classes).

Solution: Syntactic Semigroups

Part A: *aab*

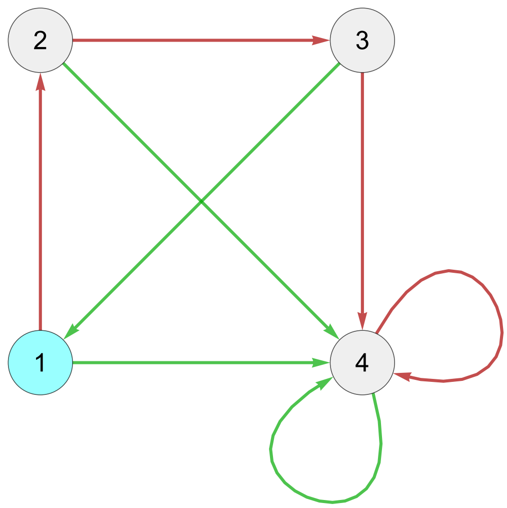
Here is the PDFA for L (add a sink 5 to get the minimal DFA).

By composition we get all transition functions:

x	δ_x
a	$2, 3, 5, 5, 5$
b	$5, 5, 4, 5, 5$
aa	$3, 5, 5, 5, 5$
ab	$5, 4, 5, 5, 5$
aab	$4, 5, 5, 5, 5$
ba	$5, 5, 5, 5, 5$
bb	$\delta_{aaa} = \delta_{aabs} = \delta_{ba}$

Hence all 5 factors of *aab* are equivalent only to themselves, all non-factors are equivalent.

Part B: *aab* Star



The possible transition functions here are

a	$2,3,4,4$	aab	$1,4,4,4$
b	$4,4,4,1$	aba	$4,2,4,4$
aa	$3,4,4,4$	baa	$4,4,4,3$
ab	$4,1,4,4$	$aaba$	$2,4,4,4$
ba	$4,4,4,2$	$abaa$	$4,3,4,4$
bb	$4,4,4,4$		

Write $L' = (aba)^\star$ and $L'' = (baa)^\star$ for the "rotations" of L. This produces the 11 equivalence classes

$$\begin{aligned} [a] &= \{a\} & [aba] &= abaL' \\ [b] &= bL & [baa] &= baaL'' \\ [aa] &= aaL'' & [aaba] &= aabaL' \\ [ab] &= abL & [abaa] &= abaaL'' \\ [ba] &= baL' & [bb] &= \text{everything else} \\ [aab] &= aabL \end{aligned}$$

Part C: Paths

Membership in L easily translates into path existence assertions in M_L:

$$\begin{aligned} u \equiv_L v &\iff \forall\, x, y \in \Sigma^\star \left(L(xuy) = L(xvy)\right) \\ &\iff \forall\, x, y \in \Sigma^\star \left(xuy \in [\![q_0]\!] \Leftrightarrow xvy \in [\![q_0]\!]\right) \\ &\iff \forall\, x, y \in \Sigma^\star \left(uy \in [\![\delta_x(q_0)]\!] \Leftrightarrow vy \in [\![\delta_x(q_0)]\!]\right) \end{aligned}$$

Using accessibility and reducedness:

$$\begin{aligned} &\iff \forall\, p \in Q, y \in \Sigma^\star \left(uy \in [\![p]\!] \Leftrightarrow vy \in [\![p]\!]\right) \\ &\iff \forall\, p \in Q \left([\![\delta_u(p)]\!] = [\![\delta_v(p)]\!]\right) \\ &\iff \delta_u = \delta_u \end{aligned}$$

Thus, two maps δ_u and δ_v in the transition semigroup agree iff $u \equiv_L v$.