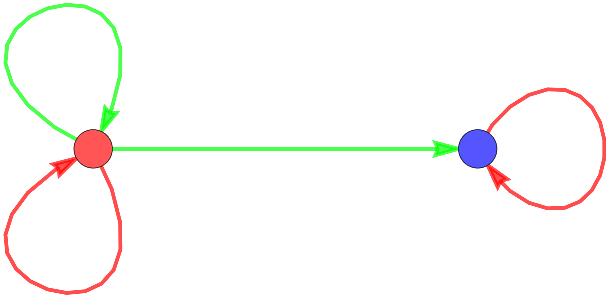
CDM

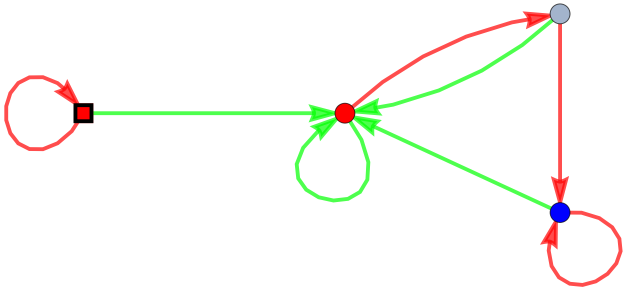
Safra's Algorithm

Klaus Sutner

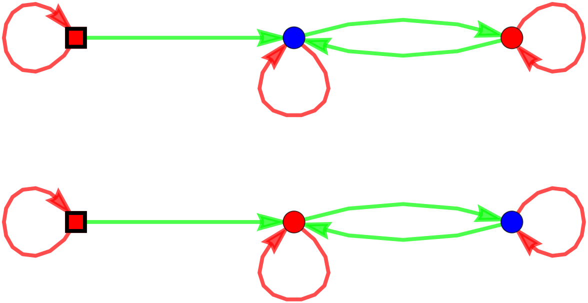
Carnegie Mellon University

Spring 2021

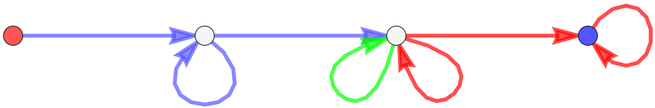




Note how an input a resets the machine. Branch first, 1 Rabin pair.

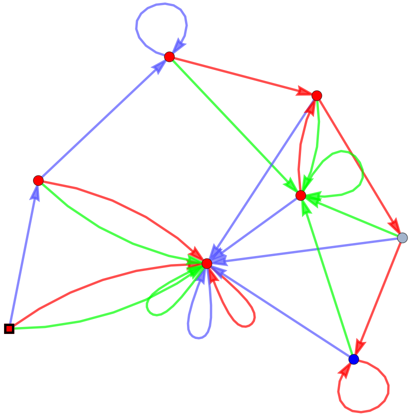


Transitions first, 2 Rabin pairs.

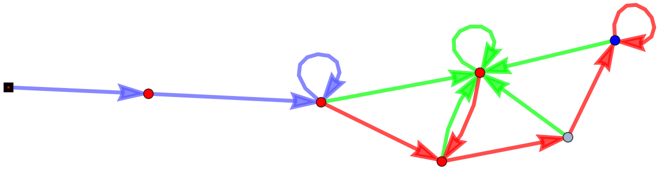


$L = ccc^{\star}(a+b)^{\star}aa^{\star}$

This could be managed by hand ...

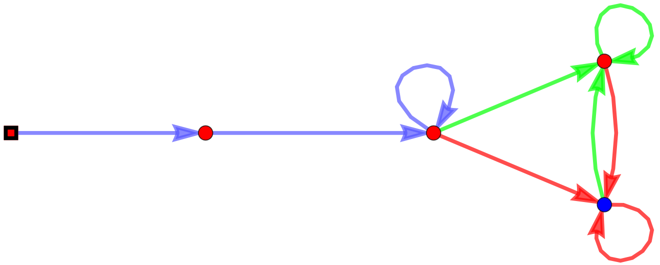


Branching first, 1 Rabin pair. Note the sink.



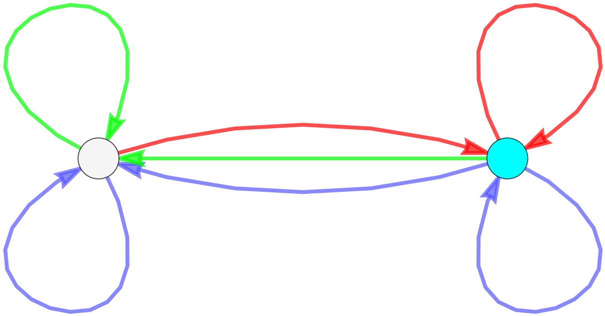
Without the sink.

b resets the right part of the automaton.

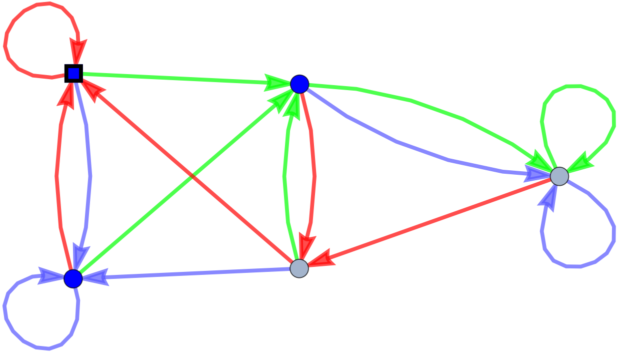


Transitions first, 1 Rabin pair. Clearly better.

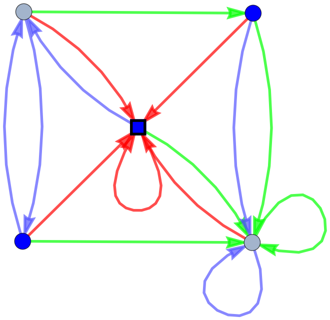
Figure out how to construct this machine from the previous one by state-merging.



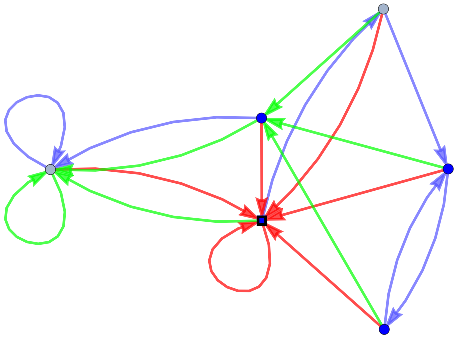
A surprisingly messy example from the Pin/Perrin book.



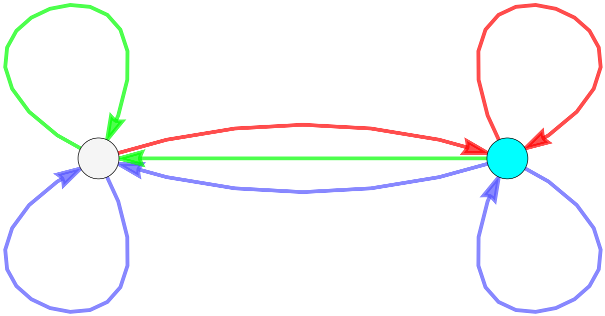
Branching first.



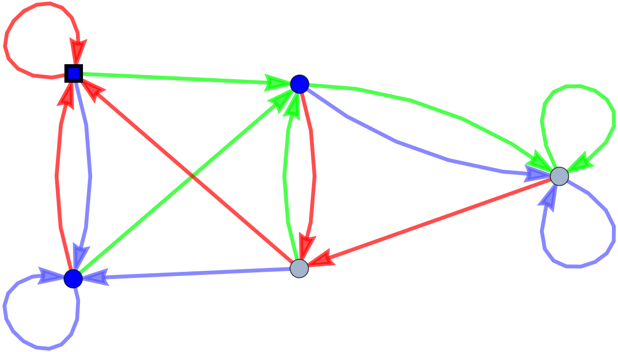
Transitions first. 2 Rabin pairs.



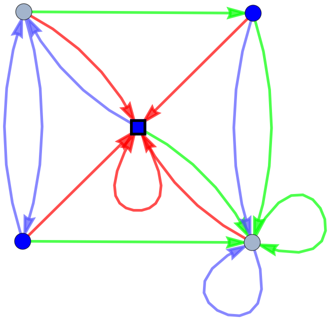
A different algorithm due to Muller and Schupp.



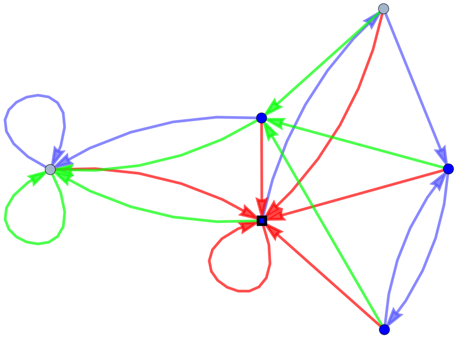
A surprisingly messy example from the Pin/Perrin book.



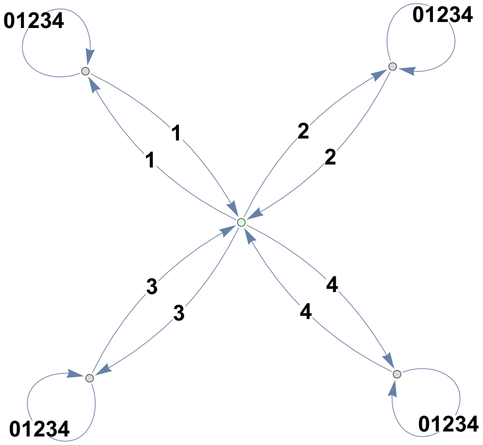
Branching first.



Transitions first. 2 Rabin pairs.

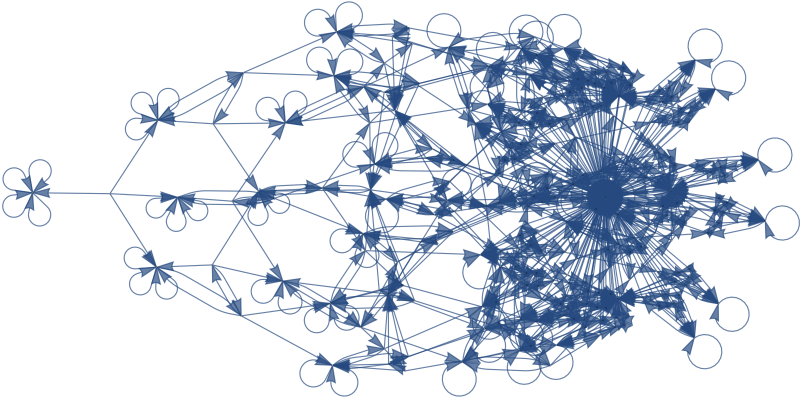


A different algorithm due to Muller and Schupp.

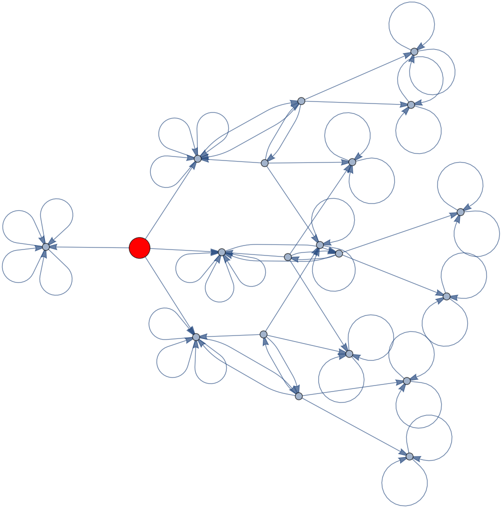


The Michel "infinite path" automaton for $k = 4$.

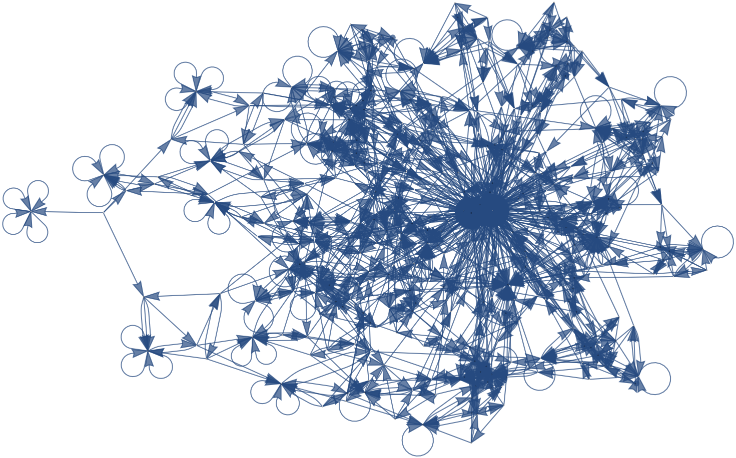
$k = 3$, Transitions First



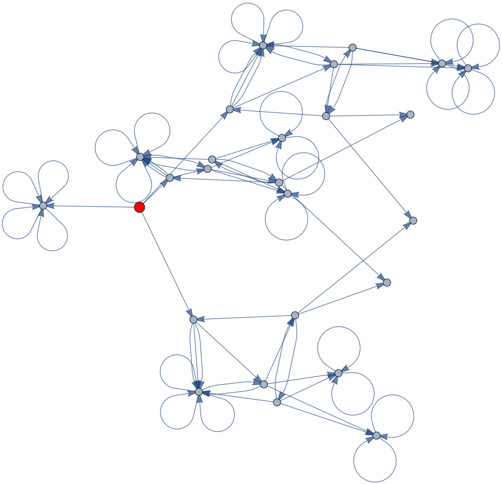
257 states—from a 4-state machine!



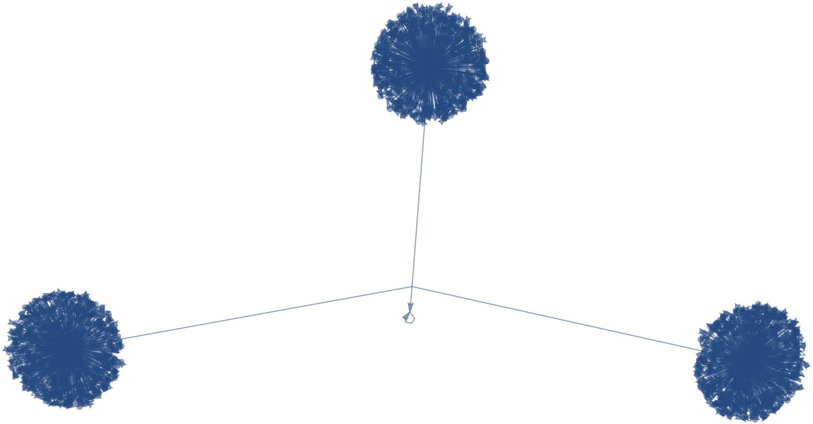
Note how the state on the left is a sink, as required.



Even worse, this time there are 385 states.



The underlying logic here is more complicated.



Muller-Schupp produces a machine with a whopping 13907 states.

The picture is the 2-neighborhood of state 1, containing 1919 states.

$n = 3$	385	$<$1 sec	branch first
	257	$<$1 sec	trans first
	13907	87 sec	Muller-Schupp
$n = 4$	13349	79 sec	branch first
	10369	72 sec	trans first

This is a straightforward implementation in Mathematica.