Synchronizing Finite Automata

Lecture IX. The Road Coloring Theorem (continued)

Mikhail Volkov

Ural Federal University / Hunter College

1. Recap

Which strongly connected digraphs may appear as underlying digraphs of synchronizing automata?

An obvious necessary condition:
all vertices should have the same out-degree.

A less obvious necessary condition called primitivity:
the g.c.d. of lengths of all cycles should be equal to 1.

The Road Coloring Conjecture claims that the two necessary conditions (constant out-degree and primitivity) are in fact sufficient. In other words: *every strongly connected primitive digraph with constant out-degree admits a synchronizing coloring.*

This was confirmed by Avraham Trahtman in August 2007 (published in: The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

1. Recap

Which strongly connected digraphs may appear as underlying digraphs of synchronizing automata?

An obvious necessary condition:
all vertices should have the same out-degree.

A less obvious necessary condition called primitivity:
the g.c.d. of lengths of all cycles should be equal to 1.

The Road Coloring Conjecture claims that the two necessary conditions (constant out-degree and primitivity) are in fact sufficient. In other words: *every strongly connected primitive digraph with constant out-degree admits a synchronizing coloring*.

This was confirmed by Avraham Trahtman in August 2007 (published in: The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

Which strongly connected digraphs may appear as underlying digraphs of synchronizing automata?

An obvious necessary condition:
all vertices should have the same out-degree.

A less obvious necessary condition called primitivity:
the g.c.d. of lengths of all cycles should be equal to 1.

The Road Coloring Conjecture claims that the two necessary conditions (constant out-degree and primitivity) are in fact sufficient. In other words: *every strongly connected primitive digraph with constant out-degree admits a synchronizing coloring.*

This was confirmed by Avraham Trahtman in August 2007 (published in: The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

Which strongly connected digraphs may appear as underlying digraphs of synchronizing automata?

An obvious necessary condition:
all vertices should have the same out-degree.

A less obvious necessary condition called primitivity:
the g.c.d. of lengths of all cycles should be equal to 1.

The Road Coloring Conjecture claims that the two necessary conditions (constant out-degree and primitivity) are in fact sufficient. In other words: *every strongly connected primitive digraph with constant out-degree admits a synchronizing coloring*.

This was confirmed by Avraham Trahtman in August 2007 (published in: The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

1. Recap

Which strongly connected digraphs may appear as underlying digraphs of synchronizing automata?

An obvious necessary condition:
all vertices should have the same out-degree.

A less obvious necessary condition called primitivity:
the g.c.d. of lengths of all cycles should be equal to 1.

The Road Coloring Conjecture claims that the two necessary conditions (constant out-degree and primitivity) are in fact sufficient. In other words: *every strongly connected primitive digraph with constant out-degree admits a synchronizing coloring*.

This was confirmed by Avraham Trahtman in August 2007 (published in: The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

Which strongly connected digraphs may appear as underlying digraphs of synchronizing automata?

An obvious necessary condition:
all vertices should have the same out-degree.

A less obvious necessary condition called primitivity:
the g.c.d. of lengths of all cycles should be equal to 1.

The Road Coloring Conjecture claims that the two necessary conditions (constant out-degree and primitivity) are in fact sufficient. In other words: *every strongly connected primitive digraph with constant out-degree admits a synchronizing coloring*.

This was confirmed by Avraham Trahtman in August 2007 (published in: The Road Coloring Problem, Israel J. Math. 172 (2009) 51–60).

2. Stability

Trahtman's proof heavily depends on a neat idea of stability (or confluence) which is due to Karel Culik II, Juhani Karhumäki and Jarkko Kari (A note on synchronized automata and Road Coloring Problem, Int. J. Found. Comput. Sci., 13 (2002) 459–471). Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA. We define the relation $\sim$ on Q as follows:

$$q \sim q' \iff \forall u \in \Sigma^* \; \exists v \in \Sigma^* \; q \, . \, uv = q' . uv .$$

$\sim$ is called the *stability relation* and any pair (q, q') such that $q \sim q'$ is called *stable*.

A coloring of a digraph with constant out-degree is *stable* if the resulting automaton has at least one stable pair (q, q') with $q \neq q'$.

Proposition CKK. *Suppose every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Then the Road Coloring Conjecture holds true.*

2. Stability

Trahtman's proof heavily depends on a neat idea of stability (or confluence) which is due to Karel Culik II, Juhani Karhumäki and Jarkko Kari (A note on synchronized automata and Road Coloring Problem, Int. J. Found. Comput. Sci., 13 (2002) 459–471). Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA. We define the relation $\sim$ on Q as follows:

$$q \sim q' \iff \forall u \in \Sigma^* \ \exists v \in \Sigma^* \ q \,.\, uv = q'.uv.$$

$\sim$ is called the *stability relation* and any pair (q, q') such that $q \sim q'$ is called *stable*.

A coloring of a digraph with constant out-degree is *stable* if the resulting automaton has at least one stable pair (q, q') with $q \neq q'$.

Proposition CKK. *Suppose every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Then the Road Coloring Conjecture holds true.*

2. Stability

Trahtman's proof heavily depends on a neat idea of stability (or confluence) which is due to Karel Culik II, Juhani Karhumäki and Jarkko Kari (A note on synchronized automata and Road Coloring Problem, Int. J. Found. Comput. Sci., 13 (2002) 459–471). Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA. We define the relation $\sim$ on Q as follows:

$$q \sim q' \iff \forall u \in \Sigma^* \ \exists v \in \Sigma^* \ q \, . \, uv = q'.uv.$$

$\sim$ is called the *stability relation* and any pair (q, q') such that $q \sim q'$ is called *stable*.

A coloring of a digraph with constant out-degree is *stable* if the resulting automaton has at least one stable pair (q, q') with $q \neq q'$.

Proposition CKK. *Suppose every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Then the Road Coloring Conjecture holds true.*

2. Stability

Trahtman's proof heavily depends on a neat idea of stability (or confluence) which is due to Karel Culik II, Juhani Karhumäki and Jarkko Kari (A note on synchronized automata and Road Coloring Problem, Int. J. Found. Comput. Sci., 13 (2002) 459–471). Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA. We define the relation $\sim$ on Q as follows:

$$q \sim q' \iff \forall u \in \Sigma^* \ \exists v \in \Sigma^* \ q \,.\, uv = q'.uv.$$

$\sim$ is called the *stability relation* and any pair (q, q') such that $q \sim q'$ is called *stable*.

A coloring of a digraph with constant out-degree is *stable* if the resulting automaton has at least one stable pair (q, q') with $q \neq q'$.

Proposition CKK. *Suppose every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Then the Road Coloring Conjecture holds true.*

2. Stability

Trahtman's proof heavily depends on a neat idea of stability (or confluence) which is due to Karel Culik II, Juhani Karhumäki and Jarkko Kari (A note on synchronized automata and Road Coloring Problem, Int. J. Found. Comput. Sci., 13 (2002) 459–471). Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA. We define the relation $\sim$ on Q as follows:

$$q \sim q' \iff \forall u \in \Sigma^* \ \exists v \in \Sigma^* \ q\,.\,uv = q'.uv.$$

$\sim$ is called the *stability relation* and any pair (q, q') such that $q \sim q'$ is called *stable*.

A coloring of a digraph with constant out-degree is *stable* if the resulting automaton has at least one stable pair (q, q') with $q \neq q'$.

Proposition CKK. *Suppose every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Then the Road Coloring Conjecture holds true.*

Trahtman has managed to prove exactly what was needed to use Proposition CKK: every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Thus, Road Coloring Conjecture holds true.

The proof is clever but not too difficult.

For brevity, we call strongly connected primitive digraphs with constant out-degree and more than 1 vertex admissible.

Trahtman has managed to prove exactly what was needed to use Proposition CKK: every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Thus, Road Coloring Conjecture holds true.

The proof is clever but not too difficult.
For brevity, we call strongly connected primitive digraphs with constant out-degree and more than 1 vertex admissible.

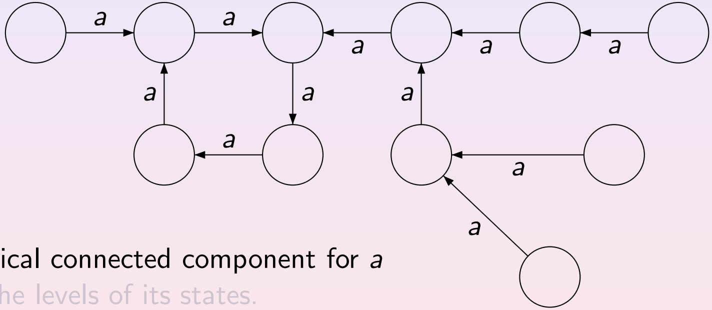
Trahtman has managed to prove exactly what was needed to use Proposition CKK: every strongly connected primitive digraph with constant out-degree and more than 1 vertex has a stable coloring. Thus, Road Coloring Conjecture holds true.

The proof is clever but not too difficult.
For brevity, we call strongly connected primitive digraphs with constant out-degree and more than 1 vertex admissible.

Let $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ be a DFA, $a \in \Sigma$. We want to assign to its states a parameter called the level w.r.t. a.

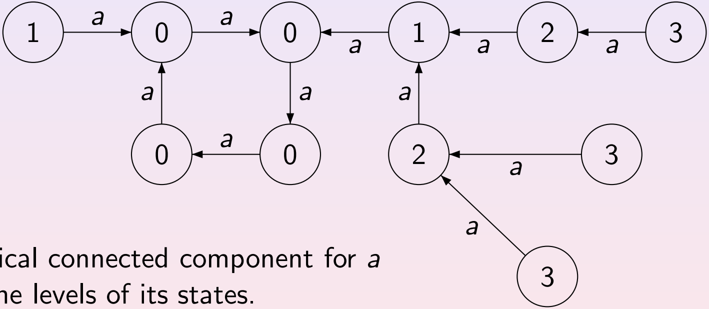
A typical connected component for a and the levels of its states.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA, $a \in \Sigma$. We want to assign to its states a parameter called the level w.r.t. a.



A typical connected component for a
and the levels of its states.

Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a DFA, $a \in \Sigma$. We want to assign to its states a parameter called the level w.r.t. a.



A typical connected component for a and the levels of its states.

Lemma 2. *Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a strongly connected automaton such that all states of maximal level $L > 0$ w.r.t. $a \in \Sigma$ belong to the same tree. Then $\mathscr{A}$ has a stable pair.*

By Lemma 2, to prove that every admissible digraph Γ has a stable coloring, it suffices to show that every such Γ may be colored into an automaton satisfying the premise of the lemma.

Take an arbitrary admissible digraph Γ. We start with an arbitrary coloring of Γ, take an arbitrary color (=letter) a, and induct on the number N of states that do not lie on any a-cycle in the initial coloring.

Lemma 2. *Let $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ be a strongly connected automaton such that all states of maximal level $L > 0$ w.r.t. $a \in \Sigma$ belong to the same tree. Then $\mathscr{A}$ has a stable pair.*

By Lemma 2, to prove that every admissible digraph Γ has a stable coloring, it suffices to show that every such Γ may be colored into an automaton satisfying the premise of the lemma.

Take an arbitrary admissible digraph Γ. We start with an arbitrary coloring of Γ, take an arbitrary color (=letter) a, and induct on the number N of states that do not lie on any a-cycle in the initial coloring.

5. Lemma on Level and Induction

Lemma 2. *Let* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *be a strongly connected automaton such that all states of maximal level* $L > 0$ *w.r.t.* $a \in \Sigma$ *belong to the same tree. Then* $\mathscr{A}$ *has a stable pair.*

By Lemma 2, to prove that every admissible digraph Γ has a stable coloring, it suffices to show that every such Γ may be colored into an automaton satisfying the premise of the lemma.

Take an arbitrary admissible digraph Γ. We start with an arbitrary coloring of Γ, take an arbitrary color (=letter) a, and induct on the number N of states that do not lie on any a-cycle in the initial coloring.

6. Induction Basis

Suppose that $N = 0$. This means that all states lie on a-cycles.

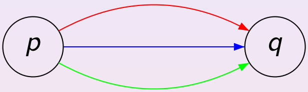
We say that a vertex p of Γ is a bunch if all edges that begin at p lead to the same vertex q.

If all vertices in Γ are bunches, then there is just one a-cycle (since Γ is strongly connected) and all cycles in Γ have the same length. This contradicts the assumption that Γ is primitive.

It is quite interesting that this is the only place in the whole proof where the primitivity condition is invoked.

6. Induction Basis

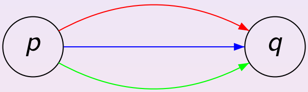
Suppose that $N = 0$. This means that all states lie on a-cycles. We say that a vertex p of Γ is a bunch if all edges that begin at p lead to the same vertex q.



If all vertices in Γ are bunches, then there is just one a-cycle (since Γ is strongly connected) and all cycles in Γ have the same length. This contradicts the assumption that Γ is primitive.
It is quite interesting that this is the only place in the whole proof where the primitivity condition is invoked.

6. Induction Basis

Suppose that $N = 0$. This means that all states lie on a-cycles. We say that a vertex p of Γ is a bunch if all edges that begin at p lead to the same vertex q.

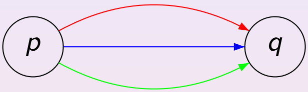


If all vertices in Γ are bunches, then there is just one a-cycle (since Γ is strongly connected) and all cycles in Γ have the same length. This contradicts the assumption that Γ is primitive.

It is quite interesting that this is the only place in the whole proof where the primitivity condition is invoked.

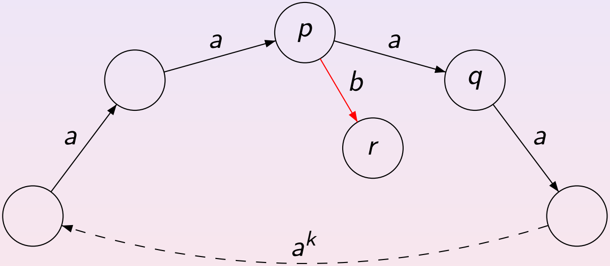
6. Induction Basis

Suppose that $N = 0$. This means that all states lie on a-cycles.
We say that a vertex p of Γ is a bunch if all edges that begin at p lead to the same vertex q.



If all vertices in Γ are bunches, then there is just one a-cycle (since Γ is strongly connected) and all cycles in Γ have the same length.
This contradicts the assumption that Γ is primitive.
It is quite interesting that this is the only place in the whole proof where the primitivity condition is invoked.

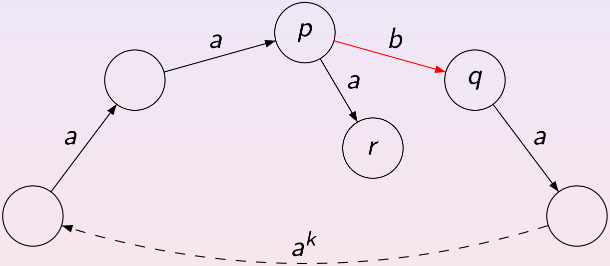
Thus, let p be a state which is not a bunch, let $q = p\,.\,a$ and let $b \neq a$ be such that $r = p\,.\,b \neq q$. We exchange the labels of the edges $p \xrightarrow{a} q$ and $p \xrightarrow{b} r$.



It is clear that in the new coloring there is only one state of maximal level w.r.t. a, namely q. Thus, the induction basis is verified.

7. Induction Basis

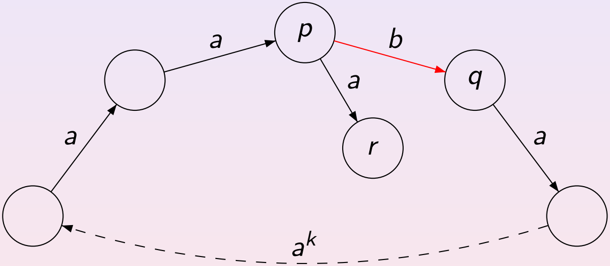
Thus, let p be a state which is not a bunch, let $q = p\,.\,a$ and let $b \neq a$ be such that $r = p\,.\,b \neq q$. We exchange the labels of the edges $p \xrightarrow{a} q$ and $p \xrightarrow{b} r$.



It is clear that in the new coloring there is only one state of maximal level w.r.t. a, namely q. Thus, the induction basis is verified.

7. Induction Basis

Thus, let p be a state which is not a bunch, let $q = p\,.\,a$ and let $b \neq a$ be such that $r = p\,.\,b \neq q$. We exchange the labels of the edges $p \xrightarrow{a} q$ and $p \xrightarrow{b} r$.



It is clear that in the new coloring there is only one state of maximal level w.r.t. a, namely q. Thus, the induction basis is verified.

Now let $N > 0$. We denote by L the maximum level of the states w.r.t. a in the initial coloring. Observe that $N > 0$ implies $L > 0$. Let p be a state of level L. Since Γ is strongly connected, there is an edge $p' \to p$ with $p' \neq p$, and by the choice of p, the label of this edge is $b \neq a$. Let $t = p' \,.\, a$. One has $t \neq p$. Let $r = p \,.\, a^L$ and let C be the a-cycle on which r lies.

The following considerations split in several cases. In each case except one we can recolor Γ by swapping the labels of two edges such the new coloring either satisfies the premise of Lemma 2 (all states of maximal level w.r.t. a belong to the same tree) or has more states on the a-cycles (and the induction assumption applies). The remaining case turns out to be easy.

Now let $N > 0$. We denote by L the maximum level of the states w.r.t. a in the initial coloring. Observe that $N > 0$ implies $L > 0$. Let p be a state of level L. Since Γ is strongly connected, there is an edge $p' \to p$ with $p' \neq p$, and by the choice of p, the label of this edge is $b \neq a$. Let $t = p' \,.\, a$. One has $t \neq p$. Let $r = p \,.\, a^L$ and let C be the a-cycle on which r lies.

The following considerations split in several cases. In each case except one we can recolor Γ by swapping the labels of two edges such the new coloring either satisfies the premise of Lemma 2 (all states of maximal level w.r.t. a belong to the same tree) or has more states on the a-cycles (and the induction assumption applies). The remaining case turns out to be easy.

8. Induction Step

Now let $N > 0$. We denote by L the maximum level of the states w.r.t. a in the initial coloring. Observe that $N > 0$ implies $L > 0$. Let p be a state of level L. Since Γ is strongly connected, there is an edge $p' \to p$ with $p' \neq p$, and by the choice of p, the label of this edge is $b \neq a$. Let $t = p' \,.\, a$. One has $t \neq p$. Let $r = p \,.\, a^L$ and let C be the a-cycle on which r lies.

The following considerations split in several cases. In each case except one we can recolor Γ by swapping the labels of two edges such the new coloring either satisfies the premise of Lemma 2 (all states of maximal level w.r.t. a belong to the same tree) or has more states on the a-cycles (and the induction assumption applies). The remaining case turns out to be easy.

8. Induction Step

Now let $N > 0$. We denote by L the maximum level of the states w.r.t. a in the initial coloring. Observe that $N > 0$ implies $L > 0$. Let p be a state of level L. Since Γ is strongly connected, there is an edge $p' \to p$ with $p' \neq p$, and by the choice of p, the label of this edge is $b \neq a$. Let $t = p' \,.\, a$. One has $t \neq p$. Let $r = p \,.\, a^L$ and let C be the a-cycle on which r lies.

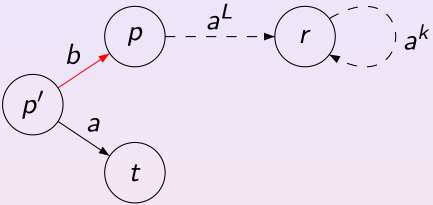
The following considerations split in several cases. In each case except one we can recolor Γ by swapping the labels of two edges such the new coloring either satisfies the premise of Lemma 2 (all states of maximal level w.r.t. a belong to the same tree) or has more states on the a-cycles (and the induction assumption applies). The remaining case turns out to be easy.

8. Induction Step

Now let $N > 0$. We denote by L the maximum level of the states w.r.t. a in the initial coloring. Observe that $N > 0$ implies $L > 0$. Let p be a state of level L. Since Γ is strongly connected, there is an edge $p' \to p$ with $p' \neq p$, and by the choice of p, the label of this edge is $b \neq a$. Let $t = p' \,.\, a$. One has $t \neq p$. Let $r = p \,.\, a^L$ and let C be the a-cycle on which r lies.

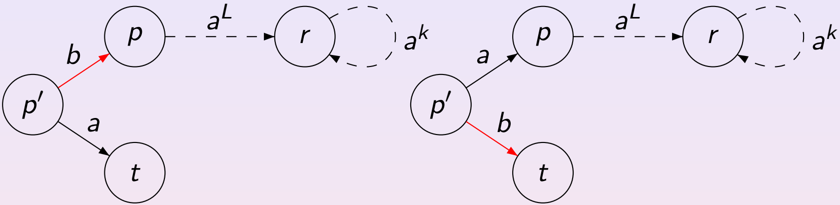
The following considerations split in several cases. In each case except one we can recolor Γ by swapping the labels of two edges such the new coloring either satisfies the premise of Lemma 2 (all states of maximal level w.r.t. a belong to the same tree) or has more states on the a-cycles (and the induction assumption applies). The remaining case turns out to be easy.

Case 1: p' is not on C.



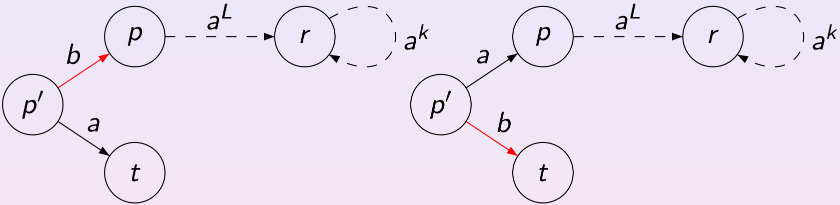
We swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If p' was on the a-path from p to r, then the swapping creates a new a-cycle increasing the number of states on the a-cycles. If p' was not on the a-path from p to r, then the level of p' w.r.t. a becomes $L+1$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of p' and thus belong to the same tree.

Case 1: p' is not on C.



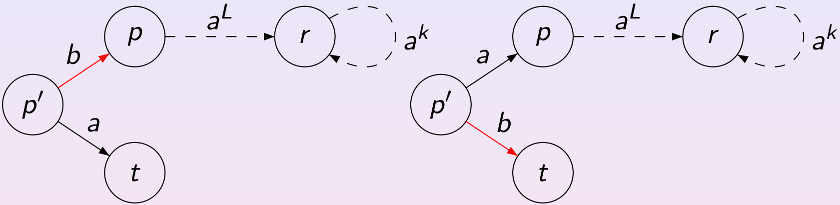
We swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If p' was on the a-path from p to r, then the swapping creates a new a-cycle increasing the number of states on the a-cycles. If p' was not on the a-path from p to r, then the level of p' w.r.t. a becomes $L+1$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of p' and thus belong to the same tree.

Case 1: p' is not on C.



We swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If p' was on the a-path from p to r, then the swapping creates a new a-cycle increasing the number of states on the a-cycles. If p' was not on the a-path from p to r, then the level of p' w.r.t. a becomes $L+1$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of p' and thus belong to the same tree.

Case 1: p' is not on C.



We swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If p' was on the a-path from p to r, then the swapping creates a new a-cycle increasing the number of states on the a-cycles. If p' was not on the a-path from p to r, then the level of p' w.r.t. a becomes $L+1$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of p' and thus belong to the same tree.

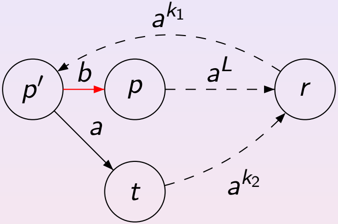
10. Induction Step: Subcase 2.1

Case 2: p' is on C. Let k_1 be the least integer such that $r \,.\, a^{k_1} = p'$. The state $t = p' \,.\, a$ is also on C. Let k_2 be the least integer such that $t \,.\, a^{k_2} = r$. Then the length of C is $k_1 + k_2 + 1$.

Subcase 2.1: $k_2 \neq L$. Again, we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If $k_2 < L$, then the swapping creates an a-cycle of length $k_1 + L + 1 > k_1 + k_2 + 1$ increasing the number of states on the a-cycles. If $k_2 > L$, then the level of t w.r.t. a becomes k_2 whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of t and thus belong to the same tree.

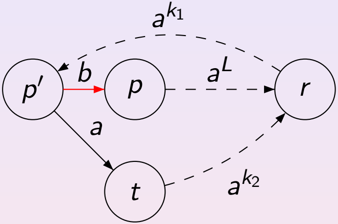
10. Induction Step: Subcase 2.1

Case 2: p' is on C. Let k_1 be the least integer such that $r\,.\,a^{k_1} = p'$. The state $t = p'\,.\,a$ is also on C. Let k_2 be the least integer such that $t\,.\,a^{k_2} = r$. Then the length of C is $k_1 + k_2 + 1$.



Subcase 2.1: $k_2 \neq L$. Again, we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If $k_2 < L$, then the swapping creates an a-cycle of length $k_1 + L + 1 > k_1 + k_2 + 1$ increasing the number of states on the a-cycles. If $k_2 > L$, then the level of t w.r.t. a becomes k_2 whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of t and thus belong to the same tree.

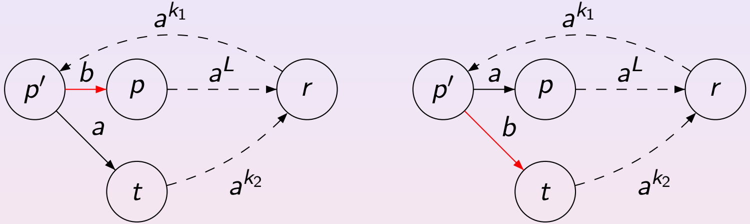
Case 2: p' is on C. Let k_1 be the least integer such that $r \,.\, a^{k_1} = p'$. The state $t = p' \,.\, a$ is also on C. Let k_2 be the least integer such that $t \,.\, a^{k_2} = r$. Then the length of C is $k_1 + k_2 + 1$.



Subcase 2.1: $k_2 \neq L$. Again, we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If $k_2 < L$, then the swapping creates an a-cycle of length $k_1 + L + 1 > k_1 + k_2 + 1$ increasing the number of states on the a-cycles. If $k_2 > L$, then the level of t w.r.t. a becomes k_2 whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of t and thus belong to the same tree.

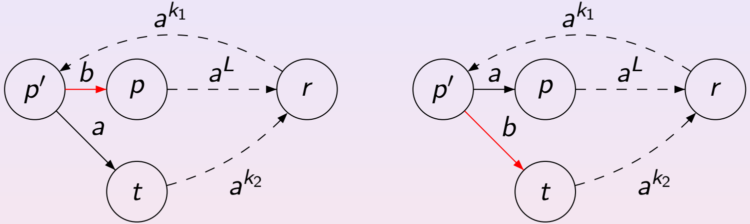
10. Induction Step: Subcase 2.1

Case 2: p' is on C. Let k_1 be the least integer such that $r \,.\, a^{k_1} = p'$. The state $t = p' \,.\, a$ is also on C. Let k_2 be the least integer such that $t \,.\, a^{k_2} = r$. Then the length of C is $k_1 + k_2 + 1$.



Subcase 2.1: $k_2 \neq L$. Again, we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If $k_2 < L$, then the swapping creates an a-cycle of length $k_1 + L + 1 > k_1 + k_2 + 1$ increasing the number of states on the a-cycles. If $k_2 > L$, then the level of t w.r.t. a becomes k_2 whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of t and thus belong to the same tree.

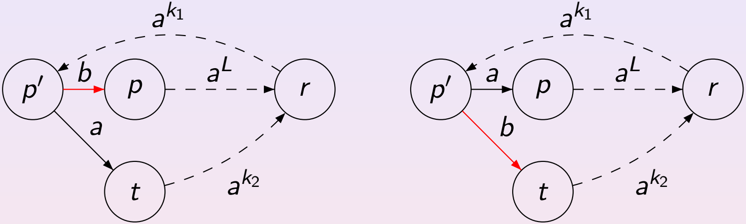
Case 2: p' is on C. Let k_1 be the least integer such that $r\,.\,a^{k_1} = p'$. The state $t = p'\,.\,a$ is also on C. Let k_2 be the least integer such that $t\,.\,a^{k_2} = r$. Then the length of C is $k_1 + k_2 + 1$.



Subcase 2.1: $k_2 \neq L$. Again, we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If $k_2 < L$, then the swapping creates an a-cycle of length $k_1 + L + 1 > k_1 + k_2 + 1$ increasing the number of states on the a-cycles. If $k_2 > L$, then the level of t w.r.t. a becomes k_2 whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of t and thus belong to the same tree.

10. Induction Step: Subcase 2.1

Case 2: p' is on C. Let k_1 be the least integer such that $r \,.\, a^{k_1} = p'$. The state $t = p' \,.\, a$ is also on C. Let k_2 be the least integer such that $t \,.\, a^{k_2} = r$. Then the length of C is $k_1 + k_2 + 1$.



Subcase 2.1: $k_2 \neq L$. Again, we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$. If $k_2 < L$, then the swapping creates an a-cycle of length $k_1 + L + 1 > k_1 + k_2 + 1$ increasing the number of states on the a-cycles. If $k_2 > L$, then the level of t w.r.t. a becomes k_2 whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of t and thus belong to the same tree.

11. Induction Step: Subcase 2.2

Let s be the state of C such that $s\,.\,a = r$.

Subcase 2.2: $k_2 = L$ and s is not a bunch. Since s is not a bunch, there is a letter c such that $s' = s\,.\,c \neq r$.

We swap the labels of $s \xrightarrow{c} s'$ and $s \xrightarrow{a} r$. If r still lies on an a-cycle, then the length of the a-cycle is at least $k_1 + k_2 + 2$ and the number of states on the a-cycles increases. Otherwise, the level of r w.r.t. a becomes at least $k_1 + k_2 + 1 > L$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of r and belong to the same tree.

11. Induction Step: Subcase 2.2

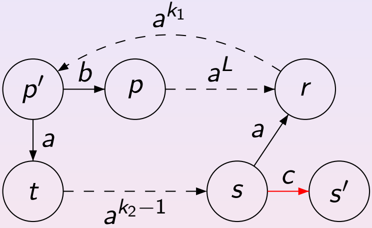
Let s be the state of C such that $s\,.\,a = r$.

Subcase 2.2: $k_2 = L$ and s is not a bunch. Since s is not a bunch, there is a letter c such that $s' = s\,.\,c \neq r$.

We swap the labels of $s \xrightarrow{c} s'$ and $s \xrightarrow{a} r$. If r still lies on an a-cycle, then the length of the a-cycle is at least $k_1 + k_2 + 2$ and the number of states on the a-cycles increases. Otherwise, the level of r w.r.t. a becomes at least $k_1 + k_2 + 1 > L$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of r and belong to the same tree.

Let s be the state of C such that $s\,.\,a = r$.

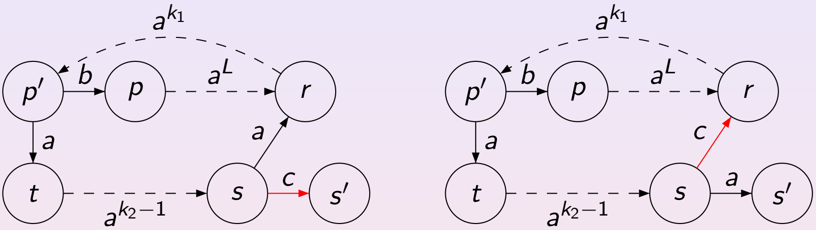
Subcase 2.2: $k_2 = L$ **and** s **is not a bunch.** Since s is not a bunch, there is a letter c such that $s' = s\,.\,c \neq r$.



We swap the labels of $s \xrightarrow{c} s'$ and $s \xrightarrow{a} r$. If r still lies on an a-cycle, then the length of the a-cycle is at least $k_1 + k_2 + 2$ and the number of states on the a-cycles increases. Otherwise, the level of r w.r.t. a becomes at least $k_1 + k_2 + 1 > L$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of r and belong to the same tree.

Let s be the state of C such that $s\,.\,a = r$.

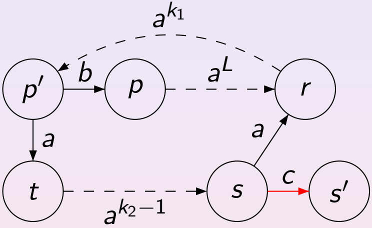
Subcase 2.2: $k_2 = L$ and s is not a bunch. Since s is not a bunch, there is a letter c such that $s' = s\,.\,c \neq r$.



We swap the labels of $s \xrightarrow{c} s'$ and $s \xrightarrow{a} r$. If r still lies on an a-cycle, then the length of the a-cycle is at least $k_1 + k_2 + 2$ and the number of states on the a-cycles increases. Otherwise, the level of r w.r.t. a becomes at least $k_1 + k_2 + 1 > L$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of r and belong to the same tree.

Let s be the state of C such that $s\,.\,a = r$.

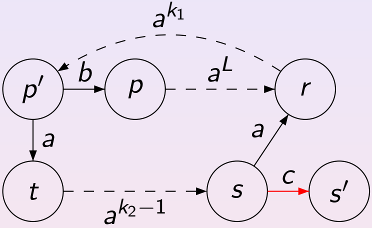
Subcase 2.2: $k_2 = L$ and s is not a bunch. Since s is not a bunch, there is a letter c such that $s' = s\,.\,c \neq r$.

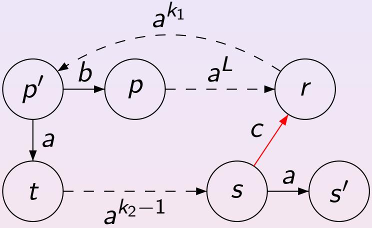


We swap the labels of $s \xrightarrow{c} s'$ and $s \xrightarrow{a} r$. If r still lies on an a-cycle, then the length of the a-cycle is at least $k_1 + k_2 + 2$ and the number of states on the a-cycles increases. Otherwise, the level of r w.r.t. a becomes at least $k_1 + k_2 + 1 > L$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of r and belong to the same tree.

Let s be the state of C such that $s\,.\,a = r$.

Subcase 2.2: $k_2 = L$ **and** s **is not a bunch.** Since s is not a bunch, there is a letter c such that $s' = s\,.\,c \neq r$.





We swap the labels of $s \xrightarrow{c} s'$ and $s \xrightarrow{a} r$. If r still lies on an a-cycle, then the length of the a-cycle is at least $k_1 + k_2 + 2$ and the number of states on the a-cycles increases. Otherwise, the level of r w.r.t. a becomes at least $k_1 + k_2 + 1 > L$ whence all states of maximal level w.r.t. a in the new automaton are a-ascendants of r and belong to the same tree.

Let q be the state on the a-path from p to r such that $q\,.\,a = r$.

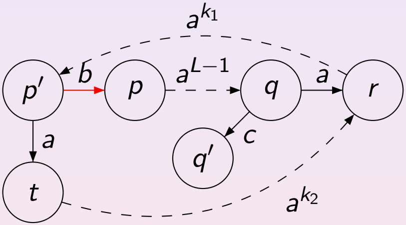
Subcase 2.3: $k_2 = L$ and q is not a bunch. Since q is not a bunch, there is a letter c such that $q' = q\,.\,c \neq r$.

If we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$, we find ourselves in the conditions of Subcase 2.2 (with q and q' playing the roles of s and s' respectively).

Let q be the state on the a-path from p to r such that $q \, . \, a = r$.
Subcase 2.3: $k_2 = L$ and q is not a bunch. Since q is not a bunch, there is a letter c such that $q' = q \, . \, c \neq r$.

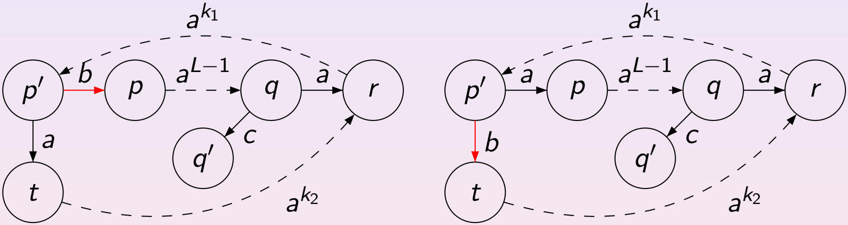
If we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$, we find ourselves in the conditions of Subcase 2.2 (with q and q' playing the roles of s and s' respectively).

Let q be the state on the a-path from p to r such that $q\,.\,a = r$.
Subcase 2.3: $k_2 = L$ and q is not a bunch. Since q is not a bunch, there is a letter c such that $q' = q\,.\,c \neq r$.



If we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$, we find ourselves in the conditions of Subcase 2.2 (with q and q' playing the roles of s and s' respectively).

Let q be the state on the a-path from p to r such that $q\,.\,a = r$. **Subcase 2.3: $k_2 = L$ and q is not a bunch.** Since q is not a bunch, there is a letter c such that $q' = q\,.\,c \neq r$.



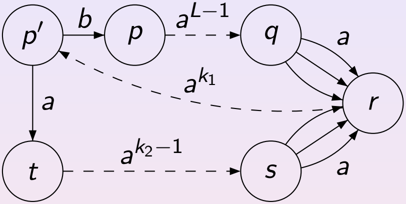
If we swap the labels of $p' \xrightarrow{b} p$ and $p' \xrightarrow{a} t$, we find ourselves in the conditions of Subcase 2.2 (with q and q' playing the roles of s and s' respectively).

Subcase 2.4: $k_2 = L$ **and both** s **and** q **are bunches.**

In this case it is clear that q and s form a stable pair.
This completes the proof.

The proof can be 'unfolded' to a quadratic (in $|V|$) algorithm to find a synchronizing coloring of a given admissible digraph $\Gamma = (V, E)$ – Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

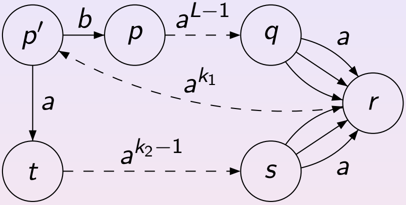
Subcase 2.4: $k_2 = L$ **and both** s **and** q **are bunches.**



In this case it is clear that q and s form a stable pair.
This completes the proof.

The proof can be 'unfolded' to a quadratic (in $|V|$) algorithm to find a synchronizing coloring of a given admissible digraph $\Gamma = (V, E)$ – Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

Subcase 2.4: $k_2 = L$ **and both** s **and** q **are bunches.**

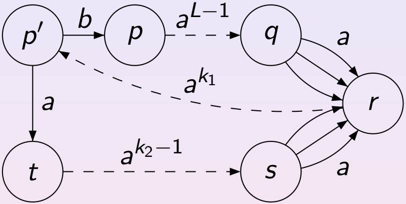


In this case it is clear that q and s form a stable pair.

This completes the proof.

The proof can be 'unfolded' to a quadratic (in $|V|$) algorithm to find a synchronizing coloring of a given admissible digraph $\Gamma = (V, E)$ – Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

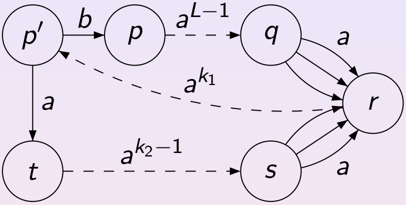
Subcase 2.4: $k_2 = L$ and both s and q are bunches.



In this case it is clear that q and s form a stable pair.
This completes the proof.

The proof can be 'unfolded' to a quadratic (in $|V|$) algorithm to find a synchronizing coloring of a given admissible digraph $\Gamma = (V, E)$ – Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

Subcase 2.4: $k_2 = L$ and both s and q are bunches.



In this case it is clear that q and s form a stable pair.
This completes the proof.

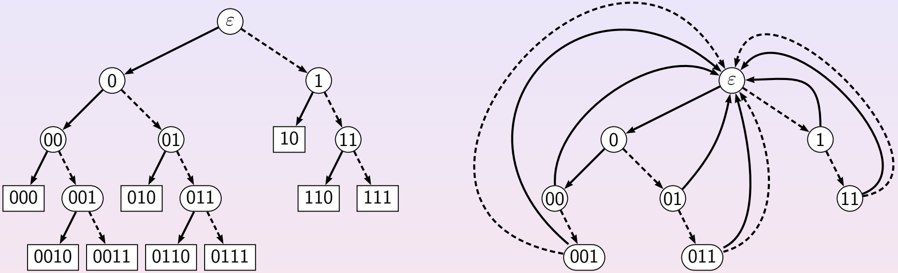
The proof can be 'unfolded' to a quadratic (in $|V|$) algorithm to find a synchronizing coloring of a given admissible digraph $\Gamma = (V, E)$ – Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

14. An Application

Recall the connection between maximal prefix codes and automata discussed in Lecture I.

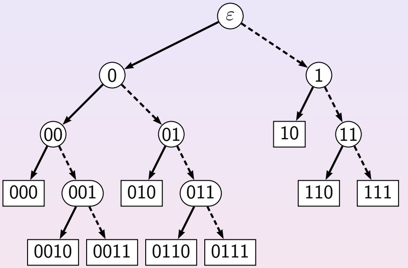
Simple cycles in the automaton on the right correspond to codewords. Thus, if a finite maximal prefix code is such that the g.c.d. of lengths of its codewords is 1, then there exists a synchronized code with the same lengths of codewords (equivalently, with the same tree.) This was proved by Dominique Perrin and Marcel-Paul Schützenberger in 1992.

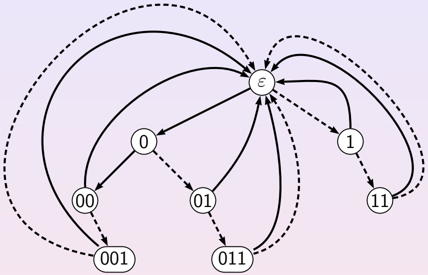
Recall the connection between maximal prefix codes and automata discussed in Lecture I.



Simple cycles in the automaton on the right correspond to codewords. Thus, if a finite maximal prefix code is such that the g.c.d. of lengths of its codewords is 1, then there exists a synchronized code with the same lengths of codewords (equivalently, with the same tree.) This was proved by Dominique Perrin and Marcel-Paul Schützenberger in 1992.

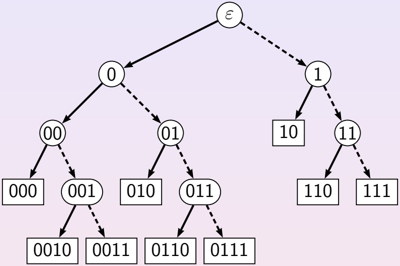
Recall the connection between maximal prefix codes and automata discussed in Lecture I.

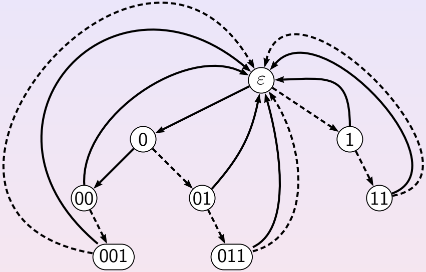




Simple cycles in the automaton on the right correspond to codewords. Thus, if a finite maximal prefix code is such that the g.c.d. of lengths of its codewords is 1, then there exists a synchronized code with the same lengths of codewords (equivalently, with the same tree.) This was proved by Dominique Perrin and Marcel-Paul Schützenberger in 1992.

Recall the connection between maximal prefix codes and automata discussed in Lecture I.

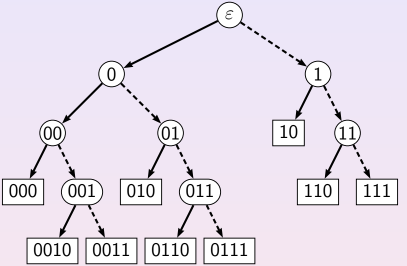


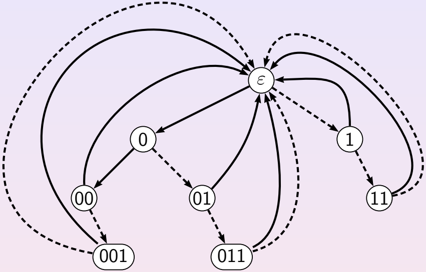


Simple cycles in the automaton on the right correspond to codewords. Thus, if a finite maximal prefix code is such that the g.c.d. of lengths of its codewords is 1, then there exists a synchronized code with the same lengths of codewords (equivalently, with the same tree.) This was proved by Dominique Perrin and Marcel-Paul Schützenberger in 1992.

14. An Application

Recall the connection between maximal prefix codes and automata discussed in Lecture I.





Simple cycles in the automaton on the right correspond to codewords. Thus, if a finite maximal prefix code is such that the g.c.d. of lengths of its codewords is 1, then there exists a synchronized code with the same lengths of codewords (equivalently, with the same tree.) This was proved by Dominique Perrin and Marcel-Paul Schützenberger in 1992.

What can be said about underlying graphs of arbitrary synchronizing automata(not necessarily strongly connected)?

Given a graph $\Gamma = (V, E)$, a vertex q is said to be reachable from a vertex p if there is a path from p to q. Clearly, the reachability relation is reflexive and transitive, and the mutual reachability relation is an equivalence on the set V. The subgraphs induced on the classes of the mutual reachability relation are strongly connected and are called the strongly connected components of Γ.

The reachability relation induces a partial order on the set of the strongly connected components: a component Γ_1 precedes a component Γ_2 in this order if some vertex of Γ_1 is reachable from some vertex of Γ_2.

Corollary. *A graph Γ with constant out-degree admits a synchronizing coloring if and only if Γ has the least strongly connected component and this component is primitive.*

What can be said about underlying graphs of arbitrary synchronizing automata(not necessarily strongly connected)? Given a graph $\Gamma = (V, E)$, a vertex q is said to be reachable from a vertex p if there is a path from p to q. Clearly, the reachability relation is reflexive and transitive, and the mutual reachability relation is an equivalence on the set V. The subgraphs induced on the classes of the mutual reachability relation are strongly connected and are called the strongly connected components of Γ. The reachability relation induces a partial order on the set of the strongly connected components: a component Γ_1 precedes a component Γ_2 in this order if some vertex of Γ_1 is reachable from some vertex of Γ_2.

Corollary. *A graph Γ with constant out-degree admits a synchronizing coloring if and only if Γ has the least strongly connected component and this component is primitive.*

What can be said about underlying graphs of arbitrary synchronizing automata(not necessarily strongly connected)? Given a graph $\Gamma = (V, E)$, a vertex q is said to be reachable from a vertex p if there is a path from p to q. Clearly, the reachability relation is reflexive and transitive, and the mutual reachability relation is an equivalence on the set V. The subgraphs induced on the classes of the mutual reachability relation are strongly connected and are called the strongly connected components of Γ. The reachability relation induces a partial order on the set of the strongly connected components: a component Γ_1 precedes a component Γ_2 in this order if some vertex of Γ_1 is reachable from some vertex of Γ_2.

Corollary. *A graph Γ with constant out-degree admits a synchronizing coloring if and only if Γ has the least strongly connected component and this component is primitive.*

What can be said about underlying graphs of arbitrary synchronizing automata(not necessarily strongly connected)? Given a graph $\Gamma = (V, E)$, a vertex q is said to be reachable from a vertex p if there is a path from p to q. Clearly, the reachability relation is reflexive and transitive, and the mutual reachability relation is an equivalence on the set V. The subgraphs induced on the classes of the mutual reachability relation are strongly connected and are called the strongly connected components of Γ. The reachability relation induces a partial order on the set of the strongly connected components: a component Γ_1 precedes a component Γ_2 in this order if some vertex of Γ_1 is reachable from some vertex of Γ_2.

Corollary. *A graph Γ with constant out-degree admits a synchronizing coloring if and only if Γ has the least strongly connected component and this component is primitive.*

15. General Case

What can be said about underlying graphs of arbitrary synchronizing automata(not necessarily strongly connected)? Given a graph $\Gamma = (V, E)$, a vertex q is said to be reachable from a vertex p if there is a path from p to q. Clearly, the reachability relation is reflexive and transitive, and the mutual reachability relation is an equivalence on the set V. The subgraphs induced on the classes of the mutual reachability relation are strongly connected and are called the strongly connected components of Γ. The reachability relation induces a partial order on the set of the strongly connected components: a component Γ_1 precedes a component Γ_2 in this order if some vertex of Γ_1 is reachable from some vertex of Γ_2.

Corollary. *A graph Γ with constant out-degree admits a synchronizing coloring if and only if Γ has the least strongly connected component and this component is primitive.*

What can be said about underlying graphs of arbitrary synchronizing automata(not necessarily strongly connected)? Given a graph $\Gamma = (V, E)$, a vertex q is said to be reachable from a vertex p if there is a path from p to q. Clearly, the reachability relation is reflexive and transitive, and the mutual reachability relation is an equivalence on the set V. The subgraphs induced on the classes of the mutual reachability relation are strongly connected and are called the strongly connected components of Γ. The reachability relation induces a partial order on the set of the strongly connected components: a component Γ_1 precedes a component Γ_2 in this order if some vertex of Γ_1 is reachable from some vertex of Γ_2.

Corollary. *A graph Γ with constant out-degree admits a synchronizing coloring if and only if Γ has the least strongly connected component and this component is primitive.*

16. Non-Primitive Case

If one drops the primitivity condition, one can prove (basically by the same method) the following generalization of the Road Coloring Theorem:

Theorem. *Suppose that d is the g.c.d. of the lengths of cycles in a strongly connected graph $\Gamma = (V, E)$ with constant out-degree. Then Γ admits a coloring for which there is a word w such that $|V\,.\,w| = d$.*

This generalization is due to Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

If one drops the primitivity condition, one can prove (basically by the same method) the following generalization of the Road Coloring Theorem:

Theorem. *Suppose that d is the g.c.d. of the lengths of cycles in a strongly connected graph $\Gamma = (V, E)$ with constant out-degree. Then Γ admits a coloring for which there is a word w such that $|V\,.\,w| = d$.*

This generalization is due to Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

16. Non-Primitive Case

If one drops the primitivity condition, one can prove (basically by the same method) the following generalization of the Road Coloring Theorem:

Theorem. *Suppose that d is the g.c.d. of the lengths of cycles in a strongly connected graph $\Gamma = (V, E)$ with constant out-degree. Then Γ admits a coloring for which there is a word w such that $|V\,.\,w| = d$.*

This generalization is due to Marie-Pierre Béal and Dominique Perrin, A quadratic algorithm for road coloring, Discr. Appl. Math. 169 (2014) 15–29.

17. Open Questions

1. Characterize totally synchronizing graphs, i.e., graphs such that every coloring makes them become synchronizing automata. E.g., the underlying graphs of the Černý automata have this feature.

2. (Quantitative Road Coloring problem) An admissible graph with n vertices and common out-degree k has k^n colorings. How many of them may be synchronizing?

Conjecture: at least one half with exactly one exception which is the Cayley graph of the symmetric group in 3 points.

1. Characterize totally synchronizing graphs, i.e., graphs such that every coloring makes them become synchronizing automata. E.g., the underlying graphs of the Černý automata have this feature.

2. (Quantitative Road Coloring problem) An admissible graph with n vertices and common out-degree k has k^n colorings. How many of them may be synchronizing?

Conjecture: at least one half with exactly one exception which is the Cayley graph of the symmetric group in 3 points.

1. Characterize totally synchronizing graphs, i.e., graphs such that every coloring makes them become synchronizing automata. E.g., the underlying graphs of the Černý automata have this feature.

2. (Quantitative Road Coloring problem) An admissible graph with n vertices and common out-degree k has k^n colorings. How many of them may be synchronizing?

Conjecture: at least one half with exactly one exception which is the Cayley graph of the symmetric group in 3 points.

17. Open Questions

1. Characterize totally synchronizing graphs, i.e., graphs such that every coloring makes them become synchronizing automata. E.g., the underlying graphs of the Černý automata have this feature.

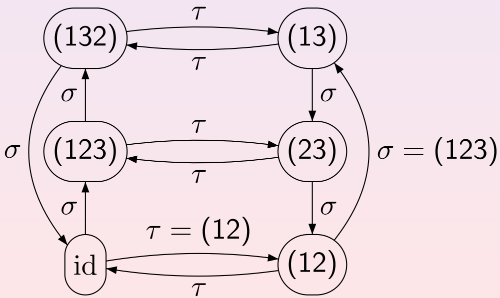
2. (Quantitative Road Coloring problem) An admissible graph with n vertices and common out-degree k has k^n colorings. How many of them may be synchronizing?

Conjecture: at least one half with exactly one exception which is the Cayley graph of the symmetric group in 3 points.

17. Open Questions

1. Characterize totally synchronizing graphs, i.e., graphs such that every coloring makes them become synchronizing automata. E.g., the underlying graphs of the Černý automata have this feature.
2. (Quantitative Road Coloring problem) An admissible graph with n vertices and common out-degree k has k^n colorings. How many of them may be synchronizing?

Conjecture: at least one half with exactly one exception which is the Cayley graph of the symmetric group in 3 points.



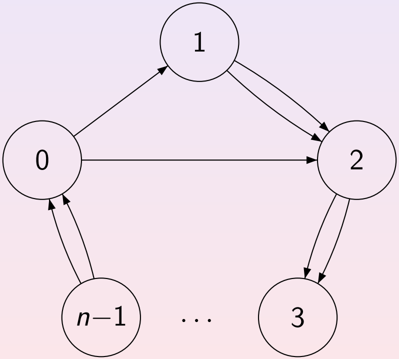
3. (Hybrid Road Coloring – Černý problem) What is the maximum value of reset thresholds for synchronizing colorings of admissible graphs with n vertices?

Conjecture: $n^2 - 3n + 3$, achieved by the Wielandt graph W_n.

3. (Hybrid Road Coloring – Černý problem) What is the maximum value of reset thresholds for synchronizing colorings of admissible graphs with n vertices?
Conjecture: $n^2 - 3n + 3$, achieved by the Wielandt graph W_n.

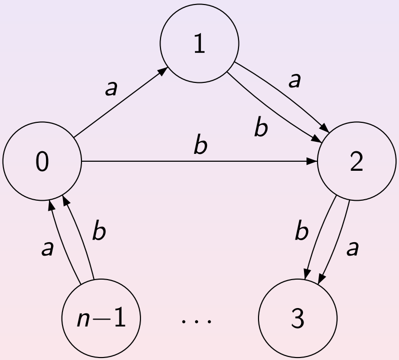
3. (Hybrid Road Coloring – Černý problem) What is the maximum value of reset thresholds for synchronizing colorings of admissible graphs with n vertices?

Conjecture: $n^2 - 3n + 3$, achieved by the Wielandt graph W_n.



3. (Hybrid Road Coloring – Černý problem) What is the maximum value of reset thresholds for synchronizing colorings of admissible graphs with n vertices?

Conjecture: $n^2 - 3n + 3$, achieved by the Wielandt graph W_n.



W_n has a unique coloring with reset threshold $n^2 - 3n + 3$.