Functionally Complete Algebras from the Computational Perspective

Gábor Horváth

Joint work with: Chrystopher L. Nehaniv,
Csaba Szabó

6^{th} September, 2009.

Functionally Complete Groups

- Theorem: *Maurer, Rhodes (1965)*
  For a finite group G the following are equivalent:

  1. G is a simple, non-Abelian group,
  2. G is functionally complete.

- A finite algebra $\mathcal{A}$ is *functionally complete*, iff every $A^n \to A$ function can be represented as an $\mathcal{A}$-polynomial.

  Ex.

$x \backslash y$	0	1
0	1	0
1	1	1

over $\mathbb{Z}_2$ is $x \cdot y + y + 1$.

Motivation

- Theorem: *Krohn, Maurer, Rhodes (1966)*
  A finite state machine based on G can compute any Boolean function f by some word $w \iff G$ is simple non-Abelian.

- proof was not algorithmic

- no bounds on $||w||$

Finite state machine based on G

G is simple, nonabelian

generated by two elements g_0 and g_1

$0 \leftrightarrow g_0, \quad 1 \leftrightarrow g_1$

$H \leq G$ is a maximal subgroup

States: right cosets $\{Hg : g \in G\}$

Initial state: H

Transitions: $Hg \xrightarrow{g_0} Hgg_0, Hg \xrightarrow{g_1} Hgg_1$

Output: States $\to \{0, 1\}$

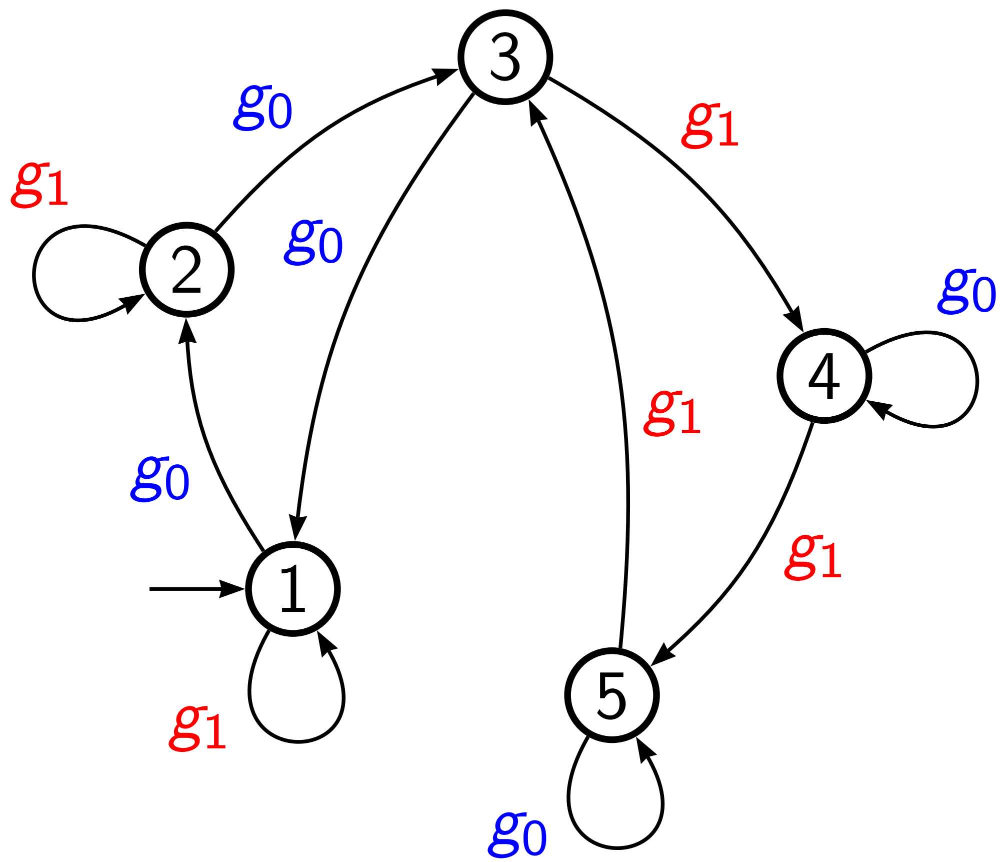
Finite state machine based on A_5

$G \simeq A_5$, $\qquad H = Stab_G(1) \simeq A_4$

$g_0 = (123)$, $\qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1$,
$\{2, 4\} \to 0$



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

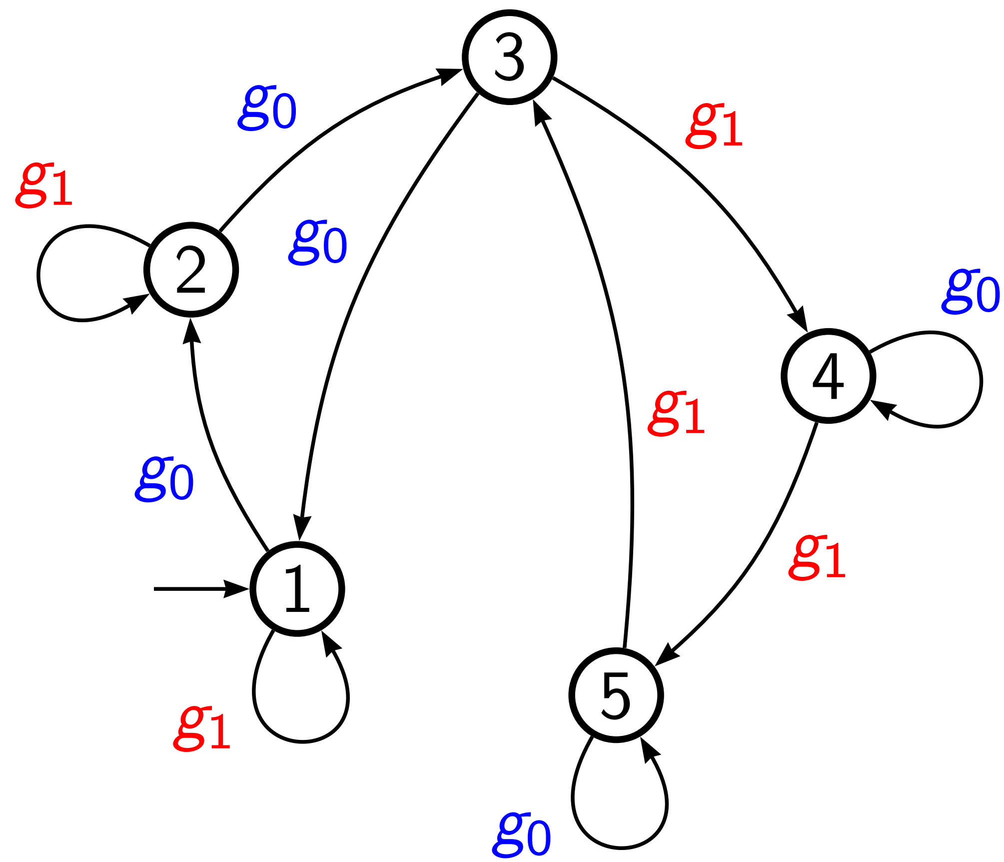
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1,$
$\{2, 4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0		
1		



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

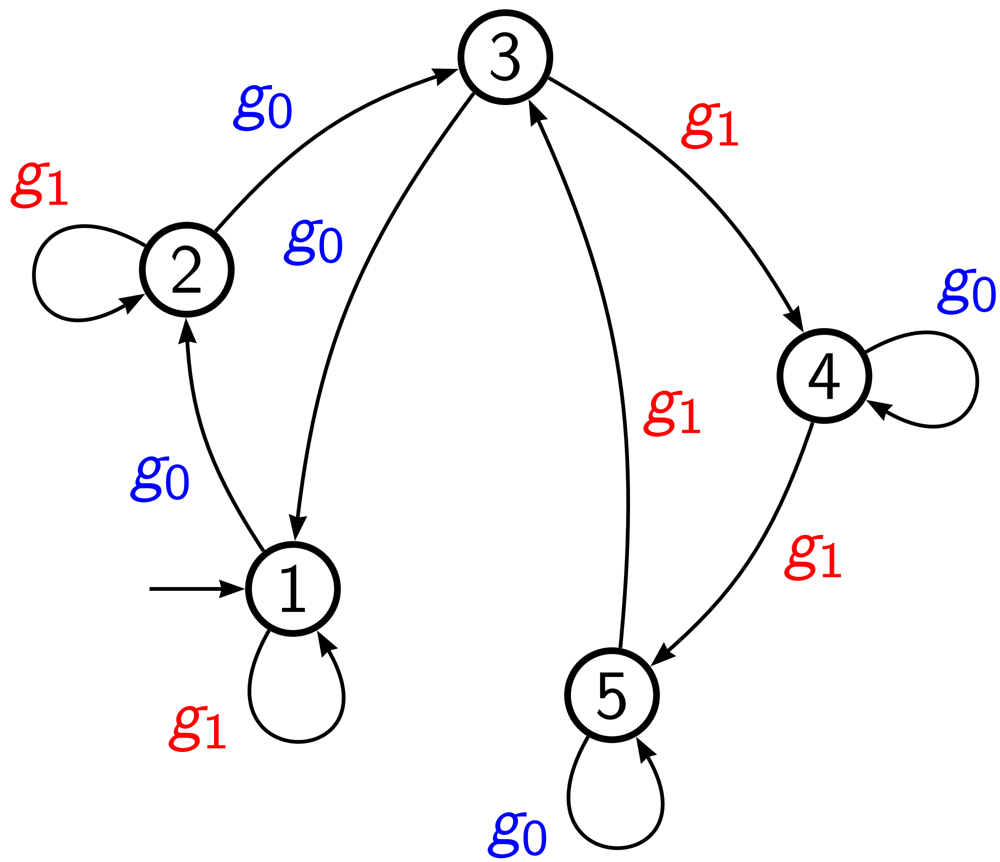
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1,$
$\{2, 4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0		
1		



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

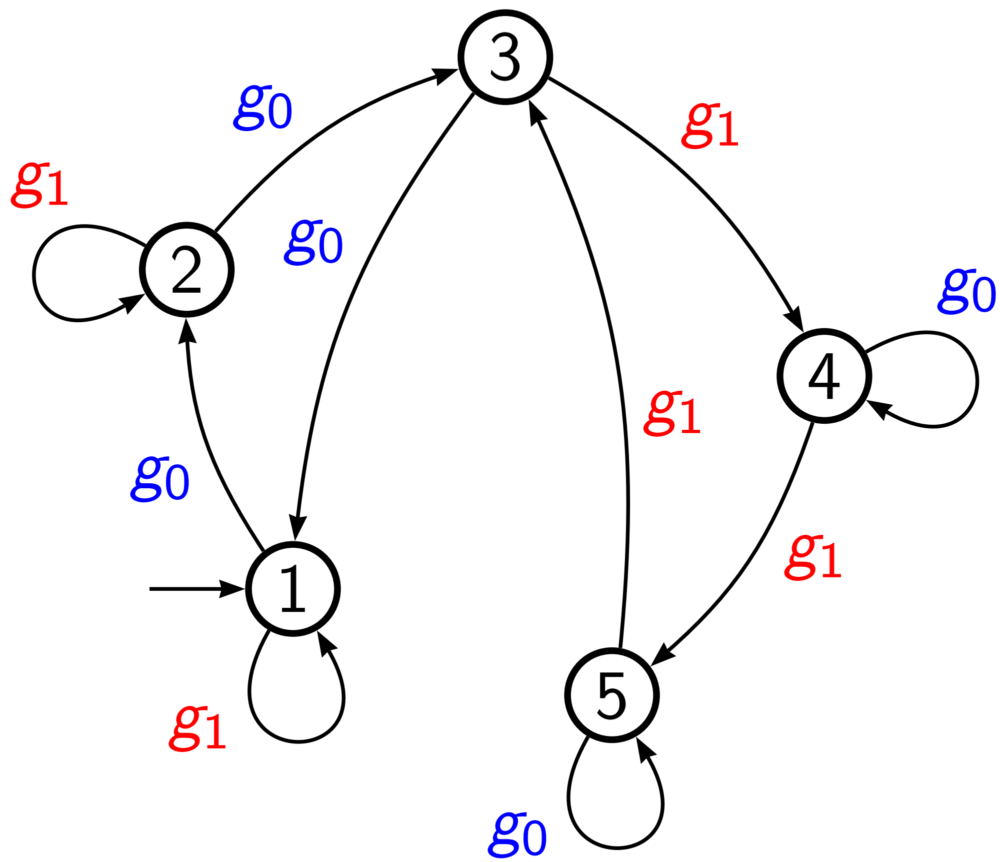
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1,$
$\{2, 4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	
1		



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

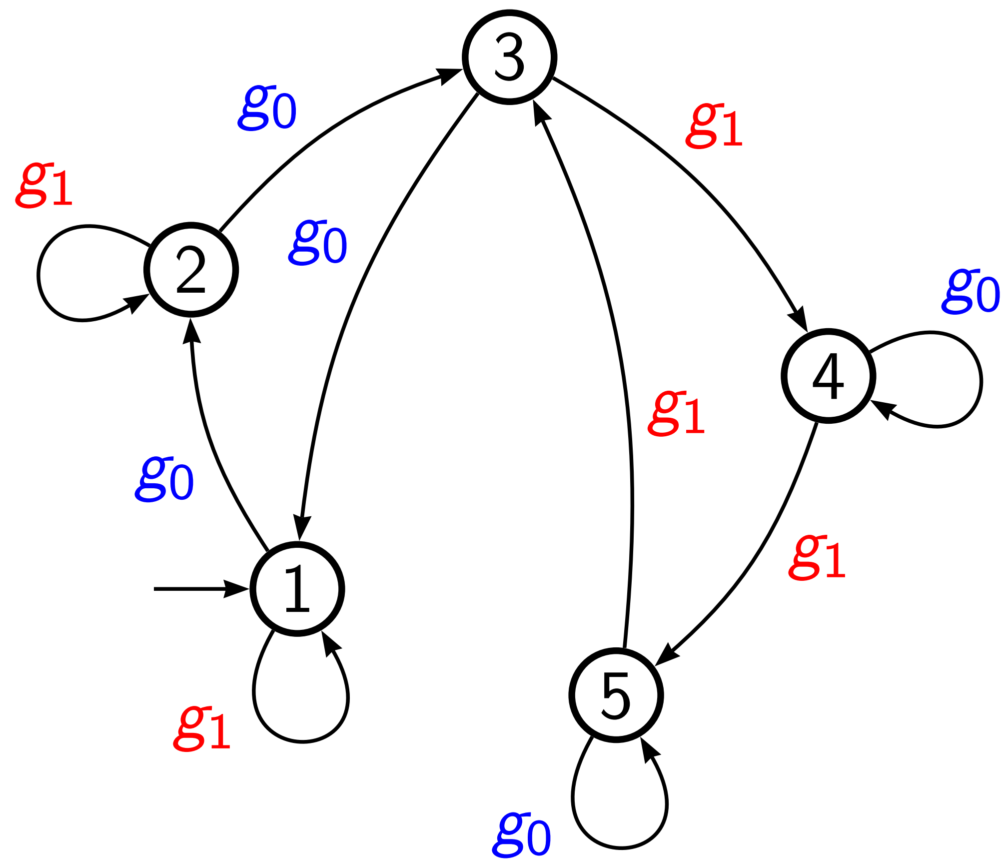
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1,$
$\{2, 4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	
1		



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

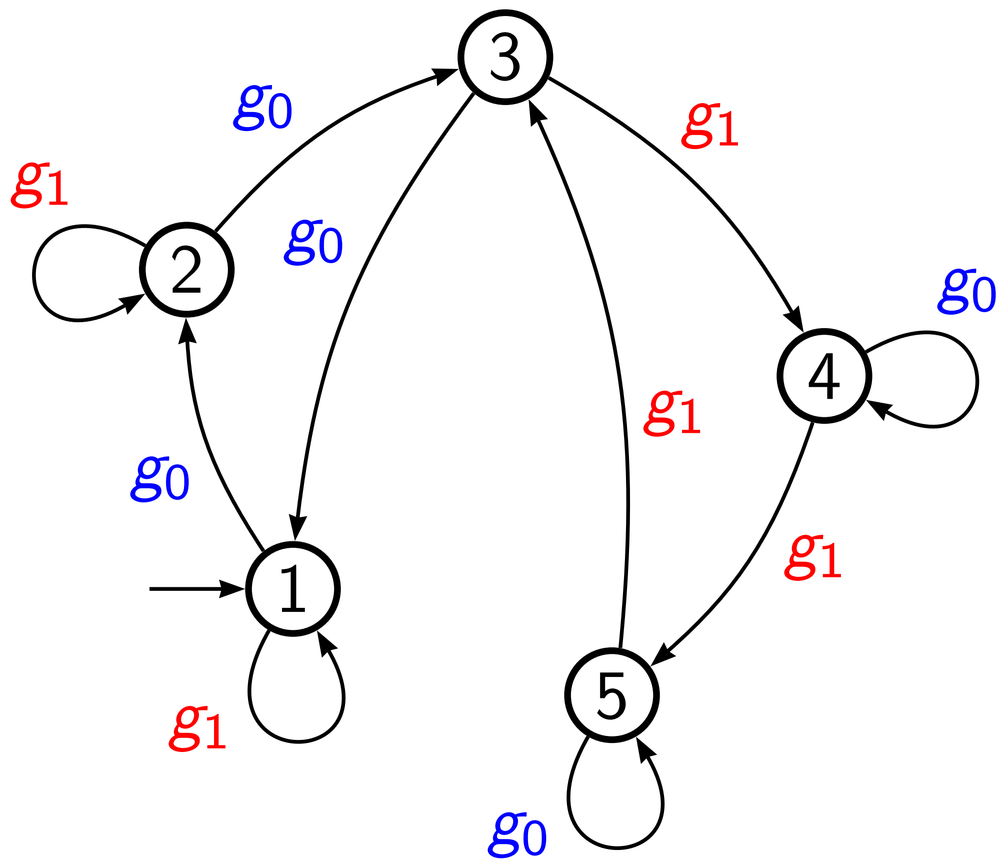
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1,3,5\} \to 1,$
$\{2,4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	0
1		



Finite state machine based on A_5

$G \simeq A_5$, $\qquad H = Stab_G(1) \simeq A_4$

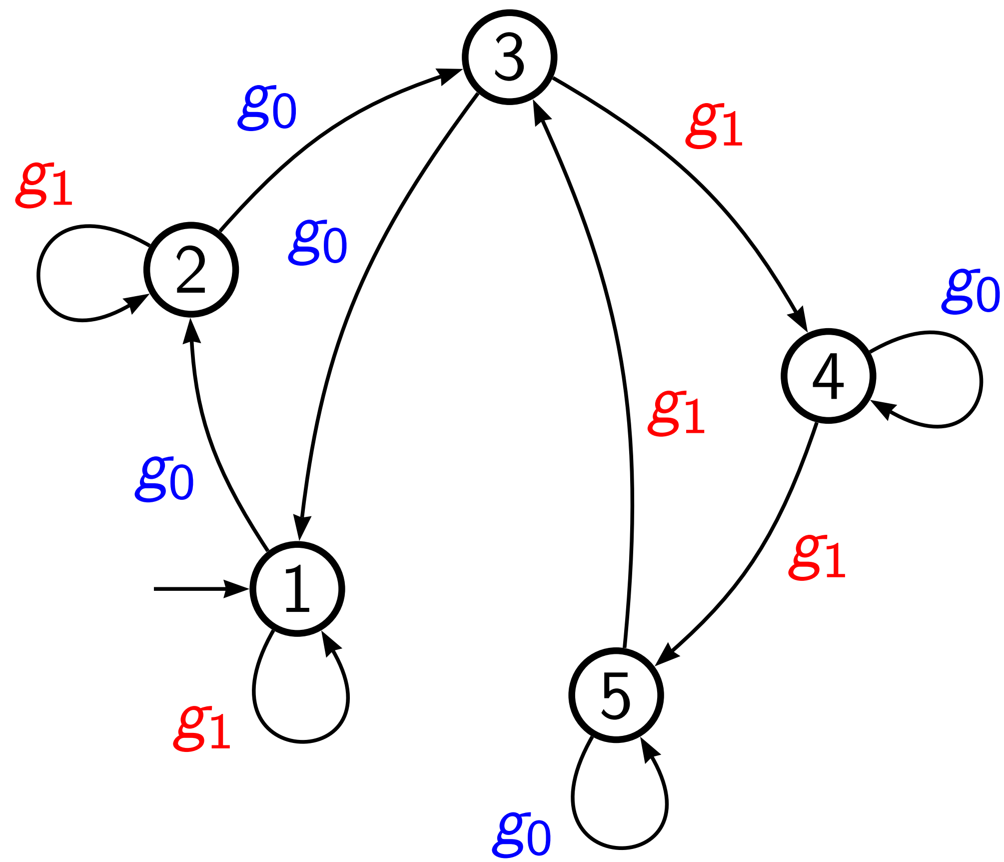
$g_0 = (123)$, $\qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1,$
$\{2, 4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	0
1		



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

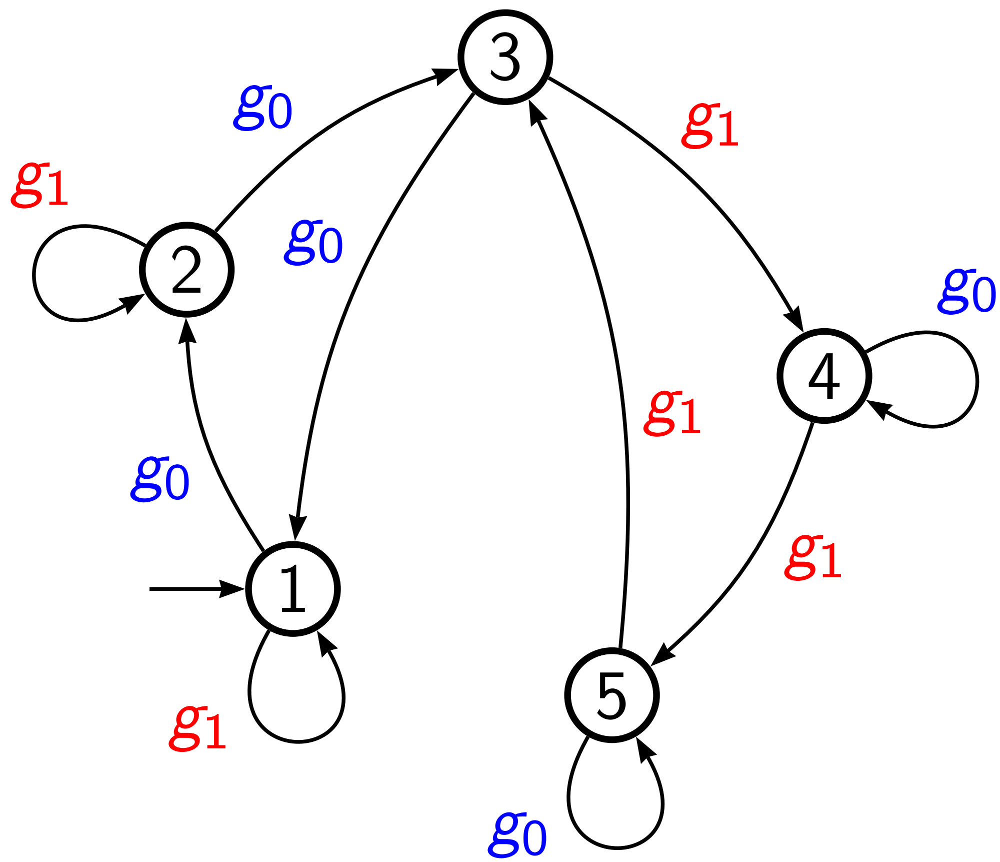
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1, 3, 5\} \to 1,$
$\{2, 4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	0
1	1	



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

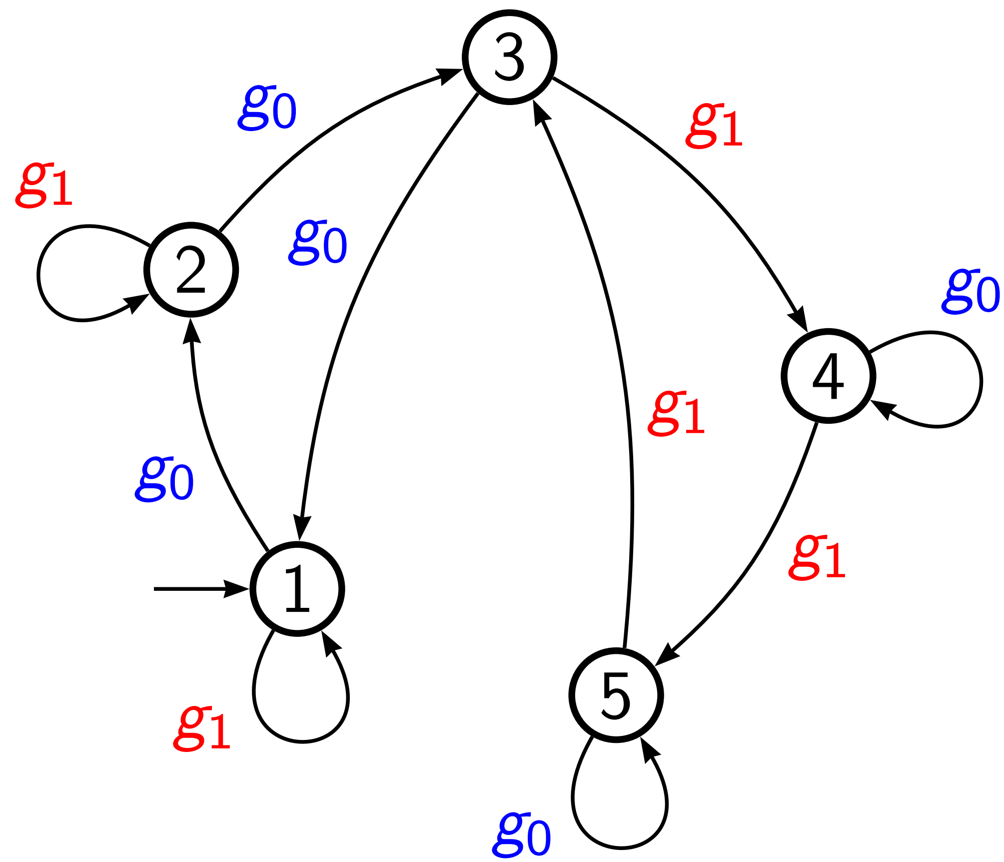
$g_0 = (123), \qquad g = (345)$

Output:

$\{1,3,5\} \to 1,$
$\{2,4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	0
1	1	



Finite state machine based on A_5

$G \simeq A_5, \qquad H = Stab_G(1) \simeq A_4$

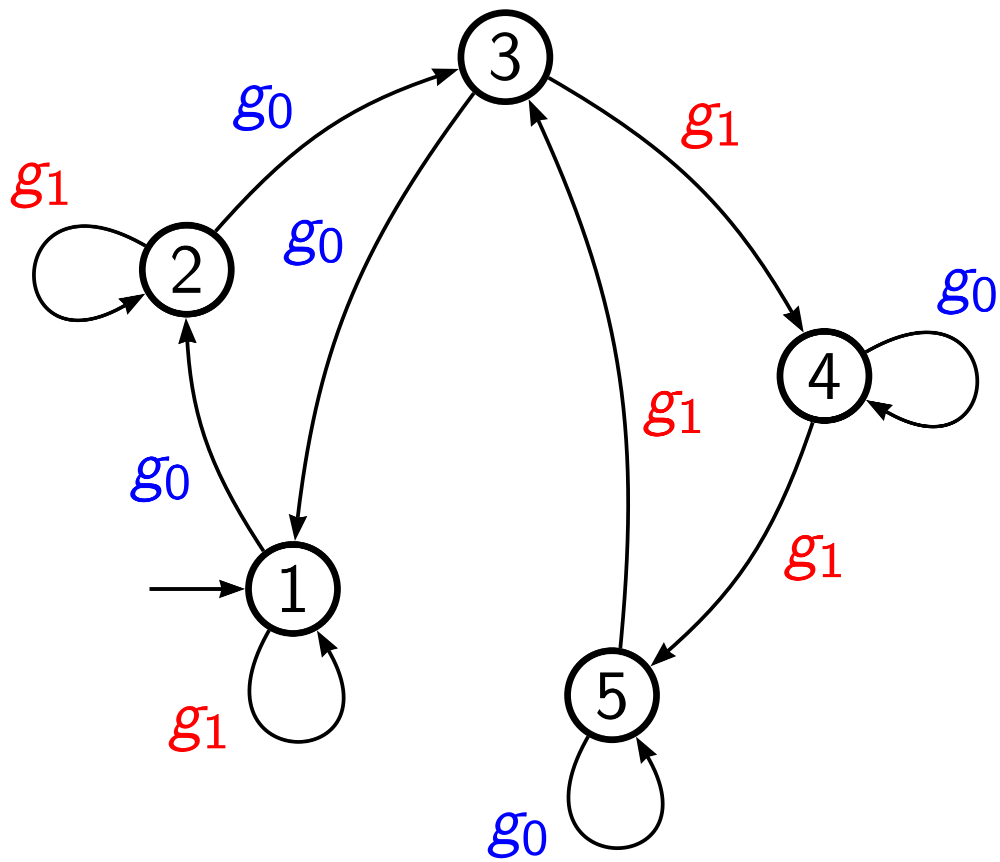
$g_0 = (123), \qquad g_1 = (345)$

Output:

$\{1,3,5\} \to 1,$
$\{2,4\} \to 0$

$g_0 \cdot g_0 \cdot x \cdot x \cdot y$

$x \backslash y$	0	1
0	1	0
1	1	1



Finite state machine based on G

Theorem: *Horváth, Nehaniv (2008)*

- G is simple
- $f\colon\ \{0,1\}^n \to \{0,1\}$
- $\left|f^{-1}(1)\right| = k$
- A finite automaton based on G can compute f by a word w
$$||w|| \leq c_1(G) \cdot n^8 \cdot k.$$

Finite state machine based on G

Theorem: *Horváth, Nehaniv (2008)*

- G is simple
- $f\colon \{0,1\}^n \to \{0,1\}$
- $\left|f^{-1}(1)\right| = k \quad = O(2^n)$
- A finite automaton based on G can compute f by a word w
$$||w|| \leq c_1(G) \cdot n^8 \cdot k.$$

Rem.: exists f s.t. for every w
$$||w|| \geq c_1'(G) \cdot 2^n / \log n.$$

Finite state machine based on A_m

Theorem: *Horváth, Nehaniv (2008)*

- A_m is simple
- $f\colon \{0,1\}^n \to \{0,1\}$
- $\left|f^{-1}(1)\right| = k \quad = O(2^n)$
- A finite automaton based on A_m can compute f by a word w

$$||w|| \le c_2(A_m) \cdot n^2 \cdot k.$$

Rem.: exists f s.t. for every w

$$||w|| \ge c_2'(A_m) \cdot 2^n / \log n.$$

Functions over groups

Cor.: *Horváth, Nehaniv (2008)*

- G is a simple group
- $f\colon G^n \to G$
- $\left|f^{-1}(G \setminus 1)\right| = k$
- f can be realized by a word w over G

$$||w|| \leq c_3(G) \cdot n^8 \cdot k.$$

- w uses iterated commutators
- f can be realized by a word w over $(G, [\cdot, \cdot])$

$$||w|| \leq c_4(G) \cdot n \cdot k.$$

Polynomials over groups

Theorem: *Horváth, Nehaniv (2008)*

Let G be a simple group. Let $p\colon G^n \to G$ an n-ary polynomial. Then there exists an n-ary polynomial $w\colon G^n \to G$, s.t.

$$w(g_1, \ldots, g_n) = p(g_1, \ldots, g_n),$$
$$||w|| \leq c_3\,(G) \cdot n^8 \cdot k.$$

There exists an n-ary polynomial $w'\colon G^n \to G$ using commutators s.t.

$$w'(g_1, \ldots, g_n) = p(g_1, \ldots, g_n),$$
$$||w'|| \leq c_4\,(G) \cdot n \cdot k.$$

Equivalence Problem

$\mathcal{A}$ finite algebra

- identity: two polynomials p_1, p_2 over $\mathcal{A}$.

$$p_1 \equiv p_2 \iff \begin{array}{l}\text{for every } a_1, \ldots, a_n \in \mathcal{A} \\ p_1(a_1, \ldots, a_n) = p_2(a_1, \ldots, a_n)\end{array}$$

- equivalence problem: (identity checking problem)

  Input: two polynomials p_1, p_2 over $\mathcal{A}$

  Question: is $p_1 \equiv p_2$ or not?

- What is the complexity? (P or coNP-complete)

Functionally complete algebras

- Theorem: *Horváth, Nehaniv, Szabó (2008)*

  functionally complete algebra $\Longrightarrow$ coNP-complete equiv.