Algebraic Theory of Automata and Semigroups

Hashem Elezabi
MACT 4134 Final Project, Fall 2018
Dr. Michel Hebert
American University in Cairo

Outline

- Introduction and motivation
- Semigroups
- The Krohn-Rhodes decomposition theorem in semigroup theory
- The Krohn-Rhodes decomposition theorem in automata theory

Outline

- Introduction and motivation
- Semigroups
- The Krohn-Rhodes decomposition theorem in semigroup theory
- The Krohn-Rhodes decomposition theorem in automata theory

Introduction

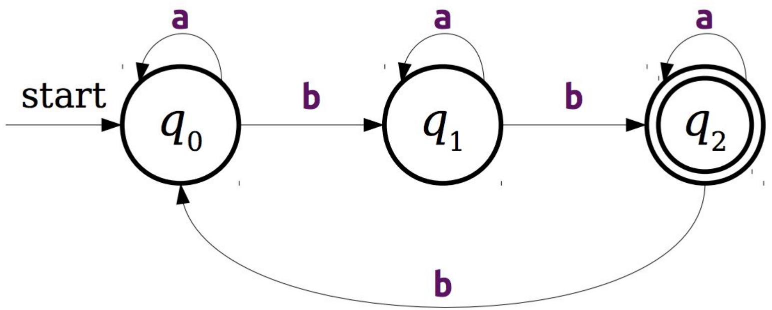
- Krohn-Rhodes decomposition theorem, 1965: every finite automaton divides a cascade product of very simple automata consisting of two types, permutation automata and reset automata (also called flip-flops).
- Proved using abstract algebra, and revealed a deep connection between *finite automata* and *finite semigroups.*
- Corresponding formulation in semigroup theory: every finite semigroup divides a wreath product of finite groups and finite aperiodic semigroups.
- We'll define division later, but think of it as a decomposition of the original automaton or semigroup into simpler components.
- Started so-called algebraic automata theory.

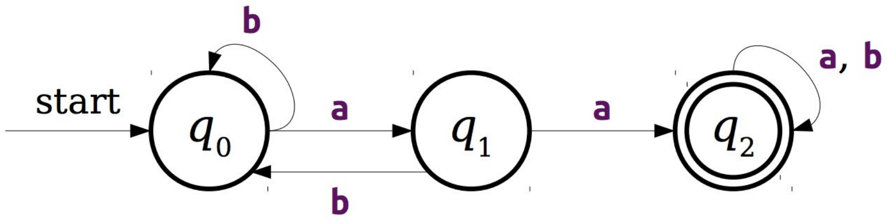
What are finite automata?

- Automata theory is a subfield of computer science dealing with automata.
- A *finite automaton* is a simple abstract model of a computing device. It is one way to mathematically model computation.
- It's a finite-state machine that accepts and rejects strings of symbols (eg. "abbaab") based on a set of rules, called *transitions.*
- Here we only study *deterministic* finite automata, or DFAs, in which there is a unique computation for every input string.

What are finite automata?

- Given a string, a DFA starts at a *start state* and moves between states according to the current letter in the string and the relevant transition. If it ends in one of its *accept states,* we say that it *accepts* the given string.
- What strings do these automata accept?

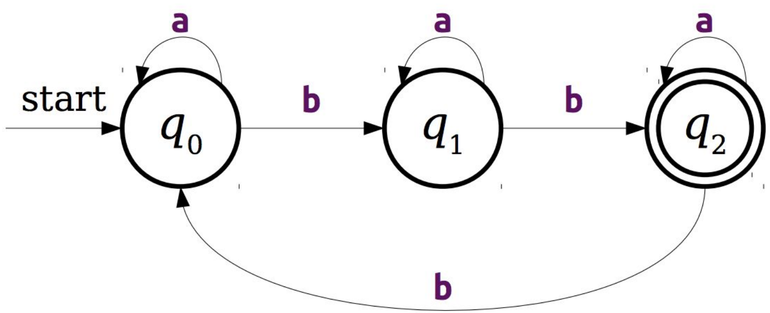




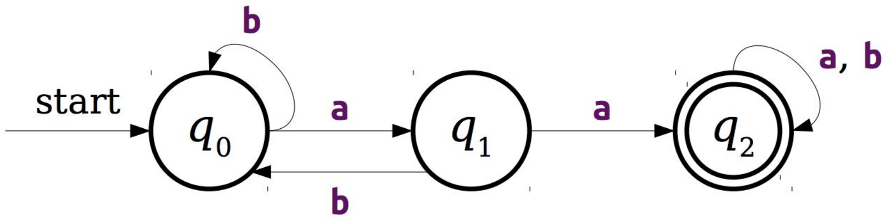
Figures taken from [1]

What are finite automata?

- Given a string, a DFA starts at a *start state* and moves between states according to the current letter in the string and the relevant transition. If it ends in one of its *accept states,* we say that it *accepts* the given string.
- What strings do these automata accept?



All strings in which the number of ‘b’s is congruent to 2 mod 3



All strings that contain the substring ‘aa’

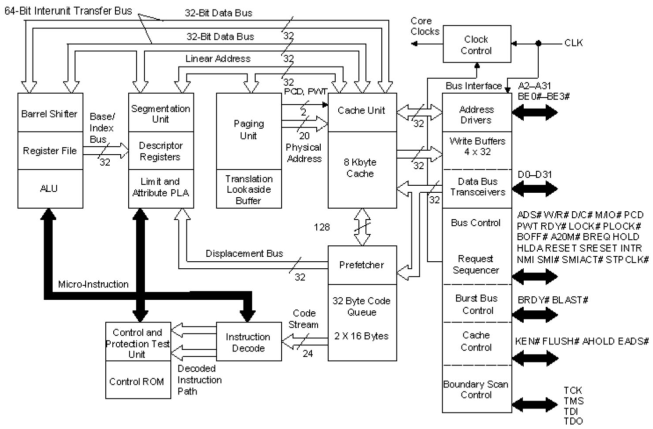
Figures taken from [1]

Formal definitions

- An *alphabet* Σ is a finite, nonempty set of symbols, which we also call *letters.*
- A *string,* or *word,* over an alphabet is a finite sequence of letters drawn from Σ. An empty string is denoted by ϵ.
- Formally, a DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ consisting of a finite set of states Q, a finite alphabet Σ, a transition function $\delta : Q \times \Sigma \rightarrow Q$, a start state $q_0 \in Q$, and a set of accept states $F \subseteq Q$.
- In the examples from last slide, we have $\Sigma = \{a, b\}$.

Why do we study automata?

- Mathematical abstraction of a computer that avoids dealing with the messy real-world intricacies.
- By studying these theoretical models, we can prove statements about whether a problem can be solved by any real-world computer, which is a powerful statement that holds regardless of technological advancements.

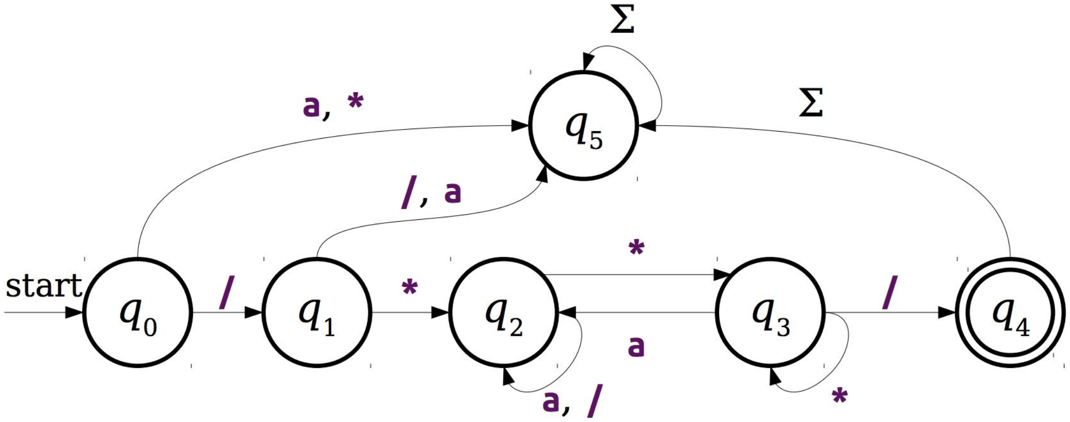


Too messy for a theoretical computer scientist to deal with mathematically!

http://www.intel.com/design/intarch/prodbref/272713.htm

Why do we study automata?

- Direct applications also exist. For example, this more elaborate DFA only accepts valid C-style comments (‘a’ represents any character other than ‘/’ and ‘*’).



- Finite automata are in fact used in software compilers for similar purposes.

Figure taken from [1]

Beyond computer science

- Few real-world systems are completely static. An important notion is the notion of *change* of a system, which is a fundamental concept in science and computation.
- Automata can capture the structure and interactions of systems and how they interact to environmental and internal changes.
- While mathematical analysis studies change in continuous settings (set of states is a continuum), automata theory studies change in discrete settings (set of states is discrete) [2].
- This means that automata theory can potentially be used to tackle some of the biggest problems in science and engineering.

Automata ⟷ Semigroups

- Krohn and Rhodes showed that we can describe and study automata algebraically.
- If we can prove results about automata using the powerful techniques of algebra, we can potentially make exciting advancements.
- A *semigroup* is a set along with an associative binary operation. It need not have an identity or inverses.
- The correspondence between finite automata and finite semigroups comes from the fact that every automaton has a corresponding *transformation semigroup*.

Transformation semigroups

- A *transformation* is a function $f : Q \to Q$ (from a set Q to itself). The permutations we studied are bijective transformations.
- Just like there are permutation groups, there are transformation semigroups that are closed under the associative operation of function composition.
- Let T_n denote the semigroup of all n^n transformations on n elements. Of course, the symmetric group $S_n \subset T_n$. Below are the 4 elements of T_2 written in (2 x n)-matrix form. The transformation takes the element in entry (1, j) to the element in entry (2, j).

$$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

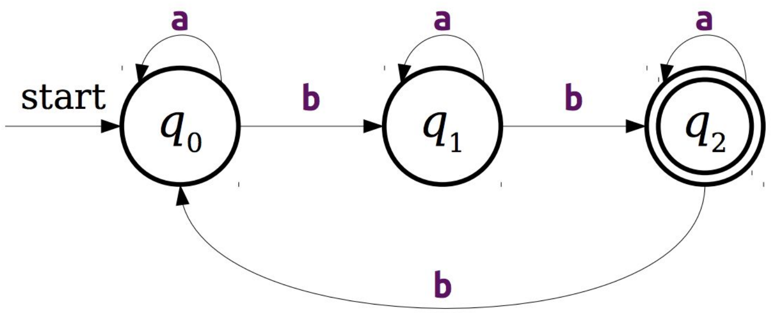
Transformation semigroups

- A *transformation* is a function $f : Q \to Q$ (from a set Q to itself). The permutations we studied are bijective transformations.
- Just like there are permutation groups, there are transformation semigroups that are closed under the associative operation of function composition.
- Let T_n denote the semigroup of all n^n transformations on n elements. Of course, the symmetric group $S_n \subset T_n$. Below are the 4 elements of T_2 written in (2 x n)-matrix form. The transformation takes the element in entry (1, j) to the element in entry (2, j).

$$S_2 \quad \boxed{\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Automata ⟷ Semigroups

- If we ignore the start and accept states of a finite automaton, we see that every finite automaton corresponds to some transformation semigroup whose generators are the transformations induced by the input letters in Σ.



Transformation induced by 'a' $\begin{pmatrix} 0 & 1 & 2 \\ 0 & 1 & 2 \end{pmatrix}$

Transformation induced by 'b' $\begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix}$

- Any other transformation corresponding to any input string can be formed by combining these two. We will formally define generators later.

Krohn-Rhodes decomposition theorem

- As a final motivating note, the idea of *decomposing* a complicated system into simpler components that are easier to understand is an important concept, seen in various fields in science and engineering, and seen many times within mathematics.
- The Krohn-Rhodes theorem is analogous to the Jordan-Hölder theorem, which shows the existence of a composition series for any finite group.
- Another example of a decomposition is the Fundamental Theorem of Finite Abelian Groups, which we studied.

Outline

- Introduction and motivation
- Semigroups
- The Krohn-Rhodes decomposition theorem in semigroup theory
- The Krohn-Rhodes decomposition theorem in automata theory

Definitions

- A *semigroup* S is a nonempty set along with an associative binary operation (under which S is closed, by definition).
- A *monoid* is a semigroup with an identity element e such that ea = ae = a for all a in S.
- We can always form a monoid from a semigroup by adjoining an identity element and defining its multiplication with all other elements appropriately. We denote this newly formed monoid by S^1.
- As we know, a group is a monoid in which every element is invertible. Clearly Groups ⊂ Monoids ⊂ Semigroups.

Examples of semigroups and monoids

- The set of positive integers is a semigroup under addition. The set of natural numbers (including 0) is a monoid under addition, with identity 0.
- The set of all nonempty strings over a finite alphabet Σ is a semigroup under the associative operation of string concatenation, called the *free semigroup over Σ*. Including the empty string yields the *free monoid over Σ*.
- Any ring is a semigroup under multiplication. Any ring with unity is a monoid under multiplication.

Definitions

- A *subsemigroup* T of a semigroup S is a nonempty subset of S that is closed under multiplication. A *submonoid* of S is a subsemigroup that is a monoid, and a *subgroup* of S is a subsemigroup that is a group.
- The set of invertible elements of a monoid M is a subgroup of M (straightforward proof).
- This subgroup of invertible elements is called the *group of units of M.*

Definitions

- A *left ideal* of a semigroup S is a nonempty subset T such that for any s $\in$ S and any t $\in$ T, st $\in$ T. A *right ideal* is the same with ts $\in$ T. An *ideal* is both a left and a right ideal.
- Of course, any ideal is a subsemigroup of S.
- A semigroup is *simple* if it contains no proper ideals, i.e. the only ideal of S is S itself. A *left simple* semigroup is one with no proper left ideals. *Right simple semigroups* are defined analogously.
    - This is a different concept from simple groups.

<X>, the subsemigroup generated by X

- Let X be a nonempty subset of a semigroup S and let T be the set of subsemigroups of S that contain X. This set T has at least one element, the semigroup S itself.
- ∩T, the intersection of all the subsemigroups in T, is nonempty since every subsemigroup in T contains X. This intersection is a subsemigroup (easy to see that it's closed).
- In fact ∩T is the smallest subsemigroup of S that contains X. **We denote this subsemigroup by <X> and call it the *subsemigroup generated by X*.**
- If <X> = S, we call X a *generating set for S* and say that *X generates S.*
- This theorem gives a helpful characterization of <X>.

Theorem 2.6. *Let X be a non-empty subset of a semigroup S. Then* $\langle X\rangle = \{x_1x_2\ldots x_n \mid n \in \mathbf{Z}^+, x_i \in X\}$.

Definitions

- If a semigroup S is generated by a single element, i.e. $S = \langle\{x\}\rangle$ (which we denote by $\langle x\rangle$), then S is called a *monogenic semigroup*. It's the same concept as a cyclic group.
- Semigroup homomorphisms and isomorphisms are defined just as they are in group theory.
- The importance of transformation semigroups as more concrete representations of abstract semigroups is analogous to the importance of permutation groups in group theory. We can prove a semigroup analogue of Cayley's theorem, which states that *any semigroup S of order n is isomorphic to a subsemigroup of T_{n+1}.* It is n + 1 instead of n because we need to adjoin an identity to S for the proof to work.

Definitions

- A *left zero* in a semigroup S is an element x such that xy = x for all y in S. A *right zero* is an element x such that yx = x for all y in S. An element x is a *zero* of S if xy = yx = x for all y in S.
- If every element of S is a left zero, i.e. xy = x for all x,y in S, then S is called a *left zero semigroup.* A *right zero semigroup* is defined analogously.
- An equivalence relation ρ on a semigroup S is
    - a *left congruence* if $\forall$ x,y,z in S, $x \rho y \rightarrow zx \rho zy$,
    - a *right congruence* if $\forall$ x,y,z in S, $x \rho y \rightarrow xz \rho yz$,
    - a *congruence* if it is both a left congruence and a right congruence.

Definitions

- Let $1_S = \{(s, s) \mid s \in S\}$ be the identity relation on S. Let A be an ideal of S. Then $\rho_A = (A \times A) \cup 1_S$ is a congruence on S (easy to see).
- We can denote the quotient set S/ρ_A by S/A and we call S/A the *Rees factor semigroup by A.* The elements of S/A are the ρ_A-classes, which comprise A and singleton sets $\{x\}$ for each $x \in S - A$, according to the definition of ρ_A above.
- For two elements [x] and [y] of S/A, where x and y are representatives of their respective ρ_A-classes, multiplication is (well) defined by $[x][y] = [xy]$.
- It's easy to see that A is a zero of S/A.

Definitions

- This figure from [3] shows how one forms S/A from S by merging elements of A to form a zero.

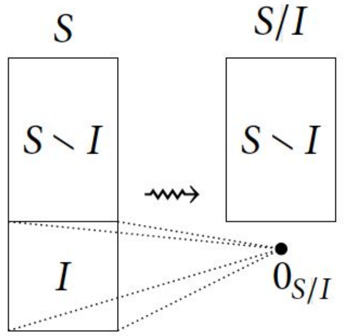


FIGURE 1.5
Forming S/I from S by merging elements of I to form a zero.

- A semigroup E is an *ideal extension* of a semigroup S by a semigroup T if S is an ideal of E and E/S ≃ T. (E/S is isomorphic to T)
- Note that since E/S contains a zero (S), T must contain a zero for this ideal extension to exist.

Semidirect product

- Let S and T be semigroups. Let '+' denote the operation ("multiplication") of S and denote the "multiplication" of T normally by t_1t_2. A *left action φ of T on S* is a function $\varphi : T \times S \rightarrow S$ defined by $(t, s) \rightarrow t * s$, where $t * s$ denotes some s' in S, such that for all $s, s_1, s_2 \in S$ and all $t, t_1, t_2 \in T$:
    - $t_1 * (t_2 * s) = (t_1t_2) * s$, and
    - $t * (s_1 + s_2) = t * s_1 + t * s_2$.
- Intuitively, the left action defines for every $t \in T$ a corresponding *endomorphism* on S (a homomorphism from S to itself). It's a homomorphism because of the second rule above.
- The *semidirect product* of S and T w.r.t φ, denoted $S \times_\varphi T$, is defined on $S \times T$ by $(s_1, t_1)(s_2, t_2) = (s_1 + t_1 * s_2, t_1t_2)$.
- If we take the trivial left action where $t * s = s$ for all $t \in T$ and $s \in S$, we get $(s_1, t_1)(s_2, t_2) = (s + s_2, t_1t_2)$, so this generalizes the direct product.

Wreath product

- New notation: Let $A = \{1, \dots, n\}$, X a set. Then X^A is the direct product of n copies of X. The copies of X are *indexed* by A.
- We can define X^A more formally (and more generally) by the set of functions from A to X. (Why?)
- Let S and T be semigroups. Define a left action φ of T on S^T by letting $y * f$ (for $y \in T$, $f \in S^T$) be such that $(y * f)(x) = f(xy)$ for all $x \in T$. (Recall: f and $(y * f)$ are functions from T to S.) This satisfies the definition of a left action.
- The *wreath product of S and T,* denoted $S \wr T$, is the semidirect product $S^T \times_\varphi T$ for the left action φ defined above.
- The product in $S \wr T$ is defined by $(f_1, t_1)(f_2, t_2) = (f_1 (t_1 * f_2), t_1 t_2)$

Division

- A semigroup S *divides* a semigroup T, denoted S ⩽ T, if S is a homomorphic image of a subsemigroup of T. So there exists a subsemigroup T' of T such that there is a surjective homomorphism from T' to S.
- Clearly ⩽ is reflexive. It's also transitive.

Proof. Let S, T, U be semigroups with $S \preceq T$ and $T \preceq U$. Then there are subsemigropus T' of T and U' of U and surjective homomorphisms $\phi : T' \to S$ and $\psi : U' \to T$. Let $U'' = \psi^{-1}(T')$, the pullback of T' under ψ. Since T' is a subsemigroup of T, U'' is a subsemigroup of U' by Lemma 2.8. Thus U'' is a subsemigroup of U. Let Φ be the restriction of ψ to U'', i.e. its domain is U''. Then the function composition $\Phi\phi : U'' \to S$ is a surjective homomorphism. Thus $S \preceq U$. □

- Lemma 2.8 is a basic result stating that the pullback of a subsemigroup under a homomorphism is a subsemigroup.

Theorem 2.10 (Prop. 7.9 in [3])

- Note: the numbering of the theorems is from my paper report.
- Let S and T be semigroups. Then S, T, and their direct product $S \times T$ divide their wreath product $S \wr T$.
- S and T divide $S \times T$ since S and T are homomorphic images of $S \times T$ under projection maps. Since division is transitive, we need only prove that $S \times T \leqslant S \wr T$.

- [Proof done on the board]

Theorems 2.11 and 2.12 (Props. 7.10 and 7.11 in [3])

- 2.11: Let M be a monoid and let E be an ideal extension of M by T. Then $E \leqslant T \wr M$.
- 2.12: If $S' \leqslant S$ and $T' \leqslant T$, then $S' \wr T' \leqslant S \wr T$.
- We will use these theorems in the proof of the Krohn-Rhodes theorem (2.12 will be used multiple times), but we don't present their proofs here.

Theorem 2.13 (Prop. 7.12 in [3])

- Let S be a semigroup and let S’ be a set in bijection with S under the mapping $x \to x'$. Define a multiplication on $S \cup S'$ as follows. Multiplication in S is as before. For all $x,y \in S$,
  a. $xy' = x'y' = y'$,
  b. $x'y = (xy)'$.
- This multiplication is associative so $S \cup S'$ is a semigroup, called the *constant extension* of S and denoted C(S).
- Th. 2.13: If $S \leqslant T$, then $C(S) \leqslant C(T)$.

- [Proof done on the board]

Th. 2.14 & Cor. 2.15 (Props. 7.13 & 7.14 in [3])

- We will also use the following two results, but we don’t prove them here.

- Theorem 2.14. Let M be a monoid and S a semigroup. Then $C(S \wr M) \leqslant C(S)^M \wr C(M)$.
- Corollary 2.15. Let M be a finite monoid and S a semigroup. Then $C(S \wr M)$ divides a wreath product of copies of C(S) and C(M).

Outline

- Introduction and motivation
- Semigroups
- The Krohn-Rhodes decomposition theorem in semigroup theory
- The Krohn-Rhodes decomposition theorem in automata theory

Definitions and Intro

- This proof follows the proof from [3].
- A semigroup is *aperiodic* if for every $x \in S$, there exists a positive integer n such that $x^n = x^{n+1}$.
- Let U_3 be the monoid obtained by adjoining an identity to a two-element right zero semigroup {a, b}. So U_3 has elements {1, a, b}.
- Note that U_3 is aperiodic since $x^2 = x$ for all x in U_3.
- Alan J. Cain, author of [3], proves a stronger statement than the one we introduced in the beginning:
- Every finite semigroup divides a wreath product of its own subgroups and copies of U_3.

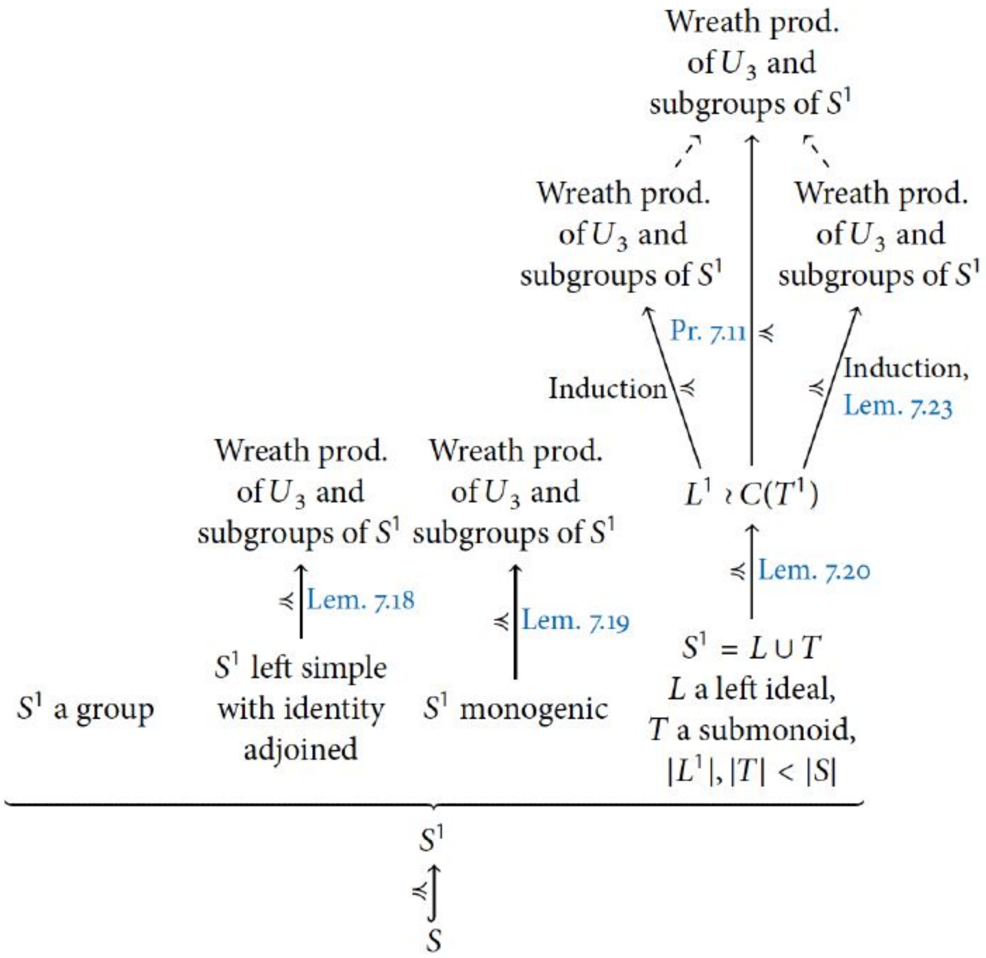
	1	a	b
1	1	a	b
a	a	a	b
b	b	a	b

TABLE 7.1
Multiplication table of U_3.

Proof outline

- Since $S \leqslant S^1$ (trivially) and division is transitive, it suffices to prove the theorem for monoids.
- We proceed by induction on the number of elements in the monoid. Lemma 3.3 forms the core of the induction; it shows that a monoid S is either a group, a left simple semigroup with an identity adjoined, monogenic, or can be decomposed as $S = L \cup T$ where L is a left ideal and T is a submonoid and L^1 and T each have fewer elements than S.
- The theorem is trivial for groups.
- Base cases:
    - 1) left simple semigroups with identities adjoined,
    - 2) monogenic semigroups.
- Inductive step:
    - $S = L \cup T$.

From [3]


Wreath prod. of U_3 and subgroups of S^1
Wreath prod. of U_3 and subgroups of S^1
Wreath prod. of U_3 and subgroups of S^1
Pr. 7.11
$\preccurlyeq$
Induction $\preccurlyeq$
$\preccurlyeq$ Induction, Lem. 7.23
Wreath prod. of U_3 and subgroups of S^1
Wreath prod. of U_3 and subgroups of S^1
$L^1 \wr C(T^1)$
$\preccurlyeq$ Lem. 7.18
$\preccurlyeq$ Lem. 7.19
$\preccurlyeq$ Lem. 7.20
S^1 a group
S^1 left simple with identity adjoined
S^1 monogenic
$S^1 = L \cup T$
L a left ideal,
T a submonoid,
$|L^1|, |T| < |S|$
S^1
$\preccurlyeq$
S

Lemma 3.1 (Lemma 7.15 in [3])

Let S be a finite semigroup. Then at least one of the following is true:

1. S is trivial;
2. S is left simple;
3. S is monogenic;
4. $S = L \cup T$, where L is a proper left ideal of S and T is a proper subsemigroup of S.

[Proof done on the board]

Lemma 3.3 (Lemma 7.16 in [3])

The following lemma is used in the proof of Lemma 3.3.

- Lemma 3.2: Let M be a finite monoid and let G be its group of units. Then M - G is either empty or an ideal. (proof omitted)

Lemma 3.3: Let S be a finite monoid. Then at least one of the following is true:

1. S is a group;
2. S is a left simple semigroup with an identity adjoined;
3. S is monogenic;
4. $S = L \cup T$, where L is a left ideal of S and T is a submonoid of S, and L^1 and T each have fewer elements than S.

[Proof done on the board]

First base case: Lemma 3.7 (Lemma 7.18 in [3])

Need three other lemmas first.

- Lemma 3.4 (7.17 in [3]). Every finite left zero semigroup divides a wreath product of copies of U_3. (proof omitted)
- Lemma 3.5. If S is a finite semigroup, then S contains an idempotent. [Proof done on board]
- Lemma 3.6 (follows from Th. 4.19 in [3]). Let S be a semigroup. If S is left simple and contains an idempotent, then S ≃ Z x G, where Z is a left zero semigroup and G is a subgroup of S. (proof omitted)

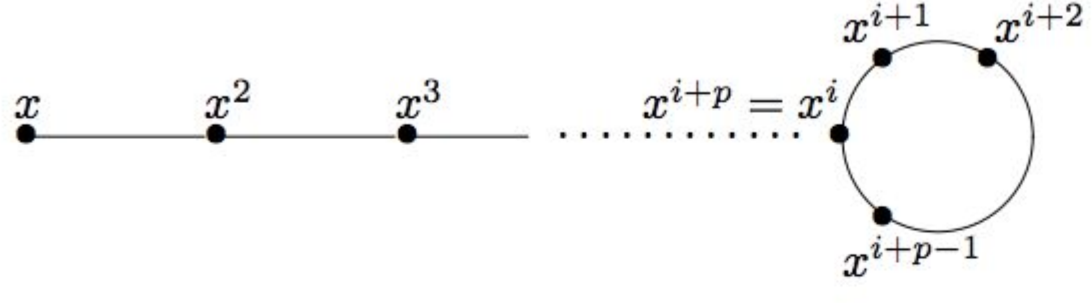
Lemma 3.7. Let S be a finite left simple semigroup. Then S^1 divides the wreath product of a subgroup of S and copies of U_3. [Proof done on board]

Second base case: Lemma 3.8 (7.19 in [3])

Let S be a finite monogenic monoid. Then S divides a wreath product of a subgroup of S and copies of U_3.

[Proof done on board]

This figure (from [4]) showing the multiplicative structure of a monogenic semigroup generated by x with *index* i and *period* p is helpful for the proof.



Two lemmas to finish the induction (proofs omitted)

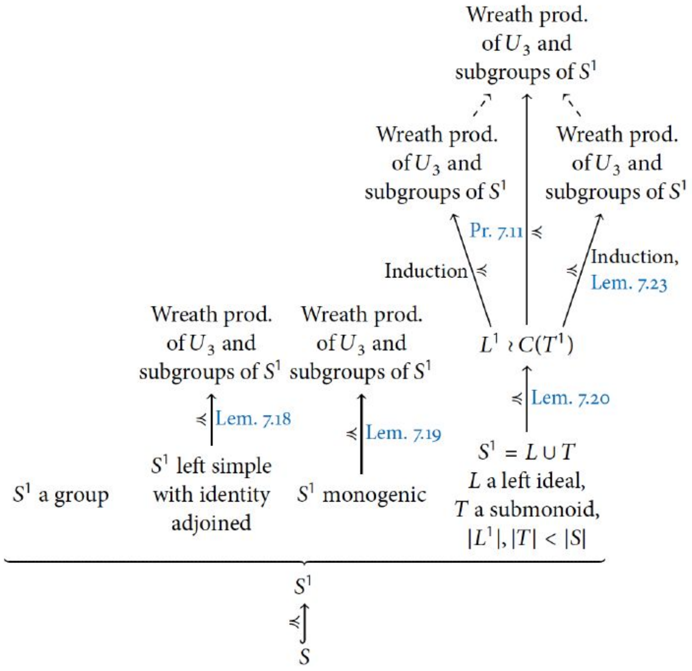
Lemma 3.9 (7.20 in [3]). Let S be a semigroup and suppose $S = L \cup T$, where L is a left ideal of S and T is a subsemigroup of S. Then $S \leqslant L^1 \wr C(T^1)$.

Lemma 3.12 (7.23 in [3]). Let S be a finite semigroup. If S divides a wreath product of groups and copies of U_3, then C(S) divides a wreath product of copies of those same groups and copies of U_3.

Krohn-Rhodes Theorem (Th. 7.24 in [3])

Every finite semigroup divides a wreath product of its own subgroups and copies of U_3.

[Proof done on board]



Outline

- Introduction and motivation
- Semigroups
- The Krohn-Rhodes decomposition theorem in semigroup theory
- The Krohn-Rhodes decomposition theorem in automata theory

Decomposition of automata

- Subgroups of S ⟵ ⟶ Permutation automata.
  - For every letter in Σ, the corresponding transformation is a permutation on the set of states of the automaton.
- Copies of U_3 ⟵ ⟶ Reset automata = flip-flops.
  - 1 = “do nothing”, a = “set”, b = “reset”.

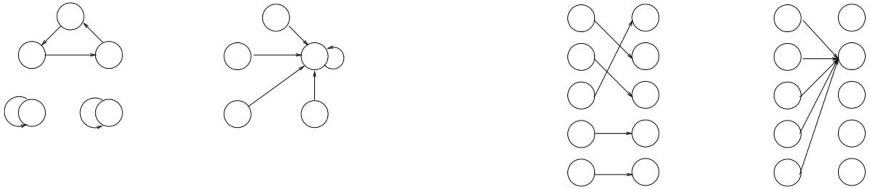


Fig. 2. A permutation and a reset illustrated as transition graphs (left) and as transformations (right).

From [5]

But what is the wreath product on automata?

- **“The *cascade* product, and its more algebraic counterpart the *wreath* product, have awkward definitions that contributed greatly to their almost total neglect from the computer science community”** [6]
- This paper [6], at long last, explains the cascade decomposition of automata in a simple way, and demonstrates small examples that make sense.
- Unfortunately, I just discovered it half an hour ago, so I didn’t have enough time to understand it and present it here.
- Thus, I refer you to [6] to learn more about the cascade decomposition, and the expressive hierarchical structure it leads to. Also, [6] does cool work on establishing a correspondence between these cascade decompositions and bit-vector algorithms, for applications in bioinformatics and other fields.

Thank you!

References

[1]: Stanford CS 103 Lecture 14.
http://web.stanford.edu/class/cs103/lectures/14/small14.pdf

[2]: A. Egri-Nagy, et. al. Computational Holonomy Decomposition of Transformation Semigroups.

[3]: Alan J. Cain. Nine Chapters on the Semigroup Art.

[4]: https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf

[5]: http://www-verimag.imag.fr/~maler/Papers/kr-new.pdf

[6]: Anne Bergeron and Sylvie Hamel.
http://www.lacim.uqam.ca/~hamel/CIAA01.ps