An Algebraic Characterization of the Strictly Piecewise Languages

Jie Fu[1], Jeffrey Heinz[2], and Herbert G. Tanner[1]

[1]Department of Mechanical Engineering
[2]Department of Linguistics and Cognitive Science

University of Delaware

May 24, 2011

TAMC 2011
University of Electro-Communications
Chofu, Japan

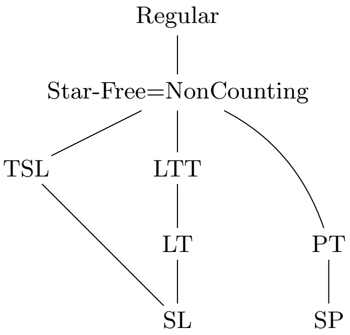
This talk

1. The Strictly Piecewise (SP) languages are those formal languages which are closed under subsequence.
2. They are a proper subclass of the regular languages; i.e. they are *subregular*.
3. This talk provides an algebraic characterization of this class: they are exactly those regular languages which are *wholly nonzero* and *right annhilating*.

*This research is supported by grant #1035577 from the National Science Foundation.

Outline

Subregular Hierarchies

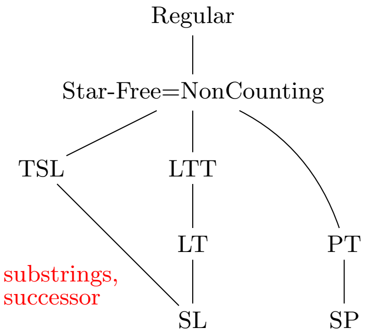


Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Subregular Hierarchies

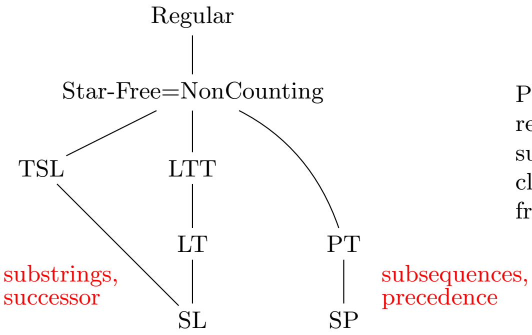


Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Subregular Hierarchies



Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

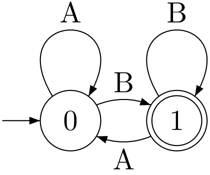
Why subregular languages?

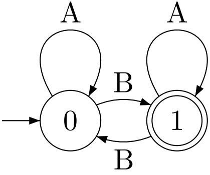
1. They provide an interesting measure of pattern complexity.
2. For particular domains, subregular language classes *better characterize* the patterns we are interested in.
   - Phonology !
   - Robotics !

We wish to obtain a better understanding of these classes. While much work characterizes subregular classes algebraically (Eilenberg, Pin, Straubing, . . .), none has addressed the SP class.

Measure of language complexity

Sequences of As and Bs which end in B	Sequences of As and Bs with an odd number of Bs
$(A+B)^*B \in SL$	$(A^*BA^*BA^*)^*A^*BA^* \notin$ star-free
Minimal deterministic finite-state automata	Minimal deterministic finite-state automata





Conclusion: The size of the DFA as given by the Nerode equivalence relation doesn't capture these distinctions.

Samala Chumash Phonotactics
Knowledge of word well-formedness

possible Chumash words	impossible Chumash words
ʃtoyonowonowaʃ	stoyonowonowaʃ
stoyonowonowas	ʃtoyonowonowas
pisotonosikiwat	pisotonoʃikiwat

1. What formal language describes this pattern?
2. By the way, *ʃtoyonowonowaʃ* means ‘it stood upright’

(Applegate 1972)

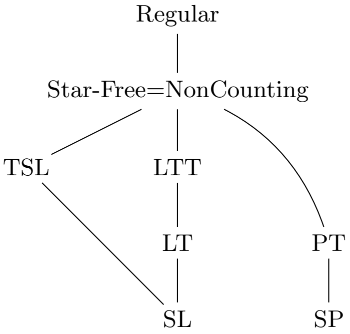
Samala Chumash Phonotactics
Knowledge of word well-formedness

possible Chumash words	impossible Chumash words
ʃtoyonowonowaʃ	stoyonowonowaʃ
stoyonowonowas	ʃtoyonowonowas
pisotonosikiwat	pisotonoʃikiwat

1. What formal language describes this pattern?
2. By the way, *ʃtoyonowonowaʃ* means ‘it stood upright’

(Applegate 1972)

Subregular Hierarchies

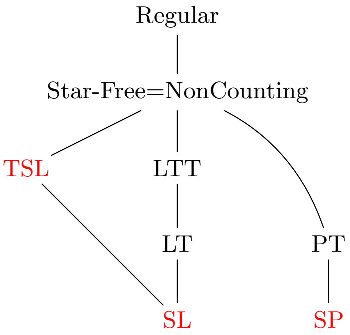


Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Subregular Hierarchies

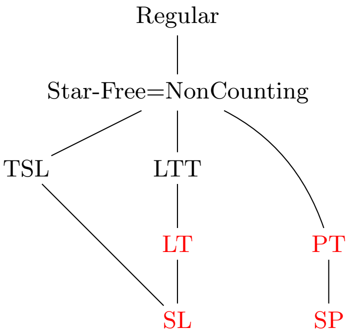


Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Subregular Hierarchies

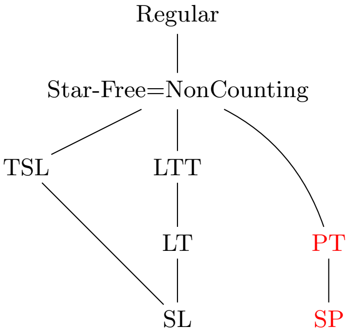


Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Subregular Hierarchies



Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Subsequences and Shuffle Ideals

Definition (Subsequence)

u is a *subsequence* of w iff $u = a_0 a_1 \cdots a_n$ and

$$w \in \Sigma^* \ a_0 \ \Sigma^* \ a_1 \ \Sigma^* \ \cdots \ \Sigma^* \ a_n \ \Sigma^*$$

We write $u \sqsubseteq_s w$.

Definition (Strictly Piecewise languages, SP)

The Strictly Piecewise languages are those closed under subsequence. I.e. $L \in SP$ if and only if for all $w \in \Sigma^*$,

$$w \in L \Leftrightarrow (\forall u \sqsubseteq_s w) \ [u \in L] \ .$$

Shuffle Ideals

Definition (Shuffle Ideal)

The *shuffle ideal* of u is

$$SI(u) = \{w : u \sqsubseteq_s w\} \ .$$

Example

$SI(aa) = \Sigma^* a \Sigma^* a \Sigma^*$.

Note $\overline{SI(u)}$ is the set of all words *not* containing the subsequence u.

Theorem (Rogers et al. 2010)

$L \in \mathrm{SP}$ iff there exists a finite set $S \subset \Sigma^*$ such that

$$L = \bigcap_{w \in S} \overline{SI(w)}\ .$$

In other words, every Strictly Piecewise language has a finite basis S, the set of ***forbidden subsequences****.*

(see also Haines 1969, Higman 1952)

Samala Chumash pattern is SP

$$L = \bigcap_{w \in S} \overline{SI(w)}$$

$$S = \{\text{sʃ}, \text{ʃs}\}$$

possible Chumash words	impossible Chumash words
ʃtoyonowonowaʃ	stoyonowonowaʃ
stoyonowonowas	ʃtoyonowonowas
pisotonosikiwat	pisotonoʃikiwat

Strictly Local

Definition (Factor)

u is a *factor* of w ($u \sqsubseteq_f w$) iff $\exists x, y \in \Sigma^*$ such that $w = xuy$.

Example

$bc \sqsubseteq_f abcd$.

Definition (Strictly Local, SL)

A language is Strictly Local$^{(*)}$ iff there is a finite set of *forbidden factors* $S \in \Sigma^*$ such that

$$L = \bigcap_{w \in S} \overline{\Sigma^* w \Sigma^*} \ .$$

Example

$L = \overline{\Sigma^* aa \Sigma^*}$ belongs to SL.

$^{(*)}$Technically, special symbols are used to demarcate the beginning and ends of words. They are ignored here for exposition.

Piecewise and Locally Testable

Subsequences

$P_{\leq k}(w) =$
$\{u : u \sqsubseteq_s w \text{ and } |u| \leq k\}$

Example

$P_{\leq 2}(abcd) =$
$\{\lambda, a, b, c, d, ab, ac, ad, bc, bd, cd\}.$

Definition: A language L is *Piecewise Testable* iff there exists some $k \in \mathbb{N}$ such that for all $u, v \in \Sigma^*$:

$$\Big[P_{\leq k}(u) = P_{\leq k}(v)\Big]$$
$$\Downarrow$$
$$\Big[u \in L \Leftrightarrow v \in L\Big]$$

Factors

$F_k(w) =$
$\{u : u \sqsubseteq_f w \text{ and } |u| = k\}$

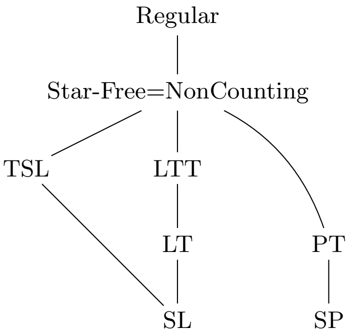
Example

$F_2(abcd) = \{ab, bc, cd\}.$

Definition: A language L is *Locally Testable* iff there exists some $k \in \mathbb{N}$ such that for all $u, v \in \Sigma^*$:

$$\Big[F_k(u) = F_k(v)\Big]$$
$$\Downarrow$$
$$\Big[u \in L \Leftrightarrow v \in L\Big]$$

Subregular Hierarchies



Proper inclusion relationships among subregular language classes (indicated from top to bottom).

TSL	Tier-based Strictly Local	PT	Piecewise Testable
LTT	Locally Threshold Testable	SL	Strictly Local
LT	Locally Testable	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum in press, Rogers et al. 2010, Heinz 2010, Heinz et al. 2011)

Outline

Semigroups, Monoids, and Zeroes

Definition

- A *semigroup* is a set with an associative operation.
- A *monoid* is a semigroup with an identity.
- A *free semigroup* (monoid) of a set S is the set of all finite sequences of one (zero) or more elements of S.
- A *zero* is an element of a semigroup such that for all $s \in S$, it is the case that $0s = s0 = 0$.

Example

Sets Σ^+ and Σ^* denote the free semigroup and free monoid of Σ, respectively.

Semigroups, Monoids, and Zeroes

Definition

- A *semigroup* is a set with an associative operation.
- A *monoid* is a semigroup with an identity.
- A *free semigroup* (monoid) of a set S is the set of all finite sequences of one (zero) or more elements of S.
- A *zero* is an element of a semigroup such that for all $s \in S$, it is the case that $0s = s0 = 0$.

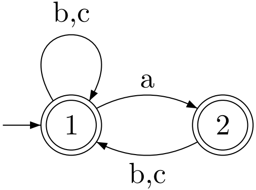
Example

Sets Σ^+ and Σ^* denote the free semigroup and free monoid of Σ, respectively.

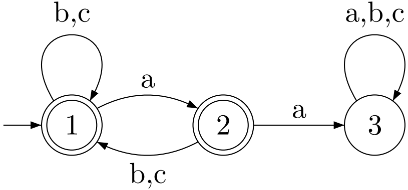
- To define the *syntactic monoid*, we need the concepts of *complete canonical automata* and the *transformation semigroup* over automata.

Complete Canonical Automata

Example



Canonical automaton of $\overline{\Sigma^* aa \Sigma^*}$.



Complete canonical automaton of $\overline{\Sigma^* aa \Sigma^*}$.

Transformation and Characteristic semigroups

Definition

Given an automaton A, its states $q_i \in Q$, and its recursively extended transition function $T : Q \times \Sigma^* \to Q$, let the *transformation* of $x \in \Sigma^*$ be

$$f_x = \begin{pmatrix} q_1 & \cdots & q_n \\ T(q_1, x) & \cdots & T(q_n, x) \end{pmatrix} .$$

Transformation Equivalence

Strings x and y are transformation-equivalent iff $f_x = f_y$.

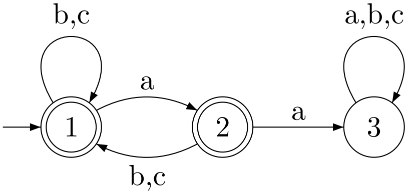
1. $F_A = \{f_x : x \in \Sigma^*\}$ is the *transformation monoid* with $f_x f_y = f_{xy}$.
2. The *characteristic monoid* is the partition of Σ^* induced by transformation equivalence with [x][y]=[xy].

Syntactic monoids

Definition (Pin 1997)

The *syntactic monoid* of a regular language L is the transformation monoid given by the complete canonical automaton.

Example



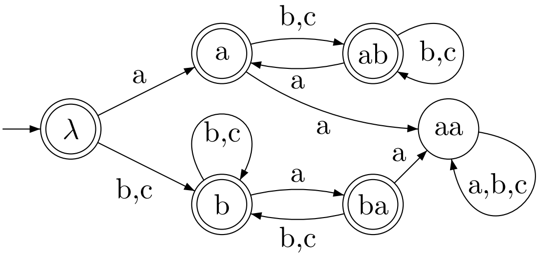
Complete canonical automaton of $\overline{\Sigma^* aa \Sigma^*}$.

F_A	1	2	-
λ	1	2	-
a	2	-	-
b	1	1	-
ab	1	-	-
ba	2	2	-
aa	-	-	-

Note $f_b = f_c, \ldots$

Monoid graphs

F_A	1	2	-
λ	1	2	-
a	2	-	-
b	1	1	-
ab	1	-	-
ba	2	2	-
aa	-	-	-



Monoid graph of the syntactic monoid of $\overline{\Sigma^* aa \Sigma^*}$.

Related Work

Theorem (Schützenberger)

A language is star-free iff its syntactic monoid is aperiodic, i.e. contains no non-trivial subgroup.

Theorem (Brzozowski and Simon, McNaughton)

A language is Locally Testable iff its syntactic monoid S is locally idempotent and commutative, i.e. for every $e, s, t \in S$ such that $e = e^2$, $(ese)^2 = (ese)$ and $(ese)(ete) = (ete)(ese)$.

Theorem (Simon)

A language is Piecewise Testable iff its syntactic monoid is $\mathcal{J}$-trivial, i.e. all cycles in the syntactic monoid are self-loops.

Outline

Zeroes

Definition (Wholly Nonzero)

An element f_x is a *zero* element of the transformation semigroup ($f_x = 0$) iff

$$f_x = \begin{pmatrix} q_1 & \dots & q_n \\ \mathbf{0} & \dots & \mathbf{0} \end{pmatrix} .$$

The corresponding zero block in the characteristic semigroup is denoted $[0]$.

Example

Considering $L = \overline{\Sigma^* aa \Sigma^*}$, it is the case that $f_{aa} = 0$ and $[0] = \Sigma^* aa \Sigma^*$.

Wholly Nonzero

Definition

Let L be a regular language, and consider its characteristic semigroup. Language L is *wholly nonzero* if and only if

$$\overline{L} = [0] \ .$$

Equivalently, for all $w \in \Sigma^*$,

$$w \in L \Leftrightarrow f_w \neq 0 \ .$$

PT Example is not Wholly Nonzero

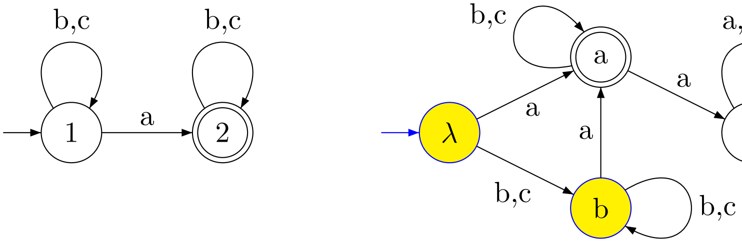


Figure: The canonical automaton and the monoid graph for $L = \{w : |w|_a = 1\}$, which is the language of all words with exactly one a.

$$f_b \neq 0 \text{ but } b \notin L.$$

Closure under prefix and suffix

Theorem

A language L is wholly nonzero if and only if L is closed under prefix and closed under suffix.

Proof sketch.

($\Rightarrow$, prefixes) Suppose L is wholly nonzero and $w = vx \in L$. If $v \notin L$ then $f_v = 0$ by assumption, contradicting $w \in L$.

($\Leftarrow$) Suppose L is closed under prefix and suffix and consider any $x \in \overline{L}$. If $f_x \neq 0$ then there are strings u, v such that $uxv \in L \Rightarrow ux \in L \Rightarrow x \in L$, contradicting the premise. $\square$

Corollary

The Strictly Piecewise languages are wholly nonzero.

Right Annhilating

Definition (Principle right ideal)

Let M be a monoid and $x \in M$. Then the *principle right ideal* generated by x is xM.

Definition (Right Annhilators)

Let M be a monoid. The set of *right annihilators* of an element $x \in M$, is $RA(x) = \{a \in M : xa = 0\}$.

Definition (Right Annihilating)

A language L is *right annihilating* iff for any element f_x in the syntactic monoid $F_A(L)$, and for all f_w in the principle right ideal generated by f_x, it is the case that

$$RA(f_x) \subseteq RA(f_w) \ .$$

Algebraic characterization of SP

Theorem

A language L is SP *iff L is wholly nonzero and right annihilating.*

Proof sketch.

($\Rightarrow$, right annhilating) Suppose L is SP. Let f_x belong to the syntactic monoid of L and let $f_x f_t = 0$. Then exists $v \sqsubseteq_s xt$ which is forbidden. For any element y in the principal right ideal of f_x it is the case that $f_x f_y f_t = 0$=0 since $v \sqsubseteq_s xyt$.

($\Leftarrow$) Suppose L is wholly nonzero and right annhilating. If $L \notin SP$ then there exists w, v such that $w \in L$ and $v \sqsubseteq_s w$ but $v \notin L$. Hence v is a zero and therefore right annhilates a prefix of w. Since L is right annhilating we can show that a suffix of v right annhilates a larger prefix of w and so on. It follows that $w \notin L$, contradicting our premise. $\square$

SP Example is Right Annihilating

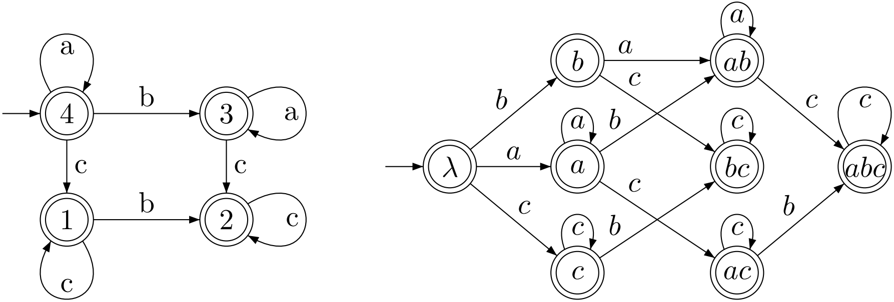


Figure: The canonical automata and the monoid graph of the syntactic monoid of $L = \overline{\mathrm{SI}(bb)} \cap \overline{\mathrm{SI}(ca)}$, i.e. the language where the subsequences bb and ca are forbidden. The 0 element is not shown in the monoid graph, but note that all missing edges go to 0.

"Missing edges propagate down" (Rogers et al. 2010)

Cayley Table for SP example

	λ	a	b	c	ab	bc	ac	abc
λ	λ	a	b	c	ab	bc	ac	abc
a	a	a	ab	ac	ab	abc	ac	abc
b	b	ab	0	bc	0	0	abc	0
c	c	0	bc	c	0	bc	0	0
ab	ab	ab	0	abc	0	0	abc	0
bc	bc	0	0	bc	0	0	0	0
ac	ac	0	abc	ac	0	abc	0	0
abc	abc	0	0	abc	0	0	0	0

Table: Cayley table for syntactic monoid for $L = \overline{\mathrm{SI}(bb)} \cap \overline{\mathrm{SI}(ca)}$.

Cayley Table for SP example

	λ	a	b	c	ab	bc	ac	abc
λ	λ	a	b	c	ab	bc	ac	abc
a	a	a	ab	ac	ab	abc	ac	abc
b	b	ab	0	bc	0	0	abc	0
c	c	0	bc	c	0	bc	0	0
ab	ab	ab	0	abc	0	0	abc	0
bc	bc	0	0	bc	0	0	0	0
ac	ac	0	abc	ac	0	abc	0	0
abc	abc	0	0	abc	0	0	0	0

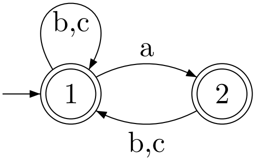
Table: Cayley table for syntactic monoid for $L = \overline{\mathrm{SI}(bb)} \cap \overline{\mathrm{SI}(ca)}$.

Cayley Table for SP example

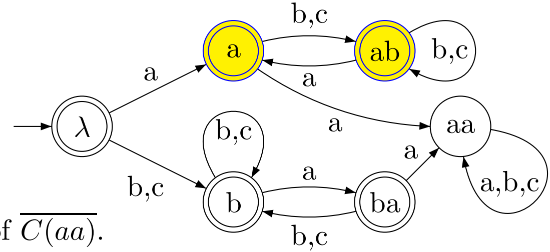
	λ	a	b	c	ab	bc	ac	abc
λ	λ	a	b	c	ab	bc	ac	abc
a	a	a	ab	ac	ab	abc	ac	abc
b	b	ab	0	bc	0	0	abc	0
c	c	0	bc	c	0	bc	0	0
ab	ab	ab	0	abc	0	0	abc	0
bc	bc	0	0	bc	0	0	0	0
ac	ac	0	abc	ac	0	abc	0	0
abc	abc	0	0	abc	0	0	0	0

Table: Cayley table for syntactic monoid for $L = \overline{\mathrm{SI}(bb)} \cap \overline{\mathrm{SI}(ca)}$.

SL example is not right annhilating



Canonical automaton of $\overline{C(aa)}$.



Monoid graph of the syntactic monoid of $\overline{\Sigma^* aa \Sigma^*}$.

Decision procedures for SP

1. These properties lead to new procedures for deciding whether a language is SP or not in time quadratic in the size of the syntactic monoid.
2. It is easy to check the wholly nonzero property.
3. It is easy to check the right annhilating property.

Open Questions

1. Are languages with syntactic monoids which are J-trivial and wholly nonzero necessarily Strictly Piecewise?
2. Are languages with syntactic monoids which are locally idempotent and commutative and wholly nonzero necessarily Strictly Local?

Summary

1. The Strictly Piecewise languages are those which are closed under subsequence.
2. They are a *subregular* class of languages.
3. Subregular classes are important in many domains, including natural language and robotics and provide a different measure of pattern complexity.
4. This talk provides an algebraic characterization of SP languages: they are exactly those regular languages which are *wholly nonzero* and *right annhilating*.

Summary

1. The Strictly Piecewise languages are those which are closed under subsequence.
2. They are a *subregular* class of languages.
3. Subregular classes are important in many domains, including natural language and robotics and provide a different measure of pattern complexity.
4. This talk provides an algebraic characterization of SP languages: they are exactly those regular languages which are *wholly nonzero* and *right annhilating*.

Thank you.