∫(͡° ͜ʖ ͡°)dx

CS 719

Topics in Mathematical Foundations of Formal Verifications

Notes By: Aryaman Maithani

Spring 2020-21

Lecture 1 (11-01-2021)

11 January 2021 09:34

Recap of Regular Languages

Different formalisms surprisingly describe the same class of lang- regular languages. Regular expressions, DFA, NFA, MSO logic.

Notation and Setup (for the rest of course)

Fix a finite alphabet Σ.

A (finite) word over Σ is a finite sequence $a_0 a_1 \cdots a_n$ of elements of Σ. $u, v, w \ldots$ are used for words.

$w = a_0 a_1 \ldots a_n$ where each $a_i \in \Sigma$.

The empty sequence corresponds to the unique word of length 0 and is denoted by ϵ, the empty word.

Σ^* = the set of all finite words over Σ. $(\epsilon \in \Sigma^*)$

$\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$ = the set of all non-empty words over Σ.

CONCATENATION of words:

$$\cdot : \Sigma^* \times \Sigma^* \longrightarrow \Sigma^*$$
$$(u, v) \longmapsto u \cdot v$$

defined in the usual manner.

→ The $\cdot$ operation on Σ^* is associative.
That is, $\forall u, v, w \in \Sigma^* : (u \cdot v) \cdot w = u \cdot (v \cdot w)$.

→ ϵ acts as an (the) identity for $\cdot$

This is an example of a monoid $(X, \star)$

$$\forall w \in \Sigma^* : \quad \epsilon \cdot w = w \cdot \epsilon = w.$$

(Another example of monoid: $(\mathbb{N}, +)$
$\mathbb{N} = \{0, 1, \dots\}$ in this course)

(Later we'll look a finite monoids.)

$$\ell : \Sigma^* \longrightarrow \mathbb{N}$$
$$u \longmapsto \text{length of } u = \ell(u)$$

Note $\quad \ell(u \cdot w) = \ell(u) + \ell(w)$
$\quad\quad \ell(\varepsilon) = \ell(0)$

Thus, ℓ is a monoid morphism.

Defn. A language L is simply a subset of Σ^*. (Language)

Given languages $L_1, L_2 \subset \Sigma^*$, we define

$$L_1 \cdot L_2 = \{ w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2 \}.$$

Regular Expressions

(Regular expressions)

$$r \equiv \phi \mid \epsilon \mid \underset{\Sigma}{\underset{\cap}{a}} \mid r_1 + r_2 \mid r_1 \cdot r_2 \mid r^*$$

$r \rightsquigarrow L(r)$ language associated to r

$L(r)$ is defined by structural induction on r.

- $L(\phi) = \phi$
- $L(\epsilon) = \{\epsilon\}$
- $L(a) = \{a\}$ $\quad (a \in \Sigma)$
- $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
- $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$ (RHS defined earlier)

- $L(r^*) = \{\varepsilon\} \cup L(r) \cup L(r)\cdot L(r) \cup L(r)\cdot L(r)\cdot L(r) \cup \cdots$

$$= \bigcup_{i=0}^{\infty} L^i \qquad \left(L^0 = \{\varepsilon\},\ L^1 = L(r),\ \underset{i \geq 1}{L^{i+1} = L^i \cdot L} \right)$$

[Example. $(ab)^* = \{\varepsilon, ab, abab, \ldots\}$.]

Def^n: A language $L \subseteq \Sigma^*$ is said to be regular if there exists a regular expression r such that

$$L(r) = L.$$

(Regular language)

Thm. Regular languages are closed under union, intersection, complementation, concatenation.

(As per our def^n using regular expressions, union & concatenation are obvious.)

Some of the above is easier to prove under diff. formalisms. One first shows that two diff. formalisms are actually same.

Def^n: (Extended reg. expressions) (Extended regular expressions)

$$r \equiv \phi \mid \epsilon \mid \underset{\Sigma}{\underset{\Uparrow}{a}} \mid r_1 + r_2 \mid r_1 \cap r_2 \mid \neg r \mid r_1 \cdot r_2 \mid r^*$$

these we can add, in view of th^m, w/o changing the class of languages

Q: What subclass of language will we get if we restrict ourselves to a subset of the operators?

Def^n: (Star-free extended reg. expressions) Exclude the $*$ operator.

(Star-free regular expressions)

Def^n: (Star-free extended reg. expressions) Exclude the $*$ operator.

(Star-free regular expressions)

Q. Which regular languages admit a star free representation?

(Non?) Example : $r = (ab)^*$

Can we rewrite this without $*$?

The "extended" is important. Else, we get trivial classes.

Lecture 2 (14-01-2021)

14 January 2021 11:35

Note that $\neg\phi = \Sigma^*$ ← can use this freely

Observe: $a^* = \neg(\underbrace{\Sigma^* \cdot b \cdot \Sigma^*}_{\text{words containing at least } b})$

Similarly $(ab)^*$ → words starting with a, ending with b, no consecutive a or b (or ϵ)

$$(ab)^* = \epsilon + [a\Sigma^* b \cap \neg(\Sigma^* aa \Sigma^* + \Sigma^* bb \Sigma^*)]$$

It is not even clear a priori whether the question "Which languages have $*$-free expression" is even decidable.

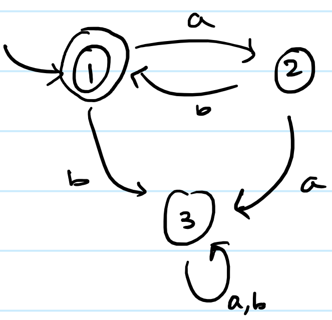
Finite state Automata (Finite state automata)

(NFA)

$$A = (Q, \Sigma, Q_0 \subseteq Q, \Delta \subseteq Q \times \Sigma \times Q, F \subseteq Q)$$

finite set — initial states — transition $(q, a, q') \in \Delta$ — final states

EXAMPLES ①



$Q = \{1, 2, 3\}$

$Q_0 = F = \{1\}$

$\Sigma = \{a, b\}$

Language accepted: $(ab)^*$

Def^n Suppose $w = a_0 \cdots a_n \in \Sigma^*$.

A run ρ of A on w is a sequence of states

$$\rho = q_0, \ldots, q_{n+1}$$

note $n+1$

such that
- $q_0 \in Q_0$
- $(q_i, a_i, q_{i+1}) \in \Delta \quad \forall\, i = 0, \ldots, n$

The run ρ is accepting if $q_{n+1} \in F$.

(Note that a word may have no run or even multiple runs.)

The language $L(A)$ of A is defined as

$$L(A) = \{ w \in \Sigma^* : A \text{ has at least one accepting run} \}.$$

A is deterministic if $|Q_0| = 1$ and

$$\forall q \in Q, \forall a \in \Sigma, \exists! \, q' \in Q \text{ s.t. } (q, a, q') \in \Delta.$$

there exists unique

In other words, $\Delta \subseteq (Q \times \Sigma) \times Q$ is a function $Q \times \Sigma \to Q$.

The example above was actually deterministic. It is called a DFA.

Thm. [TOC] (Kleene's Theorem)

Regular expressions $\equiv$ NFA $\equiv$ DFA.

(That is, all three formalisms talk about the same class of language – regular languages.)

(Recap of Proof.) Reg. Exp $\subseteq$ NFA

$$r \longmapsto A_r$$

reg exp NFA

$$L(r) = L(A_r).$$

The way to do this is by induction.

- For ε and 'a', easy.
- $r = r_1 + r_2$. We have NFAs for r_1 and r_2.
  Then, the NFA $A_{r_1} \sqcup A_{r_2}$ works.
  Allowed since non-determinism →

- $r_1 \cdot r_2$. Use ε-transitions. Idea is to take union and put ε transitions from F of A_{r_1} to Q_0 of A_{r_2}. The final states are now F of A_{r_2} and initial is Q_0 of A_{r_1}.
- r^*. Same sort of idea as above but loop on self.

NFA $\subseteq$ Reg. Exp.

$$Q = \{1, \dots, n\}$$

r_{ij} = a reg. exp. which captures the words which allow to go from i to j.

Then $r := \bigcup_{\substack{i \in Q_0 \\ j \in F}} r_{ij}$ works.

Thus, only need to figure out r_{ij}.

"Dynamic Programming"

Introduce a third parameter k.

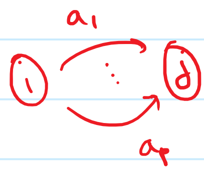
r_{ij}^k = reg. expression words w which have a

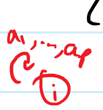
run ρ of A s.t.
(i) ρ starts at i
(ii) ρ ends at j
(iii) all intermediates states of ρ are in $\{1, \dots, k\}$.
(include $k = 0$)

Start building r_{ij}^{k} going from $k=0$ to $k=n$.
Note that $r_{ij} = r_{ij}^{n}$. ✓

$r_{ij}^{0} \equiv$ start at i, end at j, no intermediate state
$\equiv$ all those letters which allow transition from i to j. (if any)

$$r_{ij} = a_1 + a_2 + \dots + a_p \qquad (i \neq j)$$
$$= a_1 + \dots + a_p + \varepsilon \qquad (i = j)$$





$$r_{ij}^{k} = r_{ij}^{k-1} + r_{ik}^{k-1} \cdot \left(r_{kk}^{k-1}\right)^{*} \cdot r_{kj}^{k-1}$$

(We build for lower k first for all (i, j))

Thus, Reg Exp $\equiv$ NFA.

NFA $\equiv$ DFA. DFA $\subseteq$ NFA is obvious.
Converse:

$$A = (Q, \Sigma, Q_0, \Delta, F).$$

The idea to get an equivalent DFA is the powerset construction.

$$B = \left(2^{Q}, \Sigma, Q_0, \delta : 2^{Q} \times \Sigma \to 2^{Q}, F'\right)$$

Idea is to keep track of all the states that you can reach from given state.

$$\delta(X, a) = \{q \in Q : \exists q' \in X, \quad q' \xrightarrow{a} q\}.$$

Lecture 3 (18-01-2021)

18 January 2021 09:04

Today, we see another formalism to describe regular languages.
A natural way to describe a language is to give a "property" of words.

Examples:

1) Every (occurence of an) 'a' is eventually followed by a 'b'.

   aabaab ✓ bababac ✗

2) There is exactly one 'a' in the word.
3) The first position is labelled 'a'.
4) There are even number of 'a's.

We need a formal language to do so.

Formal Language : Should allow us to do "Boolean" properties like "and", "or", et cetera.

Going to use a Mathematical Logic for doing so.

First-Order Logic (over words) (First Order Logic)

Before formal defn & syntax:

An example of a formula in this logic:

predicates

unary binary unary

$$\forall x \cdot \left(\boxed{a(x)} \Rightarrow \exists y \cdot \left(\left(\boxed{x < y} \right) \wedge \boxed{b(y)} \right) \right)$$

boolean connectives

variables quantifiers

variables — going to range over positions in the word

boolean connectives

usual quantifiers

$FO[\Sigma]$ - variables :- $x, y, z, \ldots$ | range over
$x_0, x_1, x_2, \ldots$ | positions

predicates :- • letter predicates
$a \in \Sigma$, $a(x)$ says the letter 'a' at position 'x' (true / false)
• binary predicates, $y < z$
• equality $x = y$

$$\varphi \equiv a(x) \mid x < y \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x . \varphi$$

can get $\varphi \wedge \varphi$, $\varphi \Rightarrow \varphi$, $\varphi \Leftrightarrow \varphi$, $\forall x . \varphi$ using these

The above was a sentence, there was no free variable.

$$\text{first}(x) \equiv \forall y . [(x = y) \vee (x < y)]$$ ← here x is free

Given this formula, if we wish to find truth of first(x) on some word w, we need to give x.

$w = \overset{1}{a}\overset{2}{b}\overset{3}{a}\overset{4}{a}\overset{5}{b}$ $\quad w, x \leftarrow 2 \overset{?}{\models} \text{first}(x)$?

if true, we write: $w, x \leftarrow 2 \models \text{first}(x)$

else : $w, x \leftarrow 2 \not\models \text{first}(x)$

Easy to see $w, x \leftarrow 2 \not\models \text{first}(x)$. Why?

We need to check if for all positions 'p' in w:

$$w, x \leftarrow 2, y \leftarrow p \models (x = y) \vee (x < y)$$

If $P = 4$, then we check $\underbrace{\underbrace{(2=4)}_{\text{false}} \vee \underbrace{(2<4)}_{\text{true}}}_{\text{true!}}$

However, if $P = 1$, then $\underbrace{\underbrace{(2=1)}_{\text{false}} \vee \underbrace{(2<1)}_{\text{false}}}_{\text{false}}$

Then, $w, x \leftarrow 2 \not\models \text{first}(x)$. (We had the universal quantifier.)

Easy to see first (x) is true iff x is the first position.
Now, we can use first (x).

Def^n. A sentence is a formula without free variables.

(Sentence)

Example

- $\varphi \equiv \exists x . [\text{first}(x) \wedge b(x)]$ is a sentence.
  Now makes sense to ask "$aboab \models \varphi$" without any assignment. ($ababab \not\models \varphi$)
  the first position is labelled 'b'

- Exactly one 'a':
  $$\{\forall x \, \forall y \, [(a(x) \wedge a(y)) \Rightarrow x = y]\} \wedge \{\exists x . a(x)\}$$

- $(ab)^*$ ← Can you write a first order sentence which gives this regexn?

  $$\{\forall x \, [\text{first}(x) \Rightarrow a(x)]\} \wedge \{\forall x . [a(x) \Rightarrow \exists y . (s(x,y) \wedge b(y))]\}$$
  $$s(x,y) \equiv (x<y) \wedge \forall z \, (z<x \vee y<z).$$

- There are even numbers of 'a's → is regular, can come up with an automata

The other three examples were also regular. (Also expressible by FOL)
However, FOL cannot describe this logic!
But every language definable by FOL WILL be regular!

FO → FO-definable languages
REG → collection of reg. languages

$\boxed{FO \subsetneq REG}$.

We shall extend FO to MSO → Monadic Second Order (Logic).

Lecture 4 (19-01-2021)

19 January 2021 10:35

MSO (Monadic Second Order Logic - Over Words)

(MSO Monadic Second Order Logic)

FO as well

Here, we have position variables : $x, y, z, \dots, x_0, x_1, \dots$

Set of position variables : $X, Y, Z, \dots, X_0, X_1, \dots$

Predicates : $a(x)$ - $a \in \Sigma$ (Unary)

$x = y$ (Binary)

$S(x, y)$ - successor : 'y' is a successor of 'x'

(membership predicate) $X(x)$ - 'x' belongs to 'X' $[x \in X]$

$$\varphi \equiv a(x) \mid x = y \mid S(x,y) \mid X(x) \mid \varphi \vee \varphi \mid \neg \varphi \mid \exists x \cdot \varphi \mid \exists X \cdot \varphi$$

Eg of formula: $\forall X \, \exists x \cdot X(x)$

Convention (notation) : $\varphi(x_1, \dots, x_m, X_1, \dots, X_n)$ — φ is an MSO formula

$x_1, \dots, x_m$ are free posi. var

$X_1, \dots, X_n$ —"— set var.

Semantics

(Semantics) "truth"/"models" relation.

$w \in \Sigma^*$ — a finite word

$P_1, \dots, P_m$ - m positions in w,

$Q_1, \dots, Q_n$ - n sets of positions in w.

$$\boxed{w, P_1, \dots, P_m, Q_1, \dots, Q_n \models \varphi}$$

← want to define when this happens. ($x_1 \leftarrow P_1, \dots, x_m \leftarrow P_m$, $X_1 \leftarrow Q_1, \dots, X_n \leftarrow Q_n$ is understood)

($P_1, \dots, P_m$ are "concrete" positions)

($Q_1, \dots, Q_n$ —"— sets)

Defined by structural induction on φ

- $w, p_1 \models a(x_1)$ if the letter in w at position p_1 is a
- $w, p_1, Q_1 \models X_1(x_1)$ if $p_1 \in Q_1$
- $w, p_1, \ldots, p_m, Q_1, \ldots, Q_n \models \varphi(x_1, \ldots, x_m, X_1, \ldots, X_n) = \varphi_1 \vee \varphi_2$

  iff $w, p_1, \ldots, p_m, Q_1, \ldots, Q_n \models \varphi_1$ or $w, \ldots \models \varphi_2$

- $w, \ldots \models \neg\varphi$ iff $w, \ldots \not\models \varphi$

- $w, p_1, \ldots, p_m, Q_1, \ldots, Q_N \models \varphi(x_1, \ldots, x_m, X_1, \ldots, X_N) = \exists x_{m+1}\, \varphi'(x_1, \ldots, x_m, x_{m+1}, X_1, \ldots, X_n)$

  iff there exists a position p_{m+1} in w s.t.

  $w, p_1, \ldots, p_m, p_{m+1}, Q_1, \ldots, Q_N \models \varphi'(x_1, \ldots, x_{m+1}, X_1, \ldots, X_n)$.

Example $\varphi \equiv \forall X\, \exists x \cdot X(x) \wedge a(x) \equiv \forall X \cdot \underbrace{\varphi'(X)}_{\exists x \cdot X(x) \wedge a(x)}$

$aa \overset{?}{\models} \varphi$; $aa \models \varphi$ if for all subsets Q of positions in aa,

$aa, Q \models \varphi'(X)$

$aa, \{1,2\} \overset{?}{\models} \varphi' = \exists x\ X(x) \wedge a(x)$

Yes! $x = 1$ works

$aa, \emptyset \not\models \varphi'$ since $\emptyset(x)$ is never true.

Thus, $aa \not\models \varphi$.

FO: $a(x)$, $x < y$, $x = y$, boolean, $\exists x$, $\forall x$

MSO: $a(x)$, $S(x,y)$, ——"——————, $\exists X$, $\forall X$

Is FO $\subseteq$ MSO? If we had '<' in MSO, would be obvious.

As it turns out, we can write '<' in MSO, since we have set variables.

Lecture 5 (21-01-2021)

21 January 2021 11:36

$MSO[\underline{S}]$: $\varphi \equiv a(x) \mid x = y \mid \underline{S}(x,y) \mid X(x) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x \cdot \varphi \mid \exists X \cdot \varphi$

$FO[<]$: $\varphi \equiv a(x) \mid x = y \mid x < y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x \cdot \varphi$

$FO[S]$: $\varphi \equiv a(x) \mid x = y \mid S(x,y) \mid$ —"—

(Obvious semantics for all three above.)

Q. How do $FO[<]$ and $FO[S]$ compare?
Can a property in one logic be written in the other?

- If 'S' can be expressed in $FO[<]$, then $FO[S] \subseteq FO[<]$.

$$S(x,y) \equiv (x < y) \wedge \neg\big(\exists z\, ((x < z) \wedge (z < y))\big)$$

- Can '<' be expressed in $FO[S]$?
No.

Thus, $FO[S] \subsetneq FO[<]$.

- $FO[S] \subseteq MSO[S]$. Clear.

However, we also have $FO[<] \subseteq MSO[S]$.
Suffices to show '<' can be expressed in $MSO[S]$

$$x < y \equiv (\neg(x = y)) \wedge \big(\forall X\, [(X(x) \wedge SC(X)) \Rightarrow X(y)]\big)$$

$$\underset{\sim}{S}C(X) \equiv \forall z\, \forall w \{[X(z) \wedge S(z,w)] \Rightarrow X(w)\}$$

Successor closed

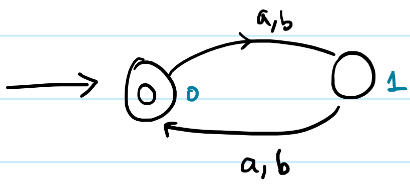
iff x is not equal to y and every subset which contains x and closed under successor also contains y.

Thus, $FO[S] \subsetneq FO[<] \subseteq MSO[S] = MSO[S,<]$.

In fact,

$$FO[<] \subsetneq MSO[S].$$

"Words of even length" can be expressed in MSO[S] but not in FO[<]. (Proof. Later. ☒)



$\longrightarrow$ Want to translate this

$\bullet \cdot \bullet \cdot \bullet \cdot \bullet \cdot \bullet \cdot$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$

$\epsilon \neq w$ has even length $\Leftrightarrow$ $\exists$ a subset X of positions in w s.t.

first(x) $\longleftarrow$ 1) X contains the first position

note this used '<' but can use that now

2) X contains every alternate position

3) X does not contain the last position

$$\exists X \Big[\big[\exists x . [\text{first}(x) \wedge X(x)]\big] \wedge \big[\forall y\, \forall z\, [\, S(y,z) \Rightarrow [X(y) \Leftrightarrow \neg X(z)]]\big] \wedge \big[\exists x . [\text{last}(x) \wedge \neg X(x)]\big] \Big]$$

[non empty words

Note $\epsilon \models \forall x . \neg(x = x)$

can OR with this

[Recall : $w = \epsilon \not\models \exists x . \varphi$
$w = \epsilon \models \forall x . \varphi$]

For convenience, we may switch to Σ^+ and forget about ϵ since we can always take care of it separately.

since we can always take care of it separately.

Def^n. Let $L \subseteq \Sigma^*$. We say L is MSO[S]-definable if $\exists$ a MSO[S] sentence φ s.t.

$$L = \{ w \mid w \models \varphi \} = L(\varphi).$$

(We will drop the "[S]" and just say "MSO")

Thm. [Büchi-Elgot] Let $L \subseteq \Sigma^*$.
L is regular iff L is MSO-definable.

More importantly, the proof (transitions b/w automata & MSO) is <u>effective</u>.

↳ Can write a program which does this conversion.

Lecture 6 (25-01-2021)

25 January 2021 02:16

Thm. (Büchi-Elgot Theorem)

L is regular iff it is MSO-definable.

can assume unique start state

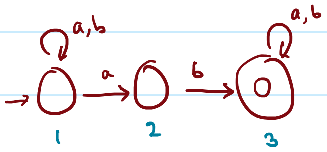
Proof. ($\Rightarrow$) Suppose $A = (Q, \Sigma, q_0, \Delta \subseteq Q \times \Sigma \times Q, F)$ be an NFA such that $L(A) = L$.

We show $\exists$ an MSO sentence φ_A s.t.

$$\forall w \in \Sigma^*, \quad w \models \varphi_A \quad \text{iff} \quad w \in L(A) = L.$$

that is, $\exists$ an accepting run of A on w

Example:



$$\varphi_A \equiv \exists x \exists y \cdot [S(x,y) \wedge a(x) \wedge b(y)]$$

(after inspecting and explicitly finding)

can't do this in general of c

		1	2	3	4	5	6	
w	$=$	a	a	b	a	b	a	
one particular run of w: ρ	$\equiv$	1	1	1	1	2	3	3
		to begin	after reading a	after a	after reading b			accepting run

Idea is to capture the state sequence using set var.

$X_1 = \{1, 2, 3, 4\}$ ← set of positions that run ρ was in state 1

$X_2 = \{5\}$

$X_3 = \{6\}$ (ignoring the final state for now)

$A = (Q, \Sigma, q_0, \Delta, F)$

$w = a_0 \; a_1 \; a_2 \; \cdots \; a_n$

$\rho = q_0 \; q_1 \; q_2 \; \cdots \; q_n \; q_{n+1}$

We encode this ρ by a set of $\{X_q\}_{q \in Q}$

X_q = the positions in ρ when it is in state q

These sets $\{X_q\}_{q\in Q}$ have the following properties

(1) $\{X_q\}_{q\in Q}$ is a partition of positions. [Some X_q may be empty, though.]

(2) The first position belongs to X_{q_0}.

(3) If two consecutive positions $p < p'$ are in the sets X_q and $X_{q'}$, respectively, then the letter at position p allows to move from q to q'.

(1) — (3) are saying that it is a valid run

Accepting run

(4) If the last position is in X_q, then there is a transition from q on the last letter to a final state.

$$Q = \{0, 1, \ldots, m\}$$

q_0

To make φ_A s.t. $w \models \varphi_A$ iff A accepts w.

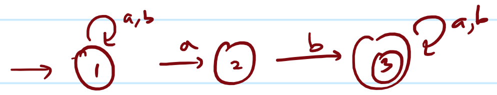
$$\begin{aligned}\varphi_A \equiv \exists X_0 \exists X_1 \cdots \exists X_m : \Big[&\{\text{partition}(X_0, X_1, \ldots, X_m)\} \wedge \\ &\{\text{first-position-is-in-}X_0\} \wedge \\ &\Big\{\forall x \forall y \Big[S(x,y) \Rightarrow \bigvee_{(q,a,q')\in\Delta} \big(X_q(x) \wedge X_{q'}(y) \wedge a(x)\big)\Big]\Big\} \wedge \\ &\Big\{\exists x \cdot \Big[\text{last}(x) \wedge \bigvee_{\substack{(q,a,q')\in\Delta \\ \text{and } q'\in F}} \big(X_q(x) \wedge a(x)\big)\Big]\Big\}\Big]\end{aligned}$$

where

$$\text{partition}(X_0, \ldots, X_m) \equiv \forall x \Big[\Big(\bigvee_{i=0}^{m} X_i(x)\Big) \wedge \Big(\bigwedge_{i\neq j} \neg\big(X_i(x) \wedge X_j(x)\big)\Big)\Big]$$

$$\text{first-position-is-in-}X_0 \equiv \exists x \,[\text{first}(x) \wedge X_0(x)]$$

For example :



$$\varphi_A \equiv \exists X_1 \exists X_2 \exists X_3 : \{\text{partition}(X_1, X_2, X_3)\} \wedge$$

$$
\begin{aligned}
\varphi_A \equiv \exists X_1\, \exists X_2\, \exists X_3 :\ & \{\text{partition}(X_1, X_2, X_3)\} \wedge \\
& \{\text{first-in-}X_1\} \wedge \\
\{\forall x\, \forall y : S(x,y) \Rightarrow \Big[\ & (X_1(x) \wedge a(x) \wedge \overset{(1,a,1)}{X_1(y)}) \vee \\
& (X_1(x) \wedge b(x) \wedge \overset{(1,b,1)}{X_1(y)}) \vee \\
& (X_1(x) \wedge a(x) \wedge \overset{(1,a,2)}{X_2(y)}) \vee \\
& (X_2(x) \wedge b(x) \wedge X_3(y)) \vee \\
& (X_3(x) \wedge a(x) \wedge X_3(y)) \vee \\
& (X_3(x) \wedge a(x) \wedge X_3(y))\Big]\} \wedge \\
\Big\{\exists x\ \text{last}(x) \wedge \Big[\ & (X_2(x) \wedge b(x)) \vee \\
& (X_3(x) \wedge a(x)) \vee \\
& (X_3(x) \wedge b(x))\Big]\Big\}
\end{aligned}
$$

Can add the empty word separately if required!

The above is a nice construction since the "length of formula" is roughly that of the automaton!

Lecture 7 (28-01-2021)

28 January 2021 11:30

Last time, we proved one direction of the Büchi-Elgot Theorem.
Namely, if L is regular, then L is MSO-definable.
Now, we see $(\Leftarrow)$.

Proof. MSO_0 - logic – eliminate position variables using 'singleton' set variables

atomic predicates: $Sing(X)$ – "X" is a singleton set

$a(x) \rightsquigarrow a(X)$ – every position in "X" is "a"

$S(x,y) \rightsquigarrow S(X,Y)$ – X and Y are singletons and the corresp. positions are related by S

$\left.\begin{matrix} x = y \\ X(x) \end{matrix}\right\} \rightsquigarrow (X \subseteq Y)$ or $subset(X,Y)$ – X is a subset of Y

Claim. MSO and MSO_0 have the same expressive power. ☒

Goal: MSO_0 sentence to automata translation.
The above is done by structural induction on the formula.

$\varphi(X_1, \ldots, X_n)$ – MSO_0-formula with n free variables
(only need to look at set variables)

$w, Q_1, \ldots, Q_n \overset{?}{\models} \varphi(X_1, \ldots, X_n)$
encode this information by a word over an extended alphabet

Example:

$$w = \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ a & b & a & a & b & a \end{array}$$

$X_1 = \{1,3,4\}$

$X_2 = \{3,4,6\}$

Construct $w' = \begin{pmatrix} a \\ 1 \\ 0 \end{pmatrix}\begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix}\begin{pmatrix} a \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} a \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix}\begin{pmatrix} a \\ 0 \\ 1 \end{pmatrix}$

We have a new alphabet $\Sigma^* = \Sigma \times \{0,1\}^n$

$\varphi(x_1, \dots, x_n) \rightsquigarrow A_\varphi$ ← construct automata s.t.

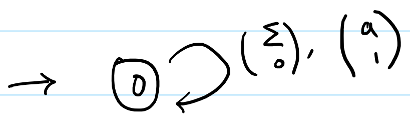
$\forall w' \in \Sigma'^*$, $\quad w' \models \varphi$ iff A_φ accepts w'

Let us now construct A_φ by structural induction.

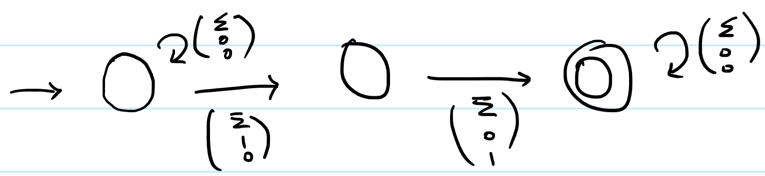
Base cases:

$\varphi(x_1) = \mathrm{Sing}(x_1) \rightsquigarrow A_\varphi$ over $\Sigma \times \{0,1\}$

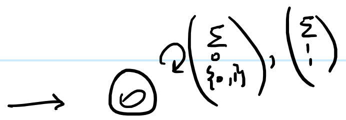
$\varphi(x_1) = a(x_1) \rightsquigarrow A_\varphi$ over $\Sigma \times \{0,1\}$



$\varphi(x_1, x_2) = S(x_1, x_2) \rightsquigarrow A_\varphi$ over $\Sigma \times \{0,1\} \times \{0,1\}$



$\varphi(x_1, x_2) = x_1 \subseteq x_2 \rightsquigarrow \Sigma \times \{0,1\}^2$



- $\varphi(x_1, \dots, x_n) = \varphi_1(x_1, \dots, x_n) \vee \varphi_2(x_1, \dots, x_n)$

can assume free variables

induction

We have A_{φ_1} and A_{φ_2}. We know how to construct union of automata. Thus, we are done.

- $\varphi \equiv \neg \psi$. Have A_ψ, can construct automata for $\neg\psi$.

  (toggle the final states, if DFA.)

- $\varphi(X_1, \ldots, X_n) = \exists X_{n+1}\ \varphi'(X_1, \ldots, X_{n+1})$

Lecture 8 (01-02-2021)

01 February 2021 09:25

φ – MSO_0 formula

$\varphi \rightsquigarrow A_\varphi$ by structural induction on φ

$$\varphi(X_1, \dots, X_n) \equiv \exists X_{n+1} \; \varphi'(X_1, \dots, X_n, X_{n+1})$$

By induction, we have $A_{\varphi'}$ over $\Sigma \times \{0,1\}^{n+1}$ such that

$\forall w \in (\Sigma \times \{0,1\}^{n+1})^*$, $w' \models \varphi' \Leftrightarrow A_{\varphi'}$ accepts w'

Goal: to construct A_φ corresponding to φ over $\Sigma \times \{0,1\}^n$.

$\forall w \in (\Sigma \times \{0,1\}^n)^*$, $w \models \varphi$

iff $\exists$ a subset of positions Q of pos. in w s.t.

$$w, X_{n+1} \leftarrow Q \models \varphi$$

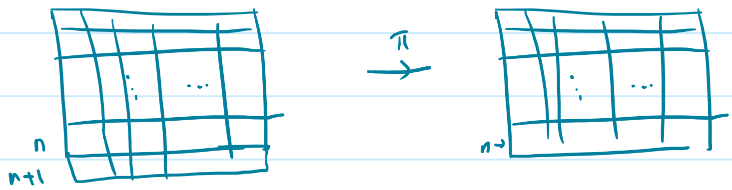
Consider the projection map $\pi : \Sigma \times \{0,1\}^{n+1} \longrightarrow \Sigma \times \{0,1\}^n$

$$(a, b_1, \dots, b_{n+1}) \longmapsto (a, b_1, \dots, b_n).$$

This extends to a map (homomorphism) (which we call π again) as

$$\pi : (\Sigma \times \{0,1\}^{n+1})^* \longrightarrow (\Sigma \times \{0,1\}^n)^*$$

which acts pointwise.



Thus, $w \models \varphi$ iff $\exists w' \in (\Sigma \times \{0,1\}^{n+1})^*$ s.t. $\pi(w') = w$

and $w' \models \varphi'$.

Note $L(\varphi') \subseteq (\Sigma \times \{0,1\}^{n+1})^*$, $L(\varphi) \subseteq (\Sigma \times \{0,1\}^n)^*$.

By our above discussion, we have:

$$\pi(L(\varphi')) = L(\varphi).$$

Note that $L(\varphi')$ regular $\Rightarrow$ $\pi(L(\varphi'))$ is regular since π is a homomorphism.

$A_{\varphi'} = (Q, \Sigma \times \{0,1\}^{n+1}, \delta', F)$ ← given

keep some initial state

$A_\varphi = (Q, \Sigma \times \{0,1\}^n, \Delta, F)$ ← construct, where

$$\Delta : q \xrightarrow{(a, b_1, \ldots, b_n)} q' \quad \text{if} \quad \exists b_{n+1} \in \{0,1\} \quad q \xrightarrow{(a, b_1, \ldots, b_{n+1})} q' \text{ in } \delta'$$

(Basically take the automaton for $A_{\varphi'}$ and erase the last bit from all transition labels.)

We assumed $A_{\varphi'}$ was a DFA but A_φ will likely be an NFA. So if we wish to stick to DFAs, this step could cause an exponential blow up.

This finishes the $MSO_0 \rightsquigarrow$ automaton construction.

Remarks about complexity:

Q. What about the size of the automata? (Asymptotic sense)

- How do we construct? NFA or DFA?

  DFA → $\neg$ is easy but $\exists$ is hard
  ↳ poly ↳ exp

  NFA → $\exists$ is easy but $\neg$ is not

- Size $2^{2^{\cdot^{\cdot^{\cdot^2}}}} \} O(n)$ where $n \to$ size of formula

  ↳ non-elementary, the length of tower is not fixed

Very bad! :(

Maybe it was our fault? Better construction exists?

Sadly, no. There is a lower bound which is also non-elementary.

MONA → software that does this translation

Connection between logic and automata very rich. Büchi did this back in '60s. Has been used in formal verification extensively.

Lecture 9 (02-02-2021)

02 February 2021 10:22

Myhill-Nerode Theorem about regular languages

Recap on equivalence relations: Fix a set X. (any cardinality)

Defn. An equivalence relation R on X is a binary relation $R \subseteq X \times X$ which is

(1) reflexive, i.e., $\forall x \in X: (x, x) \in R$ or xRx,

(2) symmetric, i.e., $\forall x, y \in X: \quad xRy \Rightarrow yRx$,

(3) transitive, i.e., $\forall x, y, z \in X: \quad xRy$ and $yRz \Rightarrow xRz$.

(Equivalence relation, equivalence class)

Fix an equivalence relation R:

For $x \in X$, we define

$$[x]_R = \{y \in X : xRy\}.$$

equivalence class of x

By reflexivity, $x \in [x]_R$. In particular, $[x]_R \neq \emptyset$.

Claim. $\forall x, y \in X: \quad [x]_R = [y]_R$ or $[x]_R \cap [y]_R = \emptyset$.

Proof. Suppose $[x]_R \cap [y]_R \neq \emptyset$. We show $[x]_R = [y]_R$.

Let $z \in [x]_R \cap [y]_R$.

Thus, xRz and yRz. $\quad yRz \Rightarrow zRy$.

xRz and $zRy \Rightarrow xRy$.

Now, if $y' \in [y]_R$, then yRy' and hence, xRy'.

$\therefore [y]_R \subset [x]_R$. Similarly, $[x]_R \subset [y]_R$. ∎

Thus, the equivalences classes of R partition X.

Usually, we use $\sim$ instead of R to denote an equivalence relation.

Def^n Let $\sim$ be an equivalence relation on X.

$X/\!\sim \; := \{ [x]_\sim : x \in X \}$

$= $ the set of all equivalence classes for $\sim$.

Example $\mathbb{Z} = \{ \dots, -2, -1, 0, 1, 2, \dots \}$.

$\sim$ on $\mathbb{Z}$: $x \sim y$ iff $3 \mid x-y$.

That is, $\exists m \in \mathbb{Z}$ s.t. $3m = x-y$.

Then, $\sim$ is an equivalence relation.

$[0]_\sim = \{ x \in \mathbb{Z} : 0 \sim x \}$

$= \{ x \in \mathbb{Z} : x$ is a multiple of $3 \}$

$= \{ \dots, -3, 0, 3, 6, \dots \}$.

$[1]_\sim = \{ \dots, -2, 1, 4, 7, \dots \}$

$[2]_\sim = \{ \dots, -1, 2, 5, 8, \dots \}$

all classes

Def^n (Finite index)

We say $\sim$ is of **finite index** if $X/\!\sim$ is finite.

Example: $\sim$ on $\mathbb{Z}$ defined above.

Σ^* – the set of all finite words over Σ.

Let $\sim$ be an equivalence relation on Σ^*.

We say:

1) $\sim$ is a **right congruence** if (right congruence)

$\forall x, y, z \in \Sigma^* : \quad x \sim y \Rightarrow xz \sim yz$

2) $\sim$ Saturates a language L if (saturates)

$\forall x, y \in \Sigma^*$: $x \sim y \Rightarrow (x \in L \Leftrightarrow y \in L)$

This basically means that either $[x]_\sim \subseteq L$ or $[x]_\sim \cap L = \emptyset$.

In particular, L is the union of (some) equivalence classes.

$$L = \bigcup_{x \in L} [x]_\sim \qquad (\subseteq, \text{ in general.})$$

Thm. (Myhill-Nerode Theorem)

A language L is regular iff there is a right congruence of finite index which saturates L.

Proof. ($\Rightarrow$) Let $L = L(A)$ where A is the DFA

$$A = (Q, q_0, \Sigma, \delta : Q \times \Sigma \to Q, F).$$

We define the relation $\sim_A$ on Σ^*:

$$x \sim_A y \quad \text{iff} \quad \delta(q_0, x) = \delta(q_0, y).$$

(Extend $\delta(q_0, \cdot)$ inductively on Σ^*.)

The above is indeed an equiv. relation. (Easy.)

- Right congruence: Suppose $x \sim_A y$. Then, $\delta(q_0, x) = \delta(q_0, y)$.
  Let $z \in \Sigma^*$ be arbitrary. We note

$$\delta(q_0, xz) = \delta(\delta(q_0, x), z)$$
$$\delta(q_0, yz) = \delta(\delta(q_0, y), z)$$

$\therefore \; xz \sim_A yz.$

- Finite index: There are at most $|Q| < \infty$ many states.

- Saturates: $x \in L \Leftrightarrow \delta(q_0, x) \in F$. Conclude. $\blacksquare$

Lecture 10 (04-02-2021)

04 February 2021 11:38

→ Satisfiability problem -

- Is there an algorithm to check if an MSO[Σ]-sentence φ is satisfiable?

(Defⁿ) φ is satisfiable if $\exists$ a finite word $w \in \Sigma^*$ such that $w \models \varphi$.

Ans. Yes! $\varphi \rightsquigarrow A_\varphi$ can be done algorithmically.
Can check if $L(A_\varphi) = \phi$ ← doable.
(decidable)

→ WS1S — weak second order theory of 1 successor

$(\mathbb{N}, +, \cdot) \rightarrow$ first order logic to write properties of natural numbers

$x, y, z, \ldots$ ← first order variables, range over $\mathbb{N}$

$$x + y = z \mid x * y = z \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x\, \varphi$$

$\text{Zero}(x) \equiv (x + x = x)$

$\text{non-prime}(x) \equiv \exists y \exists z\ (y * z = x) \wedge \neg(y = x) \wedge \neg(z = x)$

(0 and 1 possibly not considered correctly.)

$\text{even}(x) \equiv \exists y \cdot (y + y = x)$

$\varphi_0 \equiv \forall x .\ \text{even}(x) \Rightarrow \exists y\ \exists z\ \text{prime}(y) \wedge \text{prime}(z) \wedge (x = y + z)$

[Goldbach's conjecture with 0 and 2 accounted for

(Hilbert, 1900) $S = (\mathbb{N}, +, \cdot)$

$Th(S) = \{ \varphi$ a FO sentence which is true over $(\mathbb{N}, +, \cdot)$ $\}$

Is there a mechanical procedure (algorithm) for checking if a given FO-sentence is true in $(\mathbb{N}, +, \cdot)$?

(1930–1940) Gödel: No.

(Hilbert's dream shattered. :-)

(1960s) Büchi: Monadic Th $(\mathbb{N}, +)$ is also undecidable.

Is Monadic $(\mathbb{N}, S)$ decidable?

successor

$S1S = \{ \varphi$ - MSO sentence which is true in $(\mathbb{N}, S) \}$

→ Is S1S decidable? Yes. Büchi showed this.

→ WS1S – weak S1S

In the quantifiers like $\forall X\, \varphi(X)$, X only ranges over finite subsets of $\mathbb{N}$.

$$(\mathbb{N}, S) \models_{S1S} \exists X \cdot \forall x \cdot X(x)$$

$$(\mathbb{N}, S) \not\models_{WS1S} \exists X \cdot \forall x \cdot X(x)$$

Lecture 11 (08-02-2021)

08 February 2021 09:35

Myhill-Nerode: L is regular iff there is a right congruence of finite index with saturates L.

Proof Had seen ($\Rightarrow$) by taking an automaton $A = (Q, q_0, \Sigma, \delta : Q \times \Sigma \to Q, F)$ and defining $x \sim_A y \equiv \delta(q_0, x) = \delta(q_0, y)$.

($\Leftarrow$) Let $\sim$ be a right congruence of finite index

~~Then~~, ~~saturates~~ $\Sigma^*/\sim$ is finite.

Define

$$A_\sim = (Q, q_0, \Sigma, \delta : Q \times \Sigma \to Q, F)$$

where

$$q_0 = [\epsilon]_\sim,$$

$$\delta : \delta(Q[x]_\sim, \Sigma\ a) \to Q \quad [x.a]_\sim \text{ defined as}$$

well-defined ?

Yes.

If $x \sim y$, then $x.a \sim y.a$ since $\sim$ is a right congruence.

$$F = \{[w]_\sim : w \in L\}.$$

Claim. $L(A_\sim) = L$

Proof. ($\supseteq$) If $w = a_0 \cdots a_n \in L$, then $\delta(q_0, w) = [a_0 \cdots a_n]_\sim \in F$.

($\subseteq$) If $w \in L(A_\sim)$, then $w = a_0 \cdots a_n$ s.t. $[a_0 \cdots a_n]_\sim = [w']_\sim$ for some $w' \in L$. That is, $w \sim w' \in L$. By saturation, $w \in L$. ∎

Defn (Syntactic Congruence)

Let $L \subseteq \Sigma^*$ be a language, not necessarily regular.
We define $\sim_L$ on Σ^* as :

$$x \sim_L y \equiv \forall z \in \Sigma^* (xz \in L \Leftrightarrow yz \in L).$$

Straightforward check that :
- $\sim_L$ is an equivalence relation
- $\sim_L$ saturates L (take $z = \epsilon$)
- $\sim_L$ is a right congruence

Ex. "Compute" $\sim_L$ for $L = \{a^n b^n : n \geq 0\}$.

Claim. $\sim_L$ is the coarsest right congruence which saturates L.
(In other words, let $\sim$ be any right congruence saturating L, then $x \sim y \Rightarrow x \sim_L y$. (That is, $[x]_\sim \subseteq [x]_{\sim_L}$ $\forall x$.))

Proof. Let $x \sim y$. To show: $x \sim_L y$.
Let $z \in \Sigma^*$ be s.t. $xz \in L$.
Then, $xz \sim yz$ since $\sim$ is a right cong.
Then, $yz \in L$ since $\sim$ saturates L.
z was arbit. $\therefore$ $\forall z \in \Sigma^*$: $xz \in L \Rightarrow yz \in L$.
By symmetry, $\forall z \in \Sigma^*$: $xz \in L \Leftrightarrow yz \in L$.

Thus, $x \sim_L y$. ☒

Note that coarsest means the "fewest" equiv. classes.

Thm. (Myhill-Nerode) L is regular iff $\sim_L$ is of finite index.

Proof. ($\Rightarrow$) Let A be a DFA s.t. $L = L(A)$.
We had created $\sim_A$ of finite index $\rightarrow$ right congr, sat. L.
Thus, $\sim_L$ is coarser than $\sim_A$.
$\therefore$ $|\Sigma^*/\sim_L| \leq |\Sigma^*/\sim_A| < \infty$.
$\therefore$ $\sim_L$ has finite index as well.

($\Leftarrow$) By Myhill-Nerode. $\blacksquare$

Remark. The automaton $A_{\sim_L}$ corresponding to $\sim_L$ is the minimum automaton of L.

Lecture 12 (09-02-2021)

09 February 2021 10:36

$\sim$ is an equivalence relation on Σ^*.

Defn. $\sim$ is a congruence if $\forall x, y, z, w \in \Sigma^*$: (congruence)

$$x \sim y \Rightarrow zxw \sim zyw.$$

Thm. L is regular iff there is a congruence of finite index which saturates L.

Proof. ($\Leftarrow$) Follows from Myhill-Nerode since a congruence is also a right congruence.

($\Rightarrow$) $L = L(A)$ where $A = (Q, q_0, \Sigma, \delta : Q \times \Sigma \to Q, F)$.

Define $\sim_A$ on Σ^* by

$$x \sim_A y \equiv \forall q \in Q : \delta(q, x) = \delta(q, y)$$

[Given any $w \in \Sigma^*$, we get a function $f_w : Q \to Q$
(effect function) $q \mapsto \delta(q, w)$

Now, $w \sim_A w'$ iff $f_w = f_{w'}$, that is, the two functions are equal.]

→ $\sim_A$ is an equivalence relation, clearly as can be seen by looking at f_x and f_y.

→ $\sim_A$ is a congruence: Let $x, y, z, w \in \Sigma^*$ be s.t. $x \sim_A y$.

Then, $f_{zxw} = f_w \circ f_x \circ f_z \underset{x \sim_A y}{=} f_w \circ f_y \circ f_z = f_{zyw}$

$\therefore zxw \sim_A zyw$.

→ $\sim_A$ is of finite index: There are only $|Q|^{|Q|} < \infty$ many functions of the form $Q \to Q$. Thus, there are at most $|Q|^{|Q|}$ such distinct effect functions.

→ $\sim_A$ saturate L : Let $x \sim_A y$.

Then, $x \in L \Leftrightarrow f_x(q_0) \in L \Leftrightarrow f_y(q_0) \in L \Leftrightarrow y \in L$. ∎

Def^n (Syntactic congruence of a language)

Let $L \in \Sigma^*$. $x \sim_L y \equiv \forall z, w \in \Sigma^* \; (zxw \in L \Leftrightarrow zyw)$

Ex. (1) $\sim_L$ is a congruence which saturates L.

(2) L is regular iff $\sim_L$ is of finite index.

(3) If $\sim$ is a congruence which saturates L, then

$$\forall x, y : x \sim y \Rightarrow x \sim_L y.$$

That is, $\sim_L$ is the coarsest congruence which saturates L.

Def^n. The syntactic monoid of L

Let $\sim_L$ denote the syntactic congruence.

Consider the set $M_L = \Sigma^*/\sim_L$.

$$\cdot : M_L \times M_L \longrightarrow M_L$$

$$(c_1, c_2) \longmapsto c_1 \cdot c_2$$

where

$$[w_1]_{\sim_L} \cdot [w_2]_{\sim_L} = [w_1 w_2]_{\sim_L}.$$

Well defined : If $w_1 \sim w_1'$ and $w_2 \sim w_2'$, then :

$$\underbrace{w_1 w_2 \sim w_1 w_2'}_{\text{left cong}} \; \underbrace{\sim w_1' w_2'}_{\text{right cong}}$$

Then, $(M_L, \cdot, [\varepsilon])$ is a monoid, called the syntactic monoid of L.

To see that it is a monoid:

1) Associative: $([w_1]\cdot[w_2])\cdot[w_3] = [w_1w_2]\cdot[w_3]$

$$= [(w_1w_2)w_3] = [w_1(w_2w_3)]$$

$$= [w_1]\cdot[w_2w_3] = [w_1]\cdot([w_2]\cdot[w_3]).$$

Thus, $c_1\cdot(c_2\cdot c_3) = (c_1\cdot c_2)\cdot c_3 \quad \forall\, c_1, c_2, c_3 \in M_L.$

2) Unital: $[\varepsilon]\cdot[w] = [\varepsilon . w] = [w] = [w\,\varepsilon] = [w][\varepsilon] \quad \forall\, w \in M_L.$

That is, $c_0 = [\varepsilon] \in M_L$ satisfies $c_0\cdot c = c = c\cdot c_0 \quad \forall\, c \in M_L.$

Recall: A monoid is a set with a binary operation which is associative and has an identity.

Ex. (1) $(\mathbb{Z}, +, 0)$

(2) $(\mathbb{N}, +, 0)$

(3) $(\Sigma^*, \cdot, \varepsilon)$

infinite

(4) any group is a monoid

(5) $(\mathbb{Z}_n, +, 0) \longrightarrow$ finite monoid

$\mathbb{Z}_n = \{0, \dots, n-1\}$, $+$ = addition modulo n

(6) Fix a set X.

$\mathcal{F}(X)$ = the set of all functions from X to X.

$\circ : \mathcal{F}(X) \times \mathcal{F}(X) \longrightarrow \mathcal{F}(X)$

$(f, g) \longmapsto f\circ g$

$(\mathcal{F}(X), \circ, id_X)$ is a monoid.

Thm. L is regular iff M_L is finite.

Lecture 13 (11-02-2021)

11 February 2021 11:31

→ Fix a monoid $(M, \cdot, e)$.

A submonoid of M is a subset $N \subseteq M$ s.t.

(1) $e \in N$

(2) N is closed under $\circ$.

(More precisely, $(N, \cdot|_N, e)$ is the submonoid. $\cdot|_N : N \times N \to N$ makes sense. Thus, a submonoid is a monoid in itself.)

(Submonoid)

Ex. The identity element is unique.

(Proof) $e' = e \cdot e' = e$.

Def^n. (Homomorphisms between monoids)

A (homo)morphism from $(M, \cdot, e)$ to $(N, *, f)$ is a function $h: M \to N$ such that

(1) $h(e) = f$

(2) $\forall m_1, m_2 \in M: \quad h(m_1 \cdot m_2) = h(m_1) * h(m_2)$.

Example. ① Let $N \subseteq M$ be a submonoid. Then $i: N \hookrightarrow M$, $n \mapsto n$ is a homomorphism.

② $h: (\Sigma^*, \cdot, \epsilon) \longrightarrow (\mathbb{N}, +, 0)$

$h(x) = \text{length}(x)$ is a morphism.

Def^n (Recognise) Let $L \subseteq \Sigma^*$ and $h: \Sigma^* \to M$ be a morphism.

We say that h recognises L if there is a subset $X \subseteq M$ such that $h^{-1}(X) = L$.

↳ not the same as $X = h(L)$, btw!

Note that if at all, h recognises L, then $X = h(L)$ will work.

We say that L is recognised by M, if there exists a morphism

$h : \Sigma^* \longrightarrow M$ that recognises L.

Another way to see: Define $\sim_h$ on Σ^*

$x \sim_h y$ if $h(x) = h(y)$.

($\sim_h$ is indeed an equivalence relation.)

$\exists X : h^{-1}(X) = L$ iff $\sim_h$ saturates L.

That is, L is a union of $\sim_h$ equivalence classes.

Thm. L is a regular language iff L is recognised by a morphism into a finite monoid.

Proof. $(\Rightarrow)$ $L = L(A)$ where $A = (Q, q_0, \Sigma, \delta : Q \times \Sigma \to Q, F \subseteq Q)$.

Notation: Let $x \in \Sigma^*$. $\hat{\delta}_x : Q \to Q$ is a function defined by $\hat{\delta}_x(q) = \delta(q, x)$. ← transition/effect function of the word x

$\hat{\delta}_{xy} = \hat{\delta}_x \circ \hat{\delta}_y$ ← composition in reverse! $(f \circ g)(q) := g(f(q))$

Define $M = \{ \hat{\delta}_x \mid x \in \Sigma^* \}$. ← set of all transition functions

Since $\hat{\delta}_x \circ \hat{\delta}_y = \hat{\delta}_{xy}$, M is closed under $\circ$.
Moreover, $\hat{\delta}_\epsilon$ is the identity functions. Thus,

$(M, \circ, \hat{\delta}_\epsilon)$ is a monoid.

Moreover, it is finite! (There are at most $|Q|^{|Q|}$ elements)

Define $h : \Sigma^* \longrightarrow M$ by
$x \longmapsto \hat{\delta}_x$.

By construction, h is indeed a morphism.

(Our choice of composition ensures this.)

Define $X = \{ \hat{\delta}_x : x \in L \} \subseteq M$.

Then, $h^{-1}(X) = L$.

Proof. ($\supseteq$) clear.

($\subseteq$) Let $w \in h^{-1}(X)$. Then, $\hat{\delta}_w = \hat{\delta}_x$ for some $x \in L$. Then, $\hat{\delta}_w(q_0) = \hat{\delta}_x(q_0) \in F$. ☒

This monoid above is called the transition monoid of the automata A. (Transition monoid)

($\Leftarrow$) Let $h : \Sigma^* \to M$ be a homomorphism recognising L. (We have $(M, \cdot, e) \leftarrow$ monoid and $X \subseteq M$ s.t. $h^{-1}(X) = L$.)

We define the DFA A_h as

$$A_h = (M, e, \Sigma, \delta : M \times \Sigma \longrightarrow M, X)$$ where

δ is defined as

$$\delta(m, a) = m \cdot h(a).$$

Then, $L(A_h) = L$.

Proof. $a_0 \cdots a_n \in L(A_h) \Leftrightarrow h(a_0) \cdots h(a_n) \in L(A_h)$

$\Leftrightarrow h(a_0 \cdots a_n) \in L(A_h)$

$\Leftrightarrow a_0 \cdots a_n \in X$

Lecture 14 (15-02-2021)

15 February 2021 09:23

Syntactic Monoid

$L \subseteq \Sigma^*$, for $x, y \in \Sigma^*$: $x \sim_L y$ iff $\forall w \forall z (wxz \in L \Leftrightarrow wyz \in L)$

$$Syn(L) = (\Sigma^*/\sim_L, \cdot, [\epsilon]_{\sim_L}),$$

syntactic monoid

where $[x]_{\sim_L} \cdot [y]_{\sim_L} = [xy]_L$.

($\sim_L$ is a congruence, which makes this well-defined.)

$$\eta_L : \Sigma^* \longrightarrow Syn(L) \text{ is defined as}$$
$$x \longmapsto [x]_{\sim_L}.$$

Clearly, h_L is a morphism.

η_L is the syntatic morphism. (The quotient morphism.)

(Syntactic morphism)

Universal Property of $\eta_L : \Sigma^* \longrightarrow Syn(L)$:

Suppose $h : \Sigma^* \to M$ is a monoid morphism which recognises L.

Then, $h(\Sigma^*) \hookrightarrow M$ is a submonoid. We have

$$\begin{tikzcd} \Sigma^* \arrow[r, two heads, "h", "\text{onto}"'] \arrow[dr, "\eta_L"'] & h(\Sigma^*) \arrow[r, hook, "\text{one-one}"'] \arrow[d, dashed, "h_L"] & M. \\ & Syn(L) & \end{tikzcd}$$

$\exists$ a morphism $h_L : h(\Sigma^*) \to Syn(L)$ s.t. the triangle commutes.

$$h \circ h_L = \eta_L$$

(recall we write compositions in reverse.)

Def^n. We say M divides N if there exists a submonoid P of N and a surjective morphism $h: P \to M$. Denoted $M \prec N$.

(M divides N)

$$\begin{tikzcd} P \arrow[r, hook] \arrow[d] & N \\ M & \end{tikzcd}$$

Thm If M recognises L, then $\mathrm{Syn}(L) \prec M$.

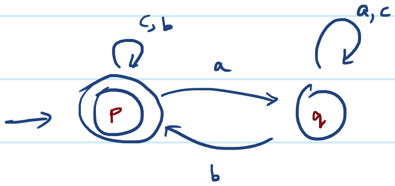
→ in some it is the gcd.

Aim: To analyse $\mathrm{Syn}(L)$ and look at algebraic properties let us see if L can be recognised by an FO-formula.

Example $\Sigma^* = \{a, b, c\}$

→ L = every 'a' is eventually followed by a 'b'.

Ex. Let $L \subseteq \Sigma^*$ be regular. $\mathrm{Syn}(L)$ is the transition monoid of the minimum automaton.



← minimum automaton for L.

$$\mathrm{Syn}\ L = \left\{ \underbrace{\delta_c : \begin{matrix} p \mapsto p \\ q \mapsto q \end{matrix}}_{\text{identity}}, \quad \delta_a : \begin{matrix} p \mapsto q \\ q \mapsto q \end{matrix}, \quad \delta_b : \begin{matrix} p \mapsto p \\ q \mapsto p \end{matrix} \right\}$$

$$\text{"}e \qquad\qquad \text{"}1 \qquad\qquad \text{"}2$$

(For any w, δ_w is one of $\delta_c, \delta_a, \delta_b$. If $w \in c^*$, $\delta_w = \delta_c = id$. Else, look at last non-c letter. It maps everything to either p or q.)

$\therefore$ What we have written above is actually $\mathrm{Syn}(L)$.

$\mathrm{Syn}(L) = (\{e, 1, 2\}, \cdot, e)$.

ρ_q → reset q

→ reset p

$r_q \to$ reset q

$2 \to$ reset p

	e	1	2
e	e	1	2
1	1	1	2
2	2	1	2

← multiplication table

$$2\cdot 1 = \delta_b \circ \delta_a = \delta_{ba} = \delta_a$$

The above monoid is called U_2, the reset-monoid.

Note that $1\cdot 1 = 1$, $2\cdot 2 = 2$.

A finite monoid typically has many idempotents.

Also, $x\cdot 1 = 1$ for all $x \in M$.

Def^n. An element $m \in M$ is called:

- an idempotent if $m\cdot m = m$,
- a right-zero if $x\cdot m = m$ for all $x \in M$,
- a left-zero if $m\cdot x = m$ for all $x \in M$.

(Idempotent, right-zero, left-zero)

Ex. Compute $Syn(L)$ for $L = (ab)^*, (aa)^*$. → list down idempotents

Lecture 15 (16-01-2021)

16 February 2021 10:35

Recall. $U_2 = (\{e, 1, 2\}, \cdot, e)$ where

$$x \cdot m = \begin{cases} x ; & m = e, \\ m ; & m \neq e. \end{cases}$$

That is,

	e	1	2
e	e	1	2
1	1	1	2
2	2	1	1

(Note that since U_2 came from an automaton, assoc. need not be checked.)

Def^n. · A monoid is said to be idempotent if every element is idempotent.

· A monoid $(M, \cdot, e)$ is said to be commutative if $x \cdot y = y \cdot x$ for all $x, y \in M$.

(Idempotent monoid, commutative monoid)

U_2 is ~~commutative~~ since $1 \cdot 2 = 2 \neq 1 = 2 \cdot 1$.

U_2 is idempotent.

$M = (\{e, p, q\}, \cdot, e)$

	e	p	q
e	e	p	q
p	p	p	q
q	q	q	q

$q \longrightarrow$ left AND right zero

Verify that this is associative.

M is both commutative and associative.

Let $\Sigma = \{a, b, c\}$. Let us define

$$h : \Sigma^* \longrightarrow M. \leftarrow \text{above } M$$

Note that Σ^* is the free monoid on Σ. It suffices to assign values to Σ. (Any function $f : \Sigma \to M$ lifts uniquely to a homomorphism $\tilde{f} : \Sigma^* \to M$.)

We define h by extending

$$a \mapsto p$$
$$b \mapsto p$$
$$c \mapsto q$$

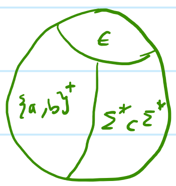
The above specifies h on Σ^*, an infinite set.

e.g. $h(aca) = h(a)h(c)\,h(a) = pqp = q.$

$h^{-1}(e) = \{\epsilon\}$

$h^{-1}(q) = \{w \mid w \text{ contain at least one } c\}$
$= \Sigma^*c\,\Sigma^*$

$h^{-1}(p) = $ non-empty words without a 'c' $= \{a,b\}^+$



Def^n. For a word w, $\alpha(w)$ = the set of letters which appear in w.

Observation: For this above h: $\alpha(w) = \alpha(w') \Rightarrow h(w) = h(w')$.

Lemma. Let M be a commutative and idempotent monoid and $h : \Sigma^* \to M$.

If $w, w' \in \Sigma^*$ are such that $\alpha(w) = \alpha(w')$, then $h(w) = h(w')$. ☒

If L is recognised by $\underset{\text{comm. + idem.}}{M}$ and $\alpha(w) = \alpha(w')$, then $[w \in L \Leftrightarrow w' \in L]$.

Def^n: $w \equiv_\alpha w'$ if $\alpha(w) = \alpha(w')$.

(This is clearly an equivalence relation.)

This is a congruence on Σ^*.

The equivalence classes of $\equiv_\alpha$ are parameterised by subsets of Σ.

Obs.

- If L is recognised by a comm. + idem. monoid, then L is a union of $\equiv_\alpha$ - eq. classes.

$$\{ w \mid \alpha(w) = A \} = A^* \setminus \bigcup_{a \in A} (A \setminus \{a\})^*$$

- If L is recognised by a comm + idem monoid, then L is a boolean combination of languages of the form A^* for $A \subseteq \Sigma$. Converse also true:

Thm. L is recognised by a comm. + idem. monoid iff L is a boolean combination of languages of the form A^* where $A \subseteq \Sigma$.

Lecture 16 (18-02-2021)

18 February 2021 11:36

Thm! Let $L \subseteq \Sigma^*$. Then, L is recognised by a commtidem monoid iff L is a boolean combination of languages of the form A^* for $A \subseteq \Sigma$.

Proof. $(\Rightarrow)$ $h : \Sigma^* \longrightarrow M$ morphism recognising L.

$\forall w, w \in \Sigma^*$, $\alpha(w) = \alpha(w') \Rightarrow h(w) = h(w')$

$\Rightarrow (w \in L \text{ iff } w' \in L)$

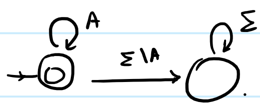
Fix $A \subseteq \Sigma$, note

$$\{ w \mid \alpha(w) = A \} = A^* \setminus \Big(\bigcup_{a \in A} (A \setminus \{a\})^* \Big)$$

boolean combination

Conclude L is the union of above form.

$(\Leftarrow)$ For A^*, we have



The corresponding monoid has two elements. It looks like:

	e	0
e	e	0
0	0	0

Called U_1

Define h by $a \mapsto e$ $\quad a \in A$

$a \mapsto 0$ $\quad a \notin A$

Then, $L = h^{-1}(\{e\})$.

- If L is recognised by M, then so is $\bar{L} = \Sigma^* \setminus L$.

- Suppose L_1 and L_2 are recognised by (h_1, M_1, X_1) and (h_2, M_2, X_2). Then, consider the monoid $M_1 \times M_2$.
  $L_1 \cap L_2$ is recognised by $(h_1 \times h_2, M_1 \times M_2, X_1 \times X_2)$.
  $L_1 \cup L_2$ by $(X_1 \times M_2) \cup (M_1 \times X_2)$.

$L_1 \cup L_2$ by $(X_1 \times M_2) \cup (M_1 \times X_2)$.

$M_1 \times M_2$: $(m_1, m_2) \cdot (m_1', m_2') = (m_1 \cdot m_1', m_2 \cdot m_2')$

$h_1 \times h_2$: $h_1 \times h_2 : \Sigma^* \to M_1 \times M_2$

$w \longmapsto (h_1(w), h_2(w))$.

Ex. If M_1 and M_2 are comm + idem, then so is $M_1 \times M_2$.

This finishes the proof. ∎

Recall Given monoids M and N, we say M divides N or $M \prec N$ if M is a homomorphic image of a submonoid of N.

$$\begin{array}{c} P \subseteq N \\ \downarrow \\ M \end{array}$$

Lemma If N is comm + idem and $M \prec N$, then M is also comm. + idem.

Proof Let $P \subseteq N$ be a submonoid s.t. $h : P \twoheadrightarrow M$.

Note P is also idem + comm.

Now, given $m_1, m_2 \in M$, $\exists p_1, p_2 \in P$ s.t. $h(p_i) = m_i$.

Then $m_1 m_2 = h(p_1) h(p_2) = h(p_1 p_2) = h(p_2) h(p_1) = m_2 m_1$ and

$m_1^2 = (h(p_1))^2 = h(p_1^2) = h(p_1) = m_1$. ∎

Cor. Given $L \subseteq \Sigma^*$, it has either of the equivalent properties of the Thm 1 iff the syntactic monoid of L is comm. + idemp.

First - Order - Logic

$FO \to$ $a(x)$, $x < y$, $x = y$, etc.

$FO^1 \longrightarrow$ first order logic with 1 variable

$a(x)$, $x < y$, $x = y$, etc.

$FO^1 \to$ first order logic with 1 variable

fix the letter: x.

$(\exists x.\ a(x)) \wedge (\exists x.\ b(x))$ is fine

$\exists x.\ (a(x) \wedge b(x))$

become very boring $x < x$ always false

$x = x$ always true

Similarly, we have FO^2, FO^3, ... Moreover,

$FO^1 \subseteq FO^2 \subseteq FO^3 \subseteq \cdots$. Is this strict?

(expressiveness)

As it turns out, $FO^1 \subsetneq FO^2 \subsetneq FO^3 = FO^4 = FO^5 = \cdots = FO$.

(wöah!!!)

Thm. 2 Let φ be an FO^1-sentence and $w, w' \in \Sigma^*$ be s.t.

$\alpha(w) = \alpha(w')$.

Then, $w \models \varphi$ iff $w' \models \varphi$.

Thm. 3 Let φ be a FO^1-formula and $w, w' \in \Sigma^*$

with $\alpha(w) = \alpha(w')$ and i, j are s.t. $w_i = w'_j$.

Then,

$$w, x \leftarrow i \models \varphi \text{ iff } w', x \leftarrow j \models \varphi$$

Proof. We prove this by structural induction.

- Base case: $\varphi = a(x)$.

  Follows since $w_i = w'_j$.

  $(w, x \leftarrow i \models a(x)$ iff $w_i = a.)$

- $\varphi_1 \vee \varphi_2$, $\neg\varphi$ follow directly.
- $\varphi = \exists x.\ \psi(x)$

  Assume w, i are s.t. $w, x \leftarrow i \models \varphi$

  $w, x \leftarrow i \models \varphi = \exists x.\ \psi(x)$

  $\Rightarrow \exists i'$ s.t. $w, x \leftarrow i' \models \psi$

  Note that $\exists j'$ s.t. $w_{i'} = w'_{j'}$ and then

$(\because \alpha(w) = \alpha(w'))$

$w', x \leftarrow j' \models \psi$ and hence,

$w, x \leftarrow j \models \varphi$

By symmetry, $w, x \leftarrow i \models \varphi$ iff $w', x \leftarrow j \models \varphi$. ∎

Thm. Let $L \subseteq \Sigma^*$. TFAE:

(1) L is definable in FO^1.

(2) L is recognised by a comm. + idem.

(3) L is a boolean combination of A^* $(A \subseteq \Sigma)$

(4) $Syn(L)$ is comm. + idemp.

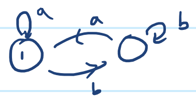
Lecture 17 (04-03-2021)

04 March 2021 11:43

Def^n. A semigroup is a set with an associative binary operation.
(We shall assume non-empty semigroups.)

(semigroup)

Any monoid is a semigroup.

Examples.
1) $(\Sigma^+, \cdot)$
2) $(\mathbb{P} = \{1, 2, \ldots\}, +)$
3) 

$\delta_a : \begin{matrix} 1 \mapsto 1 \\ 2 \mapsto 1 \end{matrix}$, $\delta_b : \begin{matrix} 1 \mapsto 2 \\ 2 \mapsto 2 \end{matrix}$

$\{\delta_a, \delta_b\}$ is a semigroup

NOT MONOIDS! No identity.

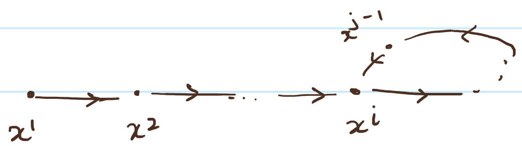
- Let S be a semigroup. Fix $x \in S$.

$X = \{x, x^2, x^3, \ldots\}$ is the subsemigroup generated by x.
(It is a cyclic semigroup.)

(semigroup generated)

Case 1. All powers are distinct. $x^i \neq x^j$.
Then, X is isomorphic to $\mathbb{P}$.

Case 2. There is a repetition in the sequence.
Choose j smallest s.t. $\exists i < j$ with $x^i = x^j$.
Thus, $x^1, x^2, \ldots, x^{j-1}$ are all distinct.
This 'i' (uniquely determined) is called the index of x.
Then, we have a repetition from that point on. (index)



This "loop" has $p = j + i$ elements in it. It is actually a group. p is called the period of x. (period)

Obs. There is a power of x which is an idempotent.

$x^i = x^{i+p}$. In fact $x^k = x^{k+p}$ $\forall k \geq i$.

Now, choose q large enough so that $k = qp \geq i$.

Then,

$$(x^k)^2 = x^{2k} = x^{k+qp} = x^{k+qp-p} = \dots = x^k.$$

Thus, k is an idempotent.

Obs: If S is a finite semigroup, then every element x has an idempotent power.

Obs. If S is a finite semigroup then there exists a positive integer π s.t. $\forall x$, x^π is idempotent.

(Note the switch of quantifiers.)

Proof What we know: $\forall x \in S$ $\exists n_x$ s.t. x^{n_x} is idemp.

Let $\pi = \underset{x \in S}{\text{LCM}}\ n_x \leftarrow$ finite.

Then, $(x^\pi)^2 = (x^{2n_x})^{\pi/n_x} = (x^{n_x})^{\pi/n_x} = x^\pi$.

Given a semigroup S, we define S' as:

$$S' = \begin{cases} S & \text{if } S \text{ is a monoid} \\ S \cup \{1\} & \text{with the mult. operation on } S \text{ extended to } S \cup \{1\} \text{ so that } (S \cup \{1\}, \cdot, 1) \text{ is a monoid} \end{cases}$$

$1 \cdot s = s \cdot 1 = s$, $s \cdot s' = s' \cdot s$ $\forall s, s' \in S$

Can check it is associative with 1 as id.

Defn. Let S be a semigroup. (right ideal)

A right ideal of S is a subset $R \subset S$ s.t.

$$RS' = R.$$

$(RS' = \{r \cdot s : r \in R, s \in S'\})$

Thus, $r \cdot s \in R \quad \forall r \in R, s \in S$.

In particular, the same is true for $s \in R$. Thus, R is a semigroup as well.

Def^n. Similarly a left ideal of S is a subset $L \subset S$ s.t.

$$S' L = L.$$

(left ideal)

Def^n. An ideal of S is a subset $I \subset S$ s.t. (ideal)

$$S' I S' = I.$$

We shall assume all types of ideals to be nonempty.

- Fix $x \in S$. What is the smallest right ideal of S which contains x?

Note that $x \cdot S'$ is a right ideal which contains x.

Moreover, if $R \ni x$ is a right ideal and $y \in S'$, then $x \cdot y \in R$. Thus, $x \cdot S' \subset R$.

$\therefore$ $x \cdot S'$ is the right ideal generated by x.

$\rightarrow$ $S'x$ is the left ideal of x.

$\rightarrow$ $S'x S'$ is the ideal of x.

Def^n. We define the following relations on S:

$x, y \in S$.

$$x \leq_{\mathcal{L}} y \quad \text{if} \quad S'x \subseteq S'y$$

"x is $\mathcal{L}$ less than y"

$$x \leq_{\mathcal{R}} y \quad \text{if} \quad xS' \subseteq yS'$$

$$x \leq_{\mathcal{J}} y \quad \text{if} \quad S'xS' \subseteq S'yS'.$$

(Script J.)

All these three relations are preorders. (preorder, pre-order)

[Preorder on a set X: A binary relation which is reflexive and transitive.]

Given a preorder $\leq$, we get the following equivalence relation $\sim$ by $x \sim y$ iff $x \leq y$ and $y \leq x$.

We can talk of the set of equivalence relations of $\sim$.
Now, we can define $\leq$ on $X/\sim$ by

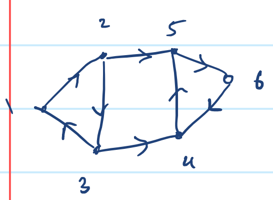
$$[x] \leq [y] \quad \text{iff} \quad x \leq y.$$

(Is well-defined!)

Now, $\leq$ on $X/\sim$ is a partial order.
↳ reflexive, transitive, anti-symmetric

Example. Let $G = (V, E)$ be a directed graph.
Let $\leq$ on V be defined by
$u \leq v$ if there is a (possibly empty) directed path from u to v.

This is a pre-order. Need not be anti-symmetric.

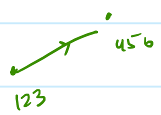


e.g.: $1 \leq 3$, $3 \leq 1$, $2 \leq 5$, $5 \not\leq 2$

Now, $u \sim v$ iff $u \leq v$ and $v \leq u$.

Then, $[1] = \{1, 2, 3\}$
$[4] = \{4, 5, 6\}$. → "Strongly connected components"

We get the poset $\{[1], [4]\}$ with $[1] \leq [4]$.



→ directed acyclic graph

Lecture 18 (08-03-2021)

08 March 2021 09:32

Recall: Given S ← semigroup, we defined S^1 and the pre-orders $\leqslant_{\mathcal{L}}$, $\leqslant_{\mathcal{R}}$, $\leqslant_{\mathcal{J}}$ as

$$s \leq_{\mathcal{L}} s' \equiv S^1 s \subseteq S^1 s';$$
$$s \leq_{\mathcal{R}} s' \equiv s S^1 \subseteq s' S^1$$
$$s \leq_{\mathcal{J}} \equiv S^1 s S^1 \subseteq S^1 s' S^1.$$

The associated equivalence relations by the letters $\mathcal{L}$, $\mathcal{R}$, $\mathcal{J}$, resp. That is:

$$s \mathcal{L} s' \Leftrightarrow (s \leq_{\mathcal{L}} s' \text{ and } s' \leq_{\mathcal{L}} s) \Leftrightarrow S^1 s = S^1 s'$$
$$\Leftrightarrow \exists m, n \in S^1 \text{ s.t. } s = ms' \text{ and } s' = ns.$$

Similarly, $s \mathcal{R} s' \Leftrightarrow sS^1 = S^1 s' \Leftrightarrow \exists m, n \in S^1$ s.t. $s = s'm$ and $s' = sn$.

Lastly, $s \mathcal{J} s' \Leftrightarrow S^1 s S^1 = S^1 s' S^1 \Leftrightarrow \exists m, m', n, n' \in S^1$ s.t.
$s = m's'm$ and
$s' = n's n$.

For an element $s \in S$: $L(s)$, $R(s)$, and $J(s)$ denote the equivalence class containing s corresp. to $\mathcal{L}$, $\mathcal{R}$, $\mathcal{J}$.

Lemma The relations $\leq_{\mathcal{R}}$ and $\mathcal{R}$ are stable on the left.

That is, $\forall s, x \in S$, we have

$$s \leq_{\mathcal{R}} s' \Rightarrow xs \leq_{\mathcal{R}} xs' \text{ and}$$
$$s \mathcal{R} s' \Rightarrow xs \mathcal{R} xs'.$$

Similarly, $\leq_{\mathcal{L}}$ and $\mathcal{L}$ are stable on right.

Proof $s \leq_{\mathcal{R}} s' \iff sS^1 \subseteq s'S^1 \iff \exists m \in S^1$ s.t. $s = s'm$

Now, $x \in S$ gives $xs = xs'm \in (xs')S^1$.

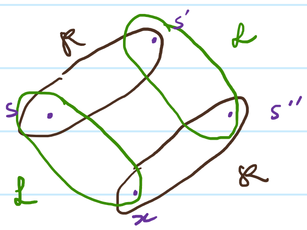
$$\Rightarrow xsS^1 \subseteq (xs')S^1$$
$$\Rightarrow xs \leq_{\mathcal{R}} xs'$$

This gives that $\mathcal{R}$ is left stable too. ∎

Lemma The relations $\mathcal{R}$ and $\mathcal{L}$ commute.

$\forall s, s', s'' \in S$, we have

$$s \mathcal{R} s' \text{ and } s' \mathcal{L} s'' \Rightarrow \exists x \in S \text{ s.t. } s \mathcal{L} x \text{ and } x \mathcal{R} s''$$



Proof. $s\mathcal{R}s' \Rightarrow \exists m, n \quad s = s'm, \quad s' = sn$

$s \mathcal{L} s'' \Rightarrow \exists p, q \quad s' = ps'', \quad s'' = qs'$

Let $x = qs'm = s''m \in s''S^1$
$= qs \in S^1 s$

Now, $px = pqs'm$
$= ps''m = s'm = s.$

Thus, $s = px \in S^1 x$.

$\therefore s \mathcal{L} x$. $\|^{ly}$ $x \mathcal{R} s''$ ∎

$\therefore\ s \mathcal{L} x$. $\quad$ "ly $\quad x \mathcal{R} s$" $\quad \blacksquare$

- Suppose we have R_1 and $R_2 \to$ equiv. relations on X. We want the smallest equiv. rel^n which contains R_1 and R_2.

This becomes easier if R_1 and R_2 commute.

Def^n. Denote by $\mathcal{D}$ the equivalence relation $\mathcal{R}\mathcal{L}$ $(= \mathcal{L}\mathcal{R})$, i.e.,

$$x \mathcal{D} z \text{ iff } \exists y \text{ s.t. } x \mathcal{R} y \text{ and } y \mathcal{L} z.$$

This is an equivalence relation since $\mathcal{R}$ and $\mathcal{L}$ commute.

$x \mathcal{D} x$ since $x \mathcal{R} x \mathcal{L} x$.

$x \mathcal{D} z \Rightarrow \exists y\ x \mathcal{R} y \mathcal{L} z \Rightarrow \exists y'\ x \mathcal{L} y' \mathcal{R} z \Rightarrow z \mathcal{R} y' \mathcal{L} x \Rightarrow z \mathcal{D} x$

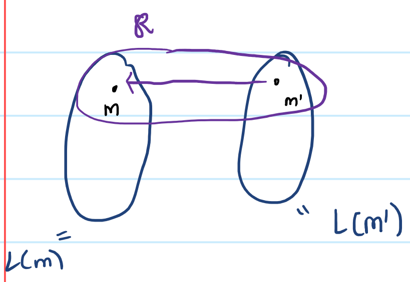
$x_1 \mathcal{D} x_2 \mathcal{D} x_3 \Rightarrow x_1 \mathcal{R} y_1 \mathcal{L} x_2 \mathcal{R} y_2 \mathcal{L} x_3 \Rightarrow x \mathcal{L} y_3 \mathcal{R} x_2 \mathcal{R} y_2 \mathcal{L} x_3$

$x_1 \mathcal{D} x_3 \Leftarrow x \mathcal{R} y_3 \mathcal{L} x_3 \Leftarrow x \mathcal{R} y_4 \mathcal{L} y_2 \mathcal{L} x_3 \Leftarrow x \mathcal{L} y_3 \mathcal{R} y_2 \mathcal{L} x_3$

Def^n. Denote by $\mathcal{H}$ the equivalence relation $\mathcal{R} \cap \mathcal{L}$.

Lemma. Let $D \subseteq S$ be a $\mathcal{D}$ class and let $m, m' \in D$ be s.t. $m \mathcal{R} m'$.
Further, choose p and q s.t. $m = m'p$ and $m' = mq$.

Follows from commutativity: Then, $x \longmapsto xp$ is a map $L(m') \longrightarrow L(m)$ and $x \longmapsto xq$ is a map $L(m) \longrightarrow L(m')$.



Moreover, these are inverses of each other. (In particular, they are bijections.)
Furthermore, they preserve $\mathcal{H}$ classes.

Proof. Let $n \in L(m)$. $[m \mathcal{L} n]$

Write $n = sm$.

Now, $(nq)p = smqp = sm'p = sm = n$.

This shows that the maps are inverses. (By symmetry.)

Lecture 19 (09-03-2021)

09 March 2021 10:34

Green's relation : $\leq_{\mathcal{L}}$, $\leq_{\mathcal{R}}$, $\leq_{\mathcal{J}}$, $\mathcal{L}$, $\mathcal{R}$, $\mathcal{J}$.

(1) $\leq_{\mathcal{R}}$, $\mathcal{R}$ stable on right, . . .

(2) $\mathcal{L}$ and $\mathcal{R}$ commute.

Ex. $(\leq_{\mathcal{L}}) \circ (\leq_{\mathcal{R}}) = \leq_{\mathcal{J}} = (\leq_{\mathcal{R}}) \circ (\leq_{\mathcal{L}})$

(3) $\mathcal{D} = \mathcal{L} \circ \mathcal{R} = \mathcal{R} \circ \mathcal{L}$.

Note that $\mathcal{D} \subseteq \mathcal{J}$ in general but $\mathcal{D} \neq \mathcal{J}$ not necessary.

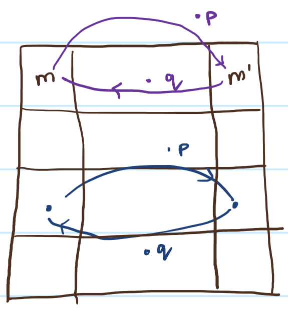
However, $\mathcal{D} = \mathcal{J}$ for finite semigroups

(4) $\mathcal{H} = \mathcal{L} \cap \mathcal{R}$.

(5) Let D be a $\mathcal{D}$-class, $m, m' \in D$ and $m \,\mathcal{R}\, m'$.

Fix $p, q \in S^1$ s.t. $m' = mp$ and $m = m'q$.

$\cdot p : L(m) \to L(m')$
$\cdot q : L(m') \to L(m)$
inverse bijection, preserving $\mathcal{H}$ classes

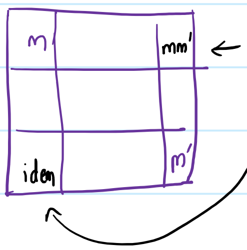


D

$\mathcal{R} \to$ rows
right ideal

$\mathcal{L} \to$ left ideal
columns

(6) Let D be a $\mathcal{D}$ class; $m, m' \in D$.



Claim. $m \cdot m' \in R(m) \cap L(m')$ iff $L(m) \cap R(m')$ contains an idempotent

Proof. $(\Longrightarrow)$ $\cdot m' : L(m) \longrightarrow L(mm')$ is a bijection, by the previous.

But $L(mm') = L(m')$.

Moreover, $\cdot m'$ preserves $\mathcal{H}$ classes.

$\therefore \exists e \in L(m) \cap R(m')$ such that

$$e \cdot m' = m'.$$

As $e \mathcal{R} m'$, $\exists x$ s.t. $m'x = e$.

Now, $e \cdot e = e \cdot (m'x) = (e \cdot m') x = m' \cdot x = e$.

($\Leftarrow$) Let $e \in L(m) \cap R(m')$ be an idempotent.

$e \mathcal{R} m' \Rightarrow \exists x \quad ex = m'$

Will use again!

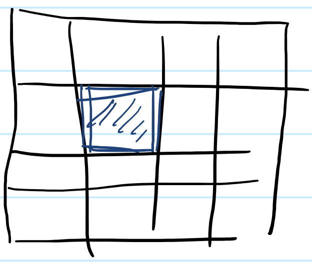
Note $em' = e(ex) = e^2 x = ex = m'$.

Thus, we may assume $x = m'$.

$\cdot m' : L(m) = L(e) \longrightarrow L(m')$ is an $\mathcal{H}$-class preserving map (in fact, a bijection)

$\Rightarrow m \cdot m' \in R(m) \cap L(m')$. ∎

(7) An $\mathcal{H}$-class H is a group (under the induced operation) iff it contains the product of two of its elements. (iff it contains an idempotent)



($\Rightarrow$) Trivial.

($\Leftarrow$) Let $m, m' \in H$ be s.t. $m \cdot m' \in H$.

But then we are in the previous scenario. (Degenerate rectangle.)

Thus, H contains an idempotent, say e.

Now, $\forall x \in H$: $xe = x = ex$. (Use the trick from earlier!)

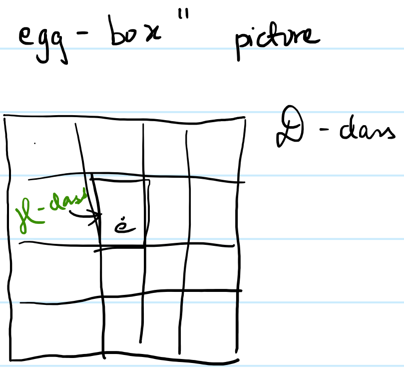
($x \in H \Rightarrow x \mathcal{R} e$ and $x \mathcal{L} e \Rightarrow \exists m', m''$ s.t. $x = em' = m''e$

but we can choose both to ...)

($x \in H \Rightarrow x \mathcal{R} e$ and $x \mathcal{L} e \Rightarrow \exists\, m', m''$ s.t. $x = e m' = m'' e$
but we can choose both to x.)

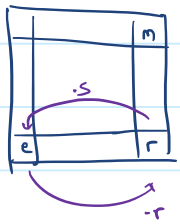
Now $\cdot x : H \to H$ is a bijection. $\therefore \exists\, y \in H$ s.t.
$yx = e$.
Similarly, so is $x \cdot : H \to H$. $\therefore xz = e$ for some z.
Thus, every elt has a left as well as right inverse.
Visual algebra tells us that they are same. ∎

(8) "egg-box" picture



All $\mathcal{H}$-classes within a $\mathcal{D}$ class have same cardinality.
(Possibly different across diff. $\mathcal{D}$ classes.)

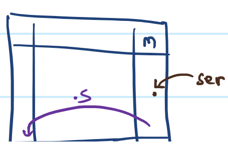
If D contains an idempotent, it contains at least one idempotent in each $\mathcal{R}$-class and each $\mathcal{L}$-class.
(Thus, if a D class contain one idem, so does every row and column.)



Proof Let $e \in D$ be an idempotent.
Let $m \in D$.
$\exists r$ s.t. $e \,\mathcal{R}\, r \,\mathcal{L}\, m$.
$e \cdot r = r$ (since e is idemp.) (same trick)

$\exists s$ s.t. $r \cdot s = e$. Now,
$(ser)^2 = ser\,ser = s e^3 r = ser$.
Thus, ser is an idempotent. Note $er = r$ and thus,
$ser = sr$.

Claim: $r \,\mathcal{L}\, (ser)$.

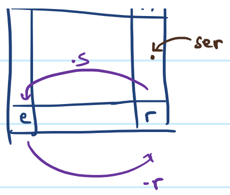


Proof. $ser = (se)r$ — (1)

Proof. $ser = (se)r$ — (1)

$r(ser) = (rs)er = e^2 r = r$

$\Rightarrow r = (r)\, ser$ — (2)



(1) and (2) show that $r \mathcal{L} (ser)$. ∎

Thus, the column $L(m)$ contains an idempotent.
$\|^{ly}$, $R(m)$ contains one. ∎

Lecture 20 (11-03-2021)

11 March 2021 11:36

Falling down in pre-orders:

- right multiplication makes you fall down in $\leq_R$.
  That is, if $x \in S$, $y \in S^1$, then $xy \leq_R x$.

- left multiplication makes you fall down in $\leq_L$.
  $zx \leq_{\mathcal{L}} x$ for $x \in S$, $z \in S^1$

- similarly, $zxy \leq_{\mathcal{J}} x$.

Note $x \mathcal{D} y \Rightarrow x \mathcal{J} y$

$\Downarrow$

$x \mathcal{L} z \mathcal{R} y \Rightarrow S^1x = S^1z,\ zS^1 = yS^1 \Rightarrow S^1xS^1 = S^1zS^1 = S^1yS^1$

Converse not true in general. Is true when $|S| < \infty$.

From now, S will denote a <u>finite</u> semigroup.

Lemma. (Simplification lemma) Let $m \in S$, $x, y \in S^1$.
If $xmy = m$, then $m \mathcal{L} xm$ and $m \mathcal{R} my$.
(We do always have $xm \leq_{\mathcal{L}} m$. Here, $xm \leq_{\mathcal{L}} m \leq_{\mathcal{L}} xm$.)

Proof. $m = xmy$

$= x(xmy)y = x^2my^2 = \cdots = x^3my^3 = \cdots = x^\ell my^\ell$

$\forall\ \ell \geq 0$

Recall that every element in finite semi. has idemp. power.
Let $i, j > 0$ be s.t. x^i, y^j are idempotent.

$$x^i = x^{2i} = x^{3i} = \dots x^{ij}, \qquad y^j = y^{2j} = \dots = y^{ij}$$

$$m = x^{ij} m y^{ij} = x^i x^{ij} \cdot m \cdot y^{ij}$$
$$= x^i \cdot m = x^{i-1} \cdot (xm)$$

$\Rightarrow$ $m \leq_{\mathcal{L}} xm$.

$\therefore$ $m \,\mathcal{L}\, xm$.

Similarly, $m \,\mathcal{R}\, my$. ∎

Lemma. $m \,\mathcal{J}\, m' \Rightarrow m \,\mathcal{D}\, m'$

Proof. $m \,\mathcal{J}\, m' \Rightarrow \exists x, y, a, b, \quad m = xm'y \,;\, m' = amb$.

$m = xm'y = (xa)m(by)$.

By simplification, $m \,\mathcal{L}\, (xa)\cdot m$, $\quad m \,\mathcal{R}\, m(by)$.

$$m \leq_{\mathcal{L}} (xa)\cdot m \leq_{\mathcal{L}} am \leq_{\mathcal{L}} m.$$

$\Rightarrow$ $m \,\mathcal{L}\, am$. $\quad$ Similarly, $m \,\mathcal{R}\, mb$.

$\Downarrow$

$am \,\mathcal{R}\, amb$

$\Rightarrow$ $m \,\mathcal{L}\, am \,\mathcal{R}\, amb \Rightarrow m \,\mathcal{D}\, amb = m'$.

$\therefore$ $m \,\mathcal{D}\, m'$, as desired. ∎

Lemma. Suppose $m \,\mathcal{J}\, m'$ (and hence, $m \,\mathcal{D}\, m'$).

(i) If $m \leq_{\mathcal{R}} m'$, then $m \,\mathcal{R}\, m'$.

Thus, two (distinct) $\mathcal{R}$ classes within a $\mathcal{J}$ class are incomparable.

(ii) If $m \leq_{\mathcal{L}} m'$, then $m \,\mathcal{L}\, m'$.

Proof. We only prove (i).

$m \mathcal{J} m'$ and $m \leq_{\mathcal{R}} m'$.

$m = m'x$ for some $x \in S^1$. $(\because m \leq_{\mathcal{R}} m')$

$m' = amb$ for some $a, b \in S^1$. $(\because m' \leq_{\mathcal{J}} m)$

$m' = am'xb$. Apply simplification to get

$$m' \mathcal{R} m'xb.$$

$$m' \leq_{\mathcal{R}} m'xb \leq_{\mathcal{R}} m'x \leq_{\mathcal{R}} m'.$$

$$\therefore \quad m' \mathcal{R} m'x = m. \qquad \blacksquare$$

Defn. A finite semigroup S is aperiodic if $\exists n > 0 \ \forall x \in S : x^n = x^{n+1}$,

or $\forall x \in S \ \exists n > 0 : x^n = x^{n+1}$.

(aperiodic)

(Both are equivalent since S is finite.)

Propn Let S be a finite semigroup.

TFAE:

(i) S is aperiodic.

(ii) Each element generates a sub-semigroup of period 1. "each element has period 1"

(iii) Each $\mathcal{H}$ class of S is trivial.

(iv) Every group in S is trivial. [Group free semigroup.]

Proof. (i) $\Rightarrow$ (ii) trivial, the loop of length p repeats.

if $p \neq 1$, it will never be $x^n = x^{n+1}$.

(iii) $\Rightarrow$ (iv) a maximal group in a semigroup is an $\mathcal{H}$-class.

(general)

(iv) $\Rightarrow$ (i) we showed the loop forms a group.

(ii) $\Rightarrow$ (iii) next class

Lecture 21 (15-03-2021)

15 March 2021 09:23

(ii) $\Rightarrow$ (iii) Have : Each element has period 1

To show : $\mathcal{H}$ relation is trivial.

Let $a \mathcal{H} b$. That is, $S^1 a = S^1 b$ and $aS^1 = bS^1$.

$\exists x, y \in S^1$ s.t. $xa = b$, $yb = a$.

$\exists p, q \in S^1$ s.t. $ap = b$, $bq = a$

Thus, $b = xa = xbq$ $\quad$ $b = xbq$

$= x^2 bq^2$

$\vdots$

$= x^n bq^n \quad \forall n \geq 1$

We know that q has period 1. Thus, $\exists m \geq 1$ s.t. $q^m = q^{m+1}$.

$b = x^m b q^m = x^m bq^{m+1}$

$= (x^m b q^m) q = bq = a.$ ∎

(iii) $\Rightarrow$ (iv) [More elaboration]

Let $G \subseteq S$ be a group.

We show $g \mathcal{H} e \quad \forall g \in G$. (e is identity of G.)

(Since $\mathcal{H}$ classes are trivial, we would get $G = \{e\}$.)

Let $g \in G$ be arbitrary. Then, $\exists g' \in G$ s.t. $g \cdot g' = e = g \cdot g'$.

Thus, we get : (*) $g \cdot g' = e$ (*) $g' \cdot g = e$

(*) $g \cdot e = g$ (*) $e \cdot g = g$

Thus, $g \mathcal{R} e$ and $g \mathcal{L} e$. ∎

In general, $\mathcal{H}$ - classes containing an idempotent are maximal groups in S.

[illegible]
in S.

Schutzenberger's Theorem

→ A language is recognised by an aperiodic monoid/semigroup iff it is expressed by a star-free expression.

→ [McNaughton-Papert Theorem]
star-free ≡ first-order logic definability

Fix Σ finite. Star-free:

$$r \equiv r_1 + r_2 \mid \neg r \mid r_1 \cap r_2 \mid r_1 \cdot r_2 \mid a \in \Sigma \mid \emptyset$$ ← star. free

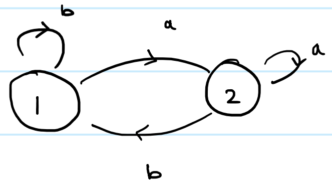
Ext. reg. exp.	Logic	Algebra	Automata
General reg exp	MSO	finite monoid/semi-group	DFA, NFA
Star-free	FO	aperiodic mon/semi	counter free aut.

Lecture 22 (16-03-2021)

16 March 2021 10:31

Examples of Green's relation

(1)



	1	2
$\epsilon = 1$	1	2
a	2	2
b	1	1

(transition functions:) $aa = a$, $bb = b$, $ab = b$, $ba = a$

$M = \{1, a, b\}$

Now, words of length $\geqslant 3$ can be reduced to a or b

$M1M = M$

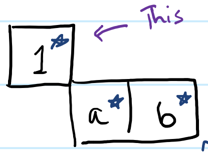
$Ma = \{a\}$; $Mb = \{b\}$; Thus, $a \not\mathcal{L} b$.

$aM = \{a, b\} = bM$; Thus, $a \mathcal{R} b$.

$MaM = \{a, b\} = MbM$; Thus, $a \mathcal{J} b$.

Thus, $\mathcal{H}$ is trivial. ($\therefore$ M is aperiodic.)

$\mathcal{D}$ class



This $\mathcal{D}$ class is $>_{\mathcal{D}}$ than

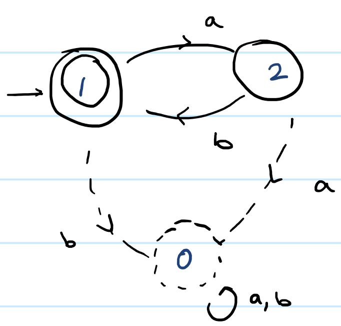
same row since $\mathcal{R}$ related

$\mathcal{D}$ class

$\star \longrightarrow$ idempotent

Another way to get $a \mathcal{R} b$ is : $b = ab \leq_{\mathcal{R}} a$ and $a = ba \leq_{\mathcal{R}} b$.

2)



$L \longleftrightarrow (ab)^*$

The 0 and bottom transitions are just to determinise. will ignore in future.

$\epsilon = 1$	1	2
a	2	0
b	0	1
ab	1	0
ba	0	2
aa	0	0
~~bb~~	~~0~~	~~0~~

(Not writing 0 column since obvious.)

$\longrightarrow$ reset to sink

Let $0 := aa \; (= bb)$.

Note $w0v = 0 \quad \forall w, v \in \Sigma^*$.

Now, $aba = a, \quad bab = b$.

Thus, everything can now be reduced.

$$M = \{1, a, b, ab, ba, 0\}.$$

$$\left.\begin{array}{l} aa = bb = 0. \\ aba = a, \quad bab = b. \end{array}\right\} \text{presentation}$$

$aaba = 0ba = 0$

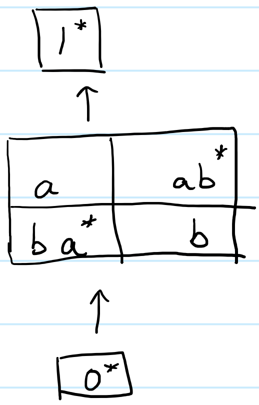
$$\left.\begin{array}{l} ab \leq_R a \\ a = ab \cdot a \leq_R ab \end{array}\right\} a \,\mathcal{R}\, ab$$

$b = bab \leq_R ba \leq_R b \Rightarrow b \,\mathcal{R}\, ba$

$a = aba \leq_L ab \leq_L a$

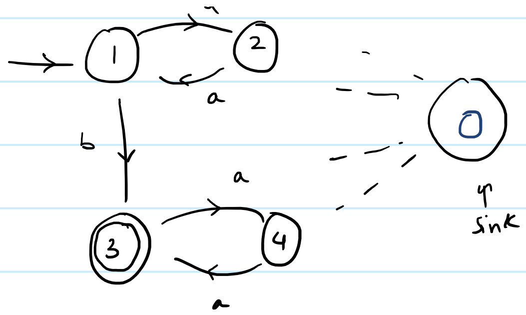
$\therefore a \,\mathcal{L}\, ab$

||ly $\quad b \,\mathcal{L}\, ba$



$\uparrow$ denotes that the above is $<_{\mathcal{D}}$ than below.

3) $K = \{ a^i b a^j \mid i \equiv 0 \bmod 2, \; j \equiv 0 \bmod 2 \}$.



1	1	2	3	4
a	2	1	4	3
b	3	0	0	0
ab	0	3	0	0
ba	4	0	0	0
~~0 = bb~~	~~0~~	~~0~~	~~0~~	~~0~~
~~1 = aa~~	~~1~~	~~2~~	~~3~~	~~4~~
aba	0			

bb = 0, aa = 1

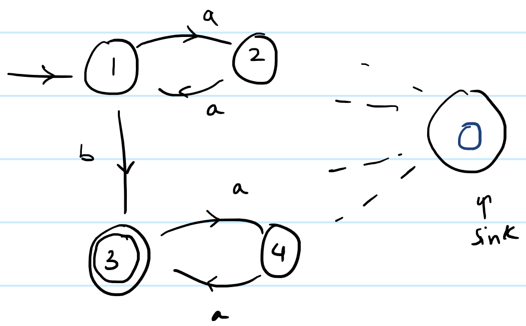
Now, we look at three letter words w/o "aa" & "bb".

Lecture 23 (18-03-2021)

18 March 2021 11:35

Green's Relations

$K = \{ a^i b a^j \mid i \equiv 0 \bmod 2,\ j \equiv 0 \bmod 2 \}.$



	1	2	3	4
a	2	1	4	3
b	3	0	0	0
ab	0	3	0	0
ba	4	0	0	0
~~0 = bb~~	~~0~~	~~0~~	~~0~~	~~0~~
~~1 = aa~~	~~1~~	~~2~~	~~3~~	~~4~~
aba	0	4	0	0
~~0 = bab~~	~~0~~	~~0~~	~~0~~	~~0~~

$bb = 0, \quad aa = 1$

Now, we look at three letter words w/o "aa" & "bb".

$bb = 0, \quad aa = 1, \quad bab = 0$

$M = \{1, a, b, ab, ba, aba, 0\}$
$bb = 0, \quad aa = 1, \quad bab = 0$ $\Big\}$ gives complete description

$a^2 = 1$. Thus, $1 \leq_R a \leq_R 1$ and same for $\mathcal{L}$.

$\therefore\ a \,\mathcal{R}\, 1 \,\mathcal{L}\, a$ and thus, $a \,\mathcal{H}\, 1$.

$\Rightarrow\ 1 \,\mathcal{D}\, a.$

First example where there is an $\mathcal{H}$ class with >1 element.
Thus, it is not aperiodic.

Def^n. A class $(\mathcal{D}, \mathcal{L}, \mathcal{R}, \mathcal{H})$ is called regular if it contains an idempotent.

Claim. Let M be a finite monoid.
Then, $J(1) = H(1)$.

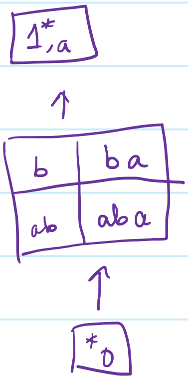
Proof. $x \,\mathcal{J}\, 1 \Rightarrow x \,\mathcal{D}\, 1$ [M is finite]
(We saw $a \,\mathcal{J}\, b$ and $a \leq_{\mathcal{R}} b$, then $a \,\mathcal{R}\, b$.)
Thus, $x \,\mathcal{R}\, 1$. ($x \leq_{\mathcal{R}} 1$ always true.)
Similarly, $x \,\mathcal{L}\, 1$
Thus, $x \,\mathcal{H}\, 1$. ∎

Back to example:

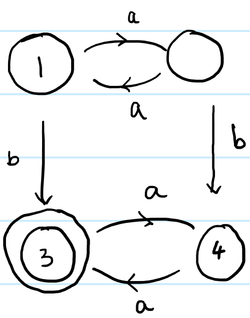
$b = a \cdot ab \leq_{\mathcal{L}} ab \leq_{\mathcal{L}} b$. $\therefore$ $b \,\mathcal{L}\, ab$.

$b = ba \cdot a \leq_{\mathcal{R}} ba \leq_{\mathcal{R}} b$. $\therefore$ $b \,\mathcal{R}\, ba$

(Show: $b \,\cancel{\mathcal{R}}\, ab$ and $b \,\cancel{\mathcal{L}}\, ba$.)



- $L = \{ a^i b a^j \mid i + j \equiv 0 \bmod 2 \}$ $\geq K$



$aa = 1$
$bb = 0$
$ba = ab$

1	1	2	3	4
a	2	1	4	3
b	3	4	0	0
ab	4	3	0	0
~~ba~~	~~4~~	~~3~~	~~0~~	~~0~~

Now, all 3 letter words are done too.

$aba = aab = b$, $bab = bba = 0$

$N = \{1, a, b, ab, 0\}$.

$aa = 1, \quad bb = 0, \quad ba = ab$.

$1 \mathcal{H} a$, as before.

$ab \leq_{\mathcal{L}} b = aab \leq_{\mathcal{L}} ab$. $\quad \therefore b \mathcal{L} ab$

$b = aab = aba \leq_{\mathcal{R}} ab = ba \leq_{\mathcal{R}} b$. $\quad \therefore b \mathcal{R} ab$.

$\boxed{{}^{*}1, a}$

$\uparrow$

$\boxed{b, ab}$ $\qquad (ab)(ab) = abab = aabb = 0 \neq ab$

$\uparrow$

$\boxed{{}^{*}0}$

Schutzenberger: star-free regex $\equiv$ recognised by an aperiodic monoid

(Finite monoid)

- A language L has a star-free regex iff $\Sigma^*/\sim_L$ is aperiodic.

① L is star-free $\Rightarrow$ L can be recognised by an aperiodic monoid.
Will do this by induction. We had seen how products recognise union/intersection. Same monoid accepts complement. Need to show for concat. Need to make sure aperiodicity is maintained.
Not difficult. Will do in tut.

② $(\Leftarrow)$ This is the difficult direction.
L recognised by aperiodic monoid $\Rightarrow$ L is star-free.
Will also show FO-definability.

$h : \Sigma^* \longrightarrow M$, $\qquad L = h^{-1}(X)$ for some $X \subseteq M$

(finite) $\qquad$ aperiodic

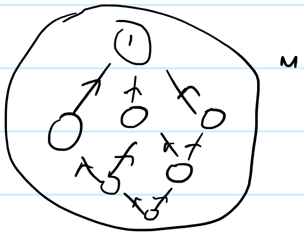
We show: $\forall m \in M$, $h^{-1}(m)$ is a star-free language.

From the above, the result follows.

Will use $\leq_J$ to do induction.

Top class will only contain 1 since $J(1) = H(1) = \{1\}$

aperiodic



$h^{-1}(\{1\}) = ?$ Suppose $a_1 \cdots a_n \in h^{-1}(\{1\})$.

Then, $h(a_1 \ldots a_n) = 1$

$\Rightarrow h(a_1) \cdots h(a_n) = 1$

$\Rightarrow h(a_i) \; J \; 1 \quad \forall i$

$\Rightarrow h(a_i) = 1 \quad \forall i$.

$\therefore h^{-1}(\{1\}) = A^*$ where $A = \{a \in \Sigma : h(a) = 1\}$.

$= \neg\left(\bigcup_{b \notin A} \Sigma^* b \Sigma^* \right)$.

Lecture 24 (22-02-2021)

22 March 2021 09:32

Schutzenberger's Theorem

Difficult direction: Aperiodic ⇒ star-free

Let $h: \Sigma^* \to M$ be a morphism to a finite aperiodic monoid.
We will show: (*) $\forall m \in M$, $h^{-1}(\{m\})$ is star-free (SF).

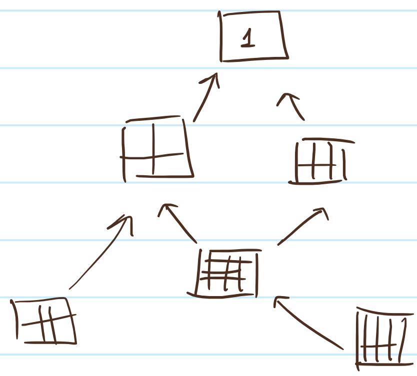
Define: $m <_{\mathcal{J}} m'$ if $m \leq_{\mathcal{J}} m'$ and $\neg(m \,\mathcal{J}\, m')$.
(Antisymmetric, irreflexive, transitive.)

We will prove (*) by induction on $<_{\mathcal{J}}$.
More precisely:

Base: 1) $h^{-1}(1)$ is SF.

Induct: 2) $\forall m \in M \; [\, (h^{-1}(n) \text{ is SF } \forall n >_{\mathcal{J}} m) \Rightarrow (h^{-1}(m) \text{ is SF})\,]$.

Note that $J(1) = H(1)$ is trivial. Thus, the topmost $\mathcal{J}$-class contains only 1.



Base case:

$h^{-1}(\{1\}) = A^*$ where $A = \{a \in \Sigma : h(a) = 1\}$.

A^* is star-free. ($\mathcal{J}$-class of 1 is trivial again.)

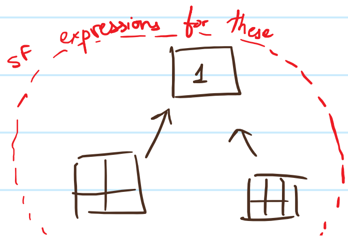
$A^* = \neg\phi\left(\neg\phi\left(\sum_{a \notin A} a\right)\neg\phi\right)$

Induction step. Notation: $L_m = h^{-1}(\{m\}) = \{w \in \Sigma^* : h(w) = m\}$.

Fix an element $1 \neq m \in M$.

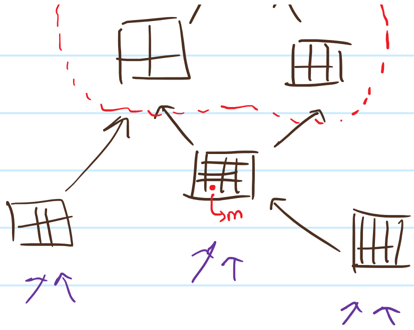
Assume: $\forall n >_{\mathcal{J}} m$, L_n is SF.

To show: L_m is SF.



To show: L_m is SF.

- Step 1. $L_{J(m)} := \{w \in \Sigma^* : h(w) \, \mathcal{J} \, m\}$ is SF.



- Step 2. $L_{R(m)} := \{w \in \Sigma^* : h(w) \, \mathcal{R} \, m\}$ is SF.

- Step 3. $L_{L(m)} := \{w \in \Sigma^* : h(w) \, \mathcal{L} \, m\}$ is SF.

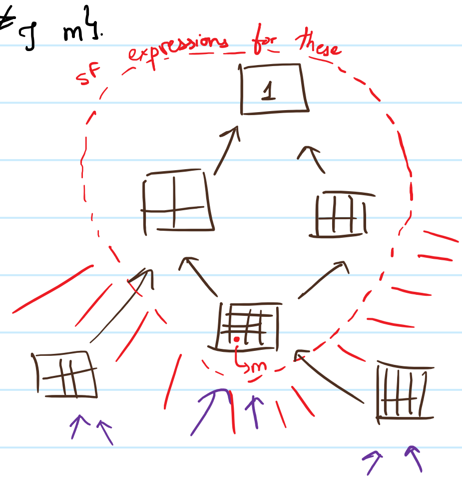
- Step 4. $L_{H(m)} := \{w \in \Sigma^* : h(w) \, \mathcal{H} \, m\}$ is SF

$$= L_{R(m)} \cap L_{L(m)}.$$

Steps 2 and 3 $\Rightarrow$ Step 4.

- Step 5. By aperiodicity, $H(m) = \{m\}$. Thus, Step 4 shows L_m is SF.

Step 1. $L_{\not\geq_J m} = \{w \in \Sigma^* : h(w) \not\geq_J m\}$.



Claim. $L_{\not\geq_J m}$ is SF.

Note: $L_{J(m)} = \left(L_{\not\geq_J m}\right)^c \setminus \left(\bigcup_{n >_J m} L_n\right)$

we show this is SF

By induct, SF

Thus, $L_{J(m)}$ is SF.

Proof (of claim). Note that $I = \{n : n \not\geq_J m\}$ is an ideal of M. $L_{\not\geq_J m} = h^{-1}(I)$.

Thus, $L_{\not\geq_J m}$ is again an ideal.

(In other words, $w \in L_{\not\geq_J m}$ and $x, y \in \Sigma^* \Rightarrow xwy \in L_{\not\geq_J m}$.)

Consider a word $w \in L_{\not\geq_J m}$ and consider a minimal factor u of w s.t. $u \in L_{\not\geq_J m}$. (Such a factor must exist. • w is one such. • there are only finitely) of w s.t. $u \in L_{\not\geq_{\mathcal{J}} m}$. (Such a factor must exist. • w is one such. • there are only finitely many.)

By minimality of u, no proper factor of u is in $L_{\not\geq_{\mathcal{J}} m}$

($u \neq \varepsilon$ since $h(\varepsilon) = 1 \geq_{\mathcal{J}} m$)

Case 1. $|u| = 1$. $h(u) \not\geq_{\mathcal{J}} m$.

• $u = a \in \Sigma$, $h(a) \not\geq_{\mathcal{J}} m$. Then, $w \in \Sigma^* X_m \Sigma^*$, where $\xrightarrow{} SF$

$X_m = \{a \in \Sigma : h(a) \not\geq_{\mathcal{J}} m\}$

Conversely, all words here map to I

Case 2. $|u| \geq 1$.

• Write $u = avb$ for $a, b \in \Sigma$ and $v \in \Sigma^*$.

$h(u) \not\geq_{\mathcal{J}} m$ but $h(av), h(v), h(vb) \geq_{\mathcal{J}} m$, by minimality.

Claim. $h(v) >_{\mathcal{J}} m$.

Proof. Suppose not. Then, $h(v) \mathcal{J} m$

Then, $h(av) \mathcal{J} m$ and $h(vb) \mathcal{J} m$ as well.

$(m \mathcal{J} h(v) \geq_{\mathcal{J}} h(av) \geq_{\mathcal{J}} m.)$

Thus, $h(v) \mathcal{J} h(vb)$. Clearly, $h(v) \geq_{\mathcal{R}} h(vb)$.

Since M is finite, we get

$$h(v) \mathcal{R} h(vb)$$

In turn, $h(av) \mathcal{R} h(avb) = h(u)$

$\Rightarrow h(av) \mathcal{R} h(u)$.

$\Rightarrow h(av) \mathcal{J} h(u)$.

$\Rightarrow m \mathcal{J} h(u)$. $\rightarrow\leftarrow$

Thus, $h(v) >_{\mathcal{J}} m$. Thus, $v \in \bigcup_{n >_{\mathcal{J}} m} L_n$.

SF language, by induction

$\therefore u \in \bigcup_{\substack{a, b \in \Sigma \\ n >_{\mathcal{J}} m}} a \cdot L_n \cdot b \xrightarrow{} SF$

$a, b \in \Sigma$

$n \geq_J m$

$h(a) \cdot n \cdot h(b) \not\geq_J m$

Conversely, all words here map to the ideal I

We have shown that $L_{\not\geq_J m}$ is SF.

Thus $L_{J(m)}$ is S.F. This finishes Step 1.

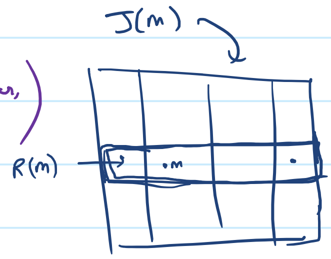
Lecture 25 (23-03-2021)

23 March 2021 10:39

Step 2. $L_{R(m)}$ is SF.

$L_{R(m)} \subseteq L_{J(m)}$.

Let $u \in L_{R(m)}$. $h(u) \, \mathcal{R} \, m$. (In particular, $h(u) \leq_R m$.)



Write $u = a_0 a_1 a_2 \cdots$

Let v be a minimal prefix of u s.t. $h(v) \leq_R m$.

- $v \neq \varepsilon$ since $h(\varepsilon) = 1$ and $1 \not\mathcal{J} m$.

- Write $v = w \cdot a$ for $w \in \Sigma^*$ and $a \in \Sigma$.
  By minimality of v, $h(w) \not\leq_R m$.

Claim. $h(w) >_J m$.

Proof. Since w is a prefix of u,
$m \leq_R h(u) \leq_R h(w)$. Thus, $m \leq_R h(w)$.
By earlier, $h(w) \not\leq_R m$.

Thus, $m \not\mathcal{R} h(w)$. That is, $m <_R h(w)$.
$m \leq_R h(w) \Rightarrow m \leq_J h(w)$.
Now, if $m \not<_J h(w)$, then $m \, \mathcal{J} \, h(w)$. But
$m \, \mathcal{J} \, h(w)$ and $m \leq_R h(w) \Rightarrow m \, \mathcal{R} \, h(w)$. $\rightarrow\leftarrow$

Thus, $m <_J h(w)$. ∎

Claim. $$L_{R(m)} = L_{J(m)} \cap \left(\bigcup_{\substack{m', n >_J m \\ n \cdot m' \leq_R m}} L_n \cdot \Sigma_{m'} \cdot \Sigma^* \right)$$

$\Sigma_{m'} = \{ a \in \Sigma : h(a) = m' \}$.

Proof Let $u \in L_{R(m)}$. $h(u) \, R \, m \Rightarrow h(u) \, J \, m \Rightarrow u \in L_{J(m)}$.

Let v be a min'l prefix of u s.t. $h(v) \leq_R m$.

Write $v = wa$ for $w \in \Sigma^*$, $a \in \Sigma$. $[v \neq \epsilon]$

Then, $n := h(w)$, $m' = h(a)$ gives $h(v) = n \cdot m' \leq_R m$. ☒

Step 3 follow similarly. So do steps 4 and 5 and we are done. ☒

Lecture 26 (25-03-2021)

25 March 2021 11:34

Thm Let L be regular. TFAE:

(i) L is recognised by a finite aperiodic monoid.

(ii) L is SF.

(iii) L is FO-definable.

(i) $\Rightarrow$ (ii) done.

(i) $\Rightarrow$ (iii) similar we do it now:

(other implications simpler. in tutorial.)

Proof. ① $h^{-1}(1)$ is FO-definable.

$h^{-1}(1) = L_{\varphi_1}$ where $\varphi_1 = \forall x \left(\bigvee_{h(a)=1} a(x) \right)$.

② $\forall m : \left[\forall n \;\; n >_J m, \; h^{-1}(n) \text{ is FO.D} \Rightarrow h^{-1}(m) \text{ is FO.D} \right]$.

Fix $m \neq 1$. Assume $h^{-1}(n)$ is FO.D $\forall\; n >_J m$.

<u>Step 1</u> $\varphi_{J(m)}$

$$\varphi_{\not\geq_J m} = \exists x . \left(\bigvee_{h(a) \not\geq_J m} a(x) \right)$$

$$\vee \exists x \exists y \bigvee_{\substack{a,b,n >_J m \\ h(a) n h(b) \not\geq_J m}} \left((x<y) \wedge a(x) \wedge b(y) \wedge \text{"the word b/w } x \text{ and } y \models \varphi_n \text{"} \right)$$

relativisation

Given sentence φ_n, can create $\hat{\varphi}_n(x, y) \equiv$ the word in (x,y) satisfies φ_n

$$\varphi_{J(m)} = \neg \varphi_{\not\geq_J m} \wedge \neg \left(\bigvee_{n >_J m} \varphi_n \right).$$

Step 2. $\varphi_{R(m)}$.

w s.t. $h(w)\ R\ m$. Then, $h(w)\ J\ m$

$w = a_0\, a_1\, a_2\, a_3 \cdots a_k$

$$h(a_0 a_1 \cdots a_k) \leq_R h(a_0 \cdots a_{k-1}) \leq_R \cdots \leq_R h(a_0) \leq_R 1.$$

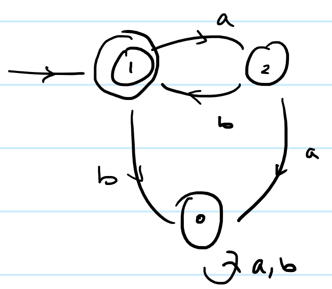
$$\varphi_{R(m)} \equiv \varphi_{J(m)} \wedge \Big(\exists x. \bigvee_{\substack{n \geq_J m \\ n \cdot h(a)\, R\, m}} \varphi_n \big|_{(-,x)} \wedge a(x) \Big)$$

relative formula to the prefix

Step 3. $\varphi_{L(m)}$ ✓

Step 4. $\varphi_m \equiv \varphi_{R(m)} \wedge \varphi_{L(m)}$. ✓ ∎

Example. $L = (ab)^*$

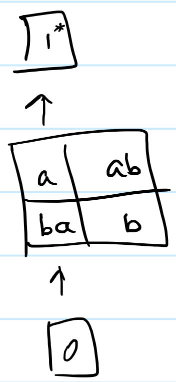


	1	2
a	2	0
b	0	1
ab	1	0
ba	0	2
aba	2	0

$M = \{1, a, b, ab, ba, 0\}$

$a^2 = b^2 = 0$

$aba = a,\ bab = b$



$h : \Sigma^* \to M$

$a \mapsto a$

$b \mapsto b$

transition function

$b \mapsto b$ transition function

letter

$$h^{-1}(1) = \{\varepsilon\} \leftrightarrow \neg \exists x \cdot (x = x)$$

Lecture 27 (30-03-2021)

30 March 2021 10:41

Have :

$\mathcal{H}$- trivial $\Leftrightarrow$ FO- definable

Comm. + idem. $\Leftrightarrow$ FO[1]- definable

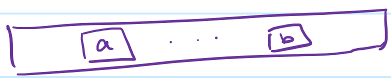
$\mathcal{J}$- trivial $\Leftrightarrow$ $\mathbb{B}[\Sigma^1]$- definable

Σ' - $\exists^*$ sentence and boolean connectives

$\exists x_1 \exists x_2 \cdots \exists x_k \underbrace{\varphi(x_1, \ldots, x_k)}_{\text{quantifier free}}$

Π' - $\forall^*$ sentence

Example. $\mathbb{B}[\Sigma']$ sentence: $\forall x \forall y \ (x<y) \wedge a(x) \wedge b(y)$



Defn. $u = a_1 \cdots a_k \in \Sigma^*$ is a subword of $v \in \Sigma^*$ if $\exists v_0, \ldots, v_k \in \Sigma^*$ s.t.

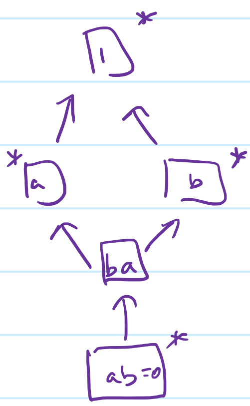
$$v = v_0 a_1 v_1 a_2 \cdots v_{k-1} a_k v_k .$$

Thus, the language above is the set of those words which have "ab" as a subword.

Ex. $M = \{1, a, b, ab, ba\}$

$a^2 = a, \quad b^2 = b, \quad aba = ab$

$bab = ab$



$\mathcal{J}$ - trivial!

Defn. Combinatorial congruence

$u, v \in \Sigma^*, \quad n \geq 0$ a parameter

$u \sim_n v$ iff u and v have same subwords of length $\leq n$

Example. $u \sim_1 v \iff u$ and v have the same set of letters

$c(u) :=$ set of letters occuring in u

$u \sim_1 v \iff c(u) = c(v)$

$ab \sim_1 ba$ but $ab \not\sim_2 ba$

$ab \not\sim_2 aba$

Lemma. $u \sim_n v$ is a congruence on Σ^* of finite index.

Proof. Let $x, y \in \Sigma^*$.

TS: $u \sim_n y \implies xuy \sim_n xvy$.

Let w be a subword of xuy s.t. $\ell(w) \leq n$.

Write $w = w_0 \, w' \, w_1$.

embeds in x, in u, in y

But $\ell(w') \leq \ell(w) \leq n$. Thus, $u \sim_n v \Rightarrow w' \hookrightarrow v$

$\therefore \; w \hookrightarrow xvy$.

By symmetry, we get $xuy \sim_n xvy$.

Finite index: There are only finitely many words of length $\leq n$. The eq. classes is parameterised (naturally) by sets of these words. ∎

Lemma. Let $u, v \in \Sigma^*$, $a \in \Sigma$, $n \geq 1$.
If $uav \sim_{2n-1} uv$, then either $ua \sim_n u$ or $av \sim_n v$.

Proof. Suppose not. That is, $ua \not\sim_n u$ and $av \not\sim_n v$.

$$\Downarrow$$

$$\exists\, x \hookrightarrow ua \text{ s.t. } |x| \leq n \text{ and } x \not\hookrightarrow u.$$

(Every subword of u is indeed that of u. Thus, this is the only possibility for $\not\sim_n$.)

Moreover, x must end in a. $x = x'a$, $|x'| \leq n-1$.

$\parallel^{ly}$ $\exists y \hookrightarrow av$, $|y| \leq n$, $y \not\hookrightarrow v$.
Again, y begins with a. $y = ay'$, $|y'| \leq n-1$

Now, the word $w = x'ay' \hookrightarrow uav$ but $w \not\hookrightarrow uv$. However, $|w| \leq 2n-1$. $\rightarrow\leftarrow$

Prop^n. Let $u, v \in \Sigma^*$ and $n > 0$.
Then, $u \sim_n vu \Leftrightarrow \exists\, u_1, \dots, u_n \in \Sigma^*$ s.t. (same n)

$$u = u_1 \cdots u_n \text{ and}$$
$$c(v) \subseteq c(u_1) \subseteq \cdots \subseteq c(u_n).$$

(Here, we get $c(u_n) = c(u) = c(vu)$.)

Proof. If $u = \varepsilon$, then it's true. ($v \sim_n u \Leftrightarrow v = \varepsilon$.)

($\Rightarrow$) By induction on n.

$n = 1$. $u \sim_1 vu \Rightarrow c(u) = c(vu) \Rightarrow c(v) \subseteq c(u)$.

Choose $u_1 = u$.

Assume true for $\leq n$.
Suppose $u \sim_{n+1} vu$.

Suppose $u \sim_{n+1} vu$.

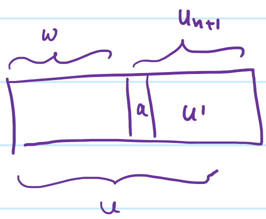
Thus, $c(u) = c(vu)$.

Let $u_{n+1} :=$ shortest suffix of u having the same content as u.

$u_{n+1} \neq \varepsilon$. Write $u_{n+1} = a \cdot u'$.

Note that choice of $u_{n+1} \Rightarrow a \notin c(u')$.

Let $w \in \Sigma^*$ be s.t. $u = w u_{n+1}$.



__Claim.__ $w \sim_n vw$.

__Proof.__ Let x be a subword of vw of length $\leq n$.

Then, $xa \hookrightarrow vu$ of length $n+1$.

Then, $xa \hookrightarrow u$.

But $a \notin u'$! Thus, $xa \hookrightarrow wa$.

$\Rightarrow x \hookrightarrow w$. ▨

By induction, factor w and get it for u. ▨

($\Leftarrow$) Induction on n. $n = 1$ is easy.

Assume true $\leq n$.

Given: u, vu s.t. $u = u_1 \cdots u_{n+1}$ with

$c(v) \subseteq c(u_1) \subseteq \cdots \subseteq c(u_{n+1})$.

To show: $u \sim_{n+1} vu$.

[Assume $u \neq \varepsilon$.]

$u_{n+1} \neq \varepsilon$ since $u \neq \varepsilon$.

Let $w = u_1 \cdots u_n$.

By induction, $w \sim_n vw$.

__Claim.__ Every subword of vu of length $\leq n+1$ is also a subword of u.

Let $x \hookrightarrow vu$ with $|x| \leq n+1$.

Lecture 28 (01-04-2021)

01 April 2021 11:36

To be added

Lecture 29 (05-04-2021)

05 April 2021 08:59

Example $f = a^3 b^3 a^3 b^3$, $g = a^2 b^4 a^4 b^2$

Check: $f \sim_4 g$ (Except for baba, all other words of length ≤ 4 can be embedded in both)

$f \wedge g = a^2$

$f = a^2 \boxed{a} b^3 a^3 b^3$
$g = a^2 \boxed{b} b^3 a^4 b^2$

$g' = a^2 a b b^3 a^4 b^2 = a^3 b^4 a^4 b^2$
$f' = a^2 b a b^3 a^3 b^3$

$f \sim_4 g'$ or $g \sim_4 f'$. (By general theorem)

baba $\hookrightarrow f'$ and baba $\not\hookrightarrow g'$

Thus, $f \sim_4 g'$

Thm Let $L \subseteq \Sigma^*$ TFAE

(i) L is recognised by a J-trivial monoid

(ii) L is a union of $\sim_n$-classes for some n [Piecewise-testable language]

(iii) L is definable in the fragment $\mathcal{B}(\Sigma')$ (boolean combinations of $\exists x_1 \cdots \exists x_k \, \varphi(x_1, \ldots, x_k)$, φ quantifier free)

Proof (ii) $\Rightarrow$ (i)

- L is a union of $\sim_n$-classes
- $\rightarrow \Sigma^*/\sim_n$ is a finite monoid

Thus, L can be recognised by $\varphi : \Sigma^* \to \Sigma^*/\sim_n$, $w \mapsto [w]$

We have "shown" that $\Sigma^*/\sim_n$ is J-trivial

(ii) $\Rightarrow$ (iii)

Fix a $\sim_n$ - class and a word $w = a_1 \cdot\ a_k$ of length $k \le n$.

$$\varphi_w = \exists x_1 \quad \exists x_k \left(\left[\bigwedge_{i<j} x_i < x_j \right] \wedge \left[\bigwedge_i a_i(x_i) \right] \right)$$

Note $u \models \varphi_w \iff w \hookrightarrow u$

Now take $\bigwedge \varphi_w$ over all w with $|w| \le n$

(iii) $\Rightarrow$ (ii)

Observe: Let $\varphi_\alpha = \exists x_1 \ldots \exists x_\ell \quad \varphi(\quad)$

q - free

Suppose $u \sim_\ell v$ Then $u \models \varphi_\alpha \iff v \models \varphi_\alpha$.

Lecture 30 (06-04-2021)

06 April 2021 10:28

(i) $\Rightarrow$ (ii) L is recognised by a morphism $\varphi\ \Sigma^* \longrightarrow M$ where M is a finite $\mathcal{J}$-trivial monoid

Let n be the maximum length of a $\mathcal{J}$-chain in M
(Can take $n = |M|$ as well)

Claim L is a union of $\sim_{2n-1}$ - classes

Proof What we show is that

$$f \sim_{2n-1} g \Rightarrow \varphi(f) = \varphi(g)$$

(This, in turn, proves the claim)

To this end, let $f, g \in \Sigma^*$ be s.t. $f \sim_{2n-1} g$

We may assume that $f \hookrightarrow g$

[$\because \exists h \in \Sigma^*$ s.t. $f \hookrightarrow h$, $g \hookrightarrow h$ and $f \sim_{2n-1} h$.]

In fact, we can even assume that $f = uv$ and $g = uav$

[Given $f \hookrightarrow g$, we can find g' s.t. $f \hookrightarrow g' \hookrightarrow g$ and $|g'| - |f| = 1$]

$uv \sim_{2n-1} uav \Rightarrow u \sim_n ua$ or $v \sim_n av$. (Recall from earlier)

- Suppose $v \sim_n av$ We show $\varphi(v) = \varphi(av)$

  (Then, $\varphi(f) = \varphi(uv) = \varphi(u)\varphi(v)$
  $= \varphi(u)\varphi(av) = \varphi(g)$)

$\rightarrow$ By the content lemma, $v = v_1 \cdots v_n$ with $\{a\} \subseteq c(v_1) \subseteq \cdots \subseteq c(v_n)$

$$\varphi(v_1 \cdots v_n) \leq_{\mathcal{J}} \cdots \leq_{\mathcal{J}} \varphi(v_{n-1}v_n) \leq_{\mathcal{J}} \varphi(v_n) \leq_{\mathcal{J}} 1$$

$\downarrow$ 1 $\quad\downarrow$ n-1 $\quad\downarrow$ n $\quad\downarrow$ n+1

By def^n of n, $\exists i$ $\varphi(v_i \cdots v_n) \mathcal{J} \varphi(v_{i+1} \cdots v_n)$

(Not all can be strict $<_J$)

But M is J trivial. Thus, $\varphi(v_1 \, v_{i+1} \, v_s) = \varphi(v_{i+1} \cdot v_n) =: s$

<u>Subclaim</u>: $\forall b \in c(v_1)$: $\varphi(b) s = s$ (Same i as above)

<u>Proof</u>. $b \in c(v_i)$, $v_i = v_i' \, b \, v_i''$.

$$s = \varphi(v_i v_{i+1} \cdots v_n) \leq_J \varphi(b v_i'' v_{i+1} \cdots v_n) \leq_J \varphi(v_i'' v_{i+1} \, v_n) \leq_J \varphi(v_{i+1} \cdot v_n) = s$$

Again, by J-triviality, all the elements above are s

Thus, $s = \varphi(b \, v_i'' \, v_{i+1} \, v_n)$

$= \varphi(b) \, \varphi(v_i'' \, v_{i+1} \, v_n) = \varphi(b) s$ □

Thus, $\varphi(av) = \varphi(a v_1 \, v_{i-1}) \varphi(v_i v_{i+1} \, v_n)$

$= \varphi(a v_1 \, v_{i-1}) s$ $\quad c(a v_1 \cdot v_{i-1}) \subseteq c(v_1)$

$= s$

Similarly, $\varphi(v) = s$

This proves $\varphi(v) = \varphi(av)$, as desired

(The case $u \sim_n ua$ is similar)

Thus, we are done ☒

<u>Simon's Theorem</u> $L \subseteq \Sigma^*$ TFAE

(1) L is piecewise-testable.

(2) The syntactic monoid of L is J-trivial

(3) L is recognised by a finite J-trivial monoid.

Lecture 31 (08-04-2021)

08 April 2021 11:37

Ordered semigroups and ordered monoids

Def^n An ordered semigroup is a semigroup $(S, \cdot)$ along with a partial order on S which is compatible with the semigroup structure.
That is,

$$\forall s_1, s_2 \in S \text{ and } \forall p, q \in S' \qquad s_1 \leq s_2 \Rightarrow ps_1q \leq ps_2q$$

An ordered monoid $(M, \cdot, \leq)$ is an ordered semigroup where $(M, \cdot)$ is a monoid

Example (1) $(\mathbb{N}, +, \leq)$ → ordered semigroup (In fact $\leq$ is a total order)
usual $\leq$

(2) $(\mathbb{N}, \max, \leq)$ → ordered semigroup

(3) $U_1 = \{1, 0\}$

	1	0
1	1	0
0	0	0

$U_1^+ \cdot\ 0 < 1$
$U_1^- \quad 1 < 0$ } ordered monoid
$U_1^= \cdot\ \leq \equiv =$

(4) In general, if $(S,)$ is a semigroup/monoid, then $(S, \cdot, =)$ is an ordered semigroup/monoid
Thus, can interpret ordinary semigroup/monoid as an ordered one.

(5) $(\Sigma^*, \ , =)$ is an ordered monoid

Def^n A morphism φ from $(S, \cdot, \leq)$ to $(T, \cdot, \leq)$ is a map $\varphi: S \longrightarrow T$, (1) φ is a semigroup/monoid morphism,
(2) $s_1 \leq s_2 \Rightarrow \varphi(s_1) \leq \varphi(s_2) \quad \forall s_1, s_2 \in S$

Example Suppose $(S, \ , \leq)$ is an ordered semigroup/monoid.

$id_S : S \longrightarrow S$ is an ordered semigroup/monoid morphism from $(S, \cdot, \Rightarrow)$ to $(S, \cdot, \leq)$.

Product of ordered semigroups

$(S_1, \leq)$ and $(S_2, \leq)$ be ordered semigroups

$(S_1 \times S_2, \preceq)$ is also an ordered semigroup with order

$$(s_1, s_2) \preceq (s_1', s_2') \equiv (s_1 \leq s_1') \text{ and } (s_2 \leq s_2')$$

Order congruence on ordered semigroups

$(S, \cdot, \leq) \rightarrow$ ordered semigroup

Def^n A congruence on $(S, \cdot, \leq)$ is a pre-order (reflexive + transitive) $\preceq$ on S s.t.

(1) $x \leq y \Rightarrow x \preceq y \quad \forall x, y \in S$

(2) $x \preceq y \Rightarrow axb \preceq ayb \quad \forall x, y \in S, \ \forall a, b \in S'$

Quotienting mod this congruence.

Let $\preceq$ be a congruence on $(S, \cdot, \leq)$

Let $\approx$ be the associated eq rel^n to $\preceq$

$$(x \approx y \Leftrightarrow x \preceq y \text{ and } y \preceq x)$$

Easy to see that $\approx$ is a congruence on the semigroup $(S, \cdot)$

[Recall that this meant : $x \approx y \Rightarrow axb \approx ayb \quad \forall x, y \in S \ \forall a, b \in S'$]

Thus, we get the semigroup $S/\approx$

[Recall the operation $[x]_\approx [y]_\approx = [xy]_\approx$ made it a semigroup]

On this, we have the relation $\preceq$ given as

$$[x]_\approx \preceq [y]_\approx \text{ iff } x \preceq y$$

[Well-defined? $x \approx x'$ and $y \approx y'$ and $x \lesssim y \Rightarrow x' \lesssim x \lesssim y \lesssim y'$
$\Downarrow$
$x' \lesssim y'$ ✓]

Moreover $\lesssim$ is a partial order

In fact, $\lesssim$ is compatible with $\cdot$

$$[s_1]_\approx \lesssim [s_2]_\approx \Rightarrow \underbrace{[a]_\approx [s_1]_\approx [b]_\approx}_{= [as_1b]_\approx} \lesssim \underbrace{[a]_\approx [s_2]_\approx [b]_\approx}_{= [as_2b]_\approx}$$

At times, we denote $S/\approx$ by $S/\lesssim$ instead

$$\pi \quad S \longrightarrow S/\approx$$
$$x \longmapsto [x]_\approx = \pi(x)$$

π is a surjective morphism from $(S, \cdot, \lesssim)$ to $(S/\approx, \cdot, \lesssim)$

$(\Sigma^*, \cdot, =) \rightarrow$ ordered monoid

$L \subseteq \Sigma^*$, $\lesssim_L$ — a congruence on $(\Sigma^*, =)$
(will define next class)

Then, we can get a (finite) ordered monoid $\Sigma^*/\lesssim_L$ (if L is regular)

Lecture 32 (12-04-2021)

12 April 2021 09:27

1) <u>Ordered automata</u> (ordered automata)

$A = (Q, q_0, \Sigma, \delta : Q \times \Sigma \to Q, F)$ automata with a partial order $\leq$ on Q such that

$$\forall a \in \Sigma \qquad p \leq q \Rightarrow \delta_a(p) \leq \delta_a(q)$$

$$\left[\forall w \in \Sigma^* \qquad p \leq q \Rightarrow \delta_w(p) \leq \delta_w(q) \right]$$

A natural <u>pre-order</u> on the states of any DFA:

$$A = (Q, q_0, \Sigma, \delta\ Q \times \Sigma \to Q, F)$$

Define $\leq$ on Q as

$$\forall p, q \in Q \qquad p \leq q \equiv \forall w\, [\delta_w(p) \in F \Rightarrow \delta_w(q) \in F]$$

<u>Lemma</u> Let A be the minimum automaton (of the language it is accepting) Then, $\leq$ is in fact a partial order

<u>Proof</u> TS: $\underbrace{p \leq q \text{ and } q \leq p}_{\Updownarrow} \Rightarrow p = q$

$$\forall w \in \Sigma^* [\ \delta_w(p) \in F \text{ iff } \delta_w(q) \in F]$$

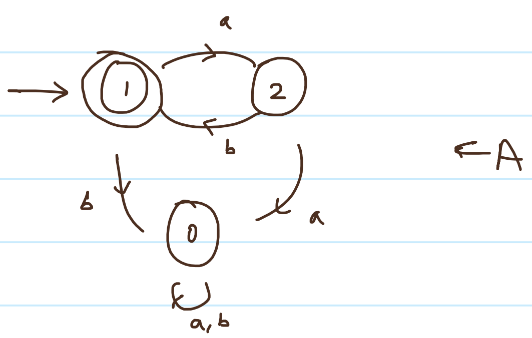
$$\Downarrow$$

$$p = q$$

property of minima automaton

∎

<u>Example</u> $L = (ab)^*$



$\leftarrow A$

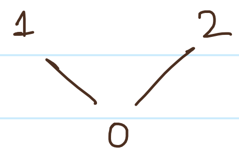
A — min'm DFA of L

$p \leq q \Leftrightarrow$ every word "accepted from p" is also "accepted from q"

$0 < 1$, $0 < 2$ clearly

$1 \not\leq 2$, look at ab

$2 \not\leq 1$, look at b



Hasse diagram

Recognition by ordered monoids

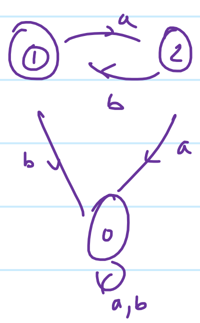
Let $\varphi : M \longrightarrow N$ be a surjective morphism of ordered monoids

A subset $Q \subseteq M$ is said to be recognised by φ if there exists an upper-set $P \subseteq N$ such that $Q := \varphi^{-1}(P)$

$$\left[\forall x, y \in N : (x \in P \text{ and } x \leq y) \Rightarrow y \in P \right]$$

Ex. Check that Q is also an upper set

Example



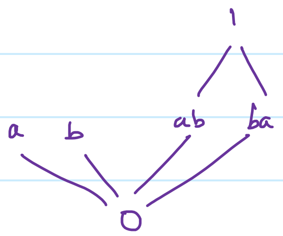
	1	2
1	1	2
a	2	0
b	0	1
ab	1	0
ba	0	2
0	0	0

$M = \{0, 1, a, b, ab, ba\}$

$a^2 = 0 = b^2$

Let us define an order $\leq$ on M:

$ab \leq 1$, $ba \leq 1$, $0 \leq x \quad \forall x \in M$



Ex $\leq$ is stable under multiplication. Thus, $(M, \cdot, \leq)$ is an ordered monoid

$$\varphi : (\Sigma^*, =) \longrightarrow (M, \leq)$$ is a morphism of ordered monoids

$$a \longmapsto a$$
$$b \longmapsto b$$

Now, note that $P = \{ab, 1\}$ is an upper set

Then, $\varphi^{-1}(P) = (ab)^* = L$

Syntactic order

Def^n Let $(M, \leq)$ be an ordered monoid

Let $P \subseteq M$ be an upper set

The syntactic order $\preceq_P$ on M is defined as

$$x \preceq_P y \equiv \left(\forall a, b \in M.\ \ axb \in P \Rightarrow ayb \in P\right)$$

(1) $\preceq_P$ is a pre-order.

(2) $\preceq_P$ contains $\leq$ $\quad x \leq y \Rightarrow x \preceq_P y$

Proof. Let $a, b \in M$ be arbit

$x \leq y \Rightarrow axb \leq ayb$ (def^n of ordered monoid)

Now, if $axb \in P$, then $ayb \in P$ since P is upward closed

$axb \preceq_P ayb$ ∎

(3) $\preceq_P$ is stable under multiplication $\quad x \preceq_P y \Rightarrow axb \preceq_P ayb$

Proof. Let $a, b \in M$. Assume $x \preceq_P y$

Now, suppose $c, d \in M$ are s.t. $c(axb)d \in P$

$$\begin{array}{c} (ca)x(bd) \\ \parallel \end{array}$$

Since $x \precsim_P y$, we get $(ca)y(bd) \in P$.

$$\begin{array}{c} \parallel \\ c(ayb)d \end{array}$$

$\boxtimes$

Lemma. $\precsim_P$ is a congruence

Proof. (1) – (3). $\boxtimes$

Thus, we may quotient to get $M/\precsim_P = (M/\approx_P, \precsim_P)$.

This was an ordered monoid

→ We have the surjective morphism $\varphi \ (M, \leq) \longrightarrow (M/\approx_P, \precsim_P)$ of ordered monoids

→ φ recognises P

In fact, $M/\precsim_P$ is the "smallest" ordered monoid which recognises P