Inverse limits of finite state automata

Michal Ferov

University of Technology, Sydney

Trees, dynamics and locally compact groups
Düsseldorf, Germany
June 29, 2018

Formal languages in discrete groups

- When a finitely generated group is given by a presentation $\langle X \| R \rangle$ we work with sequences of symbols (words) over the finite alphabet $X \cup X^{-1}$ (assuming $X \cap X^{-1} = \emptyset$).
- Sets of words are called *formal languages*.

Formal languages in discrete groups

- When a finitely generated group is given by a presentation $\langle X \| R \rangle$ we work with sequences of symbols (words) over the finite alphabet $X \cup X^{-1}$ (assuming $X \cap X^{-1} = \emptyset$).
- Sets of words are called *formal languages*.

Example

Some languages are of general interest in group theory:

word problem: $WP(G) = \{w \| w =_G 1\}$,

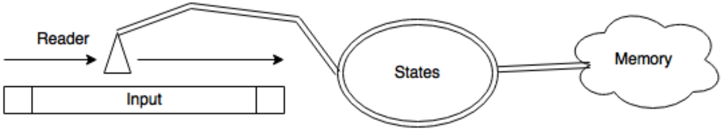
coword problem: $coWP(G) = \{w \| w \neq_G 1\}$,

multiplication table: $mult(G) = \{(u, v, w) \| uv =_G w\}$,

geodesics: $geo(G) = \{w \| \forall w' : w =_G w' \Rightarrow |w| \leq |w'|\}$.

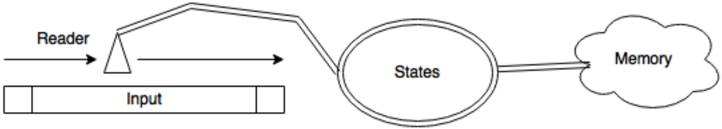
Chomsky hierarchy of languages

A natural way to understand the complexity of a formal language is by quantifying the computational strength of a machine that recognises it.



Chomsky hierarchy of languages

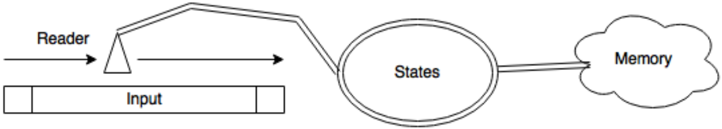
A natural way to understand the complexity of a formal language is by quantifying the computational strength of a machine that recognises it.



- We say that a machine M accepts language L if some computation ends up in a accepting state after reading a word $w \in L$.

Chomsky hierarchy of languages

A natural way to understand the complexity of a formal language is by quantifying the computational strength of a machine that recognises it.



- We say that a machine M accepts language L if some computation ends up in a accepting state after reading a word $w \in L$.

Machine	Memory	Language
Finite state automaton	N/A	Reg
Push-down automaton	Push-down stack	CF
Linear bounded automaton	Linearly bounded tape	CS
Turing machine	Infinite tape	RE

Groups and Chomsky hierarchy

Some languages in group theory have been classified within Chomsky hierarchy:

- regular (co)word problem iff finite (Anisimov),
- context-free word problem iff virtually free (Muller & Schupp),
- context-free multiplication table iff hyperbolic (Gilman),

Some languages in group theory have been classified within Chomsky hierarchy:

- regular (co)word problem iff finite (Anisimov),
- context-free word problem iff virtually free (Muller & Schupp),
- context-free multiplication table iff hyperbolic (Gilman),

Question

What about totally disconnected locally compact groups?
Is there a computational model?

An inspiration...

A group is residually finite if for every $g \in G$ there is $N \trianglelefteq G$ of finite index such that $g \notin G$.

Theorem

Mal'cev If $G = \langle X \mid R \rangle$ is a finitely presented residually finite group then G has solvable word problem.

An inspiration...

A group is residually finite if for every $g \in G$ there is $N \trianglelefteq G$ of finite index such that $g \notin G$.

Theorem

Mal'cev If $G = \langle X \mid R \rangle$ is a finitely presented residually finite group then G has solvable word problem.

Proof.

Run two algorithms in parallel:

- first to enumerate all $w' \in (X \cup X^{-1})^*$ such that $w =_G 1$;
- second to enumerate all $Cay(G/N, X)$ where $N \trianglelefteq_f G$;

An inspiration...

A group is residually finite if for every $g \in G$ there is $N \trianglelefteq G$ of finite index such that $g \notin G$.

Theorem

Mal'cev If $G = \langle X \mid R \rangle$ is a finitely presented residually finite group then G has solvable word problem.

Proof.

Run two algorithms in parallel:

- first to enumerate all $w' \in (X \cup X^{-1})^*$ such that $w =_G 1$;
- second to enumerate all $Cay(G/N, X)$ where $N \trianglelefteq_f G$;

Given a word $w \in (X \cup X^{-1})^*$,

- first algorithm will stop if it finds w,
- second algorithm will stop if it finds $N \trianglelefteq G$ such that w is not a label of a closed loop in $Cay(G/N, X)$.

Exactly one of the algorithms will stop. □

Finite state automaton over X

Definition (X-FSA)

A finite state automaton over a finite alphabet X is a tuple $M = (Q, q_0, A, \delta)$, where

- Q is a finite set of states,
- $q_0 \in Q$ is the initial state,
- $\emptyset \neq A \subseteq Q$ is the set of accepting states,
- $\delta \subseteq Q \times X \times Q$ is the transition relation.

Finite state automaton over X

Definition (X-FSA)

A finite state automaton over a finite alphabet X is a tuple $M = (Q, q_0, A, \delta)$, where

- Q is a finite set of states,
- $q_0 \in Q$ is the initial state,
- $\emptyset \neq A \subseteq Q$ is the set of accepting states,
- $\delta \subseteq Q \times X \times Q$ is the transition relation.

A word $w = x_1 \ldots x_n \in X^*$ takes state q to q' if there is a sequence of states $q_1, \ldots, q_{n-1} \in Q$ such that $(q, x_1, q_1), (q_1, x_2, q_2), \ldots, (q_{n-1}, x_n, q') \in \delta$. Denote $w(q) = \{q' \in Q \mid w \text{ takes } q \text{ to } q'\}$.

Finite state automaton over X

Definition (X-FSA)

A finite state automaton over a finite alphabet X is a tuple $M = (Q, q_0, A, \delta)$, where

- Q is a finite set of states,
- $q_0 \in Q$ is the initial state,
- $\emptyset \neq A \subseteq Q$ is the set of accepting states,
- $\delta \subseteq Q \times X \times Q$ is the transition relation.

A word $w = x_1 \dots x_n \in X^*$ takes state q to q' if there is a sequence of states $q_1, \dots, q_{n-1} \in Q$ such that $(q, x_1, q_1), (q_1, x_2, q_2), \dots, (q_{n-1}, x_n, q') \in \delta$. Denote $w(q) = \{q' \in Q \mid w \text{ takes } q \text{ to } q'\}$.
The machine M accepts word w if $w(q_0) \cap A \neq \emptyset$. The set of words accepted by M is denoted as $L(M)$.

Category of X-FSAs

Definition (morphism of X-FSAs)

Let $M = (Q, q_0, A, \delta)$ and $M' = (Q', q_0', A', \delta')$ be X-FSAs. A map $f : Q \to Q'$ is a morphism of X-FSAs if

- $f(q_0) = q_0'$,
- $f(A) \subseteq A'$,
- $(q_1, x, q_2) \in \delta \Rightarrow (f(q_1), x, f(q_2)) \in \delta'$

and we write $f : M \to M'$. By definition, $L(M) \subseteq L(M')$.

We say that a pair of words w, w' is f-compatible if $f(w(q)) \subseteq w'(f(q))$ for every $q \in Q$.
The set of pairs pair of f-compatible words is closed under coordinate-wise concatenation.

Inverse limit of X-FSAs

Definition (Profinite state automaton over X)

Let $(I, \leq)$ be a directed poset and let

$$\mathcal{M}_I = ((M_i)_{i \in I}, (f_{i,j} \colon M_j \to M_i)_{i \leq j})$$

be a directed system of X-FSAs indexed by I, i.e. $i \leq j \leq k$ implies that $f_{i,k} = f_{i,j} \circ f_{j,k}$.
We say $\hat{M}_I = \varprojlim M_i$ is a profinite state automaton.

The automaton works with sequences of words

$$\hat{W}_I = \{(w_i)_{i \in I} \mid \text{the pair } (w_j, w_i) \text{ is } f_{i,j} \text{ compatible whenever } i \leq j\}.$$

We say that $\hat{M}_I$ accepts $w \in \hat{W}_I$ if M_i accepts w_i for every $i \in I$.

Lemma

If $G = \overline{\langle X \rangle}$ is a finitely generated profinite group then there is a profinite-state-automaton over X that accepts sequences of words in X converging to the identity.

Profinite state automata from profinite groups

Lemma

If $G = \overline{\langle X \rangle}$ is a finitely generated profinite group then there is a profinite-state-automaton over X that accepts sequences of words in X converging to the identity.

Proof.

Suppose that $G = \varprojlim G_i$.
Then interpret $Cay(G_i, X)$ as an X-FSA M_i and set $\hat{M}_I = \varprojlim M_i$.
Obviously, $\hat{M}_I$ accepts $w \in \hat{W}_I$ if and only if w represents the identity in G □

Lemma

Let $G = \overline{\langle X \rangle}$ be a finitely generated group and let $\hat{M}_I = \varprojlim M_i$ what accepts $w \in \hat{W}_I$ if and only if w represents a Cauchy sequence converging to the identity. Then G is a profinite group, in particular $G = \varprojlim G_i$.

Lemma

Let $G = \overline{\langle X \rangle}$ be a finitely generated group and let $\hat{M}_I = \varprojlim M_i$ what accepts $w \in \hat{W}_I$ if and only if w represents a Cauchy sequence converging to the identity. Then G is a profinite group, in particular $G = \varprojlim G_i$.

Proof.

For every $i \in I$ can construct a X-FSA M_i' and a morphism $f_i \colon M_i \to M_i'$ such that $L(M) = L(M')$ and $M_i' \cong Cay(G_i, X)$ as a decorated graph.

Profinite groups from profinite state automata

Lemma

Let $G = \overline{\langle X \rangle}$ be a finitely generated group and let $\hat{M}_I = \varprojlim M_i$ what accepts $w \in \hat{W}_I$ if and only if w represents a Cauchy sequence converging to the identity. Then G is a profinite group, in particular $G = \varprojlim G_i$.

Proof.

For every $i \in I$ can construct a X-FSA M_i' and a morphism $f_i \colon M_i \to M_i'$ such that $L(M) = L(M')$ and $M_i' \cong Cay(G_i, X)$ as a decorated graph.

Start at the bottom and consistently work your way upwards. □

Questions?

Questions?

Thank you!