Announcements Mar 23

- Cameras on
- HW due Thu 3:30
- Office Hours Fri 2-3, appt
  + makeup 10-11 Wed
- Progress report Apr 2 ~ 1 page
- First draft Apr 9
- Talk to me about makeup points!

Today
- Howson's thm
- Regular languages
- Automata
Howson's TNN

Thm 7.32 (1954)
If $G, H$ f.g. subgps of $F_n$ then $G \cap H$ is f.g.

A "counterexample" with $F_n$ replaced by another group:

Take $F_2 \times \mathbb{Z}$, $F_2 = \langle x, y \rangle$, $\mathbb{Z} = \langle z \rangle$

$G = F_2$ (first factor)
$H = \ker (F_2 \times \mathbb{Z} \to \mathbb{Z})$

all 3 gens $\to 1$

To check:
1. $G \not\text{fg.}$  ✓
2. $H \text{fg.}$
3. $G \cap H$ not f.g.

2. Claim: $H$ is gen by $\{xz^{-1}, yz^{-1}, z\}$

Step 1. $\langle S \rangle$ normal.
   To show: $gsg^{-1} \in \langle S \rangle$
   $g = (\text{gen for } F_2 \times \mathbb{Z})^t$ $s \in S$.

example: $y(x^{-1}z)y^{-1} = (yz^{-1})(x^{-1}z)(y^{-1}z)$

Step 2. $\langle S \rangle \subseteq H$  ✓
Step 3. $(F_2 \times \mathbb{Z})/\langle S \rangle \cong \mathbb{Z}$

We get $F_2 \times \mathbb{Z}$ subject to $x = z, y = z$  ✓
\[ F_2 \times \mathbb{Z} \quad F_2 = \langle x, y \rangle \quad \mathbb{Z} = \langle z \rangle \]

G = F_2 (first factor)

H = ker (F_2 \times \mathbb{Z} \to \mathbb{Z})

all 3 gens \( \to 1 \)

Remains.

3) G\( \cap H \) not F_{G_{\mathbb{Z}}}

G\( \cap H \) is the subgp of F_2:

\[ \text{ker } F_2 \to \mathbb{Z} \]

\[ x, y \mapsto 1 \]

(exponent sum 0).

Claim. G\( \cap H \) is freely gen by

\[ x^i y^{-i} \]

Example. \( x^5 y^{-5} x^3 y^{-3} y^2 x^{-2} \)

Very similar to HW problem:

\[ \text{ker } F_2 \to \mathbb{Z}^2 \]

\[ x \mapsto (1, 0) \]

\[ y \mapsto (0, 1) \]

Freely gen by

\[ \{ x^i y^j x^{-i} y^{-j} \} \]
Hanna Neumann Conjecture (1957)

\[ \text{rk} (G \cap H) - 1 \leq (\text{rk}(G) - 1)(\text{rk}(H) - 1) \]

for \( G, H \leq F_n \)

Proved in 2011 by Friedman, Mineyev.

Our proof of Howson's thm

- uses regular languages, automata.

Today: automaton version of Howson's thm. Thu: Howson's thm.
Languages

\( S = \{x_1, \ldots, x_n\} \) “alphabet”

\( S^* = \{\text{words of finite length in } S\} \)

Any subset \( L \subseteq S^* \) is called a language.

Examples

1. \( S = \{a, \ldots, z\} \quad L = \{\text{words in } \text{OED}\} \)
2. \( S = \{a^3\} \quad L = \{a^n : 3 \mid n\} \)
3. \( S = \{a, b, c\} \quad L = \{a^i b^j c^k : i > 0, j > 0, k > 0\} \)
4. \( S = \{\text{gen set for } G\}^{\geq 1} \quad L = \{\text{words in } S \text{ that equal } \text{id} \text{ in } G\}. \)
Automata (= simple computer)
$S =$ alphabet (finite set)
An automaton $M$ over $S$ consists of a directed graph with decorations:
* some subset of vertices called start states $S$
* some subset $A$ of vertices called accept states $A$
* edges labeled by elts of $S$.
If the graph is finite, $M$ is a finite state automaton.

The language accepted by $M$ is
\[ \{ w \in S^* : w \text{ given by a directed path in } M \} \]

Examples
\[ \begin{array}{c}
\begin{array}{c}
(5) \\
(0)
\end{array}
\end{array} \]
\[ \begin{array}{c}
(5) \xrightarrow{a} (0) \xrightarrow{a} (0)
\end{array} \]

$\sim L = \{ a^i : 3 | i \}$
Poll: Is there a simpler automaton for the same language?

Deterministic! Yes!

\( h = \{ \text{words with } b\text{-exponent even} \} \)

\[ a^3b^5ab \checkmark \]
Deterministic automata

A det. aut. is a FSA with
- exactly one start state
- no two edges leaving same vertex have same label
- no edges with empty label
  (in Meier: empty = $\varepsilon$)

It is complete if each vertex has departing edges with all possible labels.

What's deterministic about it?

words $\leftrightarrow$ paths

The word $\omega$ corresponds to more than 1 path.

To see if a word is in the accepted language, start at the start state, trace out the word/path, see if land at accept state.

A language is regular if accepted by a det. FSA.
Automaton version of Howson's Thm

**Lemma 1.** \( L \) accepted by a det. FSA
(i.e. \( L \) is regular) \( \Rightarrow \) \( L \) accepted by a complete det FSA.

Pf. (exercise: add dead ends/fail states)

**Lemma 2.** \( L \) accepted by a non-det. FSA \( \Rightarrow \) \( L \) accepted by a det. FSA.

In other words: starting with a non-det FSA, Lemma 2 converts it to a det FSA, Lemma 1 converts to a complete det FSA.

**Thm 7.11** Say \( K, L \subseteq S^* \) are reg. languages. Then so are:

1. \( S^* \setminus K \)
2. \( K \cup L \)
3. \( K \cap L \)
4. \( KL = \{ w_k w_l : w_k \in K, w_l \in L \} \)
5. \( L^* = LLULULLLU \ldots \)

**reg. lang. is automaton version of f.g.**
Lemma 2. \( L \) accepted by a non-det. FSA \( \Rightarrow \) \( L \) accepted by a det. FSA.

1. Two steps:
   1. Get rid of arrows with empty labels.
   2. Get rid of \( \emptyset \) vertices.

Example. \( L = \{ a^i b^j : i, j > 0 \} \)

\( a \xrightarrow{c} \emptyset \xrightarrow{a} \emptyset \xrightarrow{b} \emptyset \)

New vertices: subsets of old vertices.
New Start vertex: set of all old start vertices.
New Accept vertices: all sets containing an accept.

\[ \emptyset \xrightarrow{a} 0 \xrightarrow{a} 0 \xrightarrow{b} 1 \xrightarrow{b} 2 \xrightarrow{a} 0 \]

\( a \xrightarrow{c} \emptyset \xrightarrow{a} 0 \xrightarrow{b} 1 \xrightarrow{a} 0 \]

\( \emptyset \xrightarrow{a} 0 \xrightarrow{a} 0 \]
yw is in vertex restriction to $V$ of $Tr G S$. Sym $T g$ gives $\mu$. Nice $G$ if you look all $g G Sym$ 54mA. Now and all $gtv$ get a finite subset of $Sym C T$. Those are the finite States IN. lt 4i j

i

i

i

i

ixo.oi

VW 0100100010000

gg lvsgilg.cn 924
We can now convert \( M_\varepsilon \) into a deterministic automaton, \( D \). The states of \( D \) consist of all the subsets of \( V(M_\varepsilon) \). The single start state of \( D \) is the subset of \( V(M_\varepsilon) \) consisting of all the start states of \( M_\varepsilon \). The accept states of \( D \) are the subsets of \( V(M_\varepsilon) \) that contain at least one accept state of \( M_\varepsilon \). In \( D \) there is an edge from \( U \) to \( U' \) labelled by \( x \) if, for each \( v \in U \), there is an edge labelled \( x \) from \( v \) to some \( v' \in U' \), and \( U' \) is entirely composed of such vertices. That is, there is an edge labelled \( x \) from \( U \) to the vertex corresponding to the set

\[
U' = \{ v' \in V(M_\varepsilon) \mid v' \text{ is at the end of an edge labelled } x \text{ that begins at some } v \in U \}.
\]