irreducible : v irr if $\nexists w$ st $v \to w$

RWS terminating : if NOT $w_1 \to w_2 \to \cdots \infty$

Confluent : q, r, s, t

if $q \Rightarrow r$, $q \Rightarrow s$ then $\exists t$

st $r \Rightarrow t$ & $s \Rightarrow t$.

RWS complete if confluent + terminating

locally confluent $q \to r$, $q \to s$, $s \Rightarrow t$, $r \Rightarrow t$

one step

many steps.

actually the same thing

Dehn alg confluent on id

- we aren't sure for others b/c

$w \neq 1_G$ can be written in many ways.

Critical pair : pair of words which overlap.

$v_1 \to w_1$, $v_2 \to w_2$ crit pair if

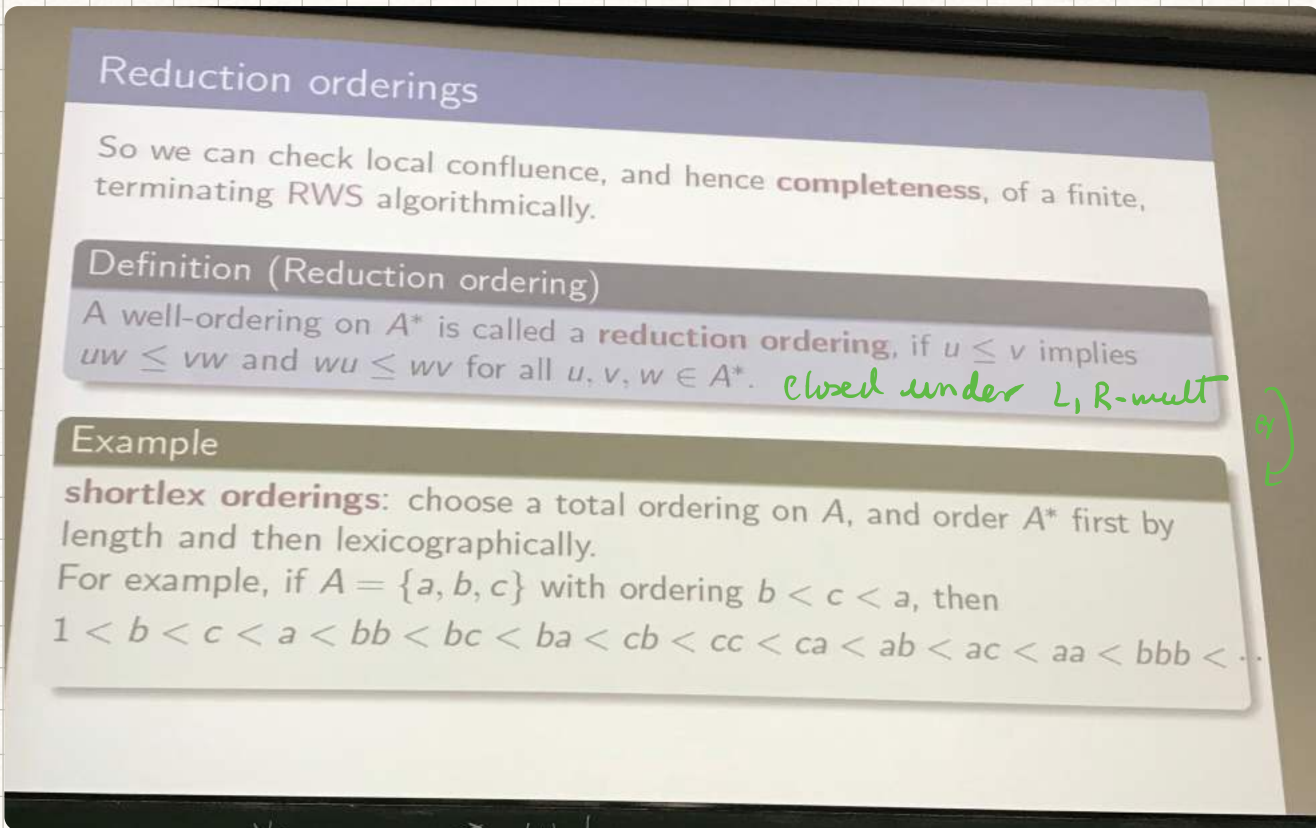
$v_1 = rs$ v_2 st

or $v_1 = rst$ $v_2 = s$

w/ $s \neq 1$

Lemma: RWS locally confluent if

iff $\forall$ crit pair ...



closed under L, R-mult

Let $<$ red. ord. Let R RWS st $v > w$ $\forall\, v \to w$. Then $w_1 > w_2$ when $w_1 \to w_2$

so R is terminating. (b/c $\exists$ a smallest word)

Knuth-Bendix

1) Start w/ RWS (finite) and choose reduction ordering st $v > w$ $\forall\, v \to w$.

2) Consider all crit pairs $rs \to w_1$, $st \to w_2$

rewrite: $w_1 t \Rightarrow w_1'$

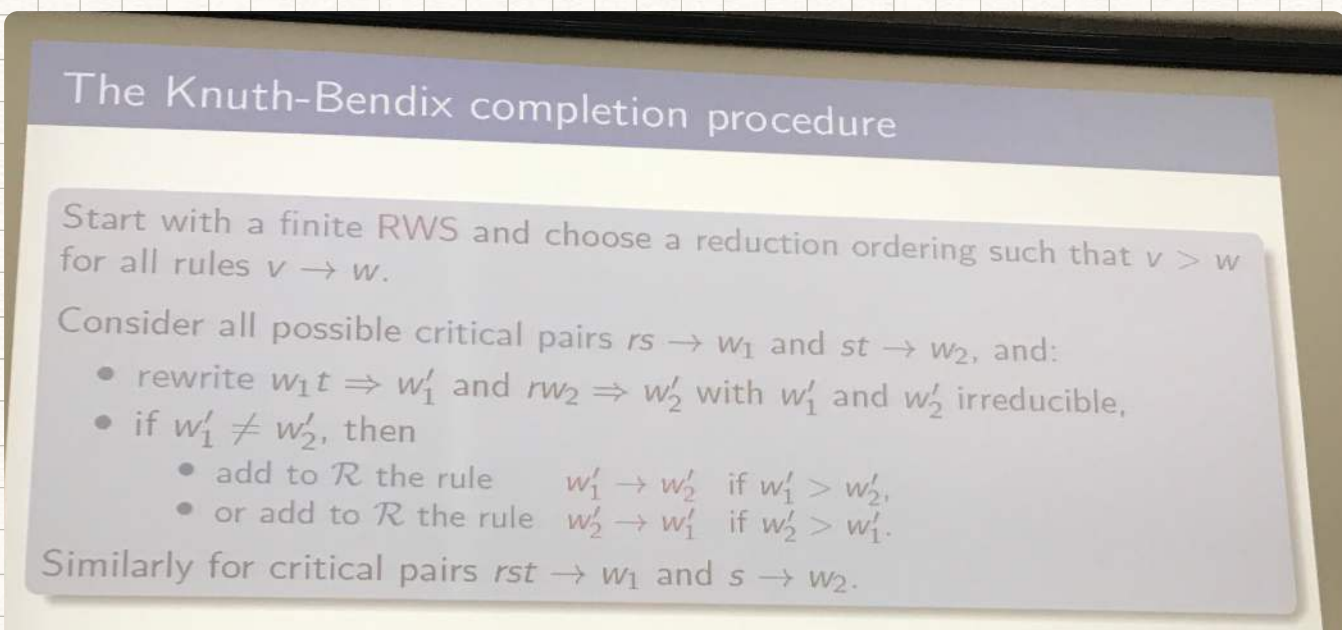
$r w_2 \Rightarrow w_2'$

w/ w_1', w_2' NR

if $w_1' \neq w_2'$

add rule $w_1' \to w_2'$ if $w_1' > w_2'$

$w_2' \to w_1'$ if $w_2' > w_1'$



- Not guarantee to term.
- If it terminates $\Rightarrow$ RWS locally confl & complete

Knuth-Bendix and groups

Let $G = \langle X \mid R \rangle$ and $A = X \cup X^{-1}$. We can present G as a monoid by adding relations $aa^{-1} = 1$ for all $a \in A$.

Choose a reduction order on A^* and define an RWS to have one rule $v \to w$ with $v > w$ for each relation $v = w$ in the monoid presentation. Then apply the Knuth-Bendix completion procedure.

If it terminates with a complete RWS, then group elements are uniquely represented by irreducible words, and we can solve the word problem in G by reducing words to their unique irreducible representatives.

- If G fin. guaranteed to complete.
- If $|G| = \infty$, then not guaranteed
  → but good ordering for polycyclic grps
  recursive path ordering,
  the RWS : power + conjugate relations.

As a simple example, consider the nonabelian group of order 6.

Example

$$G = \langle a, b \mid a^2, b^3, (ab)^2 \rangle,\ A = \{a, b, \bar{b}\},\ \bar{b} := b^{-1};$$
shortlex ordering: $a < b < \bar{b}$.

Write rules : $a^2 \to 1$, $b^2 \to \bar{b}$, $ba \to a\bar{b}$
$a\bar{a} \to 1$, $b\bar{b} \to 1$, $\bar{b}b \to 1$,

overlap: $b^2\bar{b} \to b$,
$b^2\bar{b} \to \bar{b}^2$ new rule: $\bar{b}^2 \to b$

As a simple example, consider the nonabelian group of order 6.

Example

$$G = \langle a, b \mid a^2, b^3, (ab)^2 \rangle,\ A = \{a, b, \bar{b}\},\ \bar{b} := b^{-1};$$

shortlex ordering: $a < b < \bar{b}$.

Initial rules from presentation:

$$a^2 \to 1,\ b\bar{b} \to 1,\ \bar{b}b \to 1,\ b^2 \to \bar{b},\ ba \to a\bar{b}$$

From overlap of left hand sides b^2 and $b\bar{b}$, we get:

$$b^2\bar{b} \to b,\ b^2\bar{b} \to \bar{b}^2 \quad \textbf{new rule}: \bar{b}^2 \to b$$

From overlap of left hand sides b^2 and ba, we get:

$$b^2 a \to \bar{b}a,\ b^2 a \to ba\bar{b} \to a\bar{b}^2 \to ab \quad \textbf{new rule}: \bar{b}a \to ab$$

etc. get 7 - rewrite rules. (lookup Knuth-Bendix ex's)

Knuth-Bendix good for trying to check <u>when</u> $w =_G 1$, but <u>not</u> if $w \neq_G 1$.

The Knuth-Bendix completion procedure is a useful tool for computing with groups.

Unfortunately it does not usually terminate on infinite groups. But it can help solve positive instances of the word problem in the following sense.

We can run Knuth-Bendix as described above on $G = \langle X \mid R \rangle$, and stop at some stage with an enlarged RWS $\mathcal{R}$. Then:

(i) If $w \Rightarrow_{\mathcal{R}} 1$ then $w =_G 1$.

(ii) Conversely, if $w =_G 1$ then, if we run Knuth-Bendix for long enough, we will eventually have $w \Rightarrow_{\mathcal{R}} 1$.

We turn now to another approach to solving the word problem, which works in many classes of infinite groups.

Another approach:

Finite state automata

A **finite state automaton (fsa)** is a simple computing machine M which takes as input strings in a fixed finite alphabet A.

The machine is always in one of a finite set Σ of states, one of which is designated as the start state.

The letters in the string are read successively by M. If it is in state s and reads the letter a, then it changes into the state $\tau(s, a)$, where $\tau : \Sigma \times A \to \Sigma$ is a function known as the **transition function** for M.

Some of the states are designated as **accepting** (or **final**). If M is in an accepting state after reading the string w, then it outputs the answer **yes** and is said to accept w. Otherwise it outputs the answer **no**, and rejects w. These are the only two possible outputs.

$L(M)$ = words accepted by M.

Finite state automata: example

Example

In this example $A = \{a, a^{-1}, b, b^{-1}\}$, $\Sigma = \{1, 2, 3, 4, 5, 0\}$, the start state is 1, the accepting states are $\{1, 2, 3, 4, 5\}$, and the transition function τ is given by the table below.

	a	a^{-1}	b	b^{-1}
1	2	3	4	5
2	2	0	4	5
3	0	3	4	5
4	2	3	4	0
5	2	3	0	5
0	0	0	0	0

The accepted language $L(M)$ is the set of reduced words in A^*.

Word acceptors

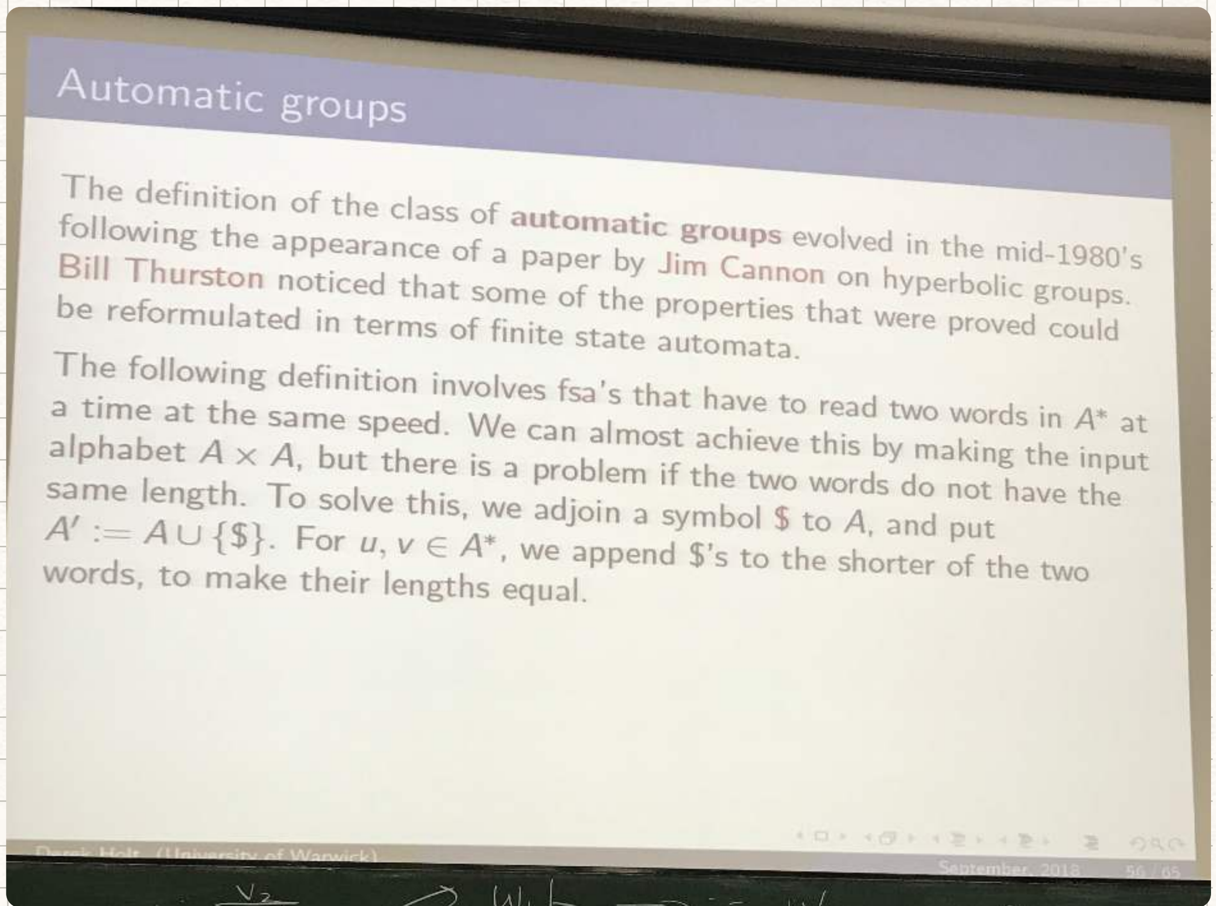
If $G = \langle X | R \rangle$ is a finitely presented group, and $A = X \cup X^{-1}$, then a fsa with input alphabet A is called a **word-acceptor** for G if it accepts at least one word in A^* for each group element.

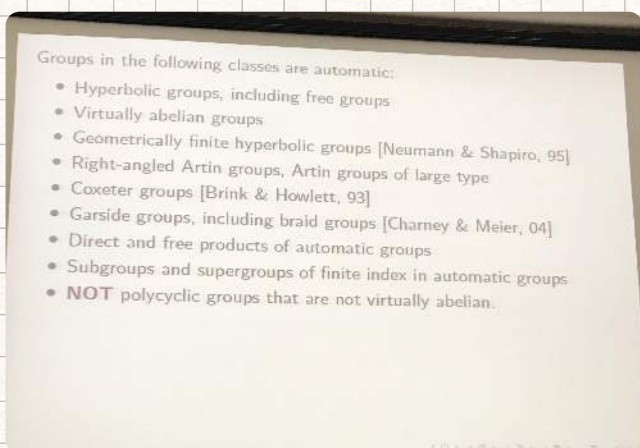
It is called a **unique word-acceptor** for G if it accepts a unique word for each group element.

The above example is a unique word acceptor for the free group $F(\{a, b\})$. It is not difficult to construct a unique word acceptor for a group for which we have a finite complete rewriting system.

One of the standard operations that one can perform on an fsa M is to enumerate its language $L(M)$ (usually up to a given word-length). In particular, we can enumerate the elements of a finitely presented group have a unique word-acceptor.

It is also straightforward to determine $|L(M)|$ (which may be infinite) and hence, if we have a unique word acceptor for G, then we can calculate $|G|$.





Nilpotent → Not auto

Automatic groups - an equivalent definition

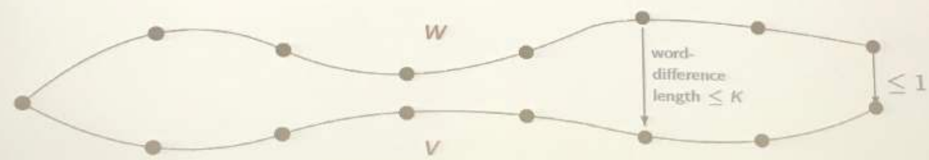
Definition

G is **automatic** if there exists an fsa W and a constant K such that:

(i) W has input alphabet A, and is a (unique) word acceptor for G;

(ii) if $w, v \in L(W)$ with $w^{-1}v \in A \cup \{1\}$, then w and v (synchronously) K-fellow travel in the Cayley graph $\Gamma := \Gamma(G, A)$ of G. That is, $d_\Gamma(w(i), v(i)) \leq K$ for all $i \geq 0$.

The finite collection of group elements $w(i)^{-1}v(i)$ arising from relations $wa =_G v$ with $a \in A \cup \{1\}$ are called the **word-differences** of these relations.



A procedure to compute shortlex automatic structures

Input: A finite presentation of a group G;
Output: The automata in a shortlex automatic structure for G or **fail**.

Step 1 Run the Knuth-Bendix procedure (with shortlex ordering) on G until the set D of all word-differences arising from all rewrite rules $v \to w$ appears to have stabilized.

Then construct fsa W_D over $A' \times A'$ with state set D, start and accept state 1, and transitions $u \xrightarrow{(a,b)} v$ with $u, v \in D$ and $a, b \in A'$ with $a^{-1}ub =_G v$.

...omatic structures

Input: A finite presentation of a group G;
Output: The automata in a shortlex automatic structure for G or **fail**.

Step 1 Run the Knuth-Bendix procedure (with shortlex ordering) on G until the set D of all word-differences arising from all rewrite rules $v \to w$ appears to have stabilized.

Then construct fsa W_D over $A' \times A'$ with state set D, start and accept state 1, and transitions $u \xrightarrow{(a,b)} v$ with $u, v \in D$ and $a, b \in A'$ with $a^{-1}ub =_G v$.

Step 2 Construct W and M_a as the fsa with

$$L(W) = \{w \in A^* : \nexists v \in A^* \text{ with } (w,v)^+ \in L(W_D) \text{ and } v < w\}.$$

$$L(M_a) = \{(w,v)^+,\ w, v \in A^* : w, v \in L(W),\ (w,v)^+ \in L(W_D),\ wa =_G v\}.$$

Step 3 Check the condition

$$\{\forall w \in L(W) : \exists v \in L(W) \text{ with } (w, v) \in L(M_a)\} = L(W)$$

for all $a \in A \cup \{1\}$. If it fails then enlarge the set D and repeat Step 2.

Step 4: Verify correctness of computed automatic struct by checking that M_a satisfy grp relations.

Applications of automatic groups software

Once the shortlex automatic structure has been computed we can use it to:

- determine whether the group is finite or infinite;
- reduce words quickly (quadratic time) to a word in $L(W)$, and hence solve the word problem in quadratic time;
- enumerate normal form words up to a specified length.
- calculate the growth series of $G = \langle X \rangle$ as a rational function: $\sum_{n \geq 0} c_n x^n$, where c_n is number of group elements with shortest representatives in A^* of length (at most) n.

We have had significant success in resolving some difficult finiteness problems. For example the **Heineken group**

$$\langle x, y, z \mid [x,[x,y]] = z,\ [y,[y,z]] = x,\ [z,[z,x]] = y \rangle.$$

is infinite (and even hyperbolic),

and also for several difficult cases of groups in the Coxeter family

$$(\ell, m, n; p) = \langle x, y \mid x^{\ell},\ y^{m},\ (xy)^{n},\ [x,y]^{p} \rangle,$$

such as $(\ell, m, n; p) = (3,4,13;2)$ and $(3,5,7;2)$.

Enumerating words has had applications to software for drawing pictures.

For example, the vertices in the picture below correspond to elements from the hyperbolic group

$$\langle a,b,c,d,e,f \mid a^4 = b^4 = c^4 = d^4 = e^4 = f^4 = \\ aba^{-1}e = bcb^{-1}f = cdc^{-1}a = ded^{-1}b = efe^{-1}c = faf^{-1}d = 1 \rangle,$$

which is the symmetry group of the tesselation in the picture.

Figure: A tessellation of hyperbolic 3-space by dodecahedra

Other recent work

A variety of theoretical complexity results on decision problems in finitely presented groups have been proved, but many of these do not lead to immediately to effective algorithms, typically because they involve enormous constants.

For example, the conjugacy problem in hyperbolic groups is known to be solvable in linear time in hyperbolc groups. A version of this has been implemented (by Joe Marshall) that works reasonably quickly for elements of infinite order.

The **generalized word problem** (GWP) is to test membership of a given element $g \in G$ in a specified subgroup $\langle Y \rangle \leq G$ (with Y finite).

A generalization of the theory of automatic groups to the cosets of a subgroup enables practical algorthms to solve this in some cases.