Formal Language Theory 2

1 Closure properties of Reg

Is Reg closed under $\cap, \cup, -$?

(1) For all $L_1, L_2 \in Reg$,
  a. $L_1 \cap L_2 \in Reg$
  b. $L_1 \cup L_2 \in Reg$
  c. $L_1 - L_2 \in Reg$

Consider intersection. Suppose $L_1, L_2 \in Reg$ to show $L_1 \cap L_2 \in Reg$.

Proof idea. Since $L_1, L_2 \in Reg$, there are autonoma A_1, A_2 such that $L(A_1) = L_1$ and $L(A_2) = L2$.

- Now we find an algorithm for creating a new FSA $A_1 \cap A_2$: $L(A_1 \cap A_2) = L_1 \cap L_2$.

Here is the algorithm:

Let $L(A_1) = L_1$ and let $L(A_2) = L_2$.

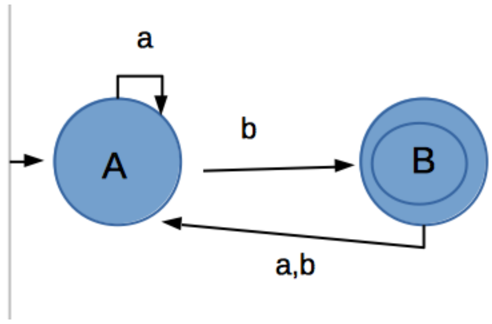
So $A_1 : \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$ and $A_2 : \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$.

Let $A_{1\cap 2} : \langle Q, \Sigma, \delta, q_0, F \rangle$, such that

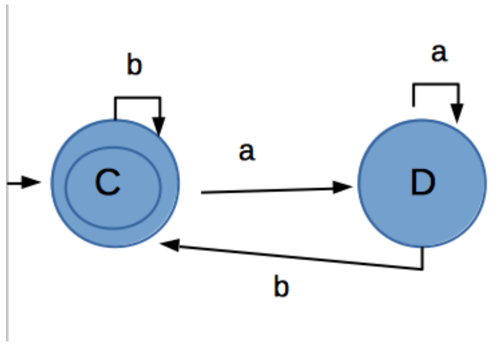
1. $Q = \{\langle r_1, r_2 \rangle : r \in Q_1 \text{ and } r_2 \in Q_2 \rangle$
2. $q_0 = \langle q_1, q_2 \rangle$
3. $F_1 \times F_2$
4. δ : for all $\langle r_1, r_2 \rangle \in Q$ and $a \in \Sigma$, $\delta(\langle \langle r_1, r_2 \rangle, a \rangle) = \langle \delta_1(r_1, a), \delta_2(r_2, a) \rangle$.

Example. Show that the intersection of the languages generated by FSAs A_1 and A_2 is also regular.

(2) A_1



(3) A_2



What are closure properties of regular languages good for?

- (Among other things) helping us establish which languages are regular.
- If $L_1, L_2 \in Reg$, then $L_1 \cap L_2 \in Reg$.

Suppose $L_1 \in Reg$ and $L_1 \cap L_2 \notin Reg$. Then $L_2 \notin Reg$.

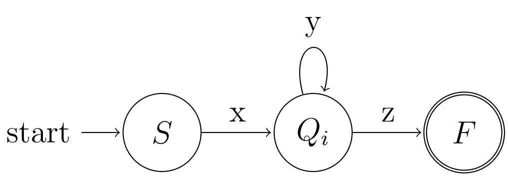
- We will use this reasoning to show that ENG $\notin Reg$.
- We pick a language that we know is regular, intersect it with English to produce another language, which we then show to $\notin Reg$.

1.1 How to show $L \notin Reg$?

We use a **Pumping Lemma**.

Consider an infinite $L \in Reg$. So L is accepted by an FSA.

- Since L is infinite, the FSA has a loop.



A accepts xyz, $xyyz, xyyyz$, etc. In other words, for all $n \geq 0$, A accepts xy^nz.

- There is a particular **structure** in L(A).

Définition 1.1 ***Pumping Lemma.*** *If $L \in Reg$, then there are strings $x, y, z \in \Sigma^*$ such that $y \neq \epsilon$ and $xy^nz \in L$, for all $n \geq 0$.*

Consider a^nb^n. Is $a^nb^n \in Reg$?

- If $a^nb^n \in Reg$, it has the structure xy^nz.
- So where is the y^n part?

(4) Cases

a. Some number of *a*s followed by some number of *b*s. $x(ab)^nz$
b. Some number of *a*s. xa^nz
c. Some number of *b*s. xb^nz

Pump these patterns:

(5) $x(ab)^n z$

a. ab n = 0
b. aabb n = 1
c. aababb $\notin a^n b^n$ n = 2

(6) $xa^n z$

a. ab n = 0
b. aab $\notin a^n b^n$ n = 1

(7) $xb^n z$

a. ab n = 0
b. abb $\notin a^n b^n$ n = 1

Therefore $a^n b^n \notin Reg$.

1.2 What about Eng(lish)?

Consider the subset of Eng:

(8) L_1

a. A famous prof hired a famous prof.
b. A famous prof a famous prof hired, hired a famous prof.
c. A famous prof a famous prof a famous prof hired hired hired a famous prof.

L_1 has the structure (a famous prof)^n(hired)^n a famous prof, so $L_1 \notin Reg$.

- $L_1 \subset Eng$.
- $L_1 \subset$ (a famous prof)*(hired)* a famous prof (L_2).
- $L_2 \cap Eng = L_1$.

We know that Reg is closed under intersection, $L_2 \in Reg$ and $L_1 \notin Reg$. Therefore, $Eng \notin Reg$ □

More generally: languages which have unbounded structures requiring matching are not regular.

2 Regular Grammars

FSAs don't look very much like grammars linguists use. So we can adopt an equivalent formalism: **Regular** (Type 3) grammars. (also known as *left/right linear* grammars).

Définition 2.1 *Let G be a regular grammar $\langle V, \Sigma, R, S\rangle$, where*

1. *V is a set of variables/non-terminals/categories.*
2. *Σ is a terminal/vocabulary symbols. V and Σ are disjoint.*
3. *R is a finite set of rules.*
4. *$S \in V$ is the start variable.*

(9) Condition on rules:

a. Right linear grammar: All rules in R are one of the following:
1.B $\rightarrow a$, where $B \in V$ and $a \in \Sigma$.

2.B $\rightarrow aC$, where $B, C \in V$ and $a \in \Sigma$.

b. Left linear grammar:

1.$B \rightarrow a$

2.$B \rightarrow Ca$

Example: G = $\langle V, \Sigma, S, R \rangle$, where

1. V = {S, A, B}
2. $\Sigma = \{a, b\}$

Sample derivation? How to do this with FSA?

3 Context-Free Grammars

Définition 3.1 *Let G be a context-sensitive grammar $\langle V, \Sigma, R, S \rangle$, where*

1. *V is a set of variables/non-terminals/categories.*
2. *Σ is a terminal/vocabulary symbols. V and Σ are disjoint.*
3. *R is a finite set of rules.*
4. *$S \in V$ is the start variable.*

Each rule has the form: Variable → string of variables and/or non-terminals.

$G_{a^n b^n} = \langle \{S\}, \{a, b\}, R, S \rangle$

- R = {
  1. S → aSb
  2. S → ϵ }

The famous professor language:

G = $\langle \{S, AP, DP, VP, NP\}, \{a, famous, professor, hired\}, S, R \rangle$
R =

1. S → DP VP
2. DP → D AP
3. AP → A NP
4. NP → N
5. VP → V DP
6. D → a
7. A → famous
8. N → professor
9. V → hired

Fun facts about CF languages:

1. There is a pumping lemma for CF languages, which is more complicated than the pumping lemma for regular languages.

   (10) $a^n b^n c^n \notin CF$

2. CF langs are closed under union, but **not** intersection.

3. CF langs are closed under intersection with **regular** languages.

   Consider the copying language: $\{xx|x \in \{a,b\}^*\}$

   - This language features cross-serial dependencies, and unbounded copying.
   - It is also not context free.

How to show a natural language is not context free:

1. We know that CF langs are closed under intersection with regular languages.
2. So we show that the intersection of a regular language and our natural language is not context-free.

A common way to do it: show that the intersection has the same structure as the copying language.

4 Argument that Bambara is not context-free (Culy 1985)

Bambara: a Mandé language spoken in Mali.

- Bambara has a free choice construction that involves copying.

(11) a. wulu 'dog' wulu o wulu 'whichever dog'
b. malo 'rice' malo o malo 'whichever rice'
c. *wulu o malo
d. *malo o wulu

There is also an agentive nominalization construction: N + V_{trans}+la.

(12) a. wulu + nyini + la = wulunyinina 'one who searches for dogs'
b. wulu + filè+ la = wulufilèla 'one who watches dogs'

The agentive construction can appear in the free choice construction:

(13) a. wulunyinina o wulunyinina 'whoever searches for dogs'
b. wulufilèla o wulufilèla 'whoever watches dogs'

(14) a. wulunyininanyinina o wulunyininanyinina 'whoever searches for dog searchers'
b. wulufilèlafilèla o wulufilèlafilèla 'whoever watches dog watchers'
c. wulufilèlanyinina o wulufilèlanyinina 'whoever searches for dog watchers'

Let R = $\{\text{wulu(filèla)}^h\text{(nyinina)}^i \text{ o wulu(filèla)}^j\text{(nyinina)}^k | h,i,j,k \geq 1\}$ **regular language**.

- BAM∩R = $\{\text{wulu(filèla)}^n\text{(nyinina)}^m \text{ o wulu(filèla)}^n\text{(nyinina)}^m | m,n \geq 1\}$, which has a similar structure to the copying and counting languages, which can be shown to be non context free.

Conclusion: NL $\not\subseteq CF$.