Pumping Lemma

L is regular $\implies (\exists \mid k \geq 0 : (\forall x, y, z \in L \wedge |y| \geq k : (\exists u, v, w | y = uvw \wedge |v| > 1 : (\forall i \mid i \geq 0 : xuv^iwz \in L))))$

- demon picks k
- you pick $x, y, z \leftarrow xyz \in L \wedge |y| \geq k$
- demon picks $u, v, w \leftarrow uvw = y \wedge |v| \geq 1$
- you pick an $i \geq 0$, and show $xuv^2wz \notin L$

Church-Turing Thesis

> Conjecture 1: All reasonable models of computation are equivalent:
>
> - perfect memory
> - finite amount of time
>
> Conjecture 2: Anything a modern digital computer can do, a Turing machine can do.

Equivalence model

- TMs with multiple tapes.
- NTMs.
- PDA with two stacks.

context-free grammar

$$G = (N, \Sigma, P, S) \quad \begin{aligned} &N : \text{non-terminal symbols} \\ &\Sigma : \text{terminal symbols } s.t\ \Sigma \cap N = \emptyset \\ &P : \text{production rules } s.t \text{a finite subset of } N \times (N \cup \Sigma)^* \\ &S : \text{start symbol} \in N \end{aligned}$$

Finite Automata from Church-Turing Thesis

Finite automata can be encoded as a string:

Let $0^n10^m10^j0^{k_1} \dots 10^{k_n}$ be a DFA with n states, m input characters, j final states, $k_1 \dots k_n$ transitions

$$A_{\text{DFA}} = \{M\#w \mid M \text{ is a DFA which accepts } w\} \quad (1)$$
$$A_{\text{TM}} = \{M\#w \mid M \text{ is a TM which accepts } w\} \quad (2)$$

M is a "recognizer" $\implies M(x) = \begin{cases} \text{accept} & \text{if } x \in L \\ \text{reject or loop} & \text{if } x \notin L \end{cases}$

M is a "decider" $\implies M(x) = \begin{cases} \text{accept} & \text{if } x \in L \\ \text{reject} & \text{if } x \notin L \end{cases}$

Properties

- $\exists$ CFG$|L(G) = L \iff L$ is a context-free language
- L is regular $\implies$ L is context-free
- L_1, L_2 are context-free $\implies L_1 \cup L_2$ are context-free
- context-free languages are not closed under complement, and $L_1 \cap L_2, \sim L_1$ are not context-free)

We know that $\{a^nb^nc^n \mid n \geq 0\}$ is not CF

Pushdown Automata PDA

$$\text{PDA} = (Q, \Sigma, \Gamma, \delta, s, \perp, F) \quad \begin{aligned} &Q : \text{Finite set of state} \\ &\Sigma : \text{Finite input alphabet} \\ &\Gamma : \text{Finite stack alphabet} \\ &\delta :\subset (Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times \Gamma^*) \\ &s : \text{start state} \in Q \\ &\perp : \text{empty stack} \in \Gamma \\ &F : \text{final state} \in Q \end{aligned}$$

Properties
$\mathcal{L}(M) = L \iff L$ is context-free

Countability

- A set S is countable infinite if $\exists$ a monotonic function $f : S \to \mathbb{N}$ (isomorphism)
- A set S is uncountable if there is **NO** injection from S

Theorem:

- The set of all PDAs is countably infinite
- Σ^* is countably infinite (list out all string n in finite time)
- The set of all TMs is countably infinite ($\Sigma = \{0, 1\}$ | set of all TMs that $S \subseteq \Sigma^*$, so does REC, DEC, CF, REG
- The set of all languages is uncountable.

Proof for A_{TM} is undeciable:

Assume A_{TM} is decidable

$\exists$ a decider for A_{TM}, D.

Let P another TM such that $P(M)$: Call D on $M\#M$

Paradox machine: P never loops: $P(M) = \begin{cases} \text{accept} & \text{if P reject M} \\ \text{reject} & \text{if P accepts M} \end{cases}$

> Note that all regular language are deciable language

Diagonalization and Problems

</>

The set of unrecognizable languages is uncountable. The set of all languages is uncountable.

Proof: I can encode a language with a infinite string. $\Sigma = \{0, 1\}$ Consider a machine N that on input $x \in \{0, 1\}^*$ such that $L^*(i)$ is undeciable from the diagonalization.

Theorem

- L is deciable $\iff$ L and $\sim L$ are both recognizable

Proof: L is deciable $\iff \sim L$ is deciable. L is deciable $\implies$ L is recognizable

Let $R_L, R_{\sim L}$ be recognizer. Create TM M that runs R_L and $R_{\sim L}$ on x concurrently. if R_L accepts $\implies$ accept, $R_{\sim L}$ accepts $\implies$ reject.

If M never halts, M decides L. If $x \in L \implies R_L(x)$ halts, and $x \notin L \implies R_{\sim L}(x)$ halts.

Decidability and Recognizability

(1) is deciable: Create a TM M' such that $M'(M\#w)$ runs M on w, therefore M' is total, or $\mathcal{L}(M) = A_{\text{DFA}}$
$M\#w \in \mathcal{L}(M') \iff M$ accepts $w \iff M\#w \in A_{\text{DFA}}$

(2) is recognizable: Create a TM M' such that $M'(M\#w)$ runs M on w
$M\#w \in \mathcal{L}(M') \iff M$ accepts $w \iff M\#w \in A_{\text{TM}} \implies \mathcal{L}(M') = A_{\text{TM}}$

Reduction on universal TMs

$\sim A_{\text{TM}} = \{M\#w \mid M \text{ does not accept w}\}$. Which implies $\sim A_{\text{TM}}$ is unrecognizable

HP is undeciable, and recognizable.

$$\text{Halting problem} = \{M\#w \mid M \text{ halts on w}\}$$

Proof: Assume HP is deciable. $\exists D_{MP}(M\#w) = \begin{cases} \text{accept} & \text{if M halts on w} \\ \text{reject} & \text{if M loops on w} \end{cases}$

Build a TM M' where $M'(M\#v)$:

```
calls $D_{MP}$ on $M\#v$:
accepts:
  - run $M$ on $v$
    - accept → accept
    - reject → reject
reject: reject
```

Therefore M' is total. Since $M\#w \in \mathcal{L}(M') \iff$ M accepts w $\iff M\#w \in A_{\text{TM}}$. Therefore $\mathcal{L}(M') = A_{\text{TM}}$. Which means M' is a decider for A_{TM} (which is a paradox) □

$$\neg(\exists x \mid R : P(x)) \equiv \forall x \mid R : \neg P(x)$$
$$\neg(\forall x \mid R : P(x)) \equiv \exists x \mid R : \neg P(x)$$

regular language

$$\hat{\delta}(q, \epsilon) = q$$
$$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

All finite languages are regular, but not all regular languages are finite

Properties

- A TM is "total" iff it halts on all input
- $\mathcal{L}(M) = L \iff (\forall s \mid s \in L \iff M \text{ accepts s})$
- L is recognizable: $\iff \exists$ TM M s.t $\mathcal{L}(M) = L$
- L is decidable: $\iff \exists$ total TM M s.t $\mathcal{L}(M) = L \land \forall s \in \Sigma^*$ M halts on s
- L is decidable $\implies$ L is recognizable

Turing machine

$$\text{TM} = (Q, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}}, \square)$$

Q : Finite set of state
Σ : Finite input alphabet
Γ : Finite tape alphabet
$\delta : (Q \times \Gamma) \to Q \times \Gamma \times \{L, R\}$
s : start state $\in Q$
q_{accept} : accept state $\in Q$
q_{reject} : reject state $\in Q$
$\square$: blank symbol $\in \Gamma$

Transition function: $\delta(q, x) = (p, y, D)$: when in state p scan symbol a, write b on tape cell, move the head in direction d and enter state q