Completeness of Resolution and Recap of computability

$$\frac{C \cup \{p\} \qquad D \cup \{\neg p\}}{C \cup D} \quad \text{Resolution}$$

Soundness Theorem If a set of clauses Γ has a resolution refutation then Γ is unsatisfiable.

Proof Suppose $C_1, C_2 \ldots C_m$ is a refutation of Γ. Consider

$$\Delta_0 = \Gamma$$

$$\Delta_i = \Delta_{i-1} \cup \{C_i\}$$

$$\Delta_m = \Gamma \cup \{C_1, C_2 \ldots C_m\}$$

$$C_m = \{\} \Rightarrow \Delta_m \text{ is unsatisfiable}$$

Resolution Lemma If C is the resolvent of two clauses in Γ then if $\Gamma \cup \{C\}$ is unsatisfiable then Γ is unsatisfiable.

Proof Assume $v \models \Gamma$ and C is the resolvent $D \cup \{p\}$ and $E \cup \{\neg p\}$.

$v \models D \cup \{p\} \qquad v \models E \cup \{\neg p\}$

WLOG $v(p) = T$

$\exists l \in E$ s.t $v[\![l]\!] = T$

Therefore $v \models C$ (since $l \in C$)

Completeness Theorem If Γ is an unsatisfiable set of clauses then Γ has a refutation.

Proof [Davis - Putnam]

Assume Γ is finite set and is unsatisfiable

Γ is non-empty

By induction on the number of propositions that appear in Γ.

<u>Base Case</u>: The set of propositions appearing in Γ is empty.

$$\{\} \in \Gamma$$

Refutation for Γ is $\{\}$

<u>Induction Step</u>: Let p be a proposition that appears in Γ.

If a clause C contains both p and $\neg p$ then C is satisfied. WLOG assume Γ does not contain such clauses.

We partition Γ into

$$\Gamma_0^p = \{C \in \Gamma \mid \{p, \neg p\} \cap C = \emptyset\}$$

$$\Gamma_+^p = \{C \in \Gamma \mid p \in C\}$$

$$\Gamma_-^p = \{C \in \Gamma \mid \neg p \in C\}$$

$$\Gamma_p = \Gamma_0^p \cup \{C \cup D \mid C \cup \{p\} \in \Gamma_+^p, D \cup \{\neg p\} \in \Gamma_-^p\}$$

Refutation Γ is going to be do all resolutions and refute Γ_p.

Lemma Γ_p is unsatisfiable.

Proof Assume for contradiction Γ_p is satisfiable.

Let $v \models \Gamma_p$.

v' be the valuation that agrees with v on all propositions except p.

$$v' \models \Gamma_p$$

WLOG $v(p) = T$ and $v'(p) = F$.

$$\Gamma_p = \Gamma_0^p \cup \{ C \cup D \mid C \cup \{p\} \in \Gamma_+^p,\ D \cup \{\neg p\} \in \Gamma_-^p \}$$

$$v \models \Gamma_0^p \qquad v' \models \Gamma_0^p$$

$$v \models \Gamma_+^p \qquad v' \models \Gamma_-^p$$

If $v \models \Gamma_-^p$ or $v' \models \Gamma_+^p$ then Γ is satisfiable

Assume $v \not\models \Gamma_-^p$ and $v' \not\models \Gamma_+^p$.

$\exists$ $C \cup \{p\} \in \Gamma_+^p$, $D \cup \{\neg p\} \in \Gamma_-^p$

s.t. $v \not\models D \cup \{\neg p\}$ $\quad v' \not\models C \cup \{p\}$.

$v, v' \not\models D$ and $v, v' \not\models C$

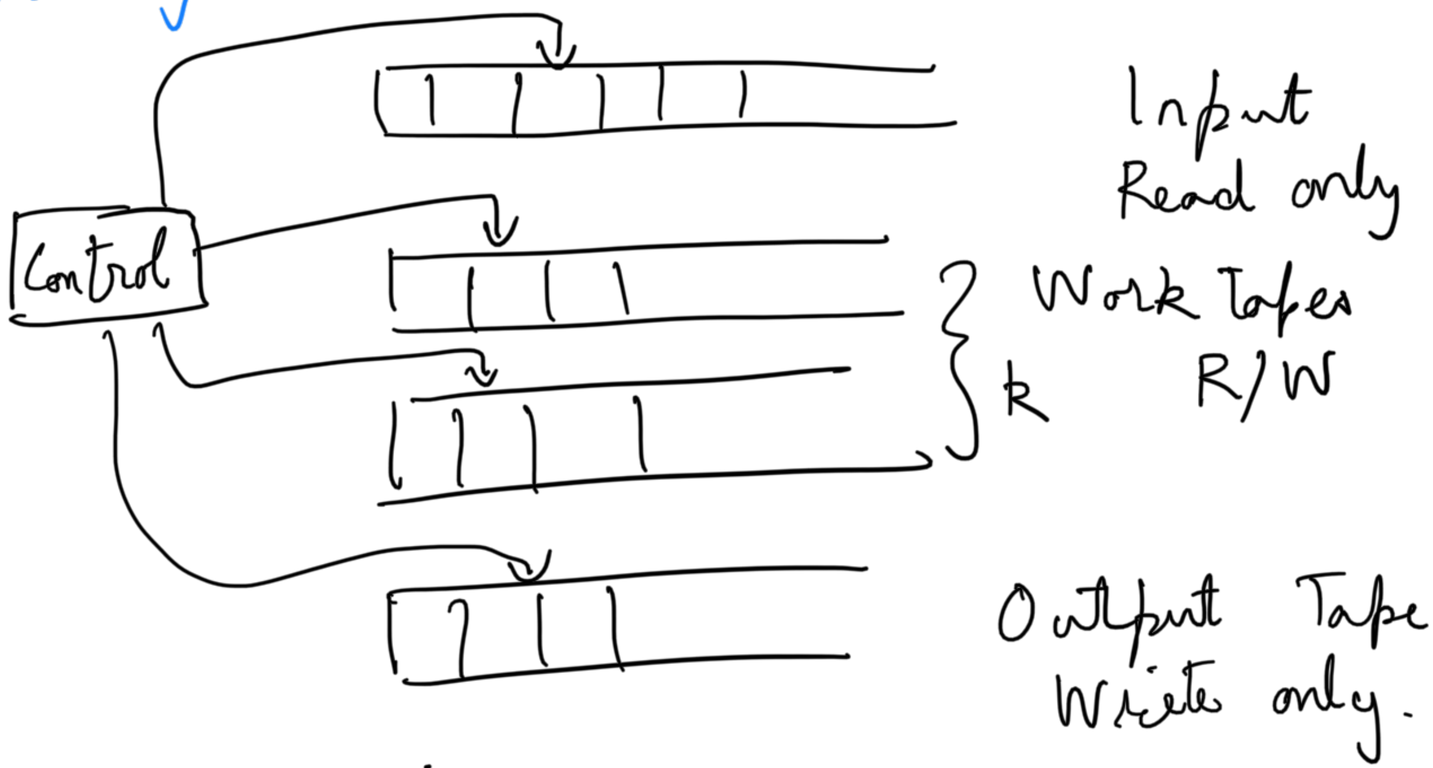
$$v, v' \not\models C \cup D \in \Gamma_p$$

Contradicts $v, v' \models \Gamma_p$.

If Γ is an infinite set of unsatisfiable clauses then compactness Theorem says that $\exists$ a finite subset $\Delta \subseteq \Gamma$ which is unsatisfiable.

Computability

Turing Machines



At the begining
- Input Tape contains input
- All other tapes are blank.
- State is initial state q_0

At any time, turing machine reads the input tape and each work-tape. Based on its current state
- Change its state
- Write symbols on each work-tape
- Move input / work tape heads either left or right.
- It may choose to write a symbol on the output tape.

Given an input, the TM can do one

of 3 things

- Run forever
- Halts but it does not accept
- Halts and accepts.

TM does not accept input

Language $L(M) = \{ w \mid M \text{ accepts } w \}$

M recognizes A iff $A = L(M)$.

Church-Turing Thesis Any mechanical procedure can be implemented on Turing Machine.

- For TM M there is sing(M) that has only one work-tape s.t

$$L(M) = L(\text{sing}(M))$$

- Nondeterministic: On any input the machine may have more than one computation.

  A NTM N accept input x if N accepts x on <u>some</u> computation.

- Fore any NTM N there is a deterministic det(N) s.t $L(N) = L(\text{det}(N))$

Recursively Enumerable A language A is r.e. if $\exists$ TM M s.t. $L(M) = A$.

Recursive A language A is recursive if $\exists$ TM M that halts on all inputs and $L(M) = A$.

Proposition If A is recursive then it is also r.e.

Proposition If A is recursive then $\bar{A}$ is also recursive.

Proof If A is recursive then $\exists$ M that halts on all inputs and $L(M) = A$.

Consider $\bar{M}$: Runs M and flips M's answer.

$\bar{M}$ halts on all inputs and $L(\bar{M}) = \bar{A}$.

Theorem A is recursive if and only if A is r.e. and $\bar{A}$ is r.e.

Proof A recursive $\Rightarrow$ A r.e

A recursive $\Rightarrow$ $\bar{A}$ recursive $\Rightarrow$ $\bar{A}$ is r.e.

($\Leftarrow$) A is recognized M_1 and $\bar{A}$ is recognized by M_2.

Algorithm for A:

On input x (dovetailing)

Run M_1 on x } parallel

Run M_2 on x

Run M_2 on x ...

If M_1 accepts then answer Yes