Part IB Logic and Proof Notes

1 Propositional Logic

- The logical connectives are

$\neg$	not
$\wedge$	and
$\vee$	or
$\implies$	implies
$\iff$	if and only if

1.1 Definitions

- **Interpretation/ truth assignment** - A function from the set of propositional symbols to $\{0, 1\}$
- An interpretation **satisfies** a formula if it makes the formula evaluate to true
- A set of formula are **valid** (or tautological) if every interpretation results in it being true
- A formula is **unsatisfiable** if it is not satisfiable
- S entails A if every satisfying interpretation of S also satisfies A, $S \models A$
- A is equivalent to B, $A \simeq B$ iff $A \models B \wedge B \models A$
- $S \models A$ iff $\{\neg A\} \cup S$ is inconsistent
- If S is inconsistent then $S \models A$ for any A
- $\models A$ iff A is valid

1.2 Normal Forms

- A literal is an atomic or its negation
- Negation normal form (NNF)
  - Formula only consists of $\wedge$, $\vee$, and $\neg$ in literals
- Conjunctive normal form (CNF)
  - $A_1 \wedge A_2 \wedge ... \wedge A_m$ where A_i is a disjunction of literals
- Disjunctive normal form (DNF)
  - $A_1 \vee A_2 \vee ... \vee A_n$ where A_i is a conjunction of literals
- Translating into normal form
  1. Eliminate $\implies$ and $\iff$
  2. Push $\neg$ in until only applies to literals
  3. Push in either $\wedge$ or $\vee$ depending on if CNF or DNF required

2 Proof Systems

- A proof system is **sound** if every theorem it generates is tautological
  - To prove a proof system sound
    1. Prove all axioms are tautological
    2. Prove every inference rule when applied to tautologies produces a tautology
- A proof system is **complete** if it can generate every tautology

2.1 Hilbert Style

- Define all connectives in terms of implication

$$\begin{aligned} \neg A &\stackrel{\text{def}}{=} A \implies f \\ A \vee B &\stackrel{\text{def}}{=} \neg A \implies B \\ A \wedge B &\stackrel{\text{def}}{=} \neg(\vee \neg B) \end{aligned}$$

- Note that $A \implies (B \implies A)$ is a tautology, as it

$$(A \implies (B \implies C) \implies ((A \implies B) \implies (A \implies C))$$

- Has only one inference rule, modus ponens

$$\frac{A \implies B \quad A}{B}$$

2.2 The Sequent Calculus

- A sequent is

$$\Gamma \implies \Delta$$

  where Γ and Δ are finite set of formula
- Sequent is true if

$$\bigwedge_{A \in \Gamma} A \implies \bigvee_{B \in \Delta} B$$

- A sequent is valid if it is true for all interpretations
- The basic sequent is always true

$$A, \Gamma \implies A, \Delta$$

- The sequent rules of propositional logic

Negation rules

$$\frac{\Gamma \implies \Delta, A}{\neg A, \Gamma \implies \Delta} \ (\neg l) \qquad \frac{A, \Gamma \implies \Delta}{\Gamma \implies \Delta, \neg A} \ (\neg r)$$

Conjunction rules

$$\frac{A, B\Gamma \implies \Delta}{A \wedge B, \Gamma \implies \Delta} (\wedge l) \qquad \frac{\Gamma \implies \Delta, A \quad \Gamma \implies \Delta, B}{\Gamma \implies \Delta, A \wedge B} (\wedge r)$$

Disjunction rules

$$\frac{A, \Gamma \implies \Delta \quad B, \Gamma \implies \Delta}{A \vee B, \Gamma \implies \Delta} (\vee l) \qquad \frac{\Gamma \implies \Delta, A, B}{\Gamma \implies \Delta, A \wedge B} (\vee r)$$

Implication rules

$$\frac{\Gamma \implies \Delta, A \quad B, \Gamma \implies \Delta}{A \to B, \Gamma \implies \Delta} (\to l) \qquad \frac{A, \Gamma \implies \Delta, B}{\Gamma \implies \Delta, A \to B} (\to r)$$

- Structural rules

Weaken rules

$$\frac{\Gamma \implies \Delta}{A, \Gamma \implies \Delta} (\text{weaken } l) \qquad \frac{\Gamma \implies \Delta, A, B}{\Gamma \implies \Delta, A} (\text{weaken } r)$$

Contract rules

$$\frac{A, A, \Gamma \implies \Delta}{A, \Gamma \implies \Delta} (\text{contract } l) \qquad \frac{\Gamma \implies \Delta, A, A}{\Gamma \implies \Delta, A} (\text{contract } r)$$

Cut

$$\frac{\Gamma \implies \Delta, A \quad A, \Gamma \implies \Delta}{\Gamma \implies \Delta} (\text{cut})$$

2.3 DPLL

1. Delete tautological clauses $\{P, \neg P, ...\}$
2. Unit propagation for every unit clause $\{L\}$
   - Delete all clauses containing L
   - Delete $\neg L$ from all clauses
3. Delete all clauses containing pure literals, a literal is pure if there is no clause containing $\neg L$
4. If the empty clause is generated then we have a refutation, conversely if all are deleted the set is satisfiable
5. If neither end conditions is met perform a case split,
   - Recursively apply the algorithm to L and $\neg L$
   - Set satisfiable if and only if one case is satisfiable

2.4 Resolution

- Method
  1. Translate into CNF
  2. Break into clauses
  3. Repeatedly apply the resolution rule
  4. If a contradiction is reached we ahve refuted $\neg A$
- Remember $\{Q, Q\}$ can become $\{Q\}$
- Clauses can be used repeatedly or not at all
- Assumptions can be incorporated by just adding them to the set of clauses (with out negating them)

3 First Order Logic

- Adds $\forall$ and $\exists$ and functions
- $\forall x A \wedge B$ is equivalent to $(\forall x A) \wedge B$
- A dot is used to weaken the binding

$$\forall x.A \wedge B \text{ is equivalent to } \forall x(A \wedge B)$$

- Free variables are effectively universally quantified
- A variable x is bound if it occurs in a subformula, A, of the form $\forall x.A$ or $\exists x.A$
- $\forall x$ and $\exists x$ are binding occurrences
- The renaming of bound variables is called α-conversion
- $A[t/x]$ - A with t for x replace all free occurrences of x with t rename all clashing names
- New sequent rules

Implication quantifier rules

$$\frac{A[t/x], \Gamma \implies \Delta \quad \forall x A, \Gamma \implies \Delta}{\forall x A, \Gamma \implies \Delta} \ (\forall l) \qquad \frac{\Gamma \implies \Delta, A}{\Gamma \implies \Delta, \forall x A} \ (\forall r)$$

- $(\forall l)$ t is any value
- $(\forall r)$ holds if x is not free in Γ or Δ

Existential quantifier rules

$$\frac{A, \Gamma \implies \Delta \quad \exists x A, \Gamma \implies \Delta}{\forall x A, \Gamma \implies \Delta} \ (\exists l) \qquad \frac{\Gamma \implies \Delta, A[t/x]}{\Gamma \implies \Delta, \exists x A} \ (\exists r)$$

- $(\exists l)$ holds if x is not free in Γ or Δ

- Apply $\exists l$ and $\forall r$ first

3.1 Formal Definition

- For a FOL language $\mathcal{L}$ ab interpretation $\mathcal{I}$ of $\mathcal{L}$ is a pair (D, I), where
  - D is the domain or universe
  - I maps symbols to individuals functions or sets
- Mapping of I

$$\begin{aligned} &\text{if } c \text{ is a constant then } I[c] \in D \\ &|f \text{ is an n-place function then } I[f] \in D^n \to D \\ &|P \text{ is an n-place relation symbol then } I[P] \in D^n \to \{10, 1\} \end{aligned}$$

- A valuation V of $\mathcal{L}$ over D is a function from variables in $\mathcal{L}$ into D (an assignment of the variables)
- Write $\mathcal{I}_V[t]$ if the value of $t \in \mathcal{L}$ with respect to I and V

$$\begin{aligned} \mathcal{I}_V[x] &\stackrel{\text{def}}{=} V(x) \qquad \text{(if } x \text{ is a variable)} \\ \mathcal{I}_V[c] &\stackrel{\text{def}}{=} I[c] \\ \mathcal{I}_V[f(t_1, ..., t_n] &\stackrel{\text{def}}{=} I[f](\mathcal{I}_V[t_1], ..., \mathcal{I}_V[t_n]) \end{aligned}$$

- Write $V\{a/x\}$ for the valuation that maps x to a

3.2 Skolemization

- An effort to remove existential and universal quantifiers
- Changes the meaning of a formula, but preserves inconsistency
- Easier if you push in quantifiers as far as possible
- Method
  1. Start at the outside and replace all existential quantifiers with functions taking all outer universal quantifiers as arguments, eg.

     $$\forall xy \exists z \forall i P(x, y, z, i) \text{ becomes } \forall xyi P(x, y, f(x, y), i)$$

     and

     $$\exists y \forall i . P(y, i) \text{ becomes } \forall i . P(a, i)$$

     ($f(x, y)$ is a skolem function and a is a skolem constant)
  2. Remove all universal quantifiers

3.3 Herbrand Interpretations

- Let C be the set of all constants in S, if there are none then let $C = \{a\}$
- Let $\mathcal{F}_n$ for $n > 0$ ve the set of all n-place functions symbols in S
- Let P_n for $n > 0$ be the set of all n-place predicate symbols in S
- The Herbrand universe H is the set of all terms that can be written with ground terms and function symbols

$$H = \bigcup_{i \geq 0} H_i$$

where H_i is defined as follows

$$\begin{aligned} H_0 &= C \\ H_{i+1} &= H_i \cup \{f(t_1, ..., t_n | t_1, ..., t_n \in H_i \text{ and } f \in \mathcal{F}_n\} \end{aligned}$$

- Every Herbrand interpretation, I_H, defines each n-place predicate symbol P to denote
$$I_H[P] \in H^n \to \{1, 0\}$$
we take
$$I_H[P(t_1, ..., t_n] = 1$$
if and only if $P(t_1, ..., t_n)$ holds in our desired interpretation of clauses
- Any interpretation $\mathcal{I} = (D, I)$ over some D can be mimicked by a Herbrand interpretation
- If an interpretation satisfies S then a Herbrand interpretation satisfies S
- A set S of clauses is unsatisfiable if and only if no Herbrand interpretation satisfies S
- The Skolem-Göbel-Herbrand Theorem states that unsatisfiablity can always be detected by a finite process

3.4 Unification

- A substitution $[t_1/x_1, ..., t_k/x_k]$ maps x_i to t_i
- Substitutions happen simultaneously
- A substitution ϕ is a unifier of terms t_1 and t_2 if
$$t_1\phi = t_2\phi$$
- A substitution θ is more general than ϕ if
$$\phi = \theta \circ \sigma \text{ for some } \sigma$$
- Unification works on trees, there are three kinds of node
  1. Variables - $x, y, ...$
  2. Constants - $a, b, ...$ also function symbols
  3. Pairs - (t, u) where t and u are terms
- When unifying a variable with a term, t, one must perform an occurs check
  - If x does not occur in t then
  $$x[t/x] = t = t[t/x]$$
  - If x does occur in t then no unifier exists
- Unifying two pairs (t_1, t_2) and (u_1, u_2) recurse on each side of the pair, to get the substitutions θ_1 and θ_2, $\theta_1 \circ \theta_2$ will then unify the pair

3.5 First-order resolution

- Each variable is scoped in its clause so can be renamed in it's clause
- Resolution rule
$$\frac{\{B, A_1, .., A_m\} \quad \{\neg D, C_1, .., C_n\}}{\{A_1, ..., A_m, C_1, ..., C_n\}\sigma} \text{ if } B\sigma = D\sigma$$
- Factoring
$$\frac{\{B_1, ..., B_k, A_1, .., A_m\}}{\{B_1, A_1, ..., A_m\}\sigma} \text{ if } B_1\sigma = ... = B_k\sigma$$
- Subsumption, for clauses C_1 and C_2
$$\frac{C_1 \quad C_2}{C_1\sigma} \text{ if } C_1\sigma \subseteq C_2\sigma$$

3.6 Prolog

- A Horn clause has at most one positive literal

$$\{\neg A_1, ..., \neg A_m, B\} \simeq (A_1, ..., A_m) \implies B$$

- If there are no negative literals then it is a fact
- If there are no positive literals then it is a goal
- Prolog repeatedly resolves the goal clause with some definite clause to produce a new goal clause
- If resolution produces an empty goal then execution succeeds
- Clauses are tried in a fixed order
- Prolog uses depth first search when preforming resolution

4 Decision Procedures and SMT Solvers

4.1 Fourier-Motzkin Variable Elimination

- In general deals with conjunctions of linear constraints over the reals adn rationals

$$\bigwedge_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

- Method
  1. Eliminate xs one by one, for every clause get x_n on one side to give

$$x_n \leq \cdots$$

     or

$$-x_n \leq \cdots$$

  2. For every oposite signed pair find the sum

$$\begin{aligned} x_n + -x_n &\leq \cdots + \cdots \\ 0 &\leq \cdots \end{aligned}$$

  3. Replace every clause containing x_n with the new clauses
  4. Repeat until one finds contradiction or a simple constraint

4.2 Satisfiablity Modulo Theory Solvers

- DPLL handles the logic part of the problem
- Reasoning about particular theories delegated to the relevant decision procedures

5 Binary Decision Diagrams (BDD)

- A binary decision tree represents the truth table for an expression
  - This may have redundancy
- A BDDs removes any redundancy
- A BDDs must satisfy the following conditions
  - Ordering - if P is tested before Q then $P < Q$
  - Uniqueness - Identical subgraphs are stored only once
  - Irredundancy - No test leads to identical subgraphs in the 1 and 0 cases
- The BDD representation of a formula is unique therefore binary decision diagrams are a canonical form
- ${}_X P_Y$ is a decision node that tests P, if true then X, else Y
- Creating a BDD

  $$Z \ op \ Z' \text{ where } Z \text{ and } Z' \text{ are already BDDs}$$

  - if $P = P'$ then recursively convert $if(P, X \ op \ X', Y \ op \ Y')$
  - if $P < P'$ then recursively convert $if(P, X \ op \ Z', Y \ op \ Z')$
  - if $P > P'$ then recursively convert $if(P', Z \ op \ X', Z \ op \ Y')$

6 Modal Logic

- Based on two parameters
  - W - The set of possible worlds
  - R - The accessibility relation between worlds
- The pair (W, R) is the modal frame
- The modal operators or modalities are $\Box$ and $\Diamond$
  - $\Box A$ means A is necessarily true
  - $\Diamond A$ means A is possibly true
- Necessarily true means true in all worlds accessible from the present one
- The two modalities are related by

  $$\neg\Diamond A \simeq \Box\neg A$$

  "It is not possible A is true" $\equiv$ "A is necessarily false"
- Complex modalities are made up of strings of modal operators
  - In S4 $\Box\Box A$ is equivalent to $\Box A$
- Under a modal frame (W, R) an interpretation, I, maps propositional letters, P, to those worlds in which P is true

- $w \Vdash A$ means A is true in the world w, it is defined as follows

$$\begin{aligned} w \Vdash A &\iff w \in I(P) \\ w \Vdash \Box A &\iff v \Vdash A \text{ for all } v \text{ such that } (w,v) \in R \\ w \Vdash \Diamond A &\iff v \Vdash A \text{ for some } v \text{ such that } (w,v) \in R \\ w \Vdash A \vee B &\iff w \Vdash A \text{ or } w \Vdash B \\ w \Vdash A \wedge B &\iff w \Vdash A \text{ and } w \Vdash B \\ w \Vdash \neg A &\iff w \Vdash A \text{ does not hold} \end{aligned}$$

- $\vDash_{W,R,I} A$ mean $w \Vdash A$ holds for all w under the interpretation I
- $\vDash_{R,I} A$ means $w \Vdash A$ for all w and all I
- A is universally valid, $\Vdash A$, if $\vDash_{W,R} A$ for all frames
- A proof system can be made to use pure modal logic (k-modal logic) by adding the following
  - The distribution axiom
$$\Box(A \implies B) \implies (\Box A \implies \Box B)$$
  - The necessitation rule (any tautology will hold in all worlds)
$$\frac{A}{\Box A}$$
  - Use the definition $\Diamond A \stackrel{\text{def}}{=} \neg\Box\neg A$

6.1 S4 Modal Logic

- Assume the accessibility relation is reflexive and transitive
- Add the following new sequent rules

$$\frac{A, \Gamma \implies \Delta}{\Box A, \Gamma \implies \Delta}\ (\Box l) \qquad \frac{\Gamma^* \implies \Delta^*, A}{\Gamma \implies \Delta, \Box A}\ (\Box r)$$

$$\frac{A, \Gamma^* \implies \Delta^*}{\Diamond A, \Gamma \implies \Delta}\ (\Diamond l) \qquad \frac{\Gamma \implies \Delta, A}{\Gamma \implies \Delta, \Diamond A}\ (\Diamond r)$$

$$\Gamma^* \stackrel{\text{def}}{=} \{\Box B | \Box B \in \Gamma\}$$
$$\Delta^* \stackrel{\text{def}}{=} \{\Diamond B | \Diamond B \in \Delta\}$$

- In S4 the dual of every operator string equivalence holds, eg.

$$\Box\Box A \simeq \Box A \text{ and } \Diamond\Diamond A \simeq \Diamond A$$

- In S4 only need to consider operator strings no longer than 3 and with alternating $\Box$ and $\Diamond$

7 The Tableaux Calculus

- Works with formula in negated normal form
- Removes all rules that move variables across the implication
- Define the following rules

$$\frac{}{\neg A, A, \Gamma \implies} \text{ (basic)} \qquad \frac{\neg A, \Gamma \implies \quad A, \Gamma \implies}{\Gamma \implies} \text{ (cut)}$$

$$\frac{A, B, \Gamma \implies}{A \wedge B, \Gamma \implies} (\wedge l) \qquad \frac{A, \Gamma \implies \quad B, \Gamma \implies}{A \vee B, \Gamma \implies} (\vee l)$$

$$\frac{A[t/x], \Gamma \implies}{\forall x A, \Gamma \implies} (\forall l) \qquad \frac{A, \Gamma \implies}{\exists x A, \Gamma \implies} (\exists l)$$

($\exists l$) holds provided x is not free in the conclusion

- Add the modal logic rules

$$\frac{A, \Gamma \implies}{\Box A, \Gamma \implies} (\Box l) \qquad \frac{A, \Gamma^* \implies}{\Diamond A, \Gamma \implies} (\Diamond l)$$

where $^* \stackrel{\text{def}}{=} \{\Box B | \Box B \in \Gamma\}$

7.1 Free-variable Tableaux Calculus

- Requires treatment of quantifiers
- Method
  - Push in universal quantifiers (makes formula easier to solve)
  - Skolemize the formula to remove existential quantifiers
  - Replace the universal quantifier rule with

$$\frac{A[z/x], \Gamma \implies}{\forall x A, \Gamma \implies} (\forall l)$$

    where z is a free variable
  - All unification in the basic sequent
- Maintain universally quantified formula as they may be used many times, eg.

$$\frac{P(z) \wedge \neg P(f(z)), \forall x.(P(x) \wedge \neg P(f(x))) \implies}{\forall x.(P(x) \wedge \neg P(f(x))) \implies}$$