CS202A Assignment 1 Solutions

Q2

(a) Following are the proof rules derived in a straightforward way from the rules of ND whose name appears on the right.

1. $\frac{\Gamma_1 \vdash \phi \quad \Gamma_2 \vdash \psi}{\Gamma_1 \cup \Gamma_2 \vdash \phi \wedge \psi} \wedge i$

2. $\frac{\Gamma \vdash \phi_1 \wedge \phi_2}{\Gamma \vdash \phi_1} \wedge e1$ $\qquad$ $\frac{\Gamma \vdash \phi_1 \wedge \phi_2}{\Gamma \vdash \phi_2} \wedge e2$

3. $\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi} \wedge i1$ $\qquad$ $\frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi} \wedge i2$

4. $\frac{\Gamma_1, \phi \vdash \chi \quad \Gamma_2, \psi \vdash \chi}{\Gamma_1 \cup \Gamma_2, \phi \vee \psi \vdash \chi} \vee e$

5. $\frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \rightarrow \psi} \rightarrow i$ $\qquad$ $\frac{\Gamma_1 \vdash \phi \quad \Gamma_2 \vdash \phi \rightarrow \psi}{\Gamma_1, \Gamma_2 \vdash \psi} \rightarrow e$

6. $\frac{\Gamma, \phi \vdash \bot}{\Gamma \vdash \neg \phi} \wedge \neg i$ $\qquad$ $\frac{\Gamma_1 \vdash \phi \quad \Gamma_2 \vdash \neg \phi}{\Gamma_1 \cup \Gamma_2 \vdash \bot} \neg e$

7. $\frac{\Gamma \vdash \bot}{\Gamma \vdash \phi} \bot e$ $\qquad$ $\frac{\Gamma \vdash \neg\neg \phi}{\Gamma \vdash \phi} \neg\neg e$

(b) No boxes are needed in proofs in the sequent system. Any assumption with which a box starts in ND is now added on the leftside of the sequent. Closing of the box corresponds to removing the assumption from the leftside using the corresponding rule in the sequent system.

(c) Given below is a proof of $p \vee \neg p$ in tree form in the system of part (a).

$p \vdash p \quad axiom$

$- - - - - - - - \quad \lor i1$

$p \vdash p \lor \neg p \qquad \neg(p \lor \neg p) \vdash \neg(p \lor \neg p) \quad axiom$

$- \neg e$

$p, \neg(p \lor \neg p) \vdash \bot$

$- - - - - - - - - - - - - - - - - \quad \neg i$

$\neg(p \lor \neg p) \vdash \neg p$

$- - - - - - - - - - - - - \lor i2$

$\neg(p \lor \neg p) \vdash p \lor \neg p \qquad \neg(p \lor \neg p) \vdash \neg(p \lor \neg p) \quad axiom$

$- \neg e$

$\neg(p \lor \neg p) \vdash \bot$

$- - - - - - - - - - - \quad \neg i$

$\vdash \neg\neg(p \lor \neg p)$

$- - - - - - - - - - - - \neg\neg e$

$\vdash p \lor \neg p$

(d) $p \vdash q \to p$ does not seem derivable from rules in part (a). Its proof in ND uses copy rule (see page 20 in the book). We have not translated copy rule of ND into any sequent rule. It can be translated as follows.

$\frac{\Gamma \vdash \phi}{\Gamma \cup \Gamma' \vdash \phi} \quad weakening$

This rule is called weakening as it introduces more assumptions in antecedent of the sequent to derive the same conclusion.

Using weakening, proof of $p \vdash q \to p$ can be given as follows.

$p \vdash p \quad axiom$

$- - - - - - - - \quad weakening$

$p, q \vdash p$

$- - - - - - - - \quad \to i$

$p \vdash q \to p$

Q3

(a) Given a set $p_1, \ldots, p_r$ of propositions,

formula $\phi \equiv (\bigvee_{i=1}^{r} p_i) \land \bigwedge_{1 \leq i < j \leq r} \neg(p_i \land p_j)$

asserts that exactly one of $p_1, \ldots, p_r$ is true.

[Note that in propositional logic only binary $\wedge, \vee$ are allowed. Notations $\bigvee_{i=1}^{r}$ and $\bigwedge_{1 \leq i < j \leq r}$ are convenient abbreviations for a formula explicitly listing all disjuncts and conjuncts. Size of ϕ is therefore $O(r^2)$. We continue to use such abbreviations below.]

- Let $\alpha \equiv \bigwedge_{v \in V}$[Exactly one of p_v^1, p_v^2, p_v^3 and p_v^4 is ture]
  (α asserts that every vertex has exactly one color)
- $\beta \equiv \bigwedge_{(u,v) \in E} \bigwedge_{i=1}^{4} \neg(p_u^i \wedge p_v^i)$.
  (β asserts that every edge has different color at its end points)

It is easy to see that $\alpha \wedge \beta$ is satisfiable iff G is 4 colorable.

Size of $\alpha \wedge \beta$ is $O(n^2)$.

(b) We define variable $p_{t,i}$ for each $t \in V$ and each i, $1 \leq i \leq n$.

Intuitively $p_{t,i}$ is true if v is the i^{th} vertex on path from u to v.

- Let $\alpha \equiv \bigwedge_{t \in V} \bigwedge_{1 \leq i < j \leq n} \neg(p_{t,i} \wedge p_{t,j})$
  [α says that for any $t \in V$, $p_{t,i}$ is ture for at most one i].
- $\beta \equiv \bigwedge_{t \in V, t \neq v} \bigwedge_{i=1}^{n} (p_{t,i} \rightarrow \bigvee_{(t,s) \in E} p_{s,i+1})$
  [β roughly says that if a path does not end at v then it can be extended].
- We let the desired formula be $\theta \equiv p_{u,1} \wedge \alpha \wedge \beta$.

Size of α is $O(n^3)$ and size of β is $O(n \cdot |E|)$, so the size of θ is $O(n^3)$.

Correctness proof:

- Let there be a simple path $v_1, v_2, \ldots, v_k$, where $v_1 = u$ and $v_k = v$. Set variables $\{p_{v_i,i} \mid i \in \{1, \ldots, k\}\}$ to true and all other variables to false. It is easy to see that this assignment satisfies θ.
- Conversely, consider a valuation which satisfies θ. For convenience let us use notation $u = v_1$. If $v \neq u$ then as $p_{u,1}$ is true, by β there is a vertex v_2 adjacent to v_1 such that $p_{v_2,2}$ is true.

  Assume inductively that variables $p_{v_1,1}, \ldots, p_{v_j,j}$ are true s.t. $v_1, \ldots, v_j$ is a simple path in G and v does not occur on this path. Then by β, there is a vertex v_{j+1} adjacent to v_j s.t. $p_{v_{j+1},j+1}$ is true.

If $v_{j+1} = v_i$ for some $i \in \{1, \ldots, j\}$ then we have both $p_{v_i,i}$ and $p_{v_i,j+1}$ true. This contradicts α, so v_{j+1} is a vertex which is not already on path $v_1, \ldots, v_j$. This shows the new path $v_1, \ldots, v_j, v_{j+1}$ to be a simple path.

So we see that there is a simple path starting at u s.t. either this path ends in v or it can be extended to another simple path. As a simple path can not be extended indefinitely (it has at most n vertices on it), it must eventually reach vertex v. This shows a path from u to v. $\square$

[Note: A simple path is one on which no vertex occurs more than once.]

Q4 Let S be the set of given clauses and let $\{q_1, \ldots q_t\}$ contain all atoms occurring in S.

We define new variables $\{p_1, \ldots p_t\}$ s.t. p_i is true iff q_i is false,

that is $p_i \leftrightarrow \neg q_i$.

Given clause $C_r \equiv q_{i_1} \rightarrow q_{j_1} \vee q_{j_2} \vee \ldots \vee q_{j_m} \in S$

we define clause D_r as $p_{j_1} \wedge p_{j_2} \wedge \ldots \wedge p_{j_m} \rightarrow p_{i_1}$

and let $T = \{D_r \mid C_r \in S\}$

Note that

$$
\begin{aligned}
& q_{i_1} \rightarrow q_{j_1} \vee q_{j_2} \vee \ldots \vee q_{j_m} \\
\Leftrightarrow\ & \neg q_{i_1} \vee q_{j_1} \vee q_{j_2} \vee \ldots \vee q_{j_m} \\
\Leftrightarrow\ & p_{i_1} \vee \neg p_{j_1} \vee \neg p_{j_2} \vee \ldots \vee \neg p_{j_m} \\
\Leftrightarrow\ & p_{i_1} \vee \neg (p_{j_1} \wedge p_{j_2} \wedge \ldots \wedge p_{j_m}) \\
\Leftrightarrow\ & p_{j_1} \wedge p_{j_2} \wedge \ldots \wedge p_{j_m} \rightarrow p_{i_1}
\end{aligned}
$$

Using the above equivalence, it is clear that S is satisfiable iff T is satisfiable. Satisfying assignment for S (T) is obtained from T (S) using equivalence $p_i \leftrightarrow \neg q_i$.

This efficiently reduces the problem of checking satisfiability of S to checking satisfiability of T. T is a set of Horn clauses for which we have seen an efficient algorithm to check satisfiability in class.

Q5 **(a)** Starting from any truth value assignment to some nodes, we can extend the assignment to other nodes as far as possible or discover a contradiction using DFS (depth first search) in linear time. Therefore the first solver works in linear time.

Now consider the second solver. Suppose at some stage second solver can't make progress by linear solver's strategy. We will upper bound the number of steps needed to get permanent mark for some unlabeled node by the second solver. For each unlabeled node m second solver considers at most two assignments (T and F). After guessing an assigment for m it uses linear solver to mark all implied node. This take $O(n)$ steps. The number of unlabeled nodes is bounded by n, so total number of steps needed to try labeling for each of them independently needs at most $O(n^2)$ steps. Therefore from the current stage in at most $O(n^2)$ steps an unlabeled node gets a permanent mark (or the algorithm halts for example, if a satisfying assignment is found in the process or no stable mark could be discovered etc.). As there are only n nodes to be marked number of steps needed to mark all of them with permanent marks is $O(n^3)$. Once every node has a permanent mark a local computation at every node determines if the assignment is consistent or contradictory.

So the algorithm halts in $O(n^3)$ steps.

(b) Consider $\neg(p_1 \wedge \neg q_1) \wedge \neg(p_2 \wedge \neg q_2)$. Initial labeling of its DAG gives two disjoint trees N_1 and N_2 representing $p_1 \wedge \neg q_1$ and $p_2 \wedge \neg q_2$ respectively with roots of both these trees labeled F (draw the dag and see it). To make progress cubic solver needs to guess labeling on an unlabeled node. Now if the guessed node is in N_i then this will not result in any unlabeled node in N_{3-i} getting labeled. So this will always result in an incomplete assignment. It is also easy to see that any such assignment will not result in a contradiction and no node gets a permanent mark. Hence cubic solver fails on this input.

(c) A Horn clause $p_1, \ldots, p_n \rightarrow q$ translates to $\neg((p_1 \wedge \ldots \wedge p_n) \wedge \neg q)$. Translation of a set of Horn clauses is a conjunction of such formulae. Initial labeling for a DAG corresponding to such set gives trees $N_1, \ldots N_k$ with their roots labeled F and N_i representing i^{th} clause say, $(p_1 \wedge \ldots \wedge p_n) \wedge \neg q$. In a partcular case, if $n = 0$ then N_i is a tree representing $\neg q$ with root labeled F this immediately assigns truth value T to q.

Also note that if all $p_1, \ldots, p_n$ get labeled T then in tree N_i representing $(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \wedge \neg q$, node for $(p_1 \wedge p_2 \wedge \ldots \wedge p_n)$ get labeled T using rule ti several times. By rule fll, $\neg q$ gets labeled F and by rule $\neg f$, q gets labeled T.

If Horn clause $p_1, \ldots, p_n \rightarrow$ is represented by tree N_i and if all $p_1, \ldots, p_n$ get labeled T then a contradiction appears at the roof of N_i as the root was already labeled F.

(Once again you may like to draw diagrams to see the above cases).

This shows that all steps of 'Marking algorithm' for Horn clauses can be simulated by linear solver. So if a given set of Horn clauses is unsatisfiable then linear SAT solver will discover a contradiction and will output 'unsatisfiable'.

However SAT solver need not be able to find a satisfying assignment when 'Marking algorithm' halts declaring the input 'satisfiable'. This can be seen from part (b), the formula exhibited there on which cubic solver fails, is conjuction of Horn clauses $p_1 \rightarrow q_1$ and $p_2 \rightarrow q_2$. So cubic SAT solver fails on any satisfiable set of Horn clauses which contains a pair of clauses at least as complex as $p_1 \rightarrow q_1$ and $p_2 \rightarrow q_2$. Note that all p_1, p_2, q_1, q_2 are distinct atoms.

Satisfiability of any single Horn clause or of the set $\{p_1 \rightarrow q, p_2 \rightarrow q\}$ or of the set $\{p_1, \ldots p_n \rightarrow q_1, p_1, \ldots p_n \rightarrow q_2\}$ can be detected by the cubic solver.

——————————x-x-x——————————