Homework 7

Li Yunzhi

October 18, 2017

1 Prove using resolution that the above sentence entails G

Proof 1 *First, we use resolution inference rules to prove that* $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) \vdash G$
$KBs : (A \vee B), (\neg A \vee C), (\neg B \vee D), (\neg C \vee G), (\neg D \vee G), \neg G$
Step 1: we choose clauses $(A \vee B)$ *and* $(\neg A \vee C)$*, after we use resolution inference rules, we get a new* $KBs : (B \vee C), (\neg B \vee D), (\neg C \vee G), (\neg D \vee G), \neg G$
Step 2: we choose clauses $(B \vee C)$ *and* $(\neg B \vee D)$*, after we use resolution inference rules, we get a new* $KBs : (C \vee D), (\neg C \vee G), (\neg D \vee G), \neg G$
Step 3 :we choose clauses $(C \vee D)$ *and* $(\neg C \vee G)$*, after we use resolution inference rules, we get a new* $KBs : (D \vee G), (\neg D \vee G), \neg G$
Step 4 :we choose clauses $(\neg D \vee G)$ *and* $(D \vee G)$*, after we use resolution inference rules, we get a new* $KBs : G, \neg G$
Step 5: we choose clauses G *and* $\neg G$*, and we get an empty clause, then it shows that* $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) \vdash G$
Finally, using the soundness of resolution we know that $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) \models G$

2 Number of semantically distinct 2-CNF clauses

Notice that we have n proposition symbols, so we have 2n different literals
(1) Every clauses have 2 places to put literals in, so if we ignore the ordering of them, there are $C_{2n}^2 = 2n^2 - n$ clauses which has two different literals
(2) Notice that $(A \vee \neg A) \equiv (B \vee \neg B)$, so we have to remove them from these clauses until one of them left. And now the number of clauses is $2n^2 - 2n + 1$
(3) Notice that $(A \vee A)$ is also a 2-CNF clause, we have to add 2n to the total number, then we reach the final number $2n^2 + 1$

3 Polynomial Time

Notice that every step we use resolution inference rules on two 2-CNF clauses, the result's size is at most 2, but the number of clauses has been reduced by 1. What's more, the number of clauses is $O(n^2)$, so it always terminates in time polynomial.

4 3-CNF

In section "Polynomial Time", we have proved that the set of 2-CNF clauses is closed which means that two 2-CNF clauses only generate a 2-CNF clause. And due to that , the size of the set of KBs is $O(n^2)$. However, when we try to use this method on 3-CNF clauses we will fail, because two 3-CNF clauses can generate a clause which length is 4, and then generate a clause which length is 5.... So, the length of clause in our KBs is O(n), which means the size of our KBs is $O(2^n)$, so we can use this proof on 3-CNF clauses.