Homework 7

Wang Huaqiang

December 5, 2018

1 a

Proof:

To prove:

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) \textit{ entails } G$$

We include $\neg G$ in this sentence:

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) \wedge \neg G$$

By using Resolution Rule:

$$(A \vee B) \wedge (\neg A \vee C) \textit{ generates } (B \vee C)$$
$$(B \vee C) \wedge (\neg B \vee D) \textit{ generates } (C \vee D)$$
$$(C \vee D) \wedge (\neg C \vee G) \textit{ generates } (D \vee G)$$
$$(D \vee G) \wedge (\neg D \vee G) \textit{ generates } (G)$$
$$(G) \wedge (\neg G) \textit{ generates } \emptyset$$

Hence:

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) \textit{ entails } G$$

Q.E.D.

2 b

For n proposition symbols:

Consider the clauses as following:

$(A \vee \neg A)$, $(B \vee \neg B)$, ... (n in total)

These clauses share the same semantic: Tautology (1)

For other clauses, such as $(A \vee \neg B)$ and $(B \vee \neg C)$, they are different in semantic in pairs.

Also, consider clauses such as $(A \vee A)$. There are n such clauses.

To avoid repeated count caused by commutative law, just use number of combination to count. As all symbols also have a negative form, the total number of symbols is 2n. Therefore, the result is:

$$\binom{2n}{2} - n + n + 1$$

i.e.

$$\binom{2n}{2} + 1$$

3 c

Combine 2-CNF clauses with the same semantic takes polynomial time. In each iteration in propositional resolution, 2 2-CNF clauses generate 1 new clause in polynomial time. As there are at most $\binom{2n}{2} + 1$ different clauses, iteration can last at most $\binom{2n}{2} + 1$ rounds, and each round needs polynomial time, the total time is at most:

$$polytime * polytime + polytime$$

Which is still polynomial time.

4 d

In the proof above-mentioned, "2 2-CNF clauses generate 1 new clause in polynomial time" is a critical condition. Yet for 3-CNFs, resolute 2 3-CNF will not necessarily generate 3-CNF or 2-CNF.
