Assignment 2

Nanyang Tang(z5103095)

**Question 1.**

(a).

The ASP program shows in Figure1.

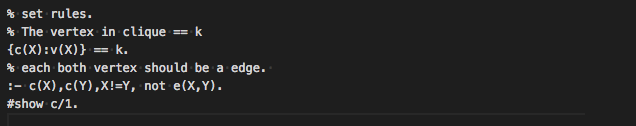
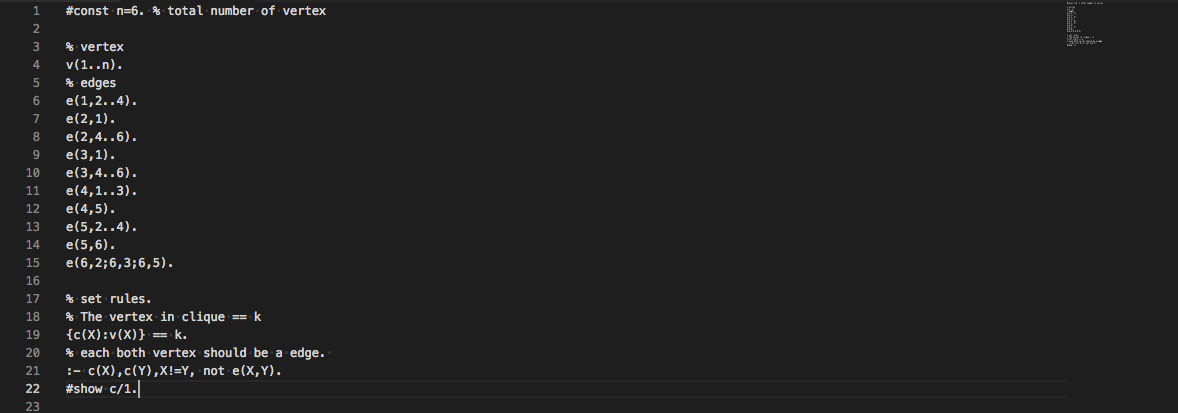


Figure 1

(b).

All programs:



When k=3.

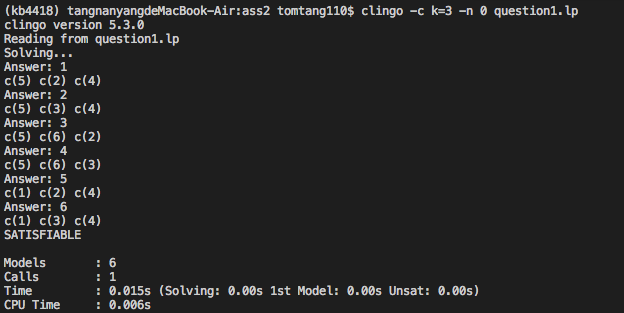


Figure 2

As can be seen from Figure2, there are 6 models satisfying including {c(5), c(2), c(4)}, {c(5), c(3), c(4)}, {c(5), c(6), c(2)}, {c(5), c(6), c(3)}, {c(1), c(2), c(4)}, {c(1), c(3), c(4)}.

When k=4.

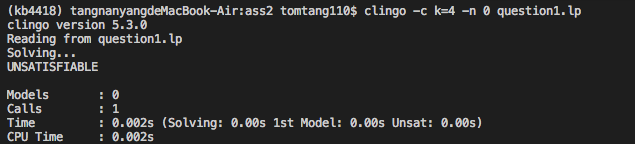


Figure 3

From Figure3, there is no model fulfilling .

When k=5.

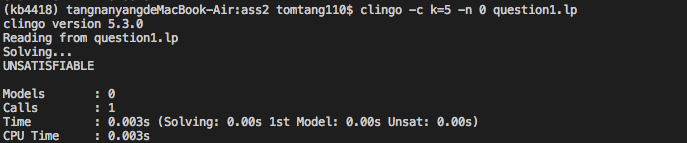


Figure 4

From Figure4, there is no model fulfilling .

When k=6.

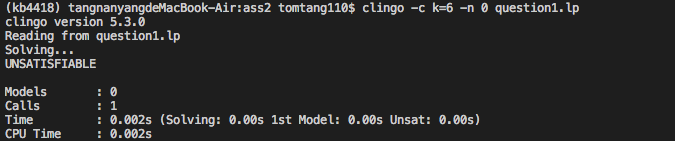


Figure 5

From Figure5, there is no model fulfilling .

**Question 2.**

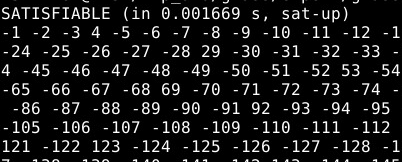
|  |  |  |
| --- | --- | --- |
| S | Reduct | Stable model |
| {a,b,c,d} |  | NO |
| {a,b,c} |  | NO |
| {a,b,d} |  | NO |
| {b,c,d} |  | NO |
| {a,b} |  | NO |
| {a,c} |  | NO |
| {a,d} |  | YES |
| {b,c} |  | NO |
| {b,d} |  | YES |
| {c,d} |  | YES |
| {a} | a. | NO |
| {b} |  | NO |
| {c} |  | NO |
| {d} |  | NO |
| {} |  | NO |

**Question 3.**

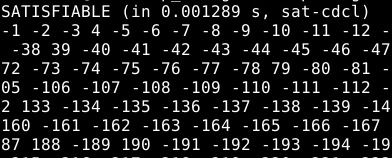
(a).

For the naïve solver, it takes more than 5 minutes to obtain answer.

For the unit propagation solver with the Watched-Literal Scheme, it speeds 0.001669s to obtain the solution.



For the clause learning solver, it speeds 0.001289s to obtain the solution.



(b).

For naïve solver, it traverses the complete binary tree. In other words, it tries all decisions, hence it takes the most time.

For unit propagation solver, it only considers unit propagation literals. This means some negated unit propagation literals are excluded.

For clause learning solver, it is added a conflict analysis procedure based on unit propagation algorithm, therefore, some conflicts would be avoided in next process. So, the time of clause learning solver is less than the time of unit propagation.

**Question 4.**

(a).

It is False.

This is because there exist transformations into CNF that avoid an exponential increase in size by preserving satisfiability rather than equivalence. For example, the above formula can be transformed into CNF by adding variables as follows:. An interpretation satisfies this formula only if at least one of the new variables is true. If this variable is , then both and are true as well. This means that every model that satisfies this formula also satisfies the original one.

(b).

It is True.

Since every problem in NP can be reduced to SAT and every problem in co-NP can be reduced to SAT, the union of these two sets contains only problems that can be solved in polynomial time with the help of one step decision box which provides both the affirmative and the negative response to SAT. This is, by definition, a SAT oracle. On the other hand, if a decision problem is not in NP ∪ co-NP, it cannot be reduced in polynomial time to SAT or SATc.

(c).

It is True.

When I is closed under unit propagation, it means all literals in I have been identified as True or False. Therefore, in order to close , we only need to inspect clause that watch . This is because clauses in I have been identified but clauses in {x} are still unknown.

**Question 5.**

|  |  |  |  |
| --- | --- | --- | --- |
| I | Clauses and Watched Literals | | |
|  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Question 6.**

(a).

Suppose K Happy K Happy is unsatisfiable.

We need to show that for all e, w does not satisfy K Happy K Happy.

e, w does not satisfy K Happy K Happy.

iff for all , e,does not satisfy Happy Happy,

iff for all , e,does not satisfy Happy and Happy,

does not satisfy K Happy K Happy

iff for all , (Happy Happy).

Therefore, it is unsatisfiable.

(b).

K (HappySad) K Happy

K (HappySad) K Happy

Suppose Happy is valid, so does not satisfy Happy.

Happy is valid iff for all , e,does not satisfy or (Happy). In all interpretations, , and satisfies or (Happy). However, does not satisfy (Happy).

Therefore, it is not valid since not all interpretations satisfy K (HappySad) K Happy, so it is satisfiable.

**Question 7.**

(a).

Positive effect:

🞎

Negative effect:

🞎

SSA:

🞎

(b).

Let

iff

iff

iff

Obviously, is True. So

.

iff

iff

iff

This is valid.

(c).

Clear(x) is redundant.