CS360 Homework 10

SAT-based Planning

1) In class, we have seen how to formulate a planning problem as a SAT (= satisfiablity) problem. The formulation allows the SAT solver to find plans that can execute actions in parallel, as long as all ways of sequentializing these parallel actions result in valid plans, that is, none of the actions deletes a precondition of another one, and none of the actions add a proposition that another action deletes. How would you modify the formulation if we want to find plans that execute actions in parallel as long as there exists at least one way of sequentializing theses parallel actions that results in a valid plan?

2) You want to find an operator sequence of length at most 2 that solves the following planning problem:

Start state:

```
In(SF),
Connected(LA,SF), Connected(SF,LA), Connected(LA,SD), Connected(SD,LA)
```

Goal state:

```
In(LA)
```

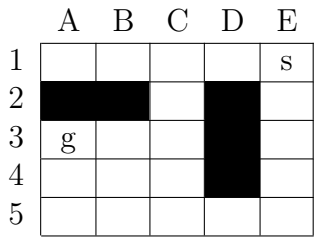
Operator:

```
Go(x,y):
Preconditions = {In(x), Connected(x,y)}
Effects = {-In(x), In(y)}
```

Formulate the corresponding propositional satisfiability problem by following the approach given in the lectures.

Breadth-First Search

3) A 4-neighbor gridworld is given below. In which order does breadth-first search (with a sensible node pruning strategy) expand the cells when searching from s to g? Ties are broken in lexicographic order. That is, A1 is preferred over A2 and B1, and A2 is preferred over B1.


A B C D E
1
2
3
4
5
s
g