CS 579 – Planning
Homework 3

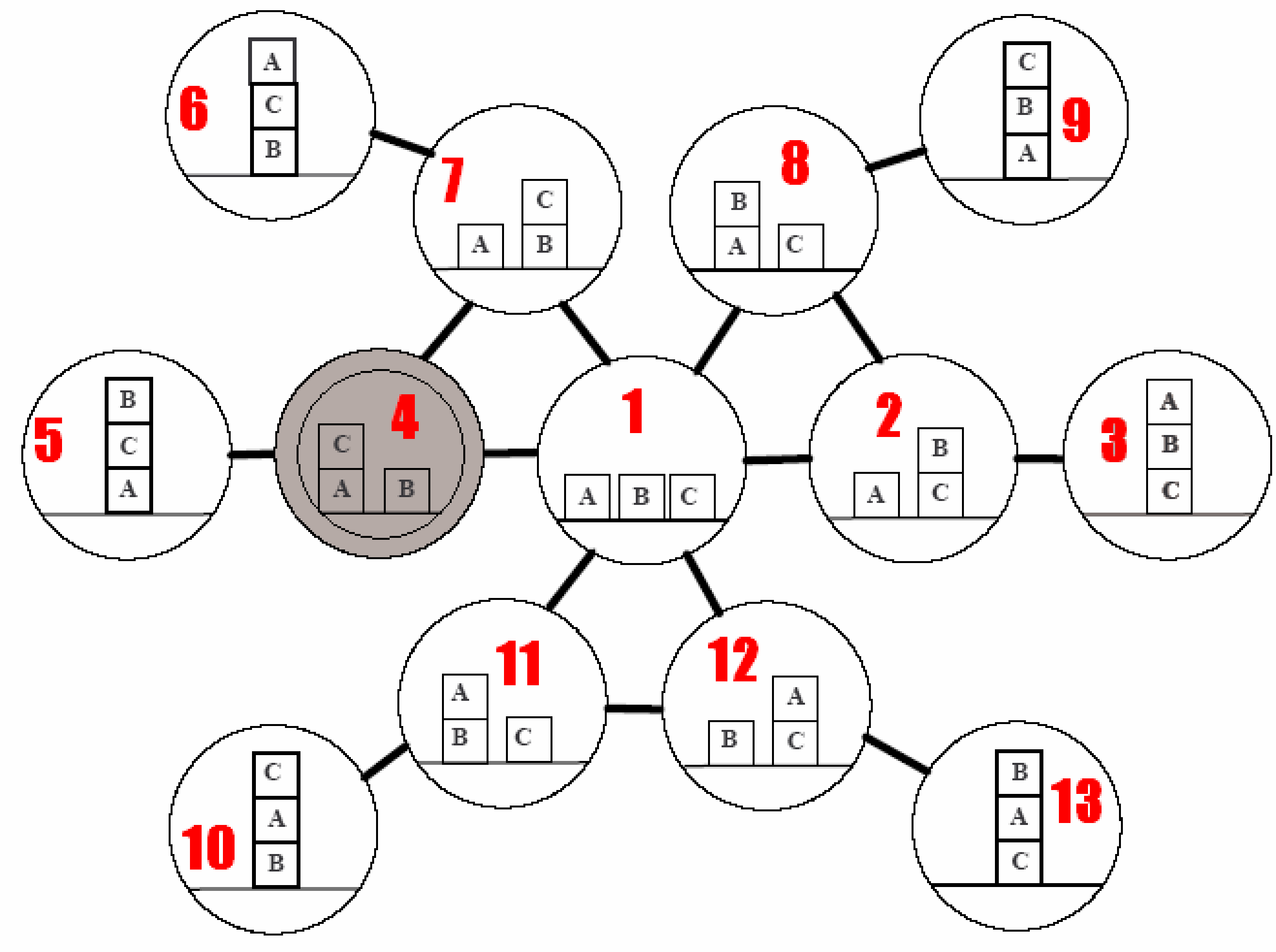
Yosep Kim

Constructing a Kripke Structure for Sussman Anomaly

Let K be a Kripke Structure for Sussman Anomaly. Then K = {W, W_0, T, L}, where

W = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
W_0 = {4}
T = {(1, 2), (2,1), (1, 8), (8,1), (1, 7), (7, 1), (1, 4), (4, 1), (1, 11), (11, 1), (1, 12), (12, 1), (4, 7), (7, 4), (4, 5), (5, 4), (6, 7), (7, 6), (2, 3), (3, 2), (2, 8), (8, 2), (8, 9), (9, 8), (11, 12), (12, 11), (10, 11), (11, 10), (12, 13), (13, 12)}
P = {AonB, BonA, AonC, ConA, BonC, ConB, AonT, BonT, ConT}
L(1) = {AonT, BonT, ConT},
L(2) = {AonT, ConT, BonC},
L(3) = {ConT, BonC, AonB},
L(4) = {BonT, AonT, ConA},
L(5) = {AonT, ConA, BonC},
L(6) = {BonT, ConB, AonC},
L(7) = {AonT, BonT, ConC},
L(8) = {ConT, AonT, BonA},
L(9) = {AonT, BonA, ConB},
L(10) = {BonT, AonB, ConA},
L(11) = {ConT, BonT, AonB},
L(12) = {BonT, ConT, AonC},
L(13) = {ConA, AonC, BonA}.

Pictorially represented is the following. Note that each edge is bi-directional and the number in bold, red font is the state number.



Using the following algorithm I can solve Sussman's Anomaly.

```
Plan(P)
// I denotes an initial state; in this case, State 4
// G denotes a goal state; in this case, State 3
// "0" denotes an empty set

CS = 0          // current state
NS = G          // next states
Plan = 0        // one state plan
while NS ≠ CS do
        if I is subset of NS then return Plan
        OS = OSP(NS, D)                     // { <s, a> : R(s, a) is in NS }
        CS = NS
        Plan = Plan U Prune (OS, NS)                // { <s, a> : s is not in NS }
        NS = NS U Proj (OS)                 // {s : <s, a> is in NS }
end while
return fail
```

Applying the algorithm, Plan(P), above on Sussman’s Anomaly, we get…

To be precise, the second component of the state-action pairs in the following table should be action. For example, *AonB* should be *move(A,B)*, *BonC* should be *move(B,C)*, etc.

Step 1	Step 2	Step 3	Step 4	Step 5
CS = 0 NS = 3 Plan = 0	CS ≠ NS I is NOT in NS OS = {<2, AonB>} Plan = {<2, AonB>} CS = {3} NS = {2, 3}	CS ≠ NS I is NOT in NS OS = {<2, AonB>, <1, BonC>, <8, BonC>, <3, AonT>} Plan = {<2, AonB>, <1, BonC>, <8, BonC>, <3, AonT>} CS = {2, 3} NS = {1, 2, 3, 8}	CS ≠ NS I is NOT in NS OS = {<2, AonB>, <1, BonC>, <8, BonC>, <3, AonT>,<7, ConT>, <4, ConT>,<11, AonT>, <12, AonT>, <2, BonT>,<8, BonT>, <9, ConT>, <1, Bon A>, <2, BonA>} Plan = {<2, AonB>, <1, BonC>, <8, BonC>, <3, AonT>, <11, AonT>, <7, ConT>, <4, ConT>, <9, ConT>, <12, AonT>} CS = {1, 2, 3, 8} NS = {1, 2, 3, 4, 7, 8, 9, 11, 12}	CS ≠ NS I is in NS So, **return Plan**

So, the solution to Sussman’s Anomaly by using a Kripke Structure and Model checking is a plan that is

Plan = {<2, AonB>, <1, BonC>, <8, BonC>, <3, AonT>, <11, AonT>, <7, ConT>, <4, ConT>, <9, ConT>, <12, AonT>}.