Name: Xiang Gu
EID: xg2847
HW11

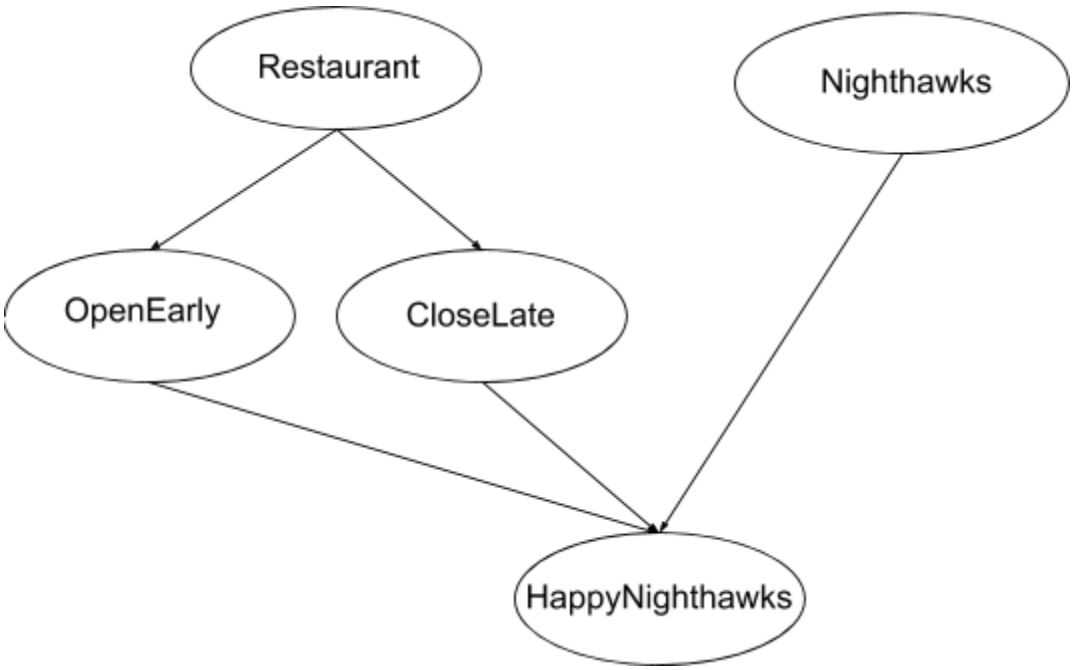
Part 1:
Review question 1:
a). OpenLate(r) <-- Restaurant(r, d, o, c) AND d = ‘Friday’ AND o > 9:00 AM
OpenLate(r) <-- Restaurant(r, d, o, c) AND d = ‘Saturday’ AND o > 9:00 AM

b). CloseLate(r) <-- Restaurant(r, d, o, c) AND d = ‘Friday’ AND c > 9:00 PM
CloseLate(r) <-- Restaurant(r, d, o, c) AND d = Saturday AND c > 9:00 PM

OpenEarly(r) <-- Restaurant(r, d, o, c) AND d = ‘Saturday’ AND o < 8:00 AM
OpenEarly(r) <-- Restaurant(r, d, o, c) AND d = Sunday AND o < 8:00 AM

HappyNighthawks(p) <-- Nighthawks(p, r) AND CloseLate(r) AND OpenEarly(r)

Review question 2:
a).



b). No, they will be no happy nighthawks because the OpenEarly relation is empty (i.e. no restaurant in Restaurant opens before 8:00 AM on Saturdays and Sundays).

c). Yes, now Tom is a happy nighthawk!

	OpenEarly	CloseLate	HappyNighthawks
Before	{}	{“UNOs 360”}	{}
After	{“Magnolia Cafe”}	{“UNOs 360”, “Magnolia Cafe”}	{Tom}

Review Question 3:
To simplify the computation notation-wise, I will use integer numbers to represent UT employees as follows:

Employee Name	Corresponding Integer
Fenves	0
Glodbart	1
Wood	2
Fussell	3
Beckner	4
Tewfik	5
Miranker	6
Mok	7
Alcook	8
Ghosh	9

Then Arc(x,y) can be represented as {(0,1), (0,2), (1,3), (1,4), (2,5), (3,6), (3,7), (4,8), (5,9)}. The following sequence shows what relation Reaches(x,y) contains at each iteration. Tuples in boldface are new ones generated at that iteration.

t = 0: Reaches(x,y) = {}

t = 1: Reaches(x,y) = {**(0,1), (0,2), (1,3), (1,4), (2,5), (3,6), (3,7), (4,8), (5,9), (0,3), (0,4), (0,5), (1,6), (1,7), (1,8), (2,9)**}

t = 2: Reaches(x,y) = {(0,1), (0,2), (1,3), (1,4), (2,5), (3,6), (3,7), (4,8), (5,9), (0,3), (0,4), (0,5), (1,6), (1,7), (1,8), (2,9), **(0,6), (0,7), (0,8), (0,9)**}

t = 3: Reaches(x,y) = {(0,1), (0,2), (1,3), (1,4), (2,5), (3,6), (3,7), (4,8), (5,9), (0,3), (0,4), (0,5), (1,6), (1,7), (1,8), (2,9), (0,6), (0,7), (0,8), (0,9)} (no new tuples generated, terminate!)

Part 2:

17.4.1

a). <START T>; <T, A, 5, 15>; <T, B, 10, 25>; <COMMIT T>

b). <START T>; <T, B, 10, 15>; <T, A, 5, 20>; <COMMIT T>

c). <START T>; <T, A, 5, 11>; <T, B, 10, 12>; <COMMIT T>

17.4.3

b).

1) U is identified as a committed transaction; T is identified as an incomplete transaction.
2) Redo U in an earliest-first order:
   - write 21 to B on disk
   - write 41 to D on disk
3) Undo T in a latest-first order:
   - write 30 to C on disk
   - write 10 to A on disk

d).

1) U, T are both identified as committed transactions.
2) Redo U in an earliest-first order:
   - write 21 to B on disk
   - write 41 to D on disk
3) Redo T in an earliest-first order:
   - write 11 to A on disk
   - write 31 to C on disk
   - write 51 to E on disk

17.4.4

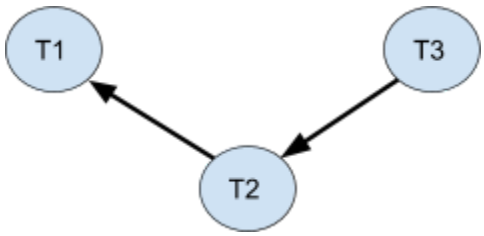
b). Value A, B, C, D *might* appear on disk

18.1.1

r(A); r(B); w(B); w(C); w(D); w(E)

18.2.4

a).



i.

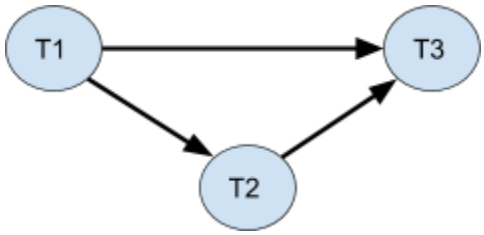
ii. Yes, T3; T2; T1

iii. No, because

T1; T2; T3	No, r_2(A) will read w_1(A) while r_2(A) originally reads initial A.
T1; T3; T2	No, r_2(C) will read w_1(C) while r_2(C) originally reads initial C
T2; T1; T3	No, r_3(B) will read w_2(B) while r_3(B) originally reads initial B
T2; T3; T1	No, r_3(B) will read w_2(B) while r_3(B) originally reads initial B
T3; T1; T2	No, r_2(C) will read w_1(C) while r_2(C) originally reads initial C
T3; T2; T1	Conflict-equivalent to S

b).

i.



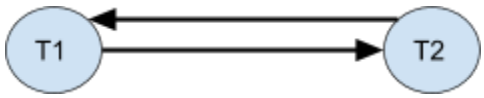
ii. Yes, T1; T2; T3

Iii. No, because

T1; T2; T3	Conflict-equivalent to S
T1; T3; T2	No, r_3(C) will read initial C while r_3(C) originally reads w_2(C)
T2; T1; T3	No, r_2(B) will read initial B while r_2(B) originally reads w_1(B)
T2; T3; T1	No, same as above
T3; T1; T2	No, r_3(C) will read initial C while r_3(C) originally reads w_2(C)
T3; T2; T1	No, same as above

d).

i.



ii. No, there is a cycle in the precedence graph.

iii. No, because

T1; T2	No, r_1(B) will read w_1(B) while r_1(B) originally reads w_2(B)
T2; T1	No, same as above

18.3.3
b). No request will get delayed
d). w_2(B) will be delayed. It (T2) will be allowed to resume after r_1(B) is finished and thereafter lock on B is released.