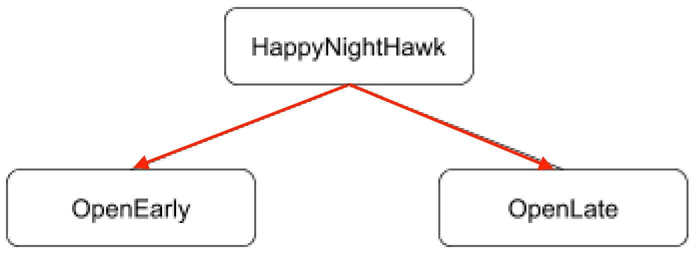
CS386D Database Systems
HW 11 Solutions

Part 1

See file hw11 Q1 solution.pdf

2. a) .



b) None.
c) Tom is the only happy NightHawk (eats at Magnolia Cafe).
Before adding:
OpenEarly = {}, CloseLate = {‘UNOs 360’}, HappyNighthawk = {}

After adding:
OpenEarly = {‘Magnolia Cafe}, CloseLate = {‘UNOs 360’, ‘Magnolia Café’},
HappyNighthawk = {‘Tom’}

3. Iteration 1:
Reaches = {}Reaches = {(Fenves, Goldbart), (Fenves, Wood), (Goldbart, Fussell), (Goldbart, Beckner), (Wood, Twefik), (Fussell, Miranker), (Fussell, Mok), (Tewfik, Ghosh), (Beckner, Alcook)}

Iteration 2:
Reaches = {(Fenves, Goldbart), (Fenves, Wood), (Goldbart, Fussell), (Goldbart, Beckner), (Wood, Twefik), (Fussell, Miranker), (Fussell, Mok), (Tewfik, Ghosh), (Beckner, Alcook), (Fenves, Fussell), (Fenves, Beckner), (Fenves, Tewfik), (Goldbart, Miranker), (Goldbart, Mok), (Goldbart, Alcook), (Wood, Ghosh)}

Iteration 3:
Reaches = {(Fenves, Goldbart), (Fenves, Wood), (Goldbart, Fussell), (Goldbart, Beckner), (Wood, Twefik), (Fussell, Miranker), (Fussell, Mok), (Tewfik, Ghosh), (Beckner, Alcook), (Fenves, Fussell), (Fenves, Beckner), (Fenves, Tewfik), (Goldbart, Miranker), (Goldbart, Mok), (Goldbart, Alcook), (Wood, Ghosh), (Fenves, Miranker), (Fenves, Mok), (Fenves, Ghosh), (Fenves, Alcook)}

Part 2

17.4.1

1. <START T>; <T,A, 5, 15>; <T, B, 10, 25>; <COMMIT T>
2. <START T >; < T, B,10,15>; < T, A,5,20>; <COMMIT T>
3. <START T >; < T, A,5,11>; < T, B,10,12>; <COMMIT T >

17.4.3

b) U is marked as committed but T as incomplete. So, we redo steps for U and undo steps for T by writing B to 21, D to 41 and A to 10, C to 30.

d) U and T are identified as committed.So we redo steps for both by writing A to 11, B to 21, C to 31, D to 41, E to 51.

17.4.4

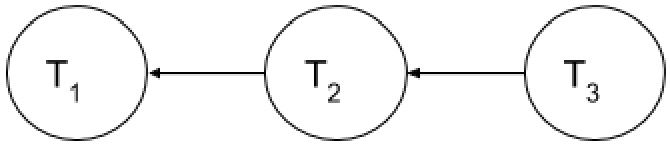
b) Before crash, no values written by T and U must appear on disk. After crash recovery, A=10, C=30, B=21, D=41 must appear on disk.

18.1.1

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

18.2.4

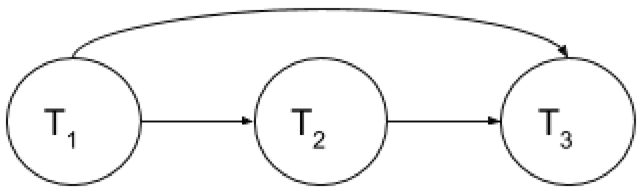
a) (i) .



(ii) Yes, it is conflict-serializable because the graph has no cycle. The equivalent serial schedule is: T3, T2, T1.

(iii) There are no serial schedules that are equivalent but not conflict-equivalent.

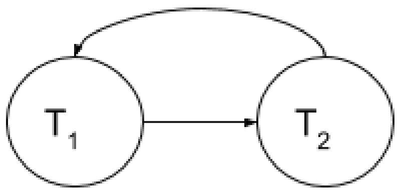
b) (i) .



(ii) Yes, it is conflict-serializable because the graph has no cycle. The equivalent serial schedule is: T1, T2, T3.

(iii) There are no serial schedules that are equivalent but not conflict-equivalent.

c) (i) .



(ii) No, it is not conflict-serializable because the graph has no cycle.

(iii) None.

18.3.3

b) No requests are delayed.

d) $w_2(B)$ would be delayed and allowed to resume after request $r_1(B)$ executes and releases the lock.