CSCI3100 Software Engineering

Assignment 3

Due – 11:59:59pm, 14th Mar, 2021 (Sunday)

Please submit the homework online through Blackboard.
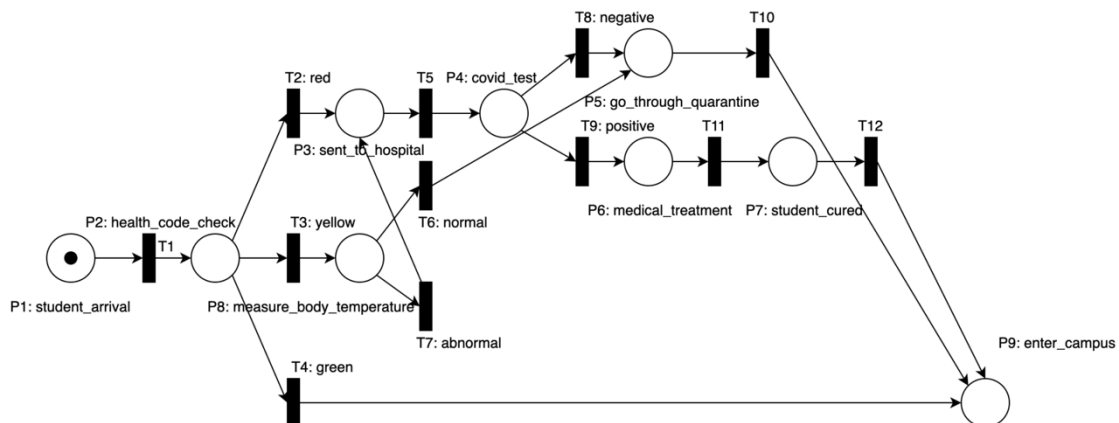
Late submission penalty within 24 hours: 50%; after 24 hours: 100%.

Remember to go through Veriguide for Academic Honesty Declaration.

Missing Veriguide report: 50% mark deduction.
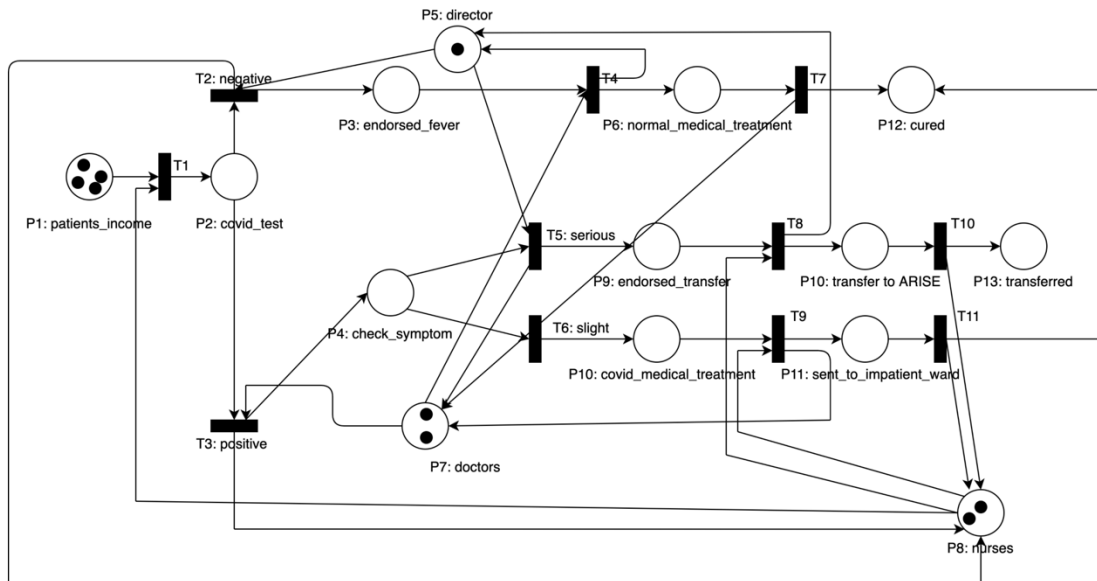
---

1. Medical Treatment Problem (25 points)
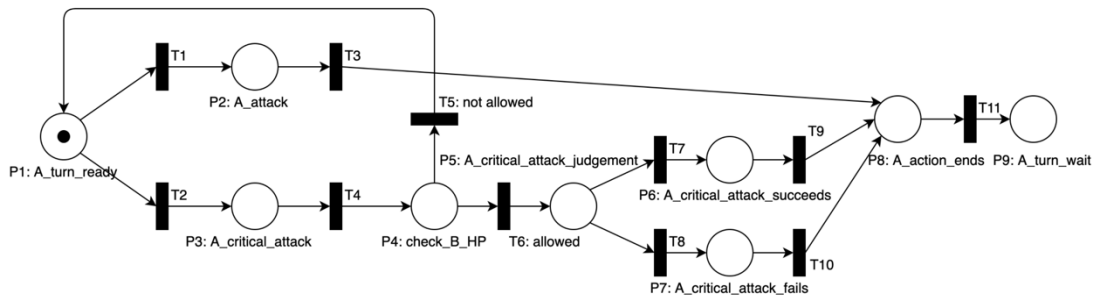
1)

Possible Firing Sequence:

T1 → T4

2)

A Possible Firing Sequence: (not the only correct answer)
(T1, T1) → (T2, T3) → (T4, T6) → (T7, T9) → T11 → (T1, T1) → (T3, T3) → (T5, T6) →
(T8, T9) → (T10, T11)

2. Turn-based Game Problem (30 points)

1)

Firing Sequence: (all the following sequences are correct)
T1 → T3 → T11
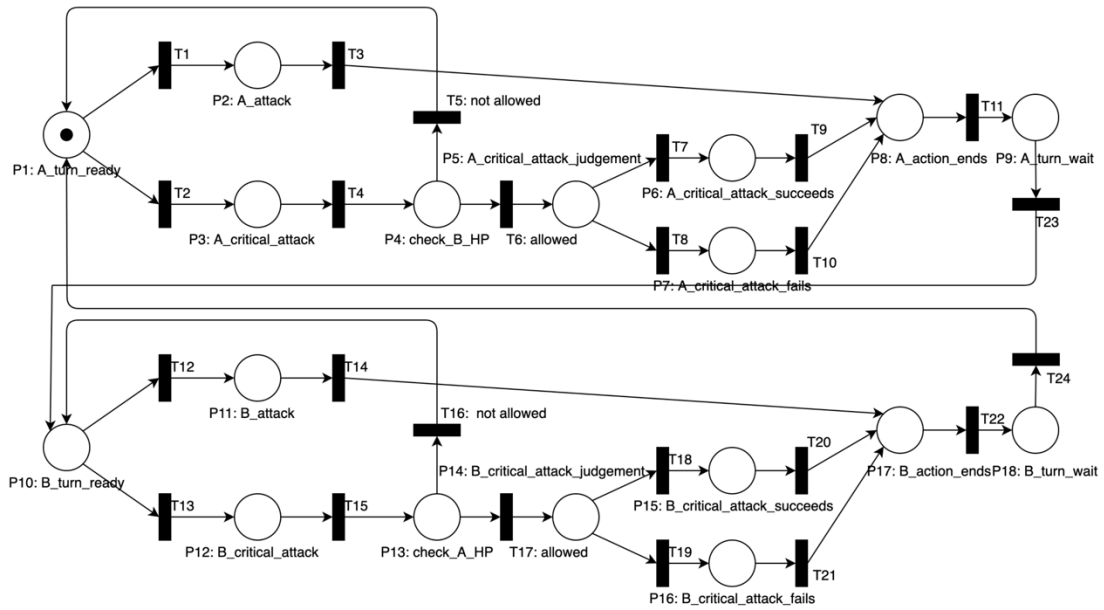T2 → T4 → T6 → T7 → T9 → T11
T2 → T4 → T6 → T8 → T10 → T11
T2 → T4 → T5 → T2 → T4 → T5 → … → T6 → T7 → T9 → T11
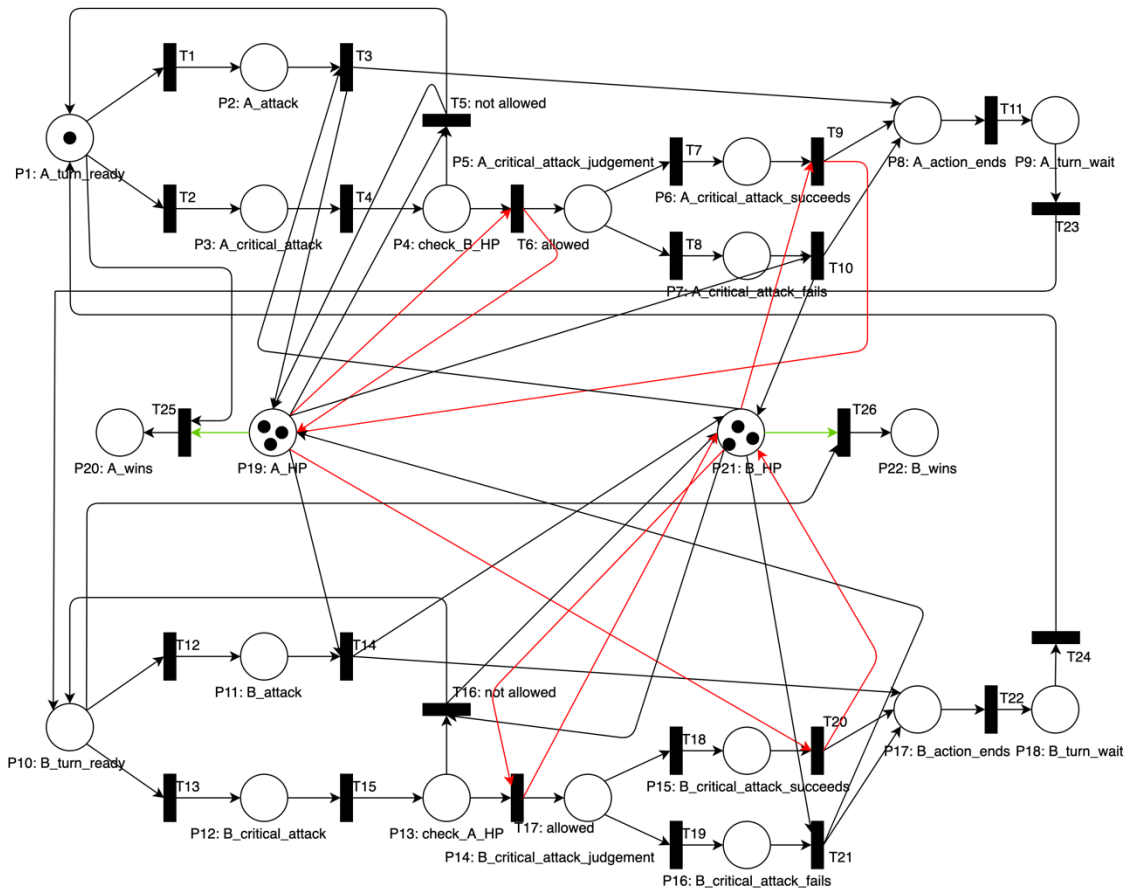T2 → T4 → T5 → T2 → T4 → T5 → … → T6 → T8 → T10 → T11

2)

A Possible Firing Sequence:
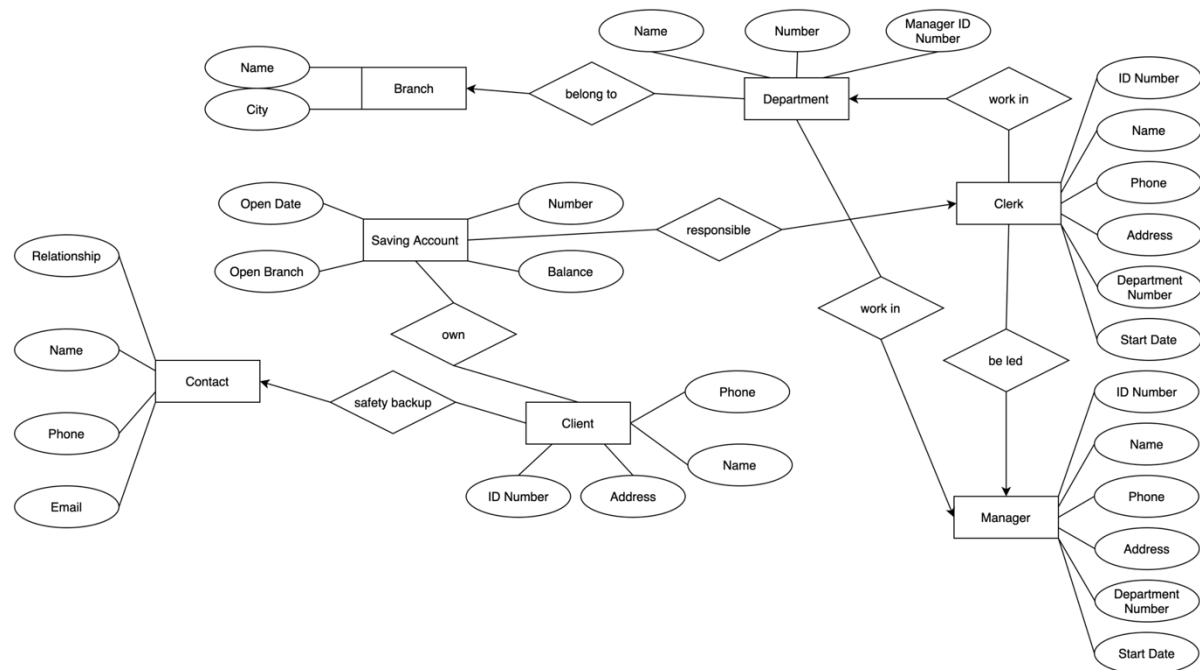
T1 → T3 → T11 → T23 → T12 → T14 → T22

3)

The red arrows in this Petri Net have weight of 2.
The green arrows in this Petri Net have weight of 6.

Other arrows have weight of 1.

3. Entity-Relationship Diagram (20 points)

4. Logic Specification (25 points)

1)
   a) False
   b) False
   c) True:
        $c = 3$
        $x,y,i,j = 2,4,5,10$ or $5,7,8,13$ or $8,10,11,16$ or $11,13,14,19$ or $14,16,17,22$
   d) False
   e) False
   f) False

2)
Any reasonable explanation is accepted. For example, the predicate check whether a pattern exists at the location xxx…

4a. Logic Specification (bonus question)

```
valid_number(m, n) ≡
for all i (m <= i <= n) implies valid_digit(i)
```

```
//number is represented with several digits, you must convert them into a number before
comparing
Is_longer(a, b, i, j) ≡
for all x (i <= x <= j) implies(
//sₓ is the accumulator
exists nₓ, sₓ  (nₓ = cₓ × 10^(j-x) and
sᵢ = nᵢ and
(i < x <=j) implies (sₓ = n_{x-1} + s_{x-1}) and
exists s_{j+1}  (s_{j+1} = n_j + s_j) and
b - a + 1 >= s_{j+1})
)
```

```
valid_answer(m,n) ≡
    // check if this sequence contains two "#" and if the first sequence is a number
    exists   i   (m  <  i  <  n  -  1)  implies  ( cᵢ =   "#"   and
valid_number(m,i - 1) and exists j (i + 1 < j < n) implies (c_j =
"#" and
    //check if the second and last sequence are words
    valid_word(i + 1,j - 1) and valid_word(j + 1,n) and
    //check if the predefined word is in the last sequence
    exists x,y (j < x <= y <= n and x - y = j - i - 2) implies
( for all k (x <= k <= y) implies c_k = c_{i+1+k-x} and
    //subcase 1: the word appears at the beginning
        ((x = j + 1 and c_y = c_{y+1}) or
    //subcase 2: the word appears at the end
        (y = n and c_x = c_{x-1}) or
    //subcase 3: there is only one word
        (y = n and x = j + 1) or
    //subcase 3: the word appears in the middle
        (x != j + 1 and y != n and c_x = c_{x-1} and c_y = c_{y+1}))) and
    //check if the length of each word is bigger than the predefined number, here we check
any word sequence whose beginning is the same with the former character and end is the same
with the latter character, this word sequence does not necessarily contain only one word, but
all of such sequences are longer than the predefined number means that every word is longer
than the predefined number
    for all a,b (j < a <= b <= n and
                ((c_a = c_{a-1} and a != j+1) or (a = j+1)) and ((c_b
                = c_{b+1} and b != n) or (b = n))
                    ) implies Is_longer(a, b, m, i - 1))

))
```