CSCI3100 Software Engineering

Assignment 3

Due – 11:00am, 16th Mar, 2020 (Monday)

Note: Please submit the homework online through blackboard.
Remember to go through Veriguide for Academic Honesty Declaration.
Late submission during 11:00am - 5:00pm: 20% mark deduction.
Missing attendance of pre-assigned tutorial session: 50% mark deduction.

Answer these problems based on lecture Topic 4 notes, from which you can consider for solutions. You can use a tool called pipe2 (http://pipe2.sourceforge.net/) for developing and drawing Petri Nets. Or you can simply use Powerpoint to draw the diagrams and import them to Word, if you use Word document to prepare your homework solution.

1. **Mobile Phone Ordering Problem (20 points)**

Let us consider a real-life business process of a telephone company, handling an incoming order for a mobile telephone. For example, a customer is ordering an iPhone from Apple Store on its website as shown in Figure 1. The processing of this order could involve two departments, i.e., an accountancy department to handle the payment, and a sales department to handle the distribution.

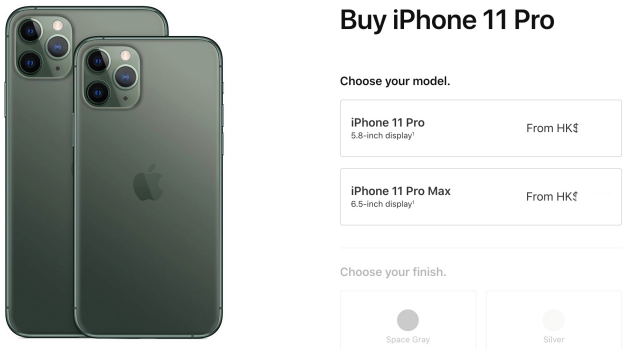


Figure 1 An online order of iPhone.

Questions:

(1) First, let us consider the workflow of only the accountancy department. Once an order is received for processing ("order_income"), the accountancy department enters "order_to_account" condition, where the standing customer is checked first ("credit_checked"), with the result to be OK or not. If the result is OK, the payment is arranged ("payment_arranged"). Then the order is accepted ("order_accepted"). Otherwise, the system will suggest to cancel this order ("cancel_suggested"). Then the order is refused ("order_refused"). For either accepted or refused, the order will be archived ("order_archived"). Please design a Petri Net for this workflow and identify

all the places and transitions you draw. Please initialize the Petri Net and give a possible firing sequence.

(2) Second, let us then consider the corporation of the two departments. Once the order is received for processing ("order_income"), it is sent to the two departments for processing in parallel. Main process of the accountancy department is described in (1). For the sales department, it will first enter "order_to_sales" condition, where a check on the order details ("order_checked") is performed. Then, the sales department (always) waits for the result of "credit_checked" from the accountancy department. If the result of "credit_checked" is OK, the sales department will go through a sequence of operations, namely, "item_picked", "item_wrapped", and "item_delivered". If "payment_arranged" from accountancy department and "item_delivered" from sales department are both satisfied, the order can be accepted ("order_accepted"). If the result of "credit_checked" in the accountancy department is not OK, "credit_unsatisfied" condition happens, and the sales department will stop the order right after "order_checked" and cancel further processing ("sales_canceled"). If "cancel_suggested" condition from the accountancy department and the "sales_canceled" condition from the sales department are both satisfied, the order is refused ("order_refused"). For either being accepted or refused, the order will be archived ("order_archived"). Please design a Petri Net for this workflow and identify all the places and transitions you draw. Please initialize the Petri Net and give *two* possible firing sequences, one for successful order and the other for unsuccessful order.

2. **Robotic Soccer Player Problem (25 points)**

Suppose that we have a robot designed for playing soccer, as shown in Figure 2. Similar to a normal soccer player, the robot should be able to move the ball and score. The purpose of this problem is to design a rule to arrange the movement of the robot and its action of scoring a goal.

Figure 2 Robotic Soccer Players.

Questions:

(1) First, let us consider the scenario of the robot moving to the ball. The scenario involves the following conditions: Initially, the robot stands by the ball ("stand_by") and does not detect the ball ("not_see_ball"). Then, if the robot sees the ball ("see_ball"), it will try to move to the ball ("move_to_ball"). During "move_to_ball", if the robot cannot see the ball for some reason, it goes back to "stand_by". Otherwise, it moves close to the ball ("close_to_ball").

Please design a Petri Net for this workflow and identify all the places and transitions in your answer. Please initialize the Petri Net and give a possible firing sequence.

(2) Now, let us move further to the action of receiving and moving the ball. The conditions are as follows: When the robot is moving and also close to the ball, it will try to take the ball ("take_ball"). If the robot fails, it does not have the ball ("not_has_ball") and will keep trying to take the ball again. Otherwise, it takes the ball successfully ("has_ball"). When the robot has the ball, it has two choices: (i) directly kick the goal ("kick_to_goal"), (ii) dribble first ("dribble_to_goal"). When the robot moves close to the opposite goal ("close_to_opp_goal"), it will kick to goal ("kick_to_goal"). During dribbling the ball, if the robot loses the ball ("not_has_ball") by accident ("accident_occurred"), it should go back to take the ball again. For both choices, after "kick_to_goal", if the ball "scores", the game is over. Otherwise, the robot will not have the ball and it will go back to take the ball again.

Please design a Petri Net for this workflow and identify all the places and transitions in your answer. Please initialize the Petri Net and give a possible firing sequence.

3. **Smart Home System Problem (25 points)**

Smart Home technology, also termed Home Automation, is the use of devices in the home with network connections, most commonly including a local LAN and/or the Internet. It uses devices such as sensors and other appliances connected to the Internet of things, which can be remotely monitored, controlled or accessed. Smart Home provides services that respond to the perceived needs of the users. Here we consider two commonly used sub-systems in a smart home, i.e., the fire alarm system and the climate control system.

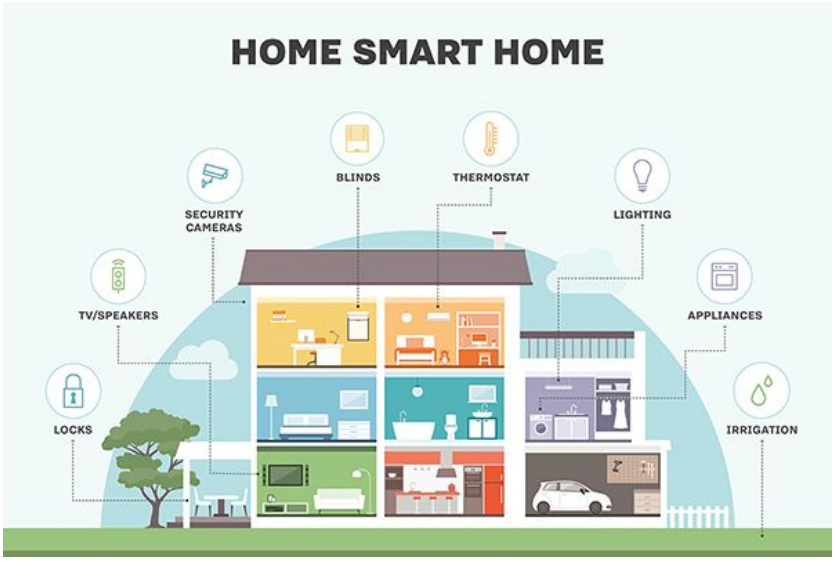


Figure 3 Smart home system.

Questions:

(1) First, we consider the fire alarm system in the Smart Home. The fire alarm system monitors temperature level, temperature rate change, and smoke, so as to detect any fire events and take appropriate actions to protect the home. The fire alarm system can be operated in the following conditions: Initially, the temperature level is normal ("temp_normal"). If the system detects an increase of the temperature level which exceeds a threshold T, the system signals an elevated temperature level ("temp_elevated"). If the temperature level goes below T, the condition returns to normal temperature. Otherwise, if the temperature level keeps increasing with a changing rate beyond a threshold TR, the alarms are launched ("alarm_launched"). At the same time, the timer of the system is started ("timer_started"). The timer records the duration of temperature increase. If the duration exceeds a preset time threshold TIME, automatic fire extinguisher will be started ("fire_ext_started"). The fire extinguisher will also be launched if the system detects a smoke during alarming ("smoke_detected"). After that, the temperature level is reset to normal. In addition, the owner of home can choose to enable or disable the fire alarm system ("fire_sys_on" or "fire_sys_off"). If the owner leaves home ("leave_home"), the fire alarm system is enabled automatically.

Please design a Petri Net for this workflow and identify all the conditions and transitions you draw. Please initialize the Petri Net and give a possible firing sequence.

(2) Climate control system enables smart control of the air conditioning (AC) systems to provide the maximum comfort and effective usage of the AC systems within the Smart Home. The system includes the following conditions: At beginning, the ACs are in the initial condition ("ACs_init"). The ACs can be turned on ("ACs_on") by the owner in manual mode ("manual_mode") and can be turned off ("ACs_off") by the owner or automatically. When the ACs are on, the owner can set them as automatic mode ("auto_mode"), which can be set back if necessary. In the meantime, in manual mode, the owner can change the ACs settings according to his/her preference ("change_by_pref"). In automatic mode, the ACs settings can be changed automatically according to the environmental condition ("change_by_envir"). For saving energy, the

ACs will detect the occupancy of place and start a timer ("in_timer"). If the place is not occupied ("not_occupied"), the ACs will be turned off once the timer runs out of time TIME. Otherwise, within TIME, if the place is occupied, the ACs will keep turning on.

Please design a Petri Net for this workflow and identify all places and transitions in your answer. Please initialize the Petri Net and give a possible firing sequence.

A l=n

x ≠ y

4. **Logic Specification (30 points)**

Please answer the questions below regarding logic specifications.

(1) Suppose we have a program that replaces an element `x` from an array A of length `n` by another element `y`. In the array A, we assume the `precondition` is that no two elements are equal, and `y` is not equal to `x` or any other elements in A.

Please use logic specification to specify the above `precondition`, and then specify a **procedure** `replace` for this operation, including its parameters. You can use old_A to represent the original array A.

(2) In this problem we assume the input contains three sequences of words, separated by a special character '#'. The start index of the first sequence of words is given as M1, and the end indices of the three sequences of words are given as N1, N2, N3, respectively. We want to check if the three sequences of words of the input satisfy the following relationships: (i) For the first sequence of words, it contains two words composed of alphabetic characters, separated by a blank. We name them as word1 and word2. Word2 can be null; but the word1 must not. (ii) The second sequence of words allows the occurrences of word1 zero or multiple times. (iii) The third sequence of words is obtained by rewriting the second sequence of words with all occurrences of word1 replaced by word2.

(a) Please use logic specification for predicate `replaced_text(M1, N1, N2, N3)` to state that input file from the M1 through N3 contains three word sequences satisfying the described relationship. To complete this question, you may refer to the predicates `input_word(m, n)` and `input_text(m, n)` introduced in the lecture Topic 4.

(b) For the `replaced_text (M1, N1, N2, N3)` defined above, we consider the following five inputs:

(i) #stone dirt#constant dripping wears away a stone#constant dripping wears away a stone#

(ii) #do#you can do it# you can do it#

(iii) #can do#you can do or cannot do it#you do do or donot do it#

(iv) #you I#do you or I agree#do I or you agree#

(v) #just please#just just never give up for yourself#please never give up for yourself#

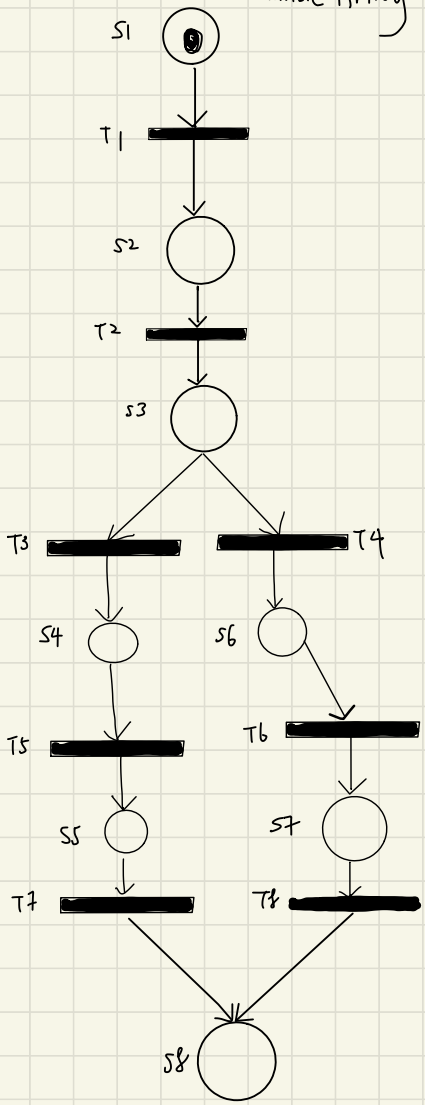
Please answer if the five inputs will satisfy the requirement predicate of `replaced_text (M1, N1, N2, N3)`; that is, whether it returns `True` or `False`. Please give your explanations.

Ng Chi Hon 1155116317

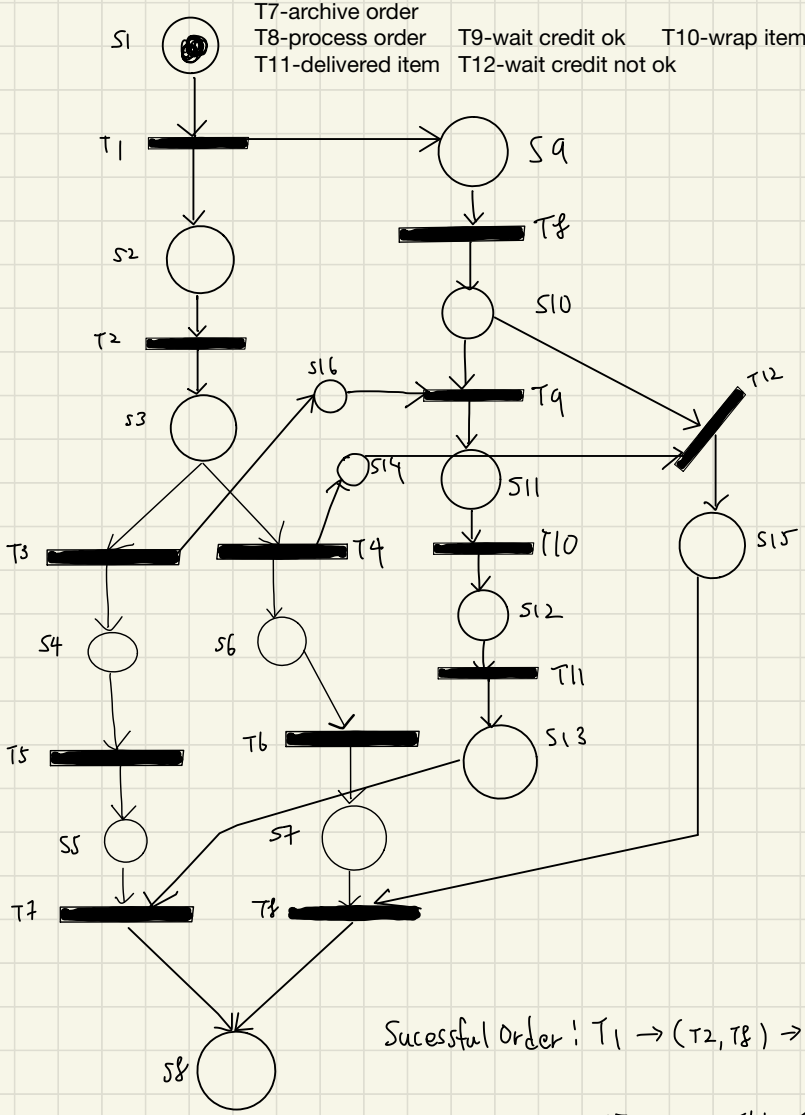
Q1) (1)

s1-order_income s2-order_to_account s3-credit_checked
s4-payment_arranged s5-order_accepted s6-cancel_suggested
s7-order_refused s8-order_achieved
T1-process order T2-customer be checked T3-result is ok
T4-result is not ok T5-accept order T6-refuse order
T7 archive order

One Possible Firing sequence: T1 → T2 → T3 → T5 → T7



s1-order_income s2-order_to_account s3-credit_checked
s4-payment_arranged s5-order_accepted s6-cancel_suggested
s7-order_refused s8-order_achieved
s9-order_to_sales s10-order_checked
s11-item_picked s12-item_wrapped s13-item_delievered
s14-credit_unsatisfied s15-sales_canceled s16-credit_satisfied
T1-process order T2-customer be checked T3-result is ok
T4-result is not ok T5-accept order T6-refuse order
T7-archive order
T8-process order T9-wait credit ok T10-wrap item
T11-delivered item T12-wait credit not ok



Sucessful Order: T1 → (T2, T8) → T3 → (T5, T9)

↓

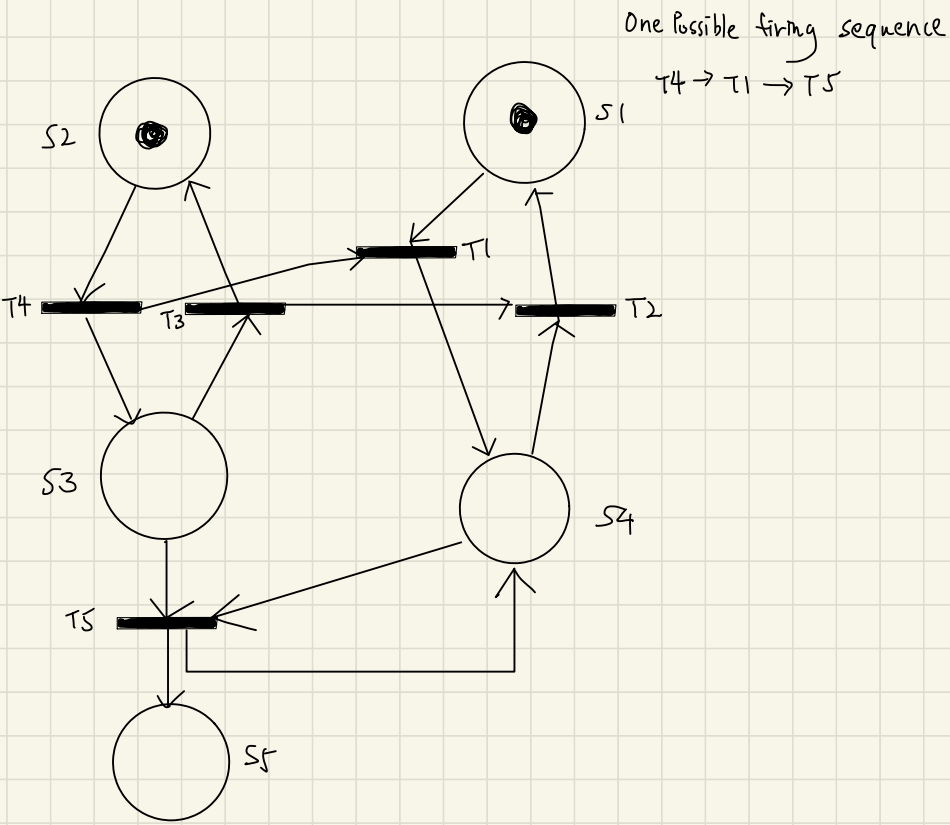
T7 ← T11 ← T10

unsucessful order: T1 → (T2, T8) → T4 → (T6, T12)

↓

T8

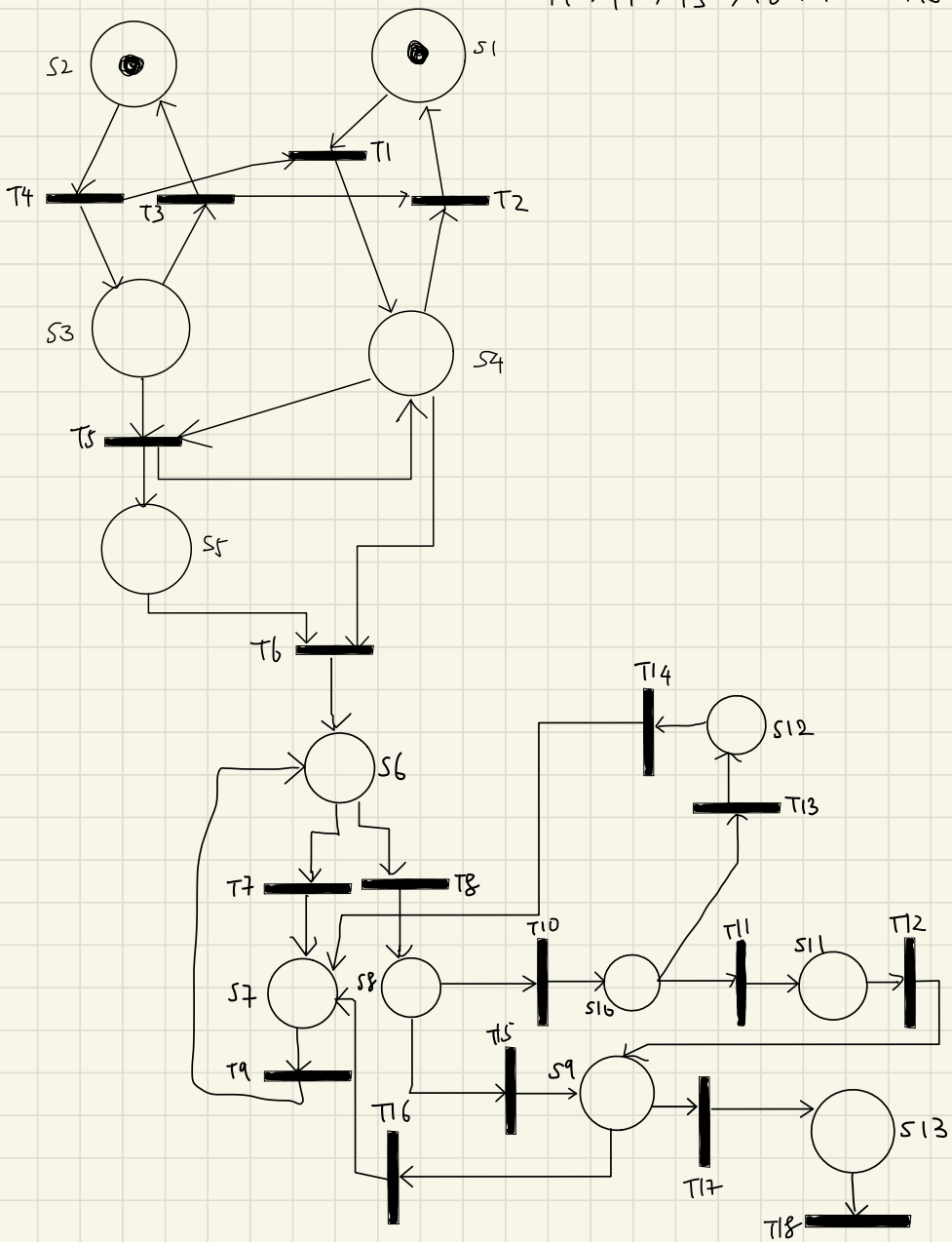
2(1) s1- stand_by s2- not_see_ball s3-see_ball s4-move_to_ball
s5- close_to_ball
t1- move t2- be stand by t3- miss ball t4- detect ball
t5- move and close to ball



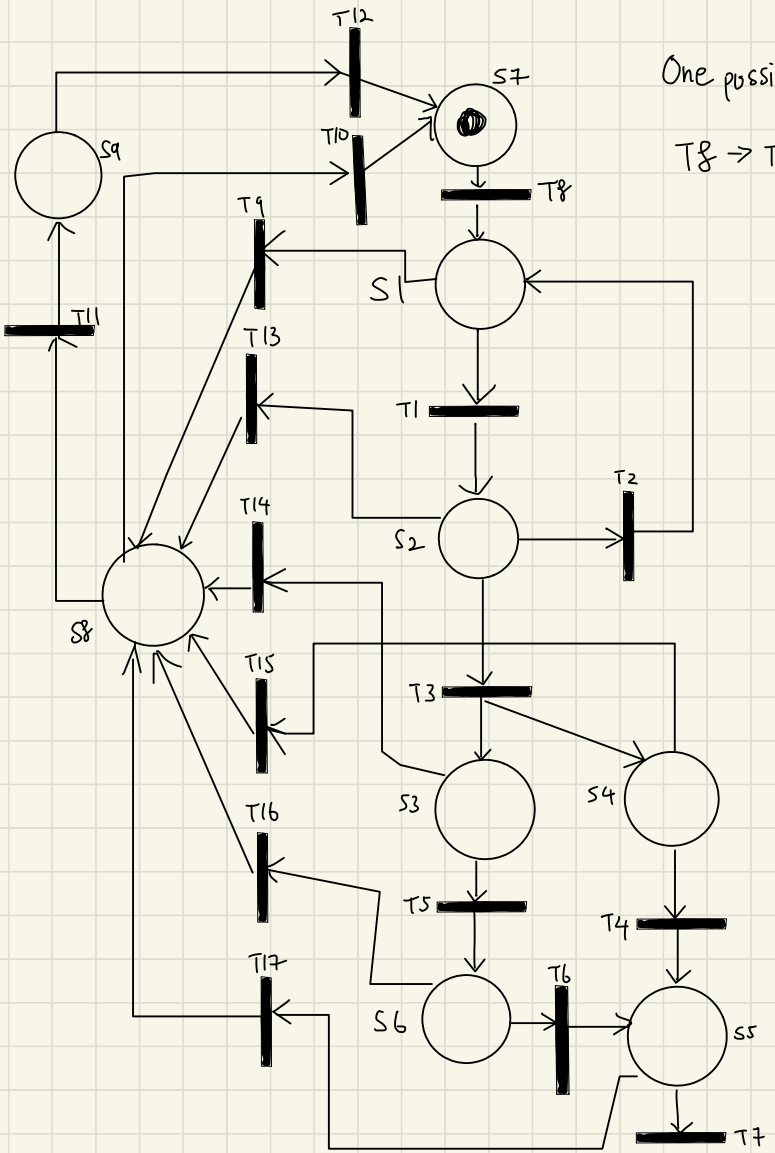
(2) s1- stand_by s2- not_see_ball s3-see_ball s4-move_to_ball
s5- close_to_ball s6- take_ball s7- not_has_ball
s8- has_ball s9- kick_to_goal s10- dribble_to_goal
s11- close_to_opp_goal s12- accident_occurred
s13- scores
t1- move t2- be stand by t3- miss ball t4- detect ball
t5- move and close to ball t6- try take ball t7- take ball fail
t8- take ball successfully t9- take ball again t10- dribble ball
t11- dribble close to goal t12- kick ball t13- face accident
t14- lose the ball(after dribble) t15-kick ball directly t16- lose the ball(after kick) t17- be scored t18- game is over

One possible firing sequence

T4 → T1 → T5 → T6 → T8 → T15 → T17
↓
T18



S1- temp_norm S2-temp_elevated S3- alarm_launched
S4- timer_started S5- fire_ext_started S6- smoke_detected
S7- fire_sys_on S8- fire_sys_off. S9- leave_home
T1- detect temperature level exceed T
T2- detect temperature level below T
T3- launch alarm and start timer due to increasing rate beyond TR
T4- duration of timer exceed TIME
T5-detect smoke T6- smoke detected
T7- Temperature reset to normal
T8- switch on system T9- disable system T10- enable system manually
T11- owner leave home T12- enable system automatically
T13 to T17 - disable system



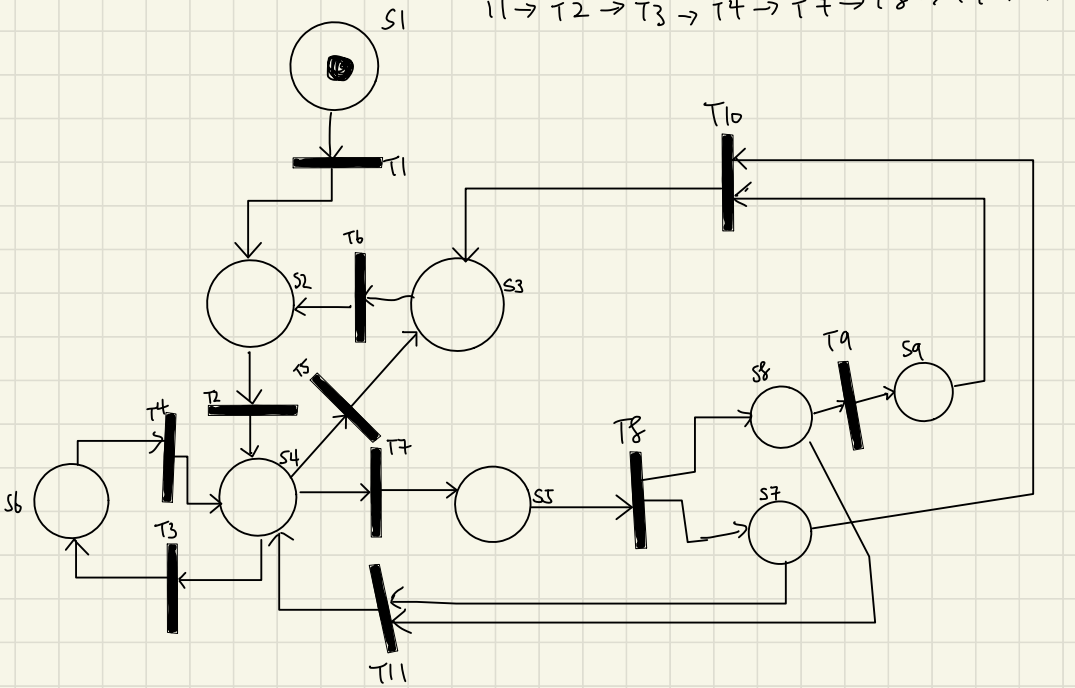
One possible firing sequence:

T8 → T1 → T3 → T4 → T7

(2)

S1- ACs_init S2- ACs_on S3-ACs_off
S4- manual_mode S5- auto_mode S6- change_by_pref
S7- change_by_envir S8- in_timer S9- not_occupied
T1- 1st time manual switch on T2- system in manual mode
T3 - change ACs setting by owner
T4 - finish changing ACs setting
T5 - manual switch off T6-not 1st time manual switch on
T7 - switch to auto mode
T8- start auto-ACs setting and saving energy
T9 - detect not occupied T10- switch off ACs due over
TIME T11- manual operation on ACs

One possible firing sequence:
T1 → T2 → T3 → T4 → T7 → T8 → T9 → T10



(1) precondit(A,y) ≡ exist j (1 ≤ j ≤ n) and forall i (1 ≤ i ≤ n implies A(j) ≠ A(i+j))
and
forall k (1 ≤ k ≤ n implies A(k) ≠ y)

replace (y i x:A' in out array element i in element)

≡ (exists i (1 ≤ i ≤ n and Old.A(i) = X and A'(i) = y)

(2) input_word ≡ (for all i (m ≤ i ≤ n) implies alphabetic (c_i))

input_text1 (m, n) ≡

(i_m = '#' and ~~i_n = '#'~~ and
(exist k (k = 2) and
exist h_1, m_1 h_2, m_2 (inputword (m_1, m_1+h_1) and
Inputword (m_2, m_2+h_2) and m_1 = m+1 and h_1 > 0 and h_2 ≥ 0
m_2 = m_1 + h_1 + 2 and i_{m_1+h_1+1} = ' ')))

input_text2 (m, n) ≡

(i_m = '#' and i_n = '#' and
exist k (for all j (1 ≤ j ≤ k) implies
exist h_j, m_j (input_word (m_j, m_j+h_j) and
m_1 = m+1 and m_k+h_k+1 = n and
(1 ≤ j < k) implies (m_{j+1} = m_j + h_j + 2 and
i_{m_j+h_j+1} = ' ')))))

Found text2 ≡

replaced_text = (input_text1 (M1, N1) input_text2 (N1, N2)
Found_text2

b) (1) False because word stone in sequence 3 not replace 'stone' to 'dirt'

(2) False because nonnessary space ' ' appear before third sequce word you

(3) True because word 'can' in sequence 3 replace by "do" correctly

(4) False because word "I" in sequence 3 should not be replaced by "you

(5) False because there are two just and should use two please replace just and result of third sequce should be # please please never give up for yourself #)