ZI 2015./16.

(1.)

$$W = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{array}{c} \begin{array}{cccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{array} \\ \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 1 & 0 & -1 \\ 1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}$$

a) $\{p_1, p_3, p_4\}$ $\quad p = [1\ 0\ 1\ 1\ 0\ 0]$ $\quad W \cdot p = [0\ 0\ 0\ 0]^T$ $\boxed{\text{JE}}$

$\{p_4, p_5, p_6\}$ $\quad p' = [0\ 0\ 0\ 1\ 1\ 1]$ $\quad W \cdot p' = [-1\ 0\ 0\ 0]^T$ $\boxed{\text{NIJE}}$

b) $S = \{p_4, p_5, p_6\}$ sifon ?

Prema definiciji $\bullet S = \{t_2, t_3, t_4\}$

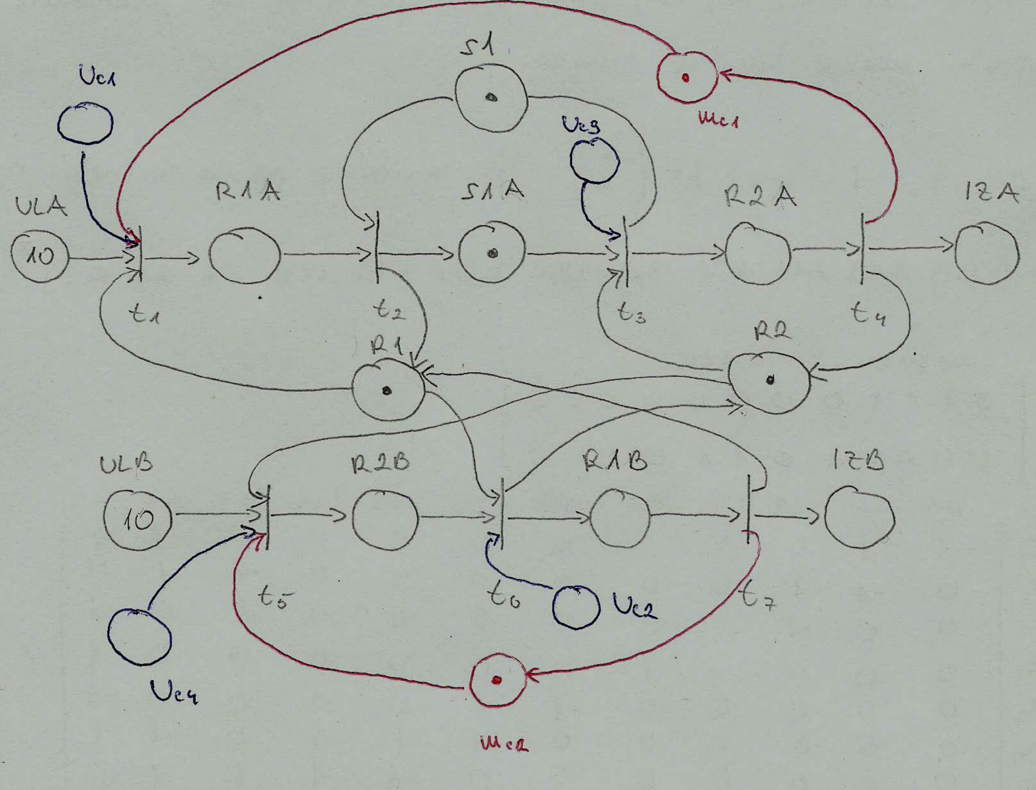
$S \bullet = \{t_1, t_2, t_3, t_4\}$

$\bullet S \subset S \bullet$ vrijedi pa zaključujemo da je S sifon.

To se može zaključiti i iz $W \cdot p' = [-1\ 0\ 0\ 0]^T$. Budući da S nije P-invarijanta, postoji mogućnost da svi tokeni „pobjegnu". Iz $W \cdot p'$ vidi se da mogu pobjeći kroz t_1 i više se ne mogu vratiti ($W \cdot p'$ sadrži samo -1 i 0, nema 1 koji bi označavao ulazni prijelaz).

(2.)

a)



b) Kružno čekanje: $C = \{R1, R2, S1\}$

Kritični podsustav:

$$T_c^i = \bullet C = \{t_2, t_3, t_4, t_6, t_7\}$$

$$T_c^o = C\bullet = \{t_1, t_2, t_3, t_5, t_6\}$$

$$T_c = \bullet C \cap C\bullet = \{t_2, t_3, t_6\}$$

$$T_s = T_c^i \setminus T_c = \{t_4, t_7\} \qquad T_Q = T_c^o \setminus T_c = \{t_1, t_5\}$$

$$\bullet T_s = \{R2A; R1B\} \qquad T_Q\bullet = \{R1A, R2B\}$$

$$J(c) = \{R1A, R1B, R2A, R2B, S1A\}$$

$$J_s(c) = J(c) \cap \bullet T_s = \{R2A, R1B\} \qquad J_Q(c) = J(c) \cap T_Q\bullet = \{R1A, R2B\}$$

$$J_{SQ}(c) = J_s(c) \cap J_Q(c) = \emptyset$$

$$J_N(c) = J(c) \setminus \{J_Q(c) \cup J_s(c)\} = \{S1A\}$$

$$J_o(c) = \{J_Q(c) \setminus J_{SQ}(c)\} \cup J_N(c) = \boxed{\{R1A, R2B, S1A\}}$$

→ A sad puno brži način:

$$J(c) = J_s(c) \cup J_o(c) \Rightarrow \boxed{J_o(c) = J(c) \setminus J_s(c)}$$

→ računate $T_c^i, T_c^o, T_c, T_s, \bullet T_s, J(c)$ i imate sve potrebno. Zamke i neutralni poslovi vam ne trebaju.

c) $b_A = 2 \quad b_B = 1 \Rightarrow b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ → proširimo sve na matrični zapis

$$m_p = [ULA\ R1A\ S1A\ R2A\ IZA\ ULB\ R2B\ R1B\ IZB\ S1\ R1\ R2]^T$$

$$L = \begin{array}{c} \begin{array}{cccccccccccc} & R1A & S1A & R2A & & & R2B & R1B & & & & \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$W_p = \begin{array}{c|cccccccccccc} & ULA & R1A & S1A & R2A & IZA & ULB & R2B & R1B & IZB & S1 & R1 & R2 \\ \hline t_1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ t_2 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ t_3 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ t_4 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ t_5 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 \\ t_6 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 \\ t_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 \end{array}$$

$$W_c = -W_p \cdot L^T = - \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} = \begin{array}{c c} \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} & \begin{array}{c} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \end{array} \\ \begin{array}{cc} m_{c1} & m_{c2} \end{array} & \end{array}$$

$$m_c(0) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} - L \cdot m_p(0) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

m_{c1} ulazi u prijelaz t_1, a izlazi iz prijelaza t_4 (označeno crvenim
m_{c2} ulazi u prijelaz t_5, a izlazi iz prijelaza t_7 mjestima)

d) U sustavu iz c) zaglavljenje nije moguće jer smo p-invarijantnim upravljanjem spriječili pražnjenje kritičnog sifona.

e) AKO (ULA!=0 && m_{c1} && R1 && R2B) ONDA $U_{c1}=1$, $U_{c2}=0$
INAČE $U_{c1}=1$, $U_{c2}=1$

(označeno plavim mjestima)

AKO (ULB!=0 && m_{c2} && R2 && S1A) ONDA $U_{c3}=1$, $U_{c4}=0$
INAČE $U_{c3}=1$, $U_{c4}=1$

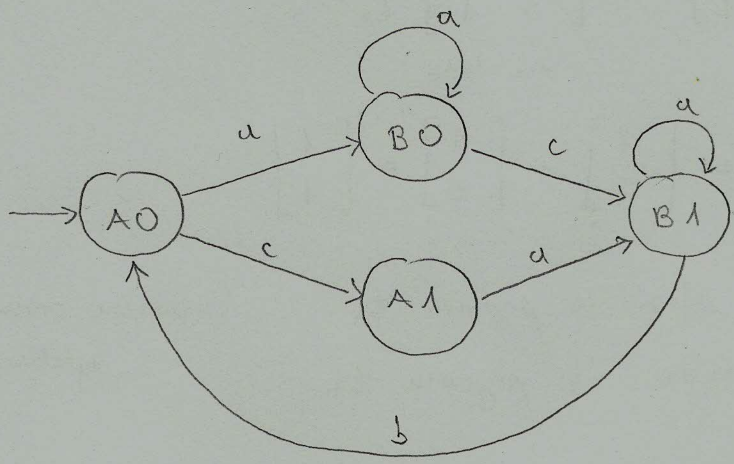
Cilj je odabrati prioritete operacija koje će roboti R1 i R2 raditi kada imaju više mogućnosti. U suprotnom, pusti sustav da radi bez ometanja.

3.

a) Kod upravljanja poslovima višeradnog resursa Petrijevom mrežom cilj je spriječiti konflikte i zaglavljenja sustava. Kod upravljanja max-plus modelom to nije problem jer je pretpostavka da je redoslijed zadataka unaprijed definiran. U tom slučaju cilj je podesiti početne uvjete tako da se minimizira ciklus grafa i poveća iskoristivost resursa.

b) $E_1 = \{a, b\}$ $E_2 = \{b, c\}$

$E_p = E_1 \cap E_2 = \{b\}$



(4.) $r = [3 \ 14 \ 12 \ 12]^T$ $N = Max - A$ $p = [P \ Q \ R \ S]$

$$A = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 1 & 0 & 3 & 0 \\ 0 & 2 & 5 & 4 \\ 0 & 1 & 3 & 2 \end{bmatrix} \qquad N = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 7 & 2 & 0 \\ 2 & 1 & 0 & 2 \\ 3 & 5 & 2 & 0 \end{bmatrix}$$

$a = [2 \ 3 \ 12 \ 8]^T$ $d_0 = r - a = [1 \ 11 \ 0 \ 4]^T$

a) 1. $d_0 \geq N(P) \rightarrow$ obavljen proces P $d_1 = d_0 + A(P) = [2 \ 11 \ 1 \ 6]^T$

2. $d_1 \geq N(R) \rightarrow$ obavljen proces R $d_2 = d_1 + A(R) = [2 \ 13 \ 6 \ 10]^T$

3. $d_2 \geq N(Q) \rightarrow$ obavljen proces Q $d_3 = d_2 + A(Q) = [3 \ 13 \ 9 \ 10]^T$

4. $d_3 \geq N(S) \rightarrow$ obavljen proces S $d_4 = d_3 + A(S) = [3 \ 14 \ 12 \ 12]^T$

Stanje je sigurno po bankarevom algoritmu.

b) Ukoliko uklonimo proces i oslobodimo njegove alocirane resurse, stanje ostaje sigurno. U originalnom stanju svi su procesi mogli završiti. Uklanjanjem jednog od njih raste vrijednost elemenata vektora d. Ako je prije vrijedilo $d \geq N(?)$, onda će to vrijediti i za $d' > d$. (?) označava neki proces.