# Zadanie 1

# 

# Zadanie 2

1. Czy jest możliwe, żeby kiedyś każdy klient był jednocześnie w stanie połączenia z sesją oraz każda sesja była jednocześnie w stanie połączenia z klientem?  
   1. **UPPAAL**:   
      𝐸 <> 𝑓𝑜𝑟𝑎𝑙𝑙(𝑖: 𝑖𝑛𝑡[0, 𝑛 − 1]) (Client(𝑖).ConnectedWithSession 𝑎𝑛𝑑 Session(𝑖).ConnectedWithClient)
   2. **CTL**:   
      𝐸𝐹 ((∀𝑖, 𝑖 ≥ 0 ∧ 𝑖 ≤ 𝑛 − 1) (Client(𝑖).ConnectedWithSession ∧ Session(𝑖).ConnectedWithClient))
   3. **Wynik**: Prawda
2. Czy jest możliwe, aby kiedyś istniał taki klient, który jest w stanie połączenia z sesją i jego czas połączenia przekroczył t2, lub żeby istniała taka sesja, która jest w czasie połączenia z klientem i jej czas połączenia przekroczył t2?  
   1. **UPPAAL**:   
      𝐸 <> 𝑒𝑥𝑖𝑠𝑡𝑠(𝑖: 𝑖𝑛𝑡[0, 𝑛 − 1]) ((Client(𝑖).ConnectedWithSession 𝑎𝑛𝑑 Client(𝑖).sessionTime > 𝑡2) 𝑜𝑟 (Session(𝑖).ConnectedWithClient 𝑎𝑛𝑑 Session(𝑖).sessionTime > 𝑡2))
   2. **CTL**:  
      𝐸𝐹 (∃ (∀𝑖, 𝑖 ≥ 0 ∧ 𝑖 ≤ 𝑛 − 1)]) ((Client(𝑖).ConnectedWithSession ∧ Client(𝑖).sessionTime > 𝑡2) ∨ (Session(𝑖).ConnectedWithClient ∧ Session(𝑖).sessionTime > 𝑡2)))
   3. **Wynik**: Fałsz
3. Czy połączenia klient-sesja na pewno się zakończą?  
   1. **UPPAAL**:  
      (Client(0).ConnectedWithSession 𝑎𝑛𝑑 Session(0).ConnectedWithClient) − −> (Client(0).Idle 𝑎𝑛𝑑 Session(0).Idle)
   2. **CTL**:  
      𝐴𝐺 ((Client(0).ConnectedWithSession 𝑎𝑛𝑑 Session(0).ConnectedWithClient) ⇒AF (Client(0).Idle 𝑎𝑛𝑑 Session(0).Idle))
   3. **Wynik**: Prawda
4. Czy możliwe jest, żeby serwer był kiedyś w stanie obsługiwania wszystkich klientów na raz?  
   1. **UPPAAL**:   
      𝐸 <> 𝑓𝑜𝑟𝑎𝑙𝑙(𝑖: 𝑖𝑛𝑡[0, 𝑛 − 1]) (Server.Busy 𝑎𝑛𝑑 Client(𝑖).WaitingForSession)
   2. **CTL**:   
      𝐸𝐹(∀𝑖, 𝑖 ≥ 0 ∧ 𝑖 ≤ 𝑛 − 1) ( Server.Busy ∧ Client(𝑖).WaitingForSession)
   3. **Wynik**: Fałsz
5. Czy na pewno zawsze na sesję może oczekiwać jednocześnie tylko jeden klient?  
   1. **UPPAAL**:   
      𝐴[] 𝑓𝑜𝑟𝑎𝑙𝑙(𝑖: 𝑖𝑛𝑡[0, 𝑛 − 1]) 𝑓𝑜𝑟𝑎𝑙𝑙(𝑗: 𝑖𝑛𝑡[0, 𝑛 − 1]) (Client(𝑖).WaitingForSession 𝑎𝑛𝑑 Client(𝑗).WaitingForSession ⇒ 𝑖 == 𝑗)
   2. **CTL**:   
      𝐴𝐺((∀𝑖, 𝑖 ≥ 0 ∧ 𝑖 ≤ 𝑛 − 1) (∀𝑗,𝑗 ≥ 0 ∧ 𝑗 ≤ 𝑛 − 1) (Client(𝑖).WaitingForSession 𝑎𝑛𝑑 Client(𝑗).WaitingForSession 𝑖 = 𝑗))
   3. **Wynik**: Prawda
6. Czy jest możliwe, żeby kiedyś któryś klient znalazł się w stanie bezczynności i jego czas połączenia był mniejszy od wybranego wcześniej t1, gdy t1 jest różne od 0?  
   1. **UPPAAL**:   
      𝐸 <> 𝑓𝑜𝑟𝑎𝑙𝑙(𝑖: 𝑖𝑛𝑡[0, 𝑛 − 1]) Client(𝑖).Idle 𝑎𝑛𝑑 (Client(𝑖).sessionTime < Client(𝑖).𝑡1) 𝑎𝑛𝑑 (Client(𝑖).𝑡1 ! = 0)
   2. **CTL**:   
      𝐸𝐹((∀𝑖, 𝑖 ≥ 0 ∧ 𝑖 ≤ 𝑛 − 1)Client(𝑖).Idle ∧ (Client(𝑖).sessionTime < Client(𝑖).𝑡1) ∧ (Client(𝑖).𝑡1 ! ≠ 0))
   3. **Wynik**: Fałsz