



Wrocław University
of Science and Technology

Modelowanie i Analiza Systemów Informatycznych

Laboratorium 11

Prowadzący: dr inż. Paweł Głuchowski
Termin laboratoria: Wtorek, 17:05 - 18:45

inż. Paweł SZYNAL 226026

Wydział Elektroniki
Kierunek: Informatyka

Wrocław 2021 r.

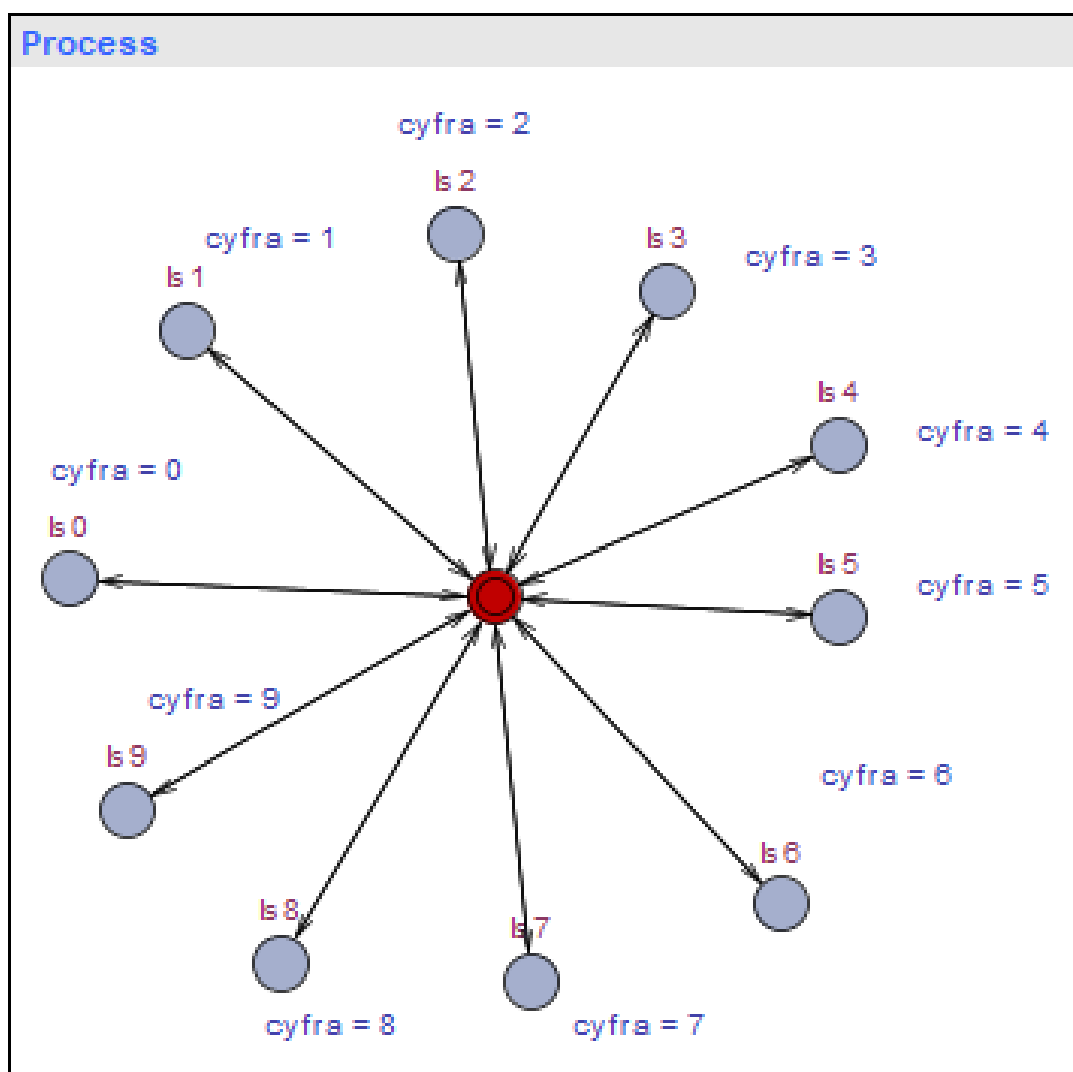
1 Cel laboratorium

Temat dziesiątego laboratorium z Modelowania i Analizy Systemów Informatycznych brzmiał ”: Logika Temporalna i Automaty Czasowe - konstrukcja prostych automatów UPPAAL.”.

2 Zadanie 2

Wykonaj automat bez końca losowo generujący cyfry z zakresu od 0 do 9 przez losowy wybór przejść. Automat generatora może pamiętać tylko ostatnią wylosowaną cyfrę, wprowadzając ją do globalnej zmiennej *cyfra*, zdefiniowanej dla wartości 0-9.

2.1 Rozwiązanie

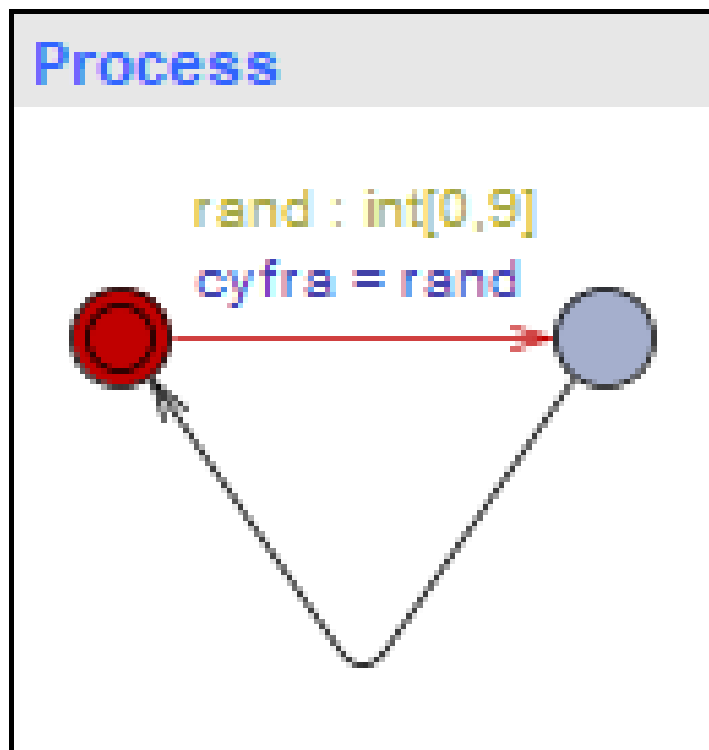


Rysunek 1: Rozwiązanie zadania 2

3 Zadanie 3

Wykonaj automat bez końca losowo JeneruMący cyfry z zakresu od 0 do 9 Przez losowy wybór wartości z tego zakresu. Automat Generators może pamiętać tylko ostatnią wylosowaną cyfrę, wprowadzając ją do globalnej zmiennej `cyfra`, zdefiniowanej dla wartości 0-9. W opisie przejść użyj etykiet `select` i `update`; innych nie używaj.

3.1 Rozwiązanie

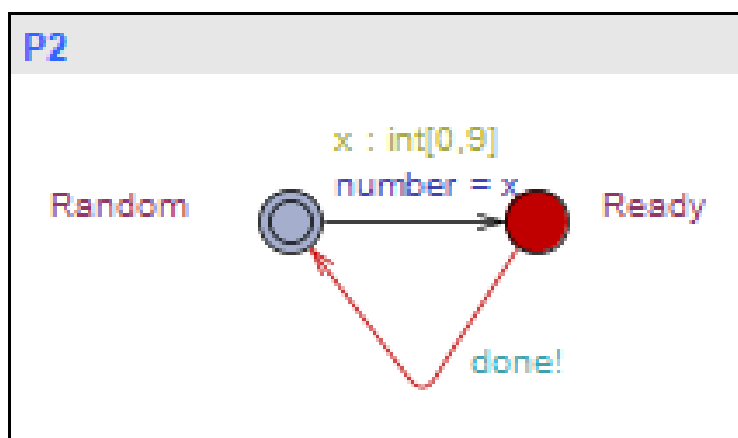


Rysunek 2: Rozwiązanie zadania 3

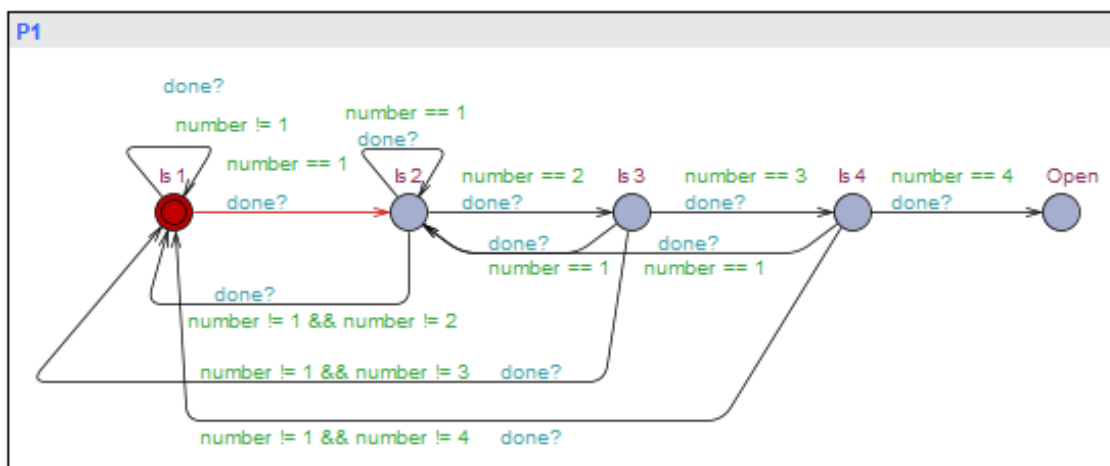
4 Zadanie 4

Wykonaj układ dwóch automatów losowego Generatora cyfr i zamka cyfrowego. Jako Generator wykorzystaj automat wykonany w zadaniu 1 lub 2, uzupełniony o możliwości synchronizacji z automatem zamka przez kanał binarny. Automat zamka każdorazowo reaguje na wygenerowanie cyfry przez generator. Otrzymanie sekwencji cyfr o dowolnej długości, która kończy się 1234, spowoduje przejście automatu zamka w końcowy stan, oznaczający otwarcie zamka. Automat generatora może pamiętać tylko ostatnią wylosowaną cyfrę, wprowadzając ją do globalnej zmiennej cyfra, zdefiniowanej dla wartości 0-9. Zsynchronizuj automaty, przy pomocy kanału binarnego wylosowano, tak aby zamek działał tylko w reakcji na wylosowanie cyfry przez generator. Upewnij się przy pomocy symulacji, że każda sekwencja cyfr, kończąca się na 1234 otwiera zamek, w tym kończąca się 11234, 121234 itp. Nie programuj zamka żadną sekwencją cyfr. Ma być losowana przez generator.

4.1 Rozwiązanie



Rysunek 3: Rozwiązanie zadania 4



Rysunek 4: Rozwiązanie zadania 4

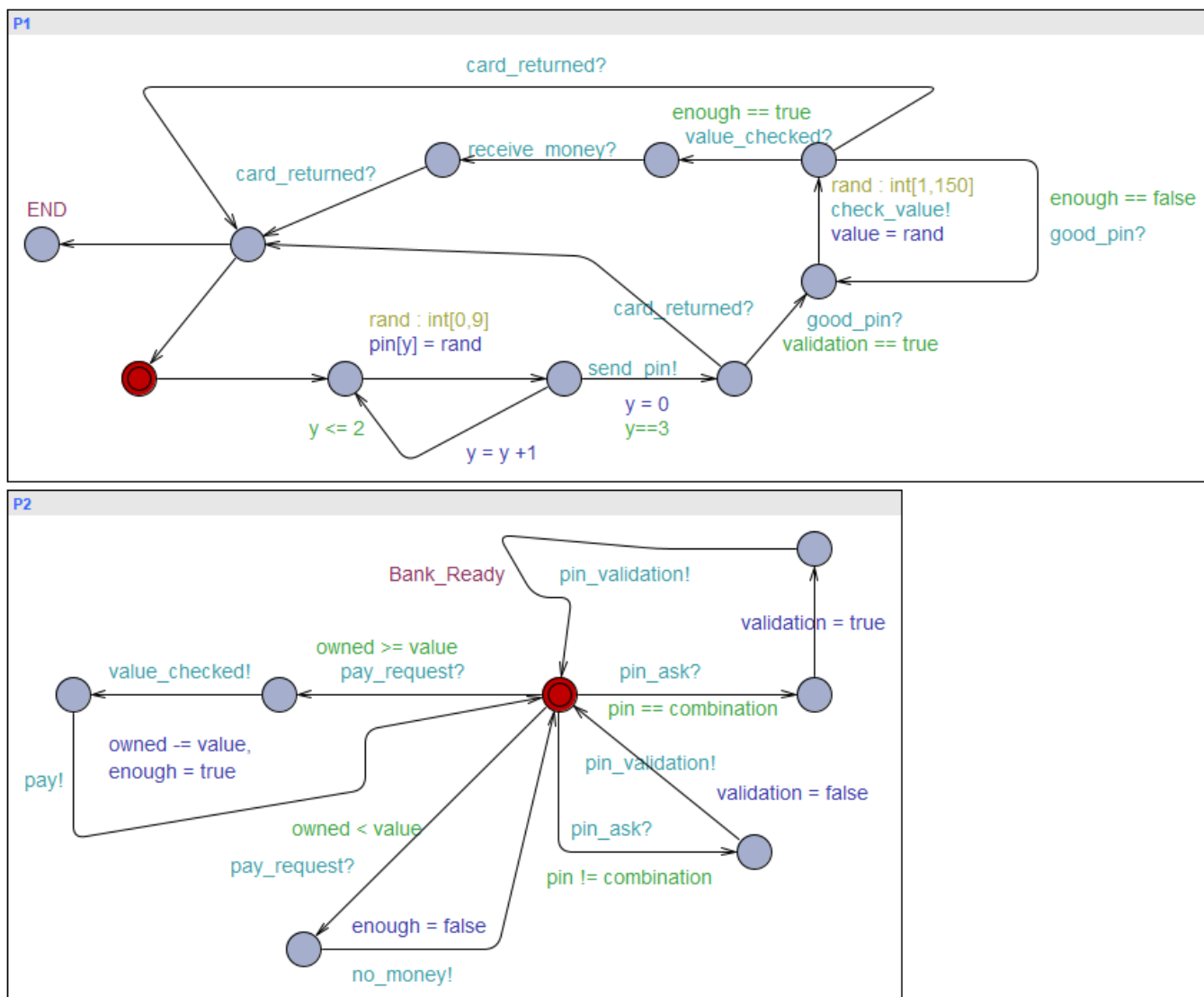
5 Zadanie 5

Wykonaj układ trzech automatów Klient, Bankomat i Bank, modelujących wypłatę pieniędzy z bankomatu, tak aby ich działanie było następujące:

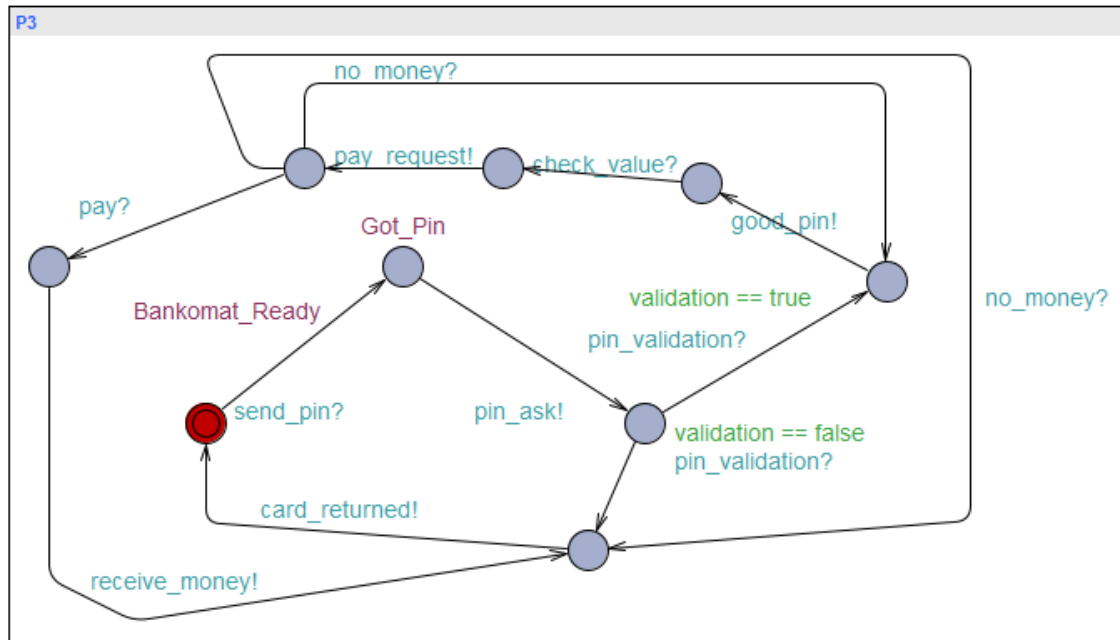
1. Klient wkłada kartę do bankomatu.
2. Klient podaje bankomatowi poprawny lub niepoprawny kod PIN.
3. Bankomat prosi bank o weryfikację kodu PIN.
4. Bank weryfikuje kod PIN.
5. Jeśli kod PIN jest niepoprawny, przejdź do 15.
6. Bank informuje bankomat o poprawności kodu PIN.
7. Klient podaje bankomatowi kwotę do wypłaty (losowa wartość z zakresu 1-150 zł).
8. Bankomat prosi bank o wypłatę podanej kwoty.
9. Bank sprawdza stan konta klienta.
10. Jeśli podana kwota jest większa niż stan konta klienta, przejdź do 16.
11. Bank zmniejsza konto klienta o podaną kwotę.
12. Bank poleca bankomatowi wypłacić podaną kwotę.
13. Bankomat wypłaca klientowi podaną kwotę.
14. Bankomat zwraca klientowi kartę. Przejdź do 1 lub koniec.
15. Bank informuje bankomat o niepoprawności kodu PIN. Przejdź do 14.
16. Bank informuje bankomat o braku pieniędzy. Przejdź do 7 lub 14 (losowo).

Przyjmij, że początkowo klient ma na koncie 100 zł. Zmienne liczbowe definiuj z zakresem ich możliwych wartości. Nazwij każdy stan tak, aby opisać to, co się dzieje w nim lub na przejściach z niego wychodzących. Zsynchronizuj zmienianie stanów przez automaty przy pomocy kanałów.

5.1 Rozwiązanie



Rysunek 5: Rozwiązanie zadania 5



Rysunek 6: Rozwiązanie zadania 5

```

// Place global declarations here.
int pin[4];
int combination[4] = {1,2,3,4};
int[0,9] cyfra;
bool validation = false;
bool enough = false;
int value = 0;
int y = 0;
chan send_pin, pin_ask, pay, no_money, pay_request, pin_validation, card_returned, good_pin, value_checked, check_value, receive_money;
int owned = 100;
  
```

Rysunek 7: Rozwiązanie zadania 5