

Zadanie 1

Symulator parkometru (alternatywnie – urządzenie do kawy).

Wykorzystując automat skończony (FA) zaproponuj program będący symulatorem parkometru.

Parkometr działa w sposób następujący :

1. Do urządzenia można wrzucać monety o nominałach : 1zł, 2zł i 5zł (10gr, 20gr i 50gr dla automatu z kawą.) – jest więc to alfabet FA, $\Sigma = \{1,2,5\}$.
 2. Monety wrzucane są pojedynczo – oznacza to, że nie można wrzucić naraz dwóch lub więcej monet.
 3. Zakładamy, że miejsce na parking kosztuje 7zł (kawa 70 gr.), jest więc to stan akceptowalny FA.
 4. Maszyna nie wydaje reszty.
 5. Jeśli suma wrzuconych monet jest większa niż 7zł (70gr) wtedy FA przechodzi do stanu nieakceptowanego – możemy założyć, iż nasza maszyna jest na tyle „głupia”, że gdy przekroczymy należną sumę, to zwraca ona wszystkie dotychczas wrzucone pieniądze i cały proces zaczynamy od nowa.
- Program powinien przyjmować z klawiatury kolejne nominały monet.
 - **Po każdorazowym wrzuceniu momenty program powinien wyświetlać aktualny stan automatu.**
 - Po zakończeniu etapu „wrzucania monet” program powinien wyświetlić aktualny stan FA, ścieżkę jaką pokonał (listę stanów).
 - Program w wersji konsolowej – na ocenę 4
 - Program w wersji z GUI - na ocenę 5
 - **Program musi wykorzystywać algorytm automatu skończonego, tzn. muszą istnieć zdefiniowane stany i przejścia pomiędzy nimi.**
 - Język programowania – dowolny – C/C++, JAVA, C# itp.
 - Oprócz programu należy dołączyć diagram przedstawiający zastosowany FA.

Zadanie 2

Wykorzystując niedeterministyczny automat skończony (NFA) opracuj program który będzie akceptował wszystkie ciągi, w których nastąpi dwukrotne powtórzenie znaku. NFA wykorzystuje następujący alfabet : $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$.

- Program powinien wczytywać z podanego pliku całą jego zawartość, oraz wskazywać wszystkie miejsca (np. pozycję pierwszego znaku ciągu), gdzie znajduje się akceptowany ciąg. Poszczególne ciągi można separować od siebie za pomocą terminatora – np. #.
- **Aktualny stan NFA powinien być wyświetlony na ekranie.**
- Program w wersji konsolowej – na ocenę 4
- Program w wersji z GUI - na ocenę 5
- Dodatkowym atutem będzie zaimplementowanie metod sprawdzających plik pod kątem poprawności (występujących w nim znaków) – tutaj też można skorzystać z NFA lub FA – wtedy na ocenę 5 wystarczy wersja konsolowa.
 - Podpowiedź – należy uwzględnić jakiś „szerszy alfabet, np. cyfry + litery”, oraz ustalić znak terminujący cały plik, np. znak EOF.
 - **Program musi wykorzystywać algorytm niedeterministycznego automatu skończonego, tzn. muszą istnieć zdefiniowane stany i przejścia pomiędzy nimi.**

- Język programowania – dowolny – C/C++, JAVA, C# itp.
- Oprócz programu należy dołączyć diagram przedstawiający zastosowany FA, w przypadku implementacji funkcji sprawdzającej poprawność pliku z wykorzystaniem NFA lub FA – należy dołączyć także diagram tego automatu.

Zadanie 3

Zmodyfikuj program z zadania pierwszego, tak aby wyeliminować jego wady (np. brak wydawania reszty, program przekroczenia należnej sumy itp.). Wykorzystaj w tym celu maszynę Turinga. Pozostałe wymagania tak jak w zadaniu 1.