

KURS PROGRAMOWANIA W C++

AUTOMATY SKOŃCZONE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie

Automat skończony FA możemy zdefiniować jako piątkę uporządkowaną $(Q, \Sigma, \delta, q_0, F)$, gdzie Q jest skończonym zbiorem stanów, Σ jest skończonym alfabetem wejściowym, $q_0 \in Q$ jest stanem początkowym, $F \subseteq Q$ jest zbiorem stanów końcowych, a $\delta : Q \times \Sigma \mapsto Q$ jest funkcją przejścia. Mówimy, że ciąg $x \in \Sigma^*$ jest akceptowany przez automat skończony $M = (Q, \Sigma, \delta, q_0, F)$, jeśli $\delta(q_0, x) \in F$.

Zdefiniuj klasę FA reprezentującą automat skończony. Obiekt klasy FA powinien działać dwufazowo: najpierw jest inicjalizowany (zbiór stanów, alfabet, funkcja przejść), a potem pracuje na strumieniu danych wejściowych. Jeśli automat będąc w jakimś stanie przeczyta symbol spoza alfabetu, albo symbol dla którego funkcja przejścia nie będzie zdefiniowana, to powinien zgłosić wyjątek (funkcja przejścia nie musi więc być zdefiniowana w całości). Rozbuduj swój automat o pamięć, tak aby przeczytane znaki były jakoś zapamiętywane (może w strumieniu `osstream`), oraz o dodatkową aktywność prześć, tak by w trakcie przechodzenia automatu od stanu do stanu były wykonywane jakieś akcje (opróżnienie bufora przeczytanych do tej pory znaków, wysłanie ich do innego strumienia, itp).

Następnie napisz programy, które będą potrafiły analizować programy napisane w języku C++ przy pomocy odpowiednio zainicjalizowanej klasy FA:

- program wypisujący treść programu w C++ z pominięciem wszystkich komentarzy (uwaga na literały napisowe, przykładowo: `"/*..."` czy `"//..."`);
- program wypisujący wszystkie literały zawarte w treści programu w C++: liczby całkowite, liczby rzeczywiste, stałe znakowe, łańcuchy znaków, wartości logiczne (uwaga na komentarze i literały postaci `j23` czy `k.true`).

Przyjmij, że dane są podawane na standardowe wejście, a wyniki zapisuj na standardowe wyjście.

Wskazówka

Wykorzystaj struktury danych i algorytmy zdefiniowane w STL.

Uwaga

Zakładamy, że programy analizowane przez automaty są napisane bezbłędnie.