Matematyczna wieża Babel.
4. Ograniczone maszyny Turinga – o językach kontekstowych
materiały do ćwiczeń

Projekt „Matematyka dla ciekawych świata”
spisał: Michał Korch

4 kwietnia 2019

1 Dodajmy kontekst!

Rozważaliśmy ostatnio gramatyki bezkontekstowe, które zawierały reguły w stylu:

$$A \to aAb$$

Oznaczało to, że zawsze, gdy wystąpi symbol A, możemy go zamienić na aAb. Wzbogaćmy naszą gramatykę w taki sposób, żeby móc opisywać kiedy można zastosować daną regułę. Nasze nowe reguły będą zawierały dodatkową rzecz, czyli kontekst. Na przykład:

$$aAa \to aaAba$$

Ta reguła oznacza, że owszem symbol A można zamienić na aAb, ale tylko jeśli stoi pomiędzy dwoma literami a.

Z przyczyn technicznych wprowadźmy jeszcze dwa ograniczenia: reguła w stylu $S \to -$ może wystąpić tylko w przypadku symbolu początkowego oraz symbolu początkowego nie wolno generować żadną regułą.

Rozważmy teraz następującą gramatykę:

$$S \to - \,|\, T$$
$$T \to aTBC \,|\, abC$$
$$CB \to PB$$
$$PB \to PC$$
$$PC \to BC$$
$$bB \to bb$$
$$bC \to bc$$
$$cC \to cc$$

W tej gramatyce można wygenerować na przykład słowo $aabbcc$. W następujący sposób:

$$\underline{S} \to \underline{T} \to a\underline{T}BC \to aab\underline{CB}C \to aab\underline{PB}C \to aab\underline{PC}C \to aa\underline{bB}CC \to aab\underline{bC}C \to aabb\underline{cC} \to aabbcc$$

Zaznaczone są fragmenty słowa odpowiadające stosowanym regułom.

Zadanie 1

Pokażcie, jak w tej gramatyce wygenerować słowo *aaaabbbbcccc*.

Zadanie 2

Oczywiście ta gramatyka opisuje język słów w których kolejno występują najpierw a, potem b, a na końcu c i liczba każdej z tych liter jest taka sama. Dlaczego tak jest?

Zauważ też, że blok

$$CB \to PB$$

$$PB \to PC$$

$$PC \to BC$$

sprowadza się do zamienienia miejscami symbolu B i C. Od tej pory zatem dla dwóch symboli roboczych X i Y, dla uproszczenia będziemy po prostu dopuszczać regułę

$$XY \to YX.$$

Zadanie 3

Zmodyfikujcie powyższą gramatykę w taki sposób, aby rozpoznawała język słów, w których występują najpierw a, potem b, a na końcu c oraz liczba liter b jest sumą liczby liter a i c.

Zadanie 4

Skonstruujcie gramatykę kontekstową generującą język wszystkich słów nad alfabetem $\{a, b\}$ postaci wvw, gdzie w jest pewnym słowem, a v jest słowem w napisanym od końca.

2 Gramatyki rosnące

Jest jeszcze jeden typ gramatyki, z jeszcze mniejszą liczbą ograniczeń. Są to gramatyki rosnące. Najważniejsza zasada jest po prostu taka, że w każdej regule przed strzałką jest nie więcej symboli i liter niż za strzałką. Jedynym dopuszczalnym wyjątkiem jest reguła $S \to -$, z tym że symbolu S nie wolno generować.

Oto przykład gramatyki rosnącej

$$S \to - \,|\, abc \,|\, aTBc$$

$$T \to aTBc \,|\, abc$$

$$cB \to Bc$$

$$bB \to bb$$

Zadanie 5

Pokażcie przykłady trzech słów, które można wygenerować przy pomocy tej gramatyki.

Zadanie 6

Zaprojektujcie gramatykę rosnącą, która generuje język wszystkich słów nad alfabetem a, które są długości będącej potęgą 2.
Wskazówka: Wskazówki do tego zadania umieszczamy na końcu skryptu!

Okazuje się, że dla każdej gramatyki rosnącej można stworzyć gramatykę kontekstową, która jest równoważna!

Zadanie 7

Zastanówcie się jak to zrobić i napiszcie gramatykę kontekstową, która jest równoważna gramatyce napisanej na początku tego rozdziału.

Zadanie 8

Spróbujcie zrobić to samo z gramatyką wymyśloną w zadaniu 6.
Wskazówka: Wskazówki do tego zadania umieszczamy na końcu skryptu!

Ograniczone maszyny Turinga

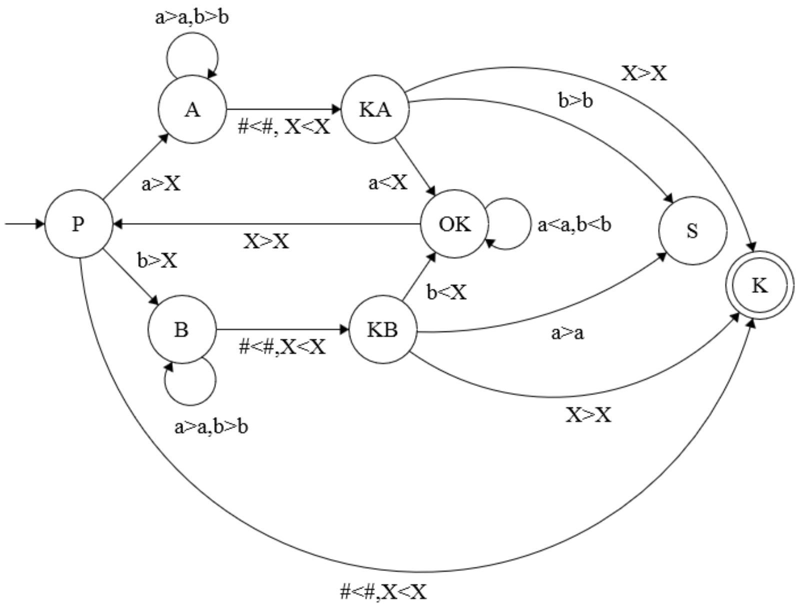
Okazuje się, że wszystko, co można wygenerować za pomocą gramatyki kontekstowej, można też rozpoznać za pomocą dość złożonego automatu nazywanego (ograniczoną) maszyną Turinga.

Maszyna Turinga, podobnie jak automat, ma pewną skończoną liczbę stanów oraz przejścia pomiędzy nimi. Ma też (podobnie jak automat ze stosem) pamięć, ale tym razem nie będzie to stos, a „taśma". Nie jest więc tak, że można odczytać tylko najświeższy zapisany symbol (tak było ze stosem). Po prostu będziemy sobie wyobrażać, że maszyna Turinga ma „głowicę", która wskazuje pewne pole na taśmie. Głowica ta może przesuwać się w prawo lub w lewo.

Nie będzie nam potrzebne w rozważaniach osobne słowo, które maszyna ma zaakceptować lub odrzucić, – po prostu uznajmy, że jest ono zapisane na taśmie. Umówmy się też, że przed i za tym słowem na taśmie stoją symbole # (żeby można było łatwo sprawdzić, że słowo się skończyło). Przymiotnik ograniczony w nazwie ograniczona maszyna Turinga, oznacza, że taśma jest ograniczona – powiedzmy, że po prostu się kończy przed początkowym znakiem # i za końcowym #. W przyszłości będziemy rozważać maszyny Turinga, których taśmy są nieograniczone (a więc, które mają nieograniczenie wiele miejsca na dodatkowe robocze „notatki").

Przejścia pomiędzy stanami będą więc opisane na przykład tak: $a < b$ – co oznacza, że maszyna może wykonać dane przejście, o ile na taśmie (w miejscu głowicy) widzi literę a. Zapisuje wtedy na to miejsce literę b i przesuwa głowicę o jeden w lewo. Przejście opisane jako $c > c$ może być wykonane tylko jeśli na taśmie w miejscu głowicy jest symbol c. Maszyna wykonując to przejście pozostawia c na taśmie i przesuwa głowicę o jeden w prawo.

Zatem to jest przykład maszyny Turinga.



Prześledźmy działanie tej maszyny dla słowa aa. To oznacza, że początkowy stan taśmy to $\#\bar{a}a\#$ (kreską na górze będziemy oznaczać, gdzie jest głowica). Działanie maszyny przebiega tak:

1. Stan P, przejście $a > X$ do stanu A, więc na taśmie mamy $\#X\bar{a}\#$.
2. Stan A, przejście $a > a$ do stanu A, więc na taśmie mamy $\#Xa\bar{\#}$.
3. Stan A, przejście $\# < \#$ do stanu KA, więc na taśmie mamy $\#X\bar{a}\#$.
4. Stan KA, przejście $a < X$ do stanu OK, więc na taśmie mamy $\#\bar{X}X\#$.
5. Stan OK, przejście $X > X$ do stanu P, więc na taśmie mamy $\#X\bar{X}\#$.
6. Stan P, przejście $X < X$ do stanu K i zaakceptowanie.

Zadanie 9

Prześledźcie działanie tej maszyny Turinga krok po kroku dla słów bab, ab oraz $abaaba$. Które z nich zostaną zaakceptowane?

Zadanie 10

Jaki język akceptuje ta maszyna Turinga? Dlaczego?

Zadanie 11

Zmodyfikujcie rozpatrywaną maszynę Turinga tak, aby akceptowała tylko te słowa, które są akceptowane przez oryginalną maszynę oraz mają parzystą liczbę liter a.

Zadanie 12

Stwórzcie ograniczoną maszynę Turinga, która akceptuje język nad alfabetem $\{a, b\}$ złożony ze słów „jąkających się”, czyli postaci ww, gdzie w jest dowolnym niepustym słowem.
Wskazówka: Wskazówka do tego zadania znajduje się na końcu skryptu.

3 Zadania dodatkowe

Zadanie 13

Rozważcie gramatykę

$$
\begin{gathered}
S \to - \,|\, aa \,|\, bb \,|\, aYT \,|\, bZT \\
T \to AYT \,|\, BZT \,|\, Aa \,|\, Bb \\
YA \to AY \\
YB \to BY \\
ZA \to AZ \\
ZB \to BZ \\
aA \to aa \\
aB \to ab \\
bA \to ba \\
bB \to bb \\
Ya \to aa \\
Yb \to ab \\
Za \to ba \\
Zb \to bb
\end{gathered}
$$

W jaki sposób w tej gramatyce wygenerować słowo $aabaaaba$?

Zadanie 14

Powyższa gramatyka generuje wszystkie słowa będące konkatenacją dwóch takich samych słów. Przeróbcie ją w taki sposób, żeby generowała słowa będące konkatenacją trzech takich samych słów.
Wskazówka: Bądźcie bardzo ostrożni przy ostatecznym zamienianiu symboli roboczych na litery. Musicie nie dopuścić do tego, żeby mogły się zamienić zanim ustawią się w dobrej kolejności!

Zadanie 15

Sprawdźcie, że wymyślona wyżej gramatyka rzeczywiście działa. To znaczy, że nie uda się wygenerować słowa nie spełniającego założonych warunków.

Zadanie 16

Wymyślcie i opiszcie ogólną procedurę pozwalającą zamienić dowolną gramatykę rosnącą na gramatykę bezkontekstową.

Zadanie 17

Prześledźcie działanie maszyny Turinga skonstruowanej w zadaniu 12 dla słów aa, aba oraz $abab$.

Zadanie 18

Udowodnijcie, że jeśli istnieje maszyna Turinga, która zawsze kończy działanie i akceptuje pewien język L nad alfabetem A, to również istnieje maszyna Turinga, która akceptuje język nad alfabetem A złożony ze wszystkich słów, które nie są w L.

4 Zadania domowe

Zadanie 19 (1 punkt)

Przekształć gramatykę, z pierwszej strony tego skryptu w taki sposób, aby generowała słowa, w których litery a i b mogą być dowolnie wymieszane, ale występują przed blokiem wszystkich liter c oraz liter a jest tyle samo co liter b oraz tyle samo, co liter c. Następnie przedstaw jak w tej gramatyce wygenerować słowo $abbacc$.

Zadanie 20 (2 punkty)

Znajdź gramatykę rosnącą która generuje język wszystkich słów nad alfabetem $\{a\}$ które mają długość $2^i 3^j$ dla pewnych $i, j \in \mathbb{N}$.
Wskazówka: Dobrym punktem wyjścia jest gramatyka stworzona w zadaniu 6.

Zadanie 21 (3 punkty)

Skonstruuj maszynę Turinga, która akceptuje język słów postaci $a \dots a$ długości będącej liczbą pierwszą.

Wskazówki

Zadanie 6

Wskazówka: Użyjcie specjalnych symboli na oznaczenie początku i końca słowa, a następnie zróbcie parę symboli, które będą podróżować pomiędzy nimi w tę i z powrotem (jeden idzie w prawo, zmienia się na drugi i wraca i znów zamienia się w ten pierwszy), i idąc w prawo podwaja każdą literę. Trzeba jeszcze odpowiednio dopracować zakończenie tej procedury.

Zadanie 8

Wskazówka: Problematyczne są:

$$XK \to ZAAA$$

oraz

$$PZ \to AAA$$

Trzeba je rozbić na kilka prostszych kontekstowych przejść z użyciem dodatkowych specjalnie wprowadzonych symboli.

Zadanie 12

Wskazówka: Najpierw znajdźcie środek słowa zapisanego na taśmie i oznaczcie go specjalnym symbolem (np. X lub Y w zależności od tego, czy to litera a, czy b). Środek ten natomiast możecie znaleźć stosując nieco modyfikowaną procedurę w porównaniu z tą, którą zastosowaliśmy w palindromach.