Metody programowania 2017

Lista zadań nr 6

Na zajęcia 4 i 5 kwietnia 2017

Zadanie 1 (1 pkt). *Kolekcje* to struktury danych służące do przechowywania elementów. Niech interfejs kolekcji składa się z następujących nazw predyktów:

- `put(+E,+S,-R)` wstawia element `E` do kolekcji `S` i zwraca nową kolekcję `R`;
- `get(+S,-E,-R)` usuwa element z kolekcji `S`, podstawia go pod `E` i zwraca nową kolekcję `R`;
- `empty(?S)` sprawdza lub tworzy pustą kolekcję;
- `addall(-E, +G, +S, -R)` wstawia wszystkie wyniki podstawień pod zmienną `E`, które spełniają cel `G` (w którym zmienna `E` występuje) do kolekcji `S` i zwraca nową kolekcję `R` (ten predykat przypomina standardowe predykaty `findall/3` i `findall/4`).

Podaj dwie implementacje tego interfejsu, jedną dla stosu (użyj list zamkniętych do reprezentowania kolekcji), drugą dla kolejek FIFO (tu użyj list różnicowych).

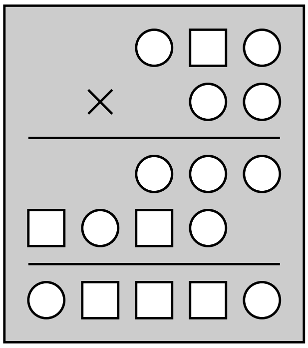
Zadanie 2 (1 pkt). Rozważmy skierowany graf $G = \langle V, E\rangle$. Jego wierzchołki ze zbioru V będziemy reprezentować w postaci atomów i liczb, a relację krawędzi E — za pomocą binarnego predykatu `e/2`. Dane są wierzchołki $v_1, v_2 \in V$. Ścieżkę z wierzchołka v_1 do wierzchołka v_2 w grafie G można znaleźć używając algorytmu przeszukiwania sparametryzowanego kolekcją przechowującą oczekujące wierzchołki:

1. Jeśli kolekcja przechowująca oczekujące wierzchołki jest pusta, to zakończ pracę.
2. W przeciwnym razie wyjmij wierzchołek v z kolekcji. Jeśli v nie został wcześniej odwiedzony, to odwiedź go i wstaw wszystkich jego e-sąsiadów do kolekcji. W przeciwnym razie odrzuć wierzchołek v. Powtórz całą procedurę.

Zaprogramuj powyższy algorytm tak, by znajdował w grafie ścieżki między podanymi wierzchołkami używając interfejsu kolekcji z poprzedniego zadania. Następnie użyj stosów, by otrzymać DFS oraz kolejek, by otrzymać BFS.

Przypuśćmy, że wierzchołkom grafu dodajemy *wagi* i używamy kolejek priorytetowych do przechowywania oczekujących wierzchołków. Zbadaj zachowanie takiego algorytmu.

Zadanie 3 (1 pkt). W tym zadaniu będziemy zakładać, że drzewa poszukiwań o etykietowanych wierzchołkach wewnętrznych nie zawierają dwóch równych etykiet. Zaprogramuj następujące predykaty: `insert/3` — wstawiający element do drzewa (bez powtórzeń), `find/2` — sprawdzający, czy element znajduje się w drzewie, `findMax/2` — ujawniający największy element w drzewie, `delMax/3` — ujawniający i usuwający największy element z drzewa, `delete/3` — usuwający podany element z drzewa oraz `empty/1` — sprawdzający, czy drzewo jest puste.



Rysunek 1: Łamigłówka z „Wiedzy i Życia”

Zadanie 4 (1 pkt). Napisz w Prologu program rozwiązujący zadania takie jak to, które pochodzi z numeru 11/2000 miesięcznika „Wiedza i Życie”: *Wpisz w kwadraty* [planszy przedstawionej na Rysunku 1] *cyfry parzyste (0, 2, 4, 6, 8), w koła zaś nieparzyste (1, 3, 5, 7, 9) tak, by otrzymać poprawny zapis mnożenia.*

Dane do programu (opis problemu) należy podać w postaci listy list atomów `s` i `c` reprezentujących kwadraty i koła. Np. dane do problemu z obrazka są następujące:

```
[[c,s,c], [c,c], [c,c,c], [s,c,s,c], [c,s,s,s,c]]
```

Zadanie 5 (1 pkt). Jaki język generuje gramatyka $G = \langle \Sigma, V, S, P\rangle$, gdzie $\Sigma = \{0,1\}$, $V = \{S\}$, zaś $P = \{S \to 0S1, S \to 0S, S \to \epsilon\}$? Wykaż, że jest ona niejednoznaczna. Zdefiniuj jednoznaczną gramatykę opisującą ten sam język. Udowodnij, że Twoja gramatyka jest jednoznaczna i że generuje ten sam język.

Zadanie 6 (1 pkt). Napisz gramatyki bezkontekstowe opisujące następujące języki nad alfabetem $\Sigma = \{0,1\}$:

1. $\{0^n 1^m 0^n \mid n, m \in \mathbb{N}\}$;
2. $\{0^n 1^n 0^m \mid n, m \in \mathbb{N}\}$;
3. $\{0^n 1^m 0^k \mid n, m, k \in \mathbb{N}, n \le m\}$;
4. $\{(01)^n 0^{2n} \mid n \in \mathbb{N}\}$;
5. zbiór ciągów zerojedynkowych, w których liczba zer i jedynek jest taka sama;
6. zbiór ciągów zerojedynkowych zawierających dwukrotnie więcej zer niż jedynek.

Zadanie 7 (1 pkt). Gramatyka bezkontekstowa jest *prawostronnie liniowa*, jeżeli wszystkie jej produkcje są postaci $A \to wB$ lub $A \to w$, gdzie $w \in \Sigma^*$ oraz $A, B \in V$.

Napisz gramatyki prawostronnie liniowe opisujące następujące języki nad alfabetem $\Sigma = \{0,1\}$:

1. $\{0^{2n} \mid n \in \mathbb{N}\}$;
2. $\{0^n 1^m \mid n, m \in \mathbb{N}\}$;
3. zbiór ciągów zerojedynkowych, które nie zawierają trzech kolejnych jedynek;
4. zbiór ciągów zerojedynkowych, w których liczba zer jest parzysta, a jedynek — dowolna;
5. zbiór ciągów zerojedynkowych, w których liczba zer jest parzysta, a jedynek — nieparzysta;
6. zbiór ciągów zerojedynkowych, w których różnica liczby zer i jedynek jest parzysta.

Czy można napisać gramatyki bezkontekstowe posiadające prostsze zbiory produkcji i opisujące powyższe języki?