Декларативно програмиране в процедурните езици

NP-пълни задачи

Владислав Ненчев

Софийски Университет “Св. Климент Охридски”
Факултет по Математика и Информатика
Катедра по Математическа Логика и Приложенията ѝ

6 Януари 2017

Подмножество с нулева сума

Условие на задачата:
Дадено е (мулти-)множество от цели числа. Да се определи, дали то има непразно подмножество със сума 0.

Подмножество с нулева сума

Условие на задачата:
Дадено е (мулти-)множество от цели числа. Да се определи, дали то има непразно подмножество със сума 0.

"Наивно" решение:

```
def subsets(set_):
  yield from chain.from_iterable(
    combinations(set_, size)
      for size in range(len(set_) + 1))

def naiveSolution(set_):
  return any(map(lambda subset:
    len(subset) > 0 and sum(subset) == 0,
      subsets(set_)))
```

Псевдо-полиномиално решение

Input: $S = \{\, x_1, x_2, \ldots, x_N \,\}$

Псевдо-полиномиално решение

$$\textbf{Input:}\ S = \{\ x_1, x_2, \ldots, x_N\ \}$$

$$A = sum\{\ x \mid x \in S\ \&\ x < 0\ \}$$
$$B = sum\{\ x \mid x \in S\ \&\ x > 0\ \}$$

$$Q(1, s) = (s == x_1) \qquad \textbf{for}\ A \leq s \leq B$$

$$Q(i, s) = Q(i-1, s) \vee (s == x_i) \vee$$
$$(A \leq s - x_i \leq B\ \&\ Q(i-1, s - x_i)) \qquad \textbf{for}\ 2 \leq i \leq N,\ A \leq s \leq B$$

$$\textbf{return}\ Q(N, 0)$$

Реализация

```
def dynamicSolution(set_):
  A = sum(filter(lambda n: n < 0, set_))
  B = sum(filter(lambda n: n > 0, set_))
  return Q(list(set_), len(set_) - 1, A, B)[0]

def Q(set_, N, A, B):
  if N == 0:
    return dict(map(lambda number:
      (number, number == set_[N]), range(A, B + 1)))
  else:
    previousQ = Q(set_, N - 1, A, B)
    return dict(map(lambda number:
      (number, previousQ[number] or
        number == set_[N] or
        (A <= number - set_[N] and
          number - set_[N] <= B and
          previousQ[number - set_[N]])),
      range(A, B + 1)))
```

Курсова задача IV - примери

Задача:
Да се реализира първата стъпка (премахване на ϵ-правилата) от алгоритъма за привеждане на контекстно-свободна граматика в Нормална Форма на Чомски. Граматиката е представена с Map обект между низове съдържащ правилата, множество от символите от азбуката, множество от нетерминалите, като е указано кой е началният.
Вход: Контекстно-свободна граматика.
Изход: Контекстно-свободната граматика след прилагането на първата стъпка.

Задача:
Да се реализира GSAT алгоритъма за проверка, дали дадена съждителна/булева формула е изпълнима.
Вход: Съждителна/булева формула.
Изход: True/False.

Курсова задача IV - примери (продължение)

Задача:
Да се реализира алгоритъма на Улман (Ulman) за проверка дали даден граф G има подграф, който е изоморфен на даден граф H.
Вход: Ориентирани графи G и H.
Изход: True/False.

Задача:
Да се минимизира даден детерминиран автомат по алгоритъма на Хопкрофт (Hopcroft). Автомата е представен с Map обект за преходите и множество на състоянията (като е отбелязано кое е финално и кое е начално).
Вход: Детерминиран автомат.
Изход: Минимизиран автомат.

Курсова задача IV - примери (продължение)

Задача:
Да се реализира алгоритъма на Таржан (Tarjan) за намиране на силно свързаните компоненти на даден ориентиран граф.
Вход: Ориентиран граф.
Изход (на екран): Всички силно свързани компоненти в графа.

Задача:
Да се реализира алгоритъма на Дийкстра (Dijkstra) - варианта с приоритетна опашка - за намиране на най-късите пътища от даден връх в граф.
Вход: Ориентиран граф с тежести на ребрата и начален връх.
Изход (на екран): Най-късите пътища от началния връх до всички останали.

Курсова задача IV - примери (продължение)

Задача:
Да се реализира псевдо-полиномиалното решение на задачата за раницата.
Вход: Множество от n предмета със съответни обем и стойност и максимален обем на раницата.
Изход: Максималното по обща стойност решение (подмножество на предметите) на задачата.

Задача:
Да се реализира алгоритъма на Хелд (Held) и Карп (Karp) за решаване на задачата за търговския пътник.
Вход: Неориентиран граф с тежести на ребрата.
Изход (на екран): Оптимално решение на задачата.

Курсова задача IV - примери (продължение)

Задача:
Да се реализира алгоритъма за намиране на най-общ унификатор на множество от атомарни формули и да се реализира операцията за намиране на резолюция (чрез най-общата унификация) на два дизюнкта.
Вход: Два дизюнкта.
Изход: Резолюцията (чрез най-общата унификация) да дадените дизюнкти.

Задача:
Да се реализира ситото на Аткин (Atkin) за намиране на простите числа до определена граница.
Вход: Горна граница за простите числа.
Изход: Всички прости числа до тази граница.

Курсова задача IV - примери (продължение)

Задача:
Да се реализира алгоритъма на Едмъндс (Edmonds) и Карп (Karp) за намиране на максимален поток в поточна мрежа.
Вход: Поточна мрежа (ориентиран граф с начален и краен връх и капацитетна функция на ребрата).
Изход (на екран): Максималния поток в мрежата.

Задача:
Да се реализира алгоритъма AC-3 за намиране на решение на система от условия (constraint satisfaction problem).
Вход: Множество от променливи, позволени стойности за всяка променлива и система от унарни и бинарни условия за променливите.
Изход: True/False.