Динамическое транзитивное замыкание

Салью Артур
371 группа

Задача 1 (a)

На лекции мы научились поддерживать инкрементальное замыкание ориентированных графов. Придумайте алгоритм для декрементального транзитивного замыкания, работающий за $O(n^2(m+n))$ суммарно на все апдейты.

Решение. Для решения задачи будем использовать вспомогательный матрицу смежности G^R, полученную инвертированием направлений дуг в исходном графе. Как и в инкрементальной версии, на стадии предпроцессинга заполним матрицу достижимости M.

Algorithm 1 Декрементальное обновление матрицы достижимости

1: **procedure** REMOVE(i, j)
2:     **if** $G[i, j] = 1$ **then**
      $G[i, j] = 0$
      $G^R[j, i] = 0$
3:     $is\_reachable \leftarrow dfs(i, G)$
4:     **for** $u : is\_reachable[u] = 0 \ \& \ M[j, u] = 1$ **do**
      $is\_reachable^R \leftarrow dfs(u, G^R)$
5:         **for** $v \in V$ **do**
          $M[v, u] \leftarrow is\_reachable^R[v]$

Пояснение: В 4 строке мы проходимся по вершинам, достижимых из j до удаления ребра, но недостижимых из i. Для этого достаточно запустить dfs из вершины i. Информация о других вершинах нас не интересует, так как удаление ребра (i, j) не повляиет на вершины не достижимые изначально из i.

Реверснутый граф нужен нам для следующего утверждения: достижимость вершины x из y в реверснутом графе эквивалентна достижмости y из x в изначальном.

Тогда запустив dfs из вершины u в реверснутом графе мы получим информацию о достижимости вершины u из других вершин v в исходном графе.

Анализ сложности: Из-за декрементальности алгоритма зануленная ячейка матрицы M никогда не станет 1. Благодаря этому, за все апдейты число итераций в цикле не может превышать n^2. Оценка dfs известна и равна $O(n + m)$. Число итераций во внутреннем цикле не превышет n, и оно будет сокрыто в $O(\cdot)$. Таким образом, сложность алгоритма $O(n^2(n + m))$ суммарно на все апдейты.