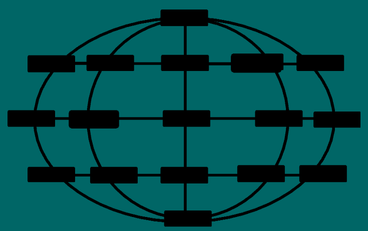
Методы построения и анализа алгоритмов



Лекция 5

Малышкин Виктор Эммануилович

Кафедра Параллельных Вычислительных Технологий
Новосибирский государственный технический университет

E_mail: malysh@ssd.sscc.ru

Телефон: 3308 994

Новосибирск

Алгоритмы на графах

Пусть задана ПВМ $\mathbf{C} = (\mathbf{X}, \mathbf{F})$, которая после трансляции представлена в виде двух таблиц TX и $ОП$. Каждая строка таблицы TX имеет вид $(x, A(x), \text{comp}(x))$, а таблицы $ОП$ —

$$(a, \text{in}(a), \text{out}(a)),\ x \in X,\ a \in \mathbf{F},\ \text{comp}(x) = \{a \in \mathbf{F} \mid x \in \text{out}(a)\},\ A(x) = \{a \in \mathbf{F} \mid x \in \text{in}(a)\}.$$

Чтобы не делать несущественных оговорок, предполагается, что $\text{in}(a) \neq \varnothing$ для любых $a \in \mathbf{F}$. Каждая строка TX содержит описание всех дуг, входящих и выходящих из переменной x. Алгоритм планирования состоит из двух частей: восходящей и нисходящей. В *восходящей* части алгоритма строятся множества переменных и операций, используемых в термах из множества $T_V = T(V, F)$.

Обозначим $V_0 = V$, тогда

$$F_0 = \{a \in \mathbf{F} \mid \text{in}(a) \subseteq V_0\} = \bigcup_{x \in V_0} \{a \in A(x) \mid \text{in}(a) \subseteq V_0\}$$

содержит все операции ПВМ такие, что $\text{in}(a) \subseteq V_0$. Далее формируется множество $V_1 = \{x \in \mathbf{X} \mid x \in \text{out}(a) \wedge a \in F_0\} \cup V_0$, на основе V_1 строится множество

$$F_1 = \bigcup_{x \in V_1 \setminus V_0} \{a \in A(x) \mid \text{in}(a) \subseteq V_1\}$$

и т. д. до тех пор, пока при некотором целом положительном k не окажется, что $F_k = \varnothing$. На этом завершается восходящая часть алгоритма планирования. Множества V_i и F_i, $i = 0, \ldots, k$, содержат все переменные и операции, используемые в термах из множества T_V.

Если $W \not\subseteq V_k$, то планирование можно прекращать, так как в этом случае существует переменная в W, которая не вычисляется никаким термом из множества T_V, и, следовательно, не существует алгоритма решения сформулированной задачи на основе имеющихся знаний о ПО. В противном случае можно начать строить множества переменных и операций, используемых в термах из T_V^W. Обозначим $F^* = \bigcup_{i=0} F_i$ и определим множества

$$G_1 = \bigcup_{x \in W} \{a \in F^* \mid a \in \mathrm{comp}(x)\}, \quad H_1 = \bigcup_{a \in G_1} \mathrm{in}(a),$$

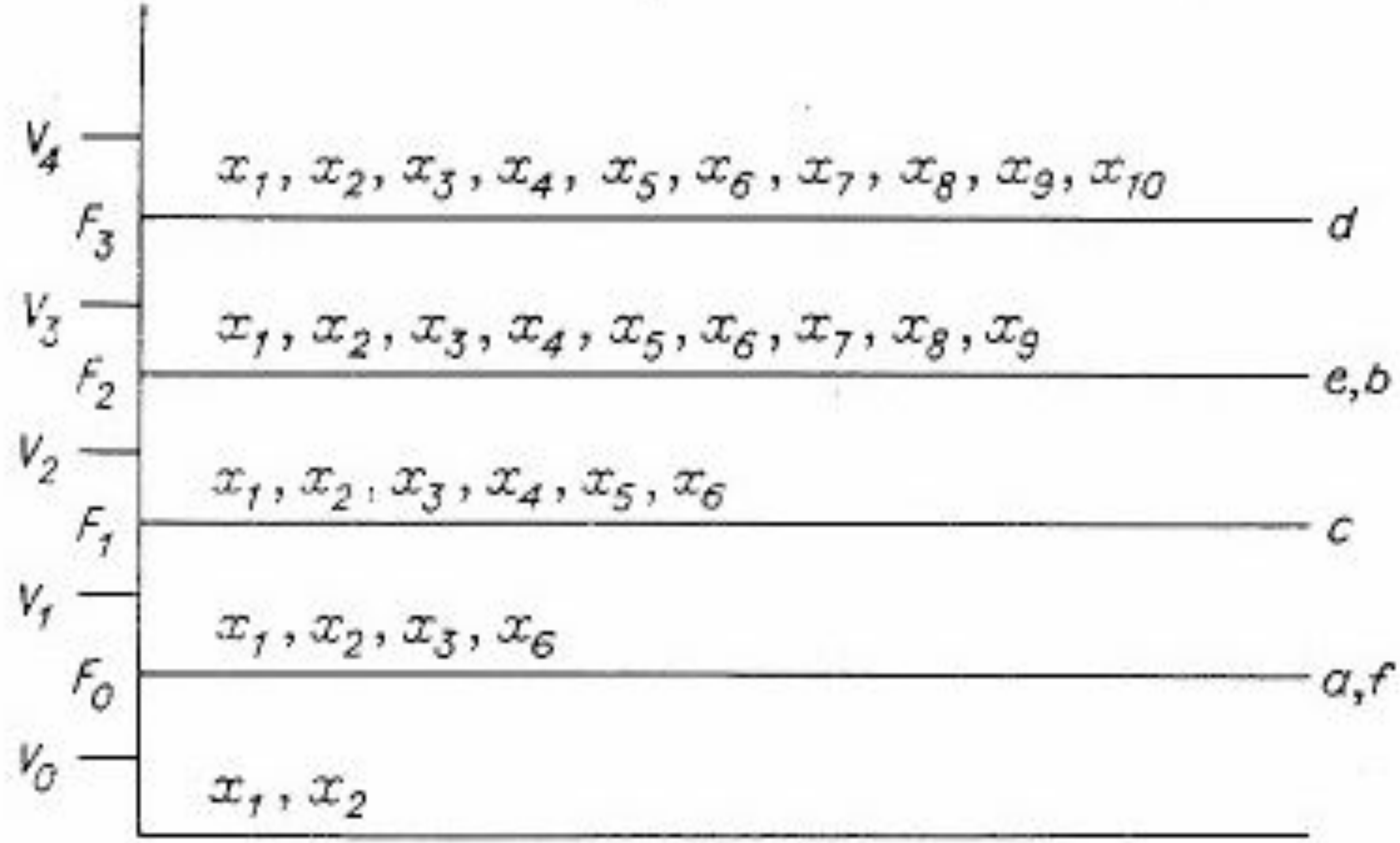


Рис. 2.6.

и далее, для $i = 2, 3, \ldots,$

$$G_i = \bigcup_{x \in H_{i-1}} \left\{ a \in F^* \mid a \in \mathrm{comp}(x) \wedge a \notin \bigcup_{m=1}^{i-1} G_m \right\}, \quad H_i = \bigcup_{a \in G_i} \mathrm{in}(a).$$

Построение множеств G_i и H_i завершается, когда при некотором целом положительном r окажется $G_r = \varnothing$. Множества H_i и G_i, $i = 1, \ldots, r$, содержат все переменные и операции, используемые в термах из множества T_V^W. Кроме того, в них остаются операции и переменные, необходимые для построения некоторых дублирующих вычислений. На рис. 2.6 показаны множества F_i и V_i, образовавшиеся в результате восходящей части алгоритма планирования на ПВМ (рис. 2.7) при $V = \{x_1, x_2\}$, $W = \{x_{10}\}$, а на рис. 2.8 — множества G_i и H_i, сформированные в нисходящей части алгоритма планирования. После завершения планирования в таблицах TX и $О\Pi$ остаются лишь переменные и операции из множеств H_i и G_i, остальные удаляются (рис. 2.9). Таким образом, результатом планирования

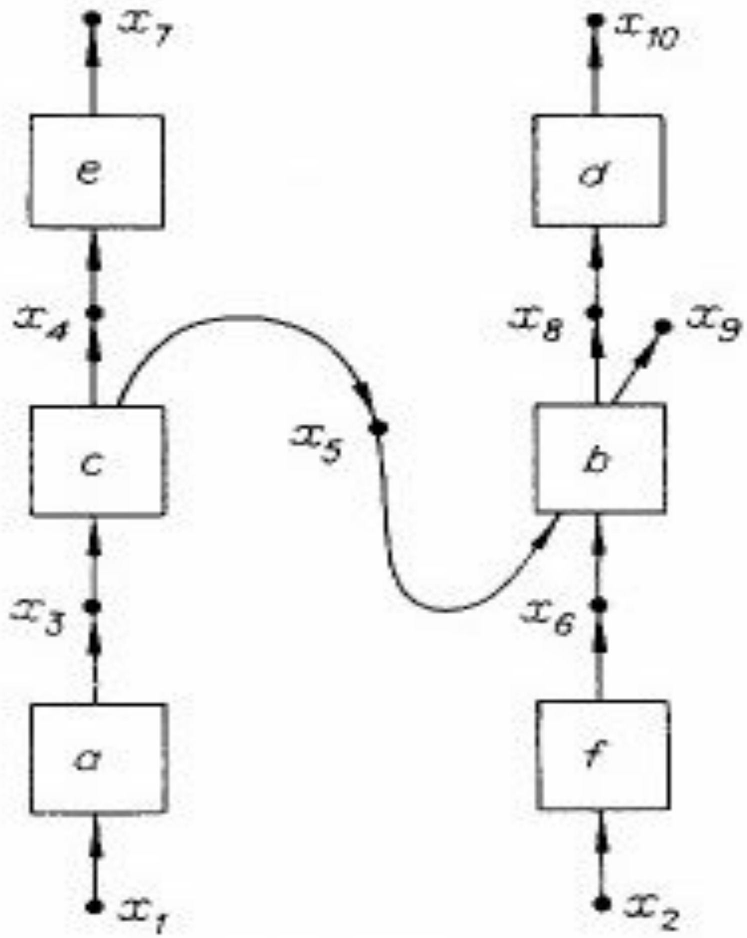


Рис. 2.7.

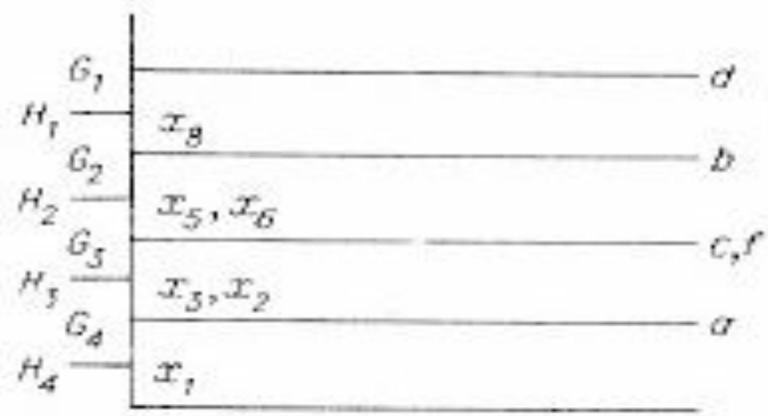


Рис. 2.8.

является ПВМ, оставшаяся от С после удаления из TX и $O\Pi$ «лишних» переменных и операций. Множество T_V^W не строится, подходящий в некотором смысле (V, W)-план T строится в каждом конкретном случае процедурой выбора алгоритма.

В случае, когда $W \nsubseteq V_k$, сформулированная задача оказывается неразрешимой и необходимо изменить формулировку задачи, т. е. либо уменьшить W, удалив из него невычислимые переменные, либо расширить V, включив в него такие новые переменные, что станут вычислимыми все переменные из W. Для уменьшения затрат на расширение V может быть использован алгоритм планирования. Для этого необходимо выполнить его *нисходящую* часть из множества переменных $W' = W \setminus V_k$ с использованием всех операций из F. Все переменные из построенных при этом множеств H_i, $i = 1, 2, \ldots, r$, являются кандидатами на включении в V. Из них человек может выбрать те, значения которых ему доступны. Нетрудно также построить человеко-машинные алгоритмы, помогающие сделать такой выбор.

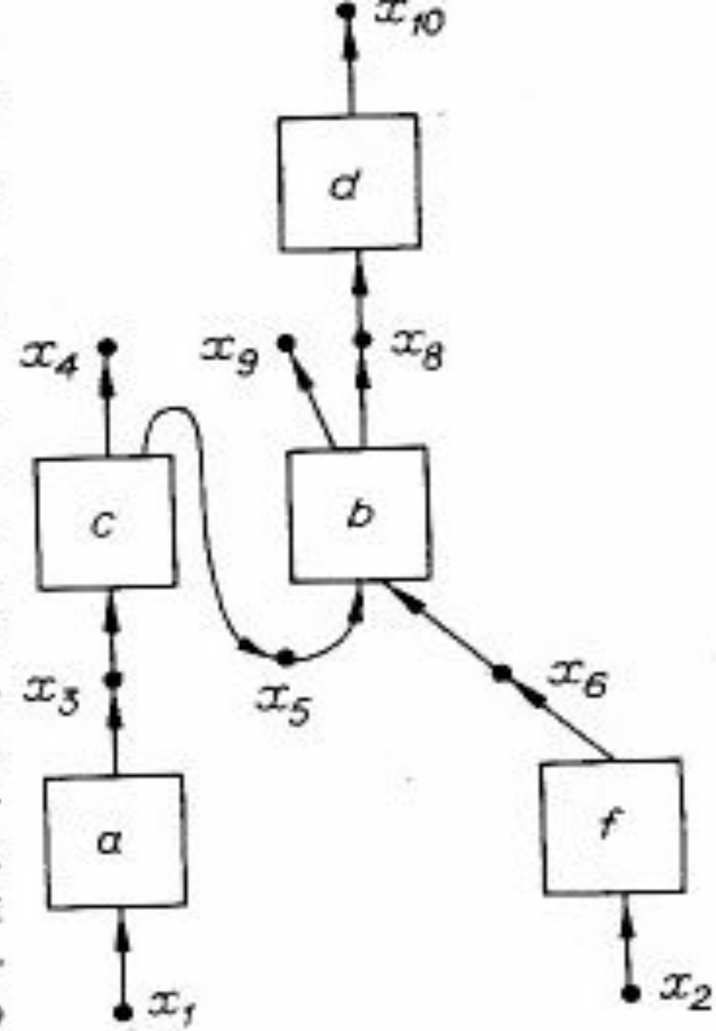


Рис. 2.9.

Из описания алгоритма следует, что проверка условия $\mathrm{in}\,(a) \subseteq V_i$ делается не более одного раза для каждой входной дуги произвольно взятой операции a, а проверка условия $\mathrm{out}\,(a) \cap H_{i-1} \neq \varnothing$ — не более одного раза для каждой выходной дуги a. Понятно, что алгоритм планирования имеет линейную относительно числа дуг в графическом представлении ПВМ временную сложность, если в качестве элементарных шагов алгоритма

Рекомендуемые учебники

- Ахо, Альфред, В., Хопкрофт, Джон, Ульман, Джеффри, Д. *Структуры данных и алгоритмы.* : Пер. с англ. : Уч. пос. — М. : Издательский дом "Вильяме", 2000. — 384 с.
- Кормен Т., Лейзерсон Ч., Риверс Р., Штайн К. *Алгоритмы. Построение и анализ* – М.: «Вильямс», 2012
- В.Э.Малышкин, В.Д.Корнеев. *Параллельное программирование мультикомпьютеров*. – В серии «Учебники НГТУ», Новосибирск, изд-во НГТУ, 2011, 296 стр. (есть в библиотеке)

ВОПРОСЫ

1. Что мы называем алгоритмом? Почему?
2. Сколько существует алгоритмов и программ, вычисляющих вычислимую функцию?
3. Задача, ее модель, алгоритм решения
4. Задача управления движением на перекрестке и ее модель
5. Три подхода к решению комбинаторной задачи
6. Задача раскраски графа. Жадный алгоритм раскраски графа
7. Абстрактные типы данных. Что такое?

ВОПРОСЫ

8.Что такое вычислительная сложность алгоритма?
9. Время работы алгоритма. От чего зависит? Верхняя оценка сложности.
10. Общая схема решения *переборных* задач .Какие алгоритмы называются эвристическими?
11. Задача/проблемы построения расписания
12, Формулировки задачи построения расписания.
13. Способы сокращения перебора.
14. Стратегии построения субоптимплльных расписаний