Методы поиска при сведении задач к совокупности подзадач.

Лекция 5.

Специальность : <u>230105</u>

Разрешимость и неразрешимость вершин.

Нахождение решающего графа основано на построении достаточно большой части “И/ИЛИ” графа, показывающей разрешимость начальной вершины.

<u>Определение.</u> Процедурами разметки разрешимых и неразрешимых вершин называются рекурсивные или итеративные процедуры, действующие на “И/ИЛИ” графе с целью отметить все разрешимые и неразрешимые вершины.

<u>Определение.</u> Перебор на “И/ИЛИ” графе считается закончившимся безуспешно, если начальная вершина может быть отмечена как неразрешимая.

<u>Определение.</u> В том случае, когда начальная вершина в конечном итоге может быть отмечена как разрешимая, решающим графом будет подграф (содержащий только разрешимые вершины), показывающий в соответствии с определением разрешимости (см. Лекцию 4) разрешимость начальной вершины.

Этапы перебора на “ИИЛИ” графах.

1) Начальная вершина ассоциируется с начальным описанием задачи.
2) Строятся множества дочерних вершин для начальной вершины с помощью тех операторов сведения задач к подзадачам, которые применимы. Обозначим как γ такой комбинированный оператор, применение которого дает все вершины, дочерние для данной, а сам процесс применения γ к вершине – раскрытием вершины.
3) От каждой дочерней вершине назад к ее родительским вершинам проводятся указатели. Эти указатели используются при попытке разметить разрешимые и неразрешимые вершины, а после окончания они дают решающий граф.
4) Процесс раскрытия вершин и установки указателей продолжается до тех пор, пока начальная вершина не может быть помечена как разрешимая или неразрешимая.

В процессе перебора порождается структура из вершин и указателей, представляющая собой “И/ИЛИ” граф. Этот граф будем в дальнейшем называть графом перебора.

Основные отличия процесса раскрытия вершин.

- Более сложными являются контроль окончания поиска и методы упорядочения вершин;
- Вместо поиска целевой вершины строится решающий граф;
- Уменьшается объем вычислений за счет за счет запоминания разрешимых из ранее раскрытых вершин;
- Ввиду отсутствия необходимости поиска более чем одного решения задачи имеет смысл отбросить на поисковом графе все неразрешимые вершины (включая следующие за ними), дочерние для разрешимых;

Взаимные различия методов перебора на “И/ИЛИ” графах.

Состоят в способе упорядочения вершин перед раскрытием.

- Полный перебор – вершины раскрываются в порядке построения;
- Перебор в глубину – в первую очередь раскрываются вершины, построенные последними;
- Методы упорядоченного перебора – для упорядочения вершин при их раскрытии используются оценочные функции

Опишем варианты указанных методов перебора для перебора на деревьях.

Алгоритм полного перебора.

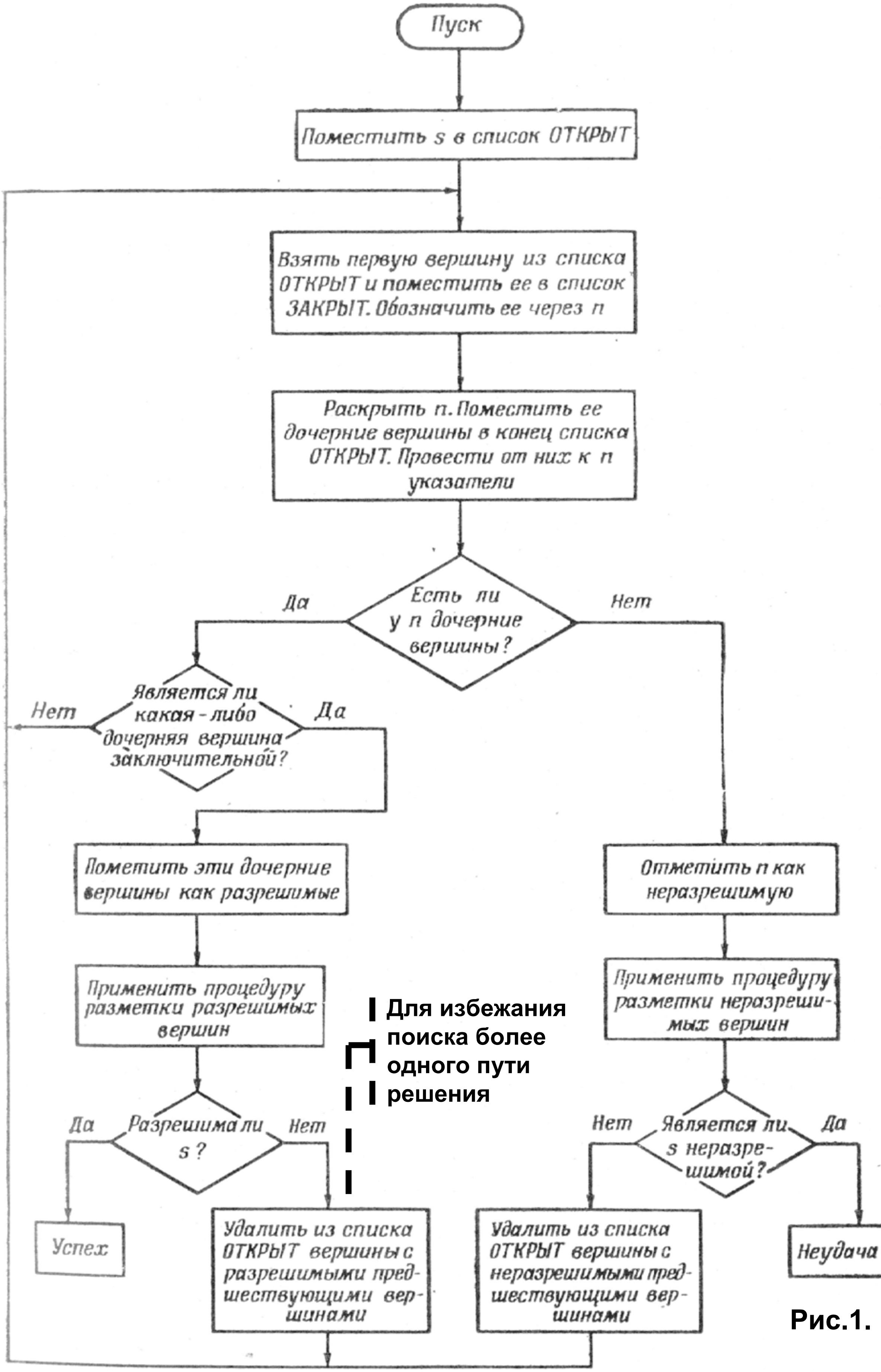

Рис.1.

Пример последовательности раскрытия вершин.

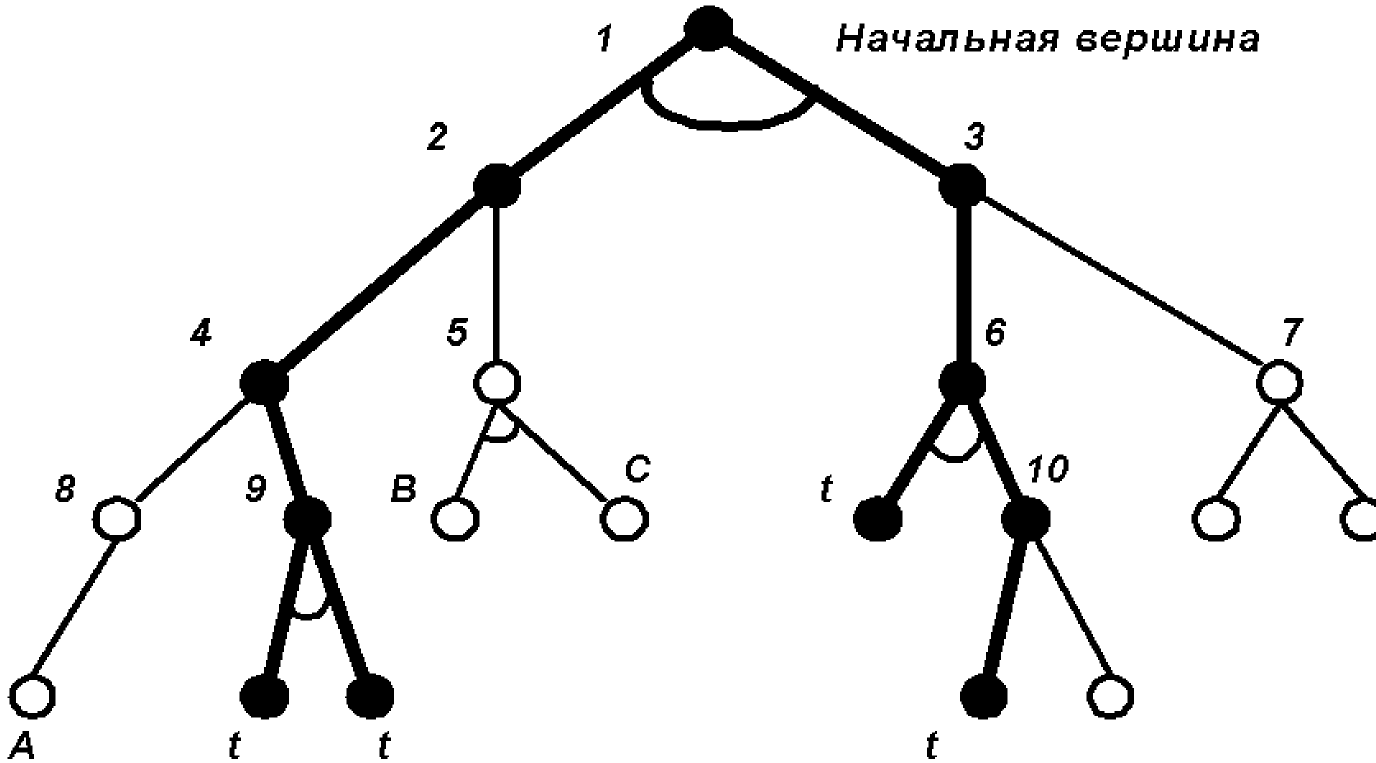

Рис.2.

Числа, стоящие около вершин дерева на рис.2, указывают очередность их раскрытия. Заключительные вершины обозначены символами *t*. Разрешимые вершины зачернены, найденное дерево решения выделено жирными ветвями. Заметим, что после раскрытия девятой вершины и установления того, что ее дочерние вершины являются заключительными, вершины А, В и С удаляются из списка ОТКРЫТ.

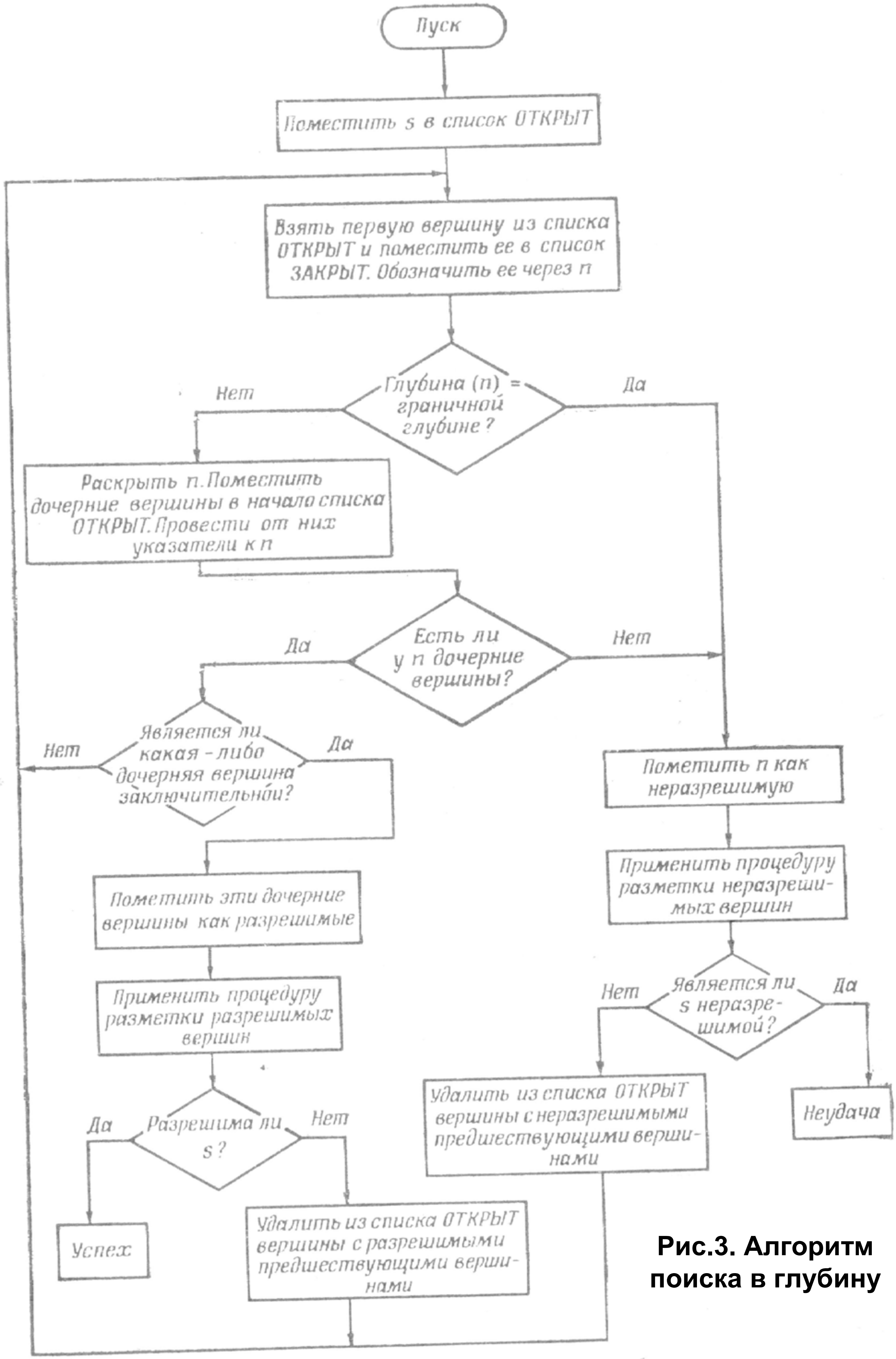

Рис.3. Алгоритм поиска в глубину

Пример последовательности раскрытия вершин.

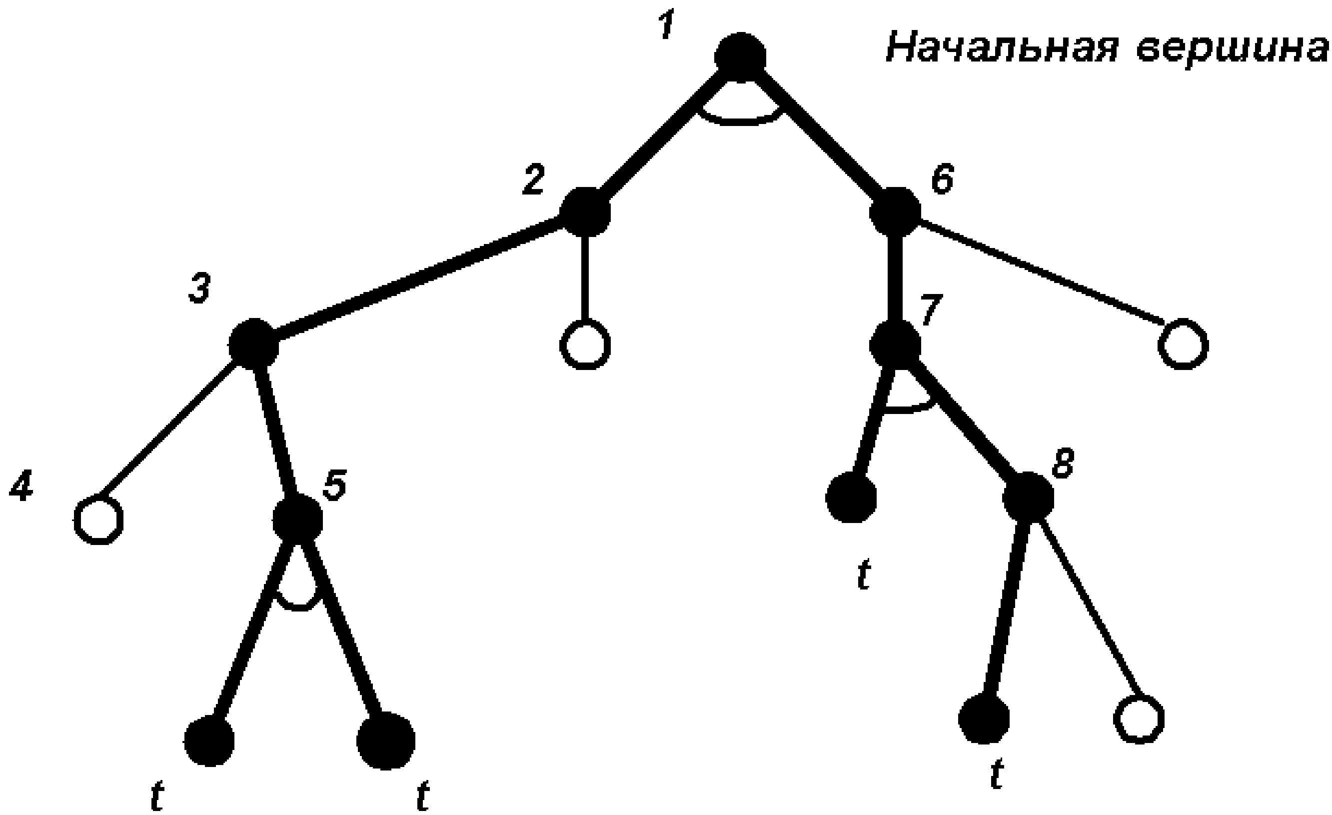

Рис.4.

Как и при переборах в пространстве состояний, с целью построения дерева решений в пределах определенной граничной глубины в первую очередь раскрываются только что построенные вершины. В приведенном на рис.4 примере граничная глубина равна 4.

Перебор на графах “И/ИЛИ”.

- **Основной причиной наличия у вершин “И/ИЛИ” графа более чем одной родительской вершины является наличие идентичных подзадач среди тех, на которые разбивается исходная задача. Если есть возможность распознавания случаев эквивалентности описаний задач, то задачу поиска можно в общем случае существенно упростить за счет однократного решения идентичных подзадач. Трудность – в исключении из рассмотрения некоторых верных нециклических решений ([1], стр.139, рис.5.6).**

- **Вторая трудность – в организации корректного анализа структуры указателей графа перебора при наличии указателей, идущих от данной вершины к более чем одной родительской. Некоторые из этих указателей могут не потребоваться в решении.**

- **При построении дочерних вершин следует проверить, нет ли уже каких-нибудь из этих вершин в списке ЗАКРЫТ и не были ли они прежде помечены как разрешимые или как неразрешимые. Если при этом мы пришли к уже разрешимой (или неразрешимой) дочерней вершине по новому пути, то необходимо воспользоваться процедурой разметки в целях проверки разрешимости/неразрешимости начальной вершины.**

Стоимости деревьев решений.

Для "И/ИЛИ" деревьев мы будем рассматривать суммарную и максимальную стоимости.

<u>Определение.</u> Суммарная стоимость представляет собой сумму стоимостей всех дуг в дереве решения.

<u>Определение.</u> Путем по дереву с корнем в вершине *n* будем называть последовательность вершин (n_1, n_2, … ,n_k) дерева, где n_1=*n*, n_k – заключительная вершина, а вершина n_j – дочерняя для n_{j-1} при *j*=2, … ,*k*.

<u>Определение.</u> Стоимость пути есть сумма стоимостей дуг, связывающих вершины этого пути.

<u>Определение.</u> Максимальная стоимость – стоимость пути по дереву решения, имеющего максимальную стоимость.

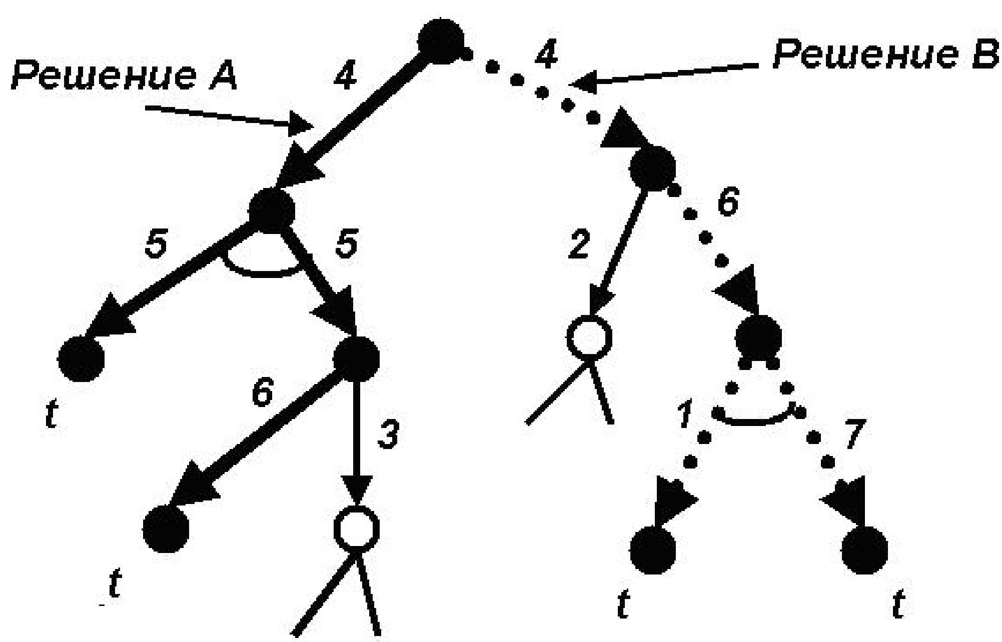

Рис.5.

На рис.5 для решения А имеем суммарную стоимость=20, максимальную стоимость=15; для решения В суммарную стоимость=18, максимальную стоимость=17.

Оптимальное дерево.

Определение. Оптимальным деревом для “И/ИЛИ” дерева будем называть дерево решения с минимальной стоимостью.

Обозначим стоимость оптимального дерева с корнем в начальной вершине *s* как *h(s)*. Дадим рекурсивное определение минимальной стоимости *h(s)* через через минимальную стоимость *h(n)* дерева решения с корнем в любой вершине *n*. Будем обозначать стоимость пути между вершиной n_i и ее дочерней вершиной n_j через $c(n_i,n_j)$.

1. Если *n* является заключительной вершиной (отвечает элементарной задаче), то

$$h(n) = 0$$

2. Если *n* не является заключительной вершиной и имеет в качестве своих дочерних вершин вершины $n_1, \ldots , n_k$ типа “ИЛИ”, то

$$h(n) = \min_i \left[c(n, n_i) + h(n_i) \right]$$

3. Если *n* не является заключительной вершиной и имеет в качестве своих дочерних вершин вершины $n_1, \ldots , n_k$ типа “И”, то для суммарных стоимостей

$$h(n) = \sum_{i=1}^{k} \left[c(n, n_i) + h(n_i) \right],$$

а для максимальных стоимостей

$$h(n) = \max_i \left[c(n, n_i) + h(n_i) \right]$$

Как следует из определения, функция *h(n)* не определена для неразрешимых *n*.

Использование оценок стоимости для прямого перебора.

Пусть $\hat{h}$ - функция оценки величины $h(n)$ для каждой вершины *n*, которая не является неразрешимой.

<u>Определение.</u> Вершину дерева перебора будем считать концевой в одном из трех случаев :

- **Вершина признана заключительной;**
- **Вершина не признана заключительной и не имеет дочерних вершин;**
- **В процессе перебора для вершины еще не были построены дочерние вершины.**

Теперь опишем, каким образом для различных типов вершин дерева перебора (концевых и неконцевых) должна строиться эвристическая функция $\hat{h}(n)$

Случи различных типов вершин.

n – концевая вершина.

1).Если вершина *n* является заключительной, то $\hat{h}(n)=0$

2).Если установлено, что *n* не является заключительной и у нее нет дочерних вершин, то функция $\hat{h}(n)$ **не определена.**

3). Если у вершины еще не были построены дочерние вершины, то за оценочную функцию $\hat{h}(n)$ **принимается некоторая эвристическая оценка стоимости величины *h(n)* оптимального дерева решения с корнем в вершине *n*. Здесь используется эвристическая информация, определяемая характером решаемой задачи.**

n не является концевой и имеет дочерние вершины n_1, …, n_k типа “ИЛИ”.

$$\hat{h}(n)=\min_i\left[c(n,n_i)+\hat{h}(n_i)\right]$$

n не является концевой и имеет дочерние вершины n_1, …, n_k типа “И”.

для суммарных стоимостей :

$$\hat{h}(n)=\sum_{i=1}^{k}\left[c(n,n_i)+\hat{h}(n_i)\right]$$

для максимальных стоимостей :

$$\hat{h}(n)=\max_i\left[c(n,n_i)+\hat{h}(n_i)\right]$$

Потенциальное дерево решения.

Ключевым понятием в эвристическом процессе перебора для деревьев "И/ИЛИ" является потенциальное дерево решения τ_0, которое определяется следующим образом :

- **Корень расположен в начальной вершине *s*;**
- **По оценке служит верхней частью оптимального дерева решения с корнем в *s*.**

Процедура извлечения потенциального дерева решения основана на следующем определении.

1. **Начальная вершина входит в τ_0.**
2. **Если у вершины *n*, входящей в τ_0, в дереве перебора есть :**

а). дочерние вершины $n_1, \ldots, n_k$ типа "ИЛИ", то та из них, для которой значение $\left[c(n, n_i) + \hat{h}(n_i)\right]$ минимально, также входит в τ_0.

б).дочерние вершины типа "И", то все они входят в τ_0

Для извлечения τ_0 в ходе перебора необходимо следить за величинами оценки *h* в каждой вершине дерева поиска. После извлечения дерева τ_0 раскрываются его еще не выбранные для раскрытия вершины и для выросшего дерева пересчитываются величины $\hat{h}$ Здесь достаточно пересчитать $\hat{h}$ для вершин, предшествующих только что раскрытой вершине, для других вершин они не изменятся.

Алгоритм упорядоченного перебора для деревьев “И/ИЛИ”.

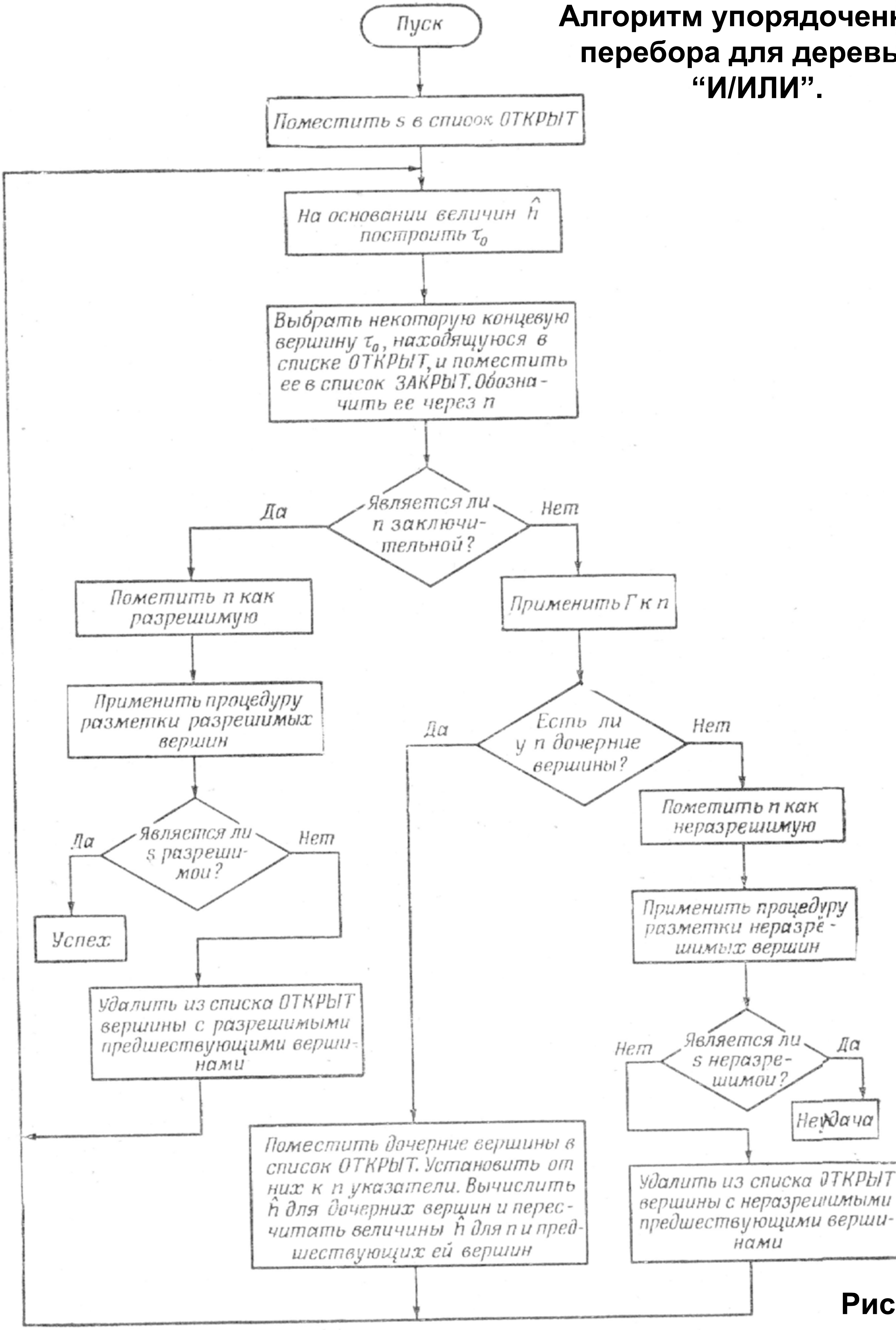

Рис.6

Выбор вершины в τ_0 для очередного раскрытия.

В целях сокращения перебора при построении оптимального дерева следует раскрывать ту из концевых вершин в τ_0, раскрытие которой с наибольшей вероятностью опровергает гипотезу о том, что τ_0 окажется верхней частью оптимального дерева решения, что позволяет с наибольшей вероятностью обнаружить ошибку выбора τ_0.

Непригодность τ_0 при использовании суммарных стоимостей с наибольшей вероятностью может показать открытая вершина в τ_0 с наибольшим $\hat{h}$

В случае максимальных стоимостей для вершины *n* с несколькими дочерними вершинами (типа "И") n_1, n_2, … ,n_k осуществляем переход к той из неразрешимых дочерних вершин, которая максимизирует величину $\left[c(n, n_i) + \hat{h}(n_i)\right]$ Достигнутая

в результате такого перемещения по дереву τ_0 концевая вершина есть вершина, раскрытие которой с наибольшей вероятностью внесет ошибку в оценку максимальной стоимости дерева τ_0.

Следует отметить, что для "И/ИЛИ" деревьев при любом способе выбора открытой вершины в τ_0 эффективность перебора выше для тех функций $\hat{h}$ которые лучше аппроксимируют истинную величину *h*.

Резюме.

В случае отсутствия вершин типа “И” описываемый алгоритмом на рис.6 перебор дерева идентичен упорядоченному перебору в пространстве состояний. Если стоимости дуг равны единице, $\hat{h} \equiv 0$ для концевых вершин и используется определение максимальной стоимости, то описываемая алгоритмом на рис.6 процедура перебора “И/ИЛИ” дерева аналогична процедуре полного перебора.

Алгоритм упорядоченного перебора для деревьев типа “И/ИЛИ” можно сделать более практическим в конкретных ситуациях, если вместо построения после каждого раскрытия вершины нового дерева τ_0 не прерываясь раскрыть одну или более вершин в τ_0 и несколько их дочерних вершин, а после этого строить новое дерево τ_0. Недостаток : возникает риск ненахождения некоторых из раскрытых вершин на наилучшем потенциальном дереве решения.

Другая стратегия модификации рассматриваемого алгоритма состоит в поиске внутри дерева перебора деревьев с наибольшей оцениваемой стоимостью. Необходимое пространство в памяти ЭВМ при этом можно периодически очищать путем отсева деревьев с наибольшей оцениваемой стоимостью. Недостаток : отброшенное дерево может оказаться верхней частью истинного дерева минимальной стоимости.

Литература.

Нильсон Н. Искусственный интеллект : Пер. с англ. - М.: Мир, 1973. С.91-171.