Сведение задачи к совокупности подзадач.

Лекция 4.

Специальность : 230105

Описание задач.

Основная идея подхода состоит в сведении задачи к совокупности тривиальных задач. Причем форма описания задачи может быть любой.

Наиболее удобным является представление в пространстве состояний как совокупность трех составляющих :

- **Множество начальных состояний *S*;**
- **Множество *F* операторов отображения (преобразования) описаний состояний в описания состояний;**
- **Множество *G* целевых состояний.**

При этом задача описывается тройкой (*S, F, G*).

$\forall f_i \in F$ преобразует описание задачи во множество результирующих, или дочерних описаний задач. Это преобразование таково, что решение всех дочерних задач обеспечивает решение исходной родительской задачи. Применение каждого оператора порождает альтернативные множества подзадач. Конечная цель всякого рода сведений задачи к подзадачам состоит в получении такого набора элементарных подзадач, решение которых очевидно. Эти элементарные задачи позволяют сократить перебор при построении альтернативных множеств

Графическое представление множеств подзадач.

Для изображения расчленения задачи на альтернативные множества результирующих задач используются графоподобные структуры, называемые “И/ИЛИ” графами.

Пусть дана задача *A*, которая может быть решена либо путем решения задач *B* и *C*, либо путем решения задач *D* и *E*, либо путем решения задачи *F*. Это соотношение изображается структурой на рис.1.

Каждая группа задач, составляющих альтернативное множество более чем из одной задачи (*B* и *C*, *D* и E), помечается специальным значком в виде дуги. В целях наглядности в подобные структуры вводятся дополнительные родительские вершины для каждого из альтернативных множеств (рис.2).

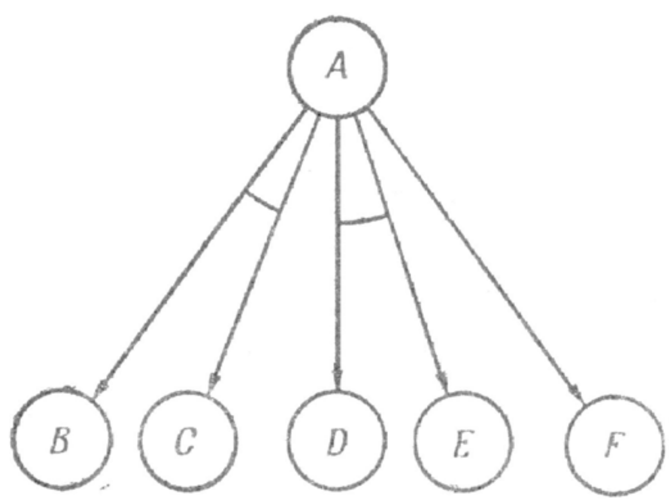


Рис.1. Различные множества подзадач для задачи *A*.

“И/ИЛИ” граф.

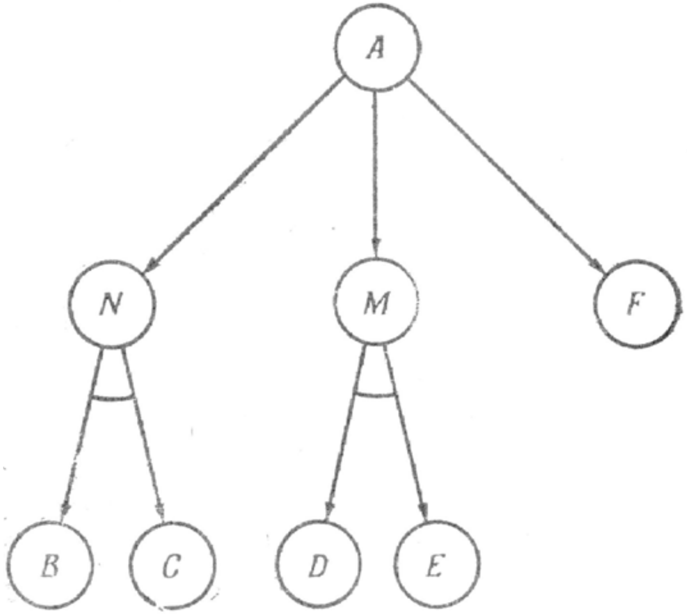


Рис.2. “И/ИЛИ” граф.

Если считать, вершины *M* и *N* играют роль описаний задач, то получается, что задача *A* сводится к одиночным альтернативным подзадачам *N*, *M* и *F*. По этой причине вершины *N*, *M* и *F* называются вершинами типа “ИЛИ”. В то же время, например, для решения задачи *N* необходимо решить и задачу *B*, и задачу *C*. Поэтому вершины : *B* и *C*, *D* и *E* называются вершинами типа “И”.

Структура на рис.2 носит название “И/ИЛИ” графа. Такой граф характеризуется тем, что если вершина имеет непосредственно следующие за ней вершины, то либо все они являются “ИЛИ” вершинами, либо все они – “И”. Если вершины “И” отсутствуют, то получаем обычный граф перебора в пространстве состояний.

На языке “И/ИЛИ” графов применение одиночного оператора $f_i \in F$ к некоторому описанию задачи будет означать, что по очереди сначала будет построена промежуточная “ИЛИ” вершина, а затем непосредственно следующие за ней “И” вершины.

Разрешимость вершин в “И/ИЛИ” графе.

Начальная вершина “И/ИЛИ” графа соответствует описанию исходной задачи. Те вершины, которые соответствуют описаниям элементарных задач, называются заключительными вершинами. Цель поиска на “И/ИЛИ” графе – показать, что начальная вершина разрешима.

<u>Определение</u> разрешимости вершины в “И/ИЛИ” графе дается рекурсивно следующим образом :

- **Заключительные вершины разрешимы как соответствующие элементарным задачам.**
- **Если у вершины, не являющейся заключительной, непосредственно следующие за ней вершины оказались вершинами “ИЛИ”, то она разрешима только тогда, когда разрешима по крайней мере одна из этих вершин.**
- **Если у вершины, не являющейся заключительной, непосредственно следующие за ней вершины оказались вершинами “И”, то она разрешима только тогда, когда разрешима каждая из этих вершин.**

<u>Определение.</u> Решающий граф определяется как подграф из разрешимых вершин, который показывает, что начальная вершина разрешима в соответствии с приведенным определением.

Неразрешимые вершины в “И/ИЛИ” графе.

<u>Определение</u> неразрешимой вершины – дается рекурсивно :

- **Вершины, не являющиеся заключительными и не имеющие следующих за ними вершин, неразрешимы.**
- **Если у вершины, не являющейся заключительной, непосредственно следующие за ней вершины оказались вершинами “ИЛИ”, то она неразрешима тогда и только тогда, когда неразрешима каждая из этих вершин.**
- **Если у вершины, не являющейся заключительной, непосредственно следующие за ней вершины оказались вершинами “И”, то она неразрешима тогда и только тогда, когда неразрешима по крайней мере одна из этих “И” вершин.**

Аналогично решению задач в пространстве состояний, граф “И/ИЛИ”, как правило, определяется неявно посредством описания исходной задачи и операторов редукции задачи. Как и для поиска в пространстве состояний, удобно ввести оператор γ построения непосредственно следующих (дочерних) вершин, его применение означает применение всех применимых операторов $f_i \in F$.

Пример – задача о пирамидке.

Граф состояний для частного случая с тремя дисками (рис.3) представлен на рис.4. (*ijk*) – наибольший диск находится на колышке *i*, диск средней величины – на колышке *j*, самый маленький диск – на колышке *k*. Больший диск – на меньшем !

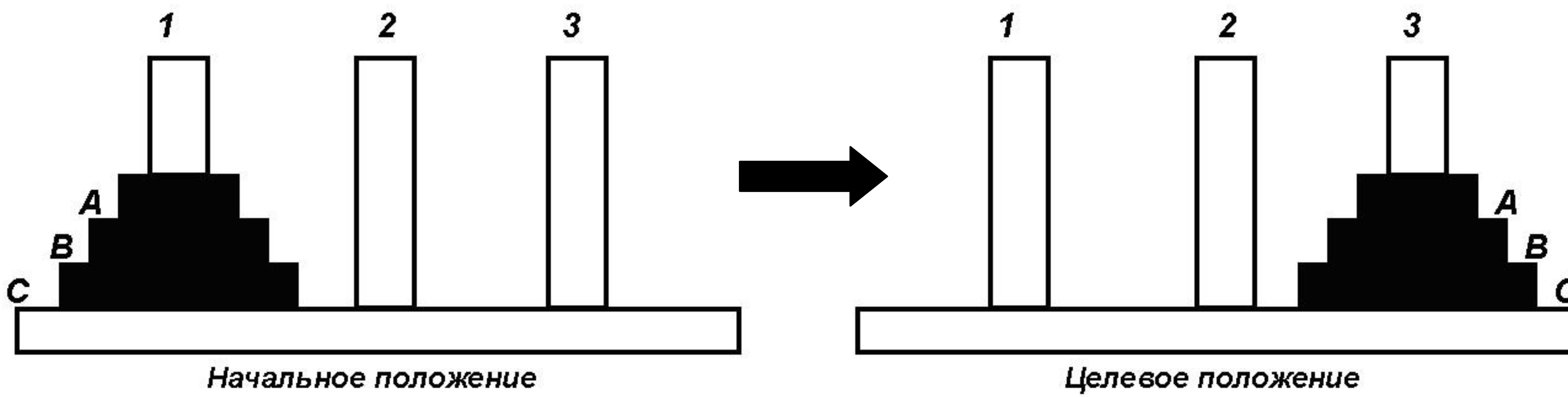


Рис.3. Задача о пирамидке.

Граф на рис.3 отображает полное пространство состояний для этой задачи и имеет 27 вершин. Решим задачу о пирамидке простым методом сведения задачи к совокупности подзадач так, как это показано на рис.5.

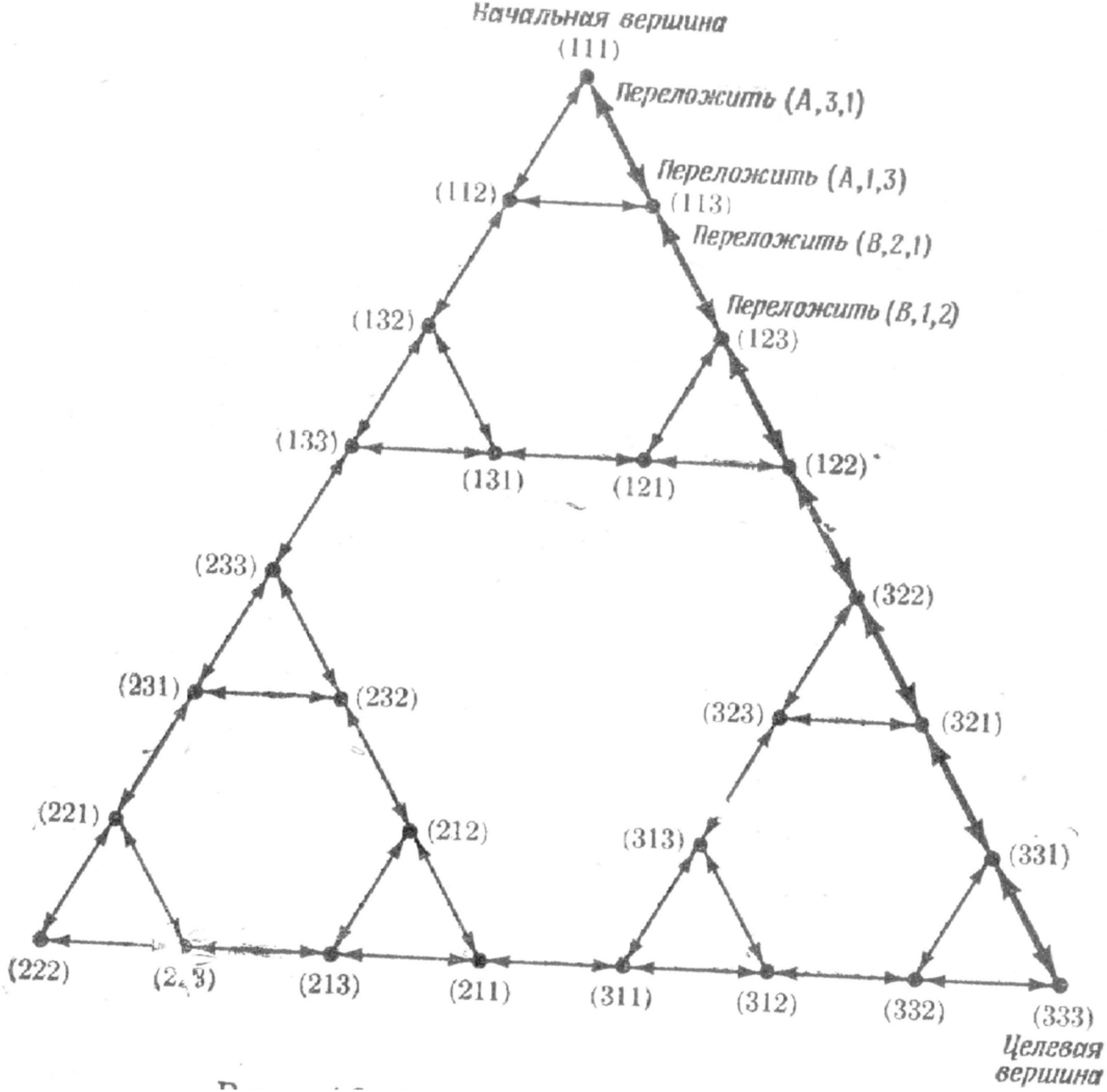


Рис.4. Граф состояний для случая с тремя дисками.

Задача о пирамидке : сведение к подзадачам.

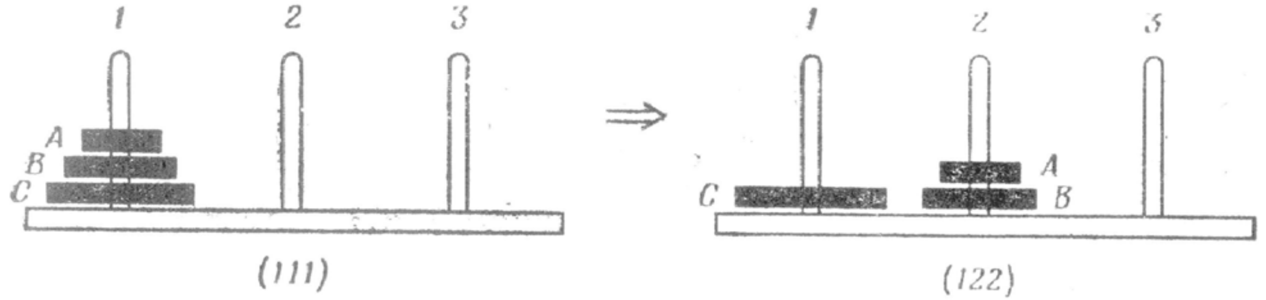
Воспользуемся следующей цепочкой рассуждений.

- **Для перемещения всех трех дисков на колышек 3 необходимо переложить на этот колышек диск *C*, причем в момент, предшествующий перекладыванию диска *C* на колышек 3, последний должен быть свободным.**
- **Для перемещения диска *C* с колышка 1 необходимо снять диски *A* и *B*. Эти два диска лучше всего переместить на колышек 2.**
- **Решение оставшейся задачи начинается с перемещения диска *C* на колышек 3, что положит начало целевому расположению дисков.**

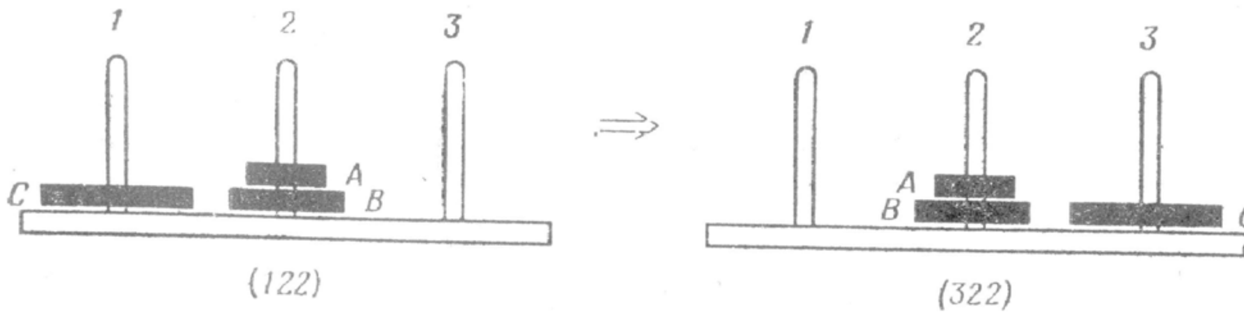
Приведенные рассуждения позволяют позволяют свести исходную задачу к трем показанным на рис.5 подзадачам. Каждая из трех полученных задач меньше исходной. В частности, задача 2 может рассматриваться как элементарная, ее решение состоит ровно из одного хода. Используя подобную цепочку рассуждений, задачи 1 и 3 также можно свести к элементарным так, как это показано на рис.6.

Такая же схема сведения задачи о пирамидке к совокупности подзадач может быть применена и для случая задачи о пирамидке с произвольным числом дисков.

1. Задача с двумя дисками о перемещении дисков A и B на колышек 2:



2. Задача с одним диском о перемещении диска C на колышек 3:



3. Задача с двумя дисками о перемещении дисков A и B на колышек 3:

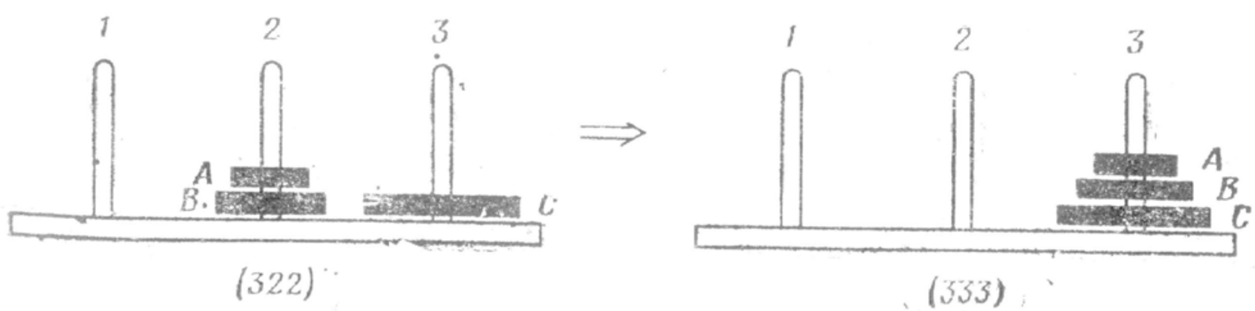


Рис.5. Подзадачи для задачи о пирамидке.

“И/ИЛИ” граф для задачи о пирамидке.

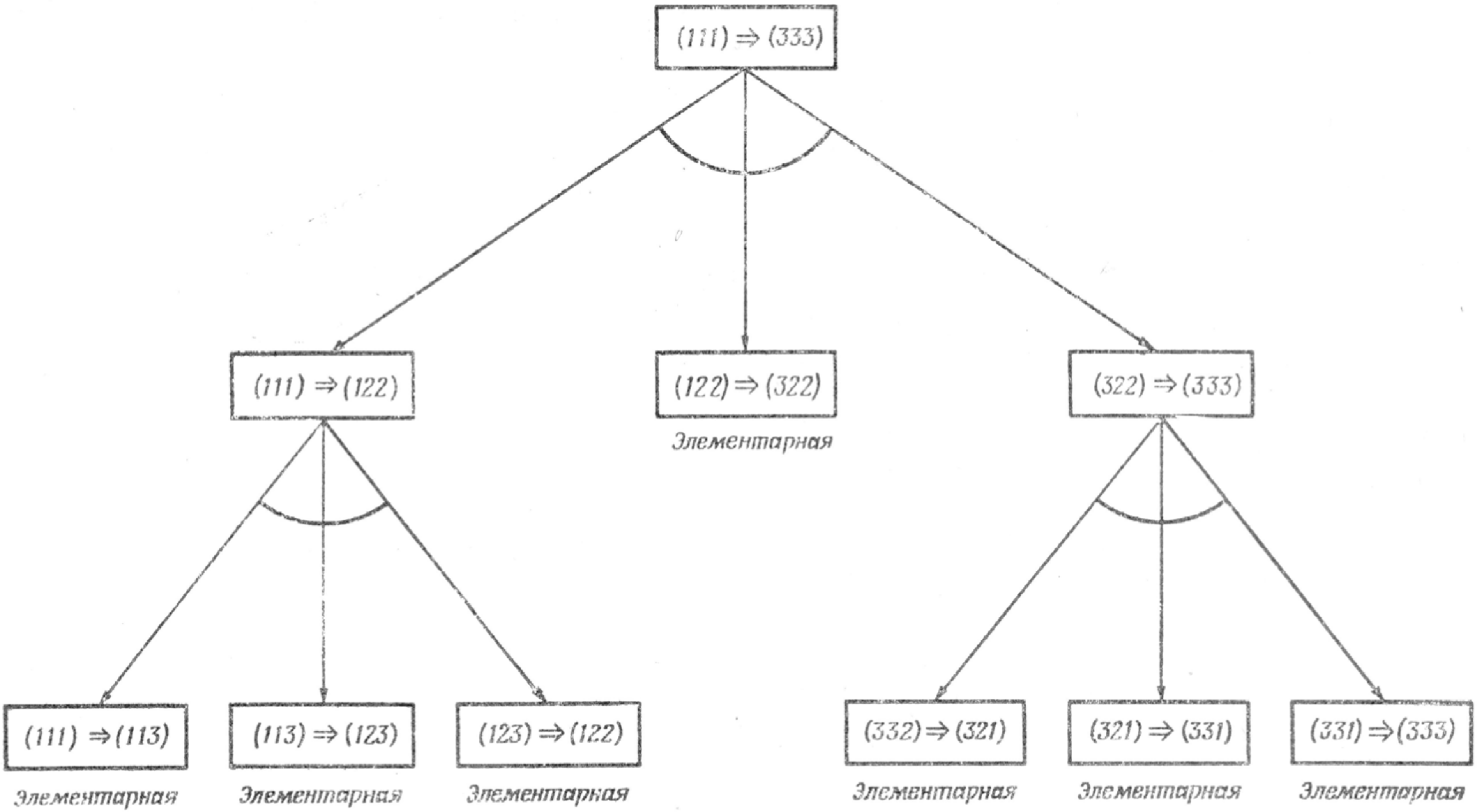


Рис.6

Использование механизмов планирования.

Предположим, что задачу (S, F, G) поиска в пространстве состояний требуется свести к совокупности более простых задач поиска в пространстве состояний.

Если имеется возможность выделить последовательность “*основных промежуточных состояний*” (milestones) g_1, g_2, … ,g_N, то первоначальная задача может быть сведена к множеству задач, определяемых тройками :

(S, F ,{g_1}), ({g_1}, F, {g_2}), … ,({g_N}, F, G). Решение этих задач эквивалентно решению первоначальной задачи.

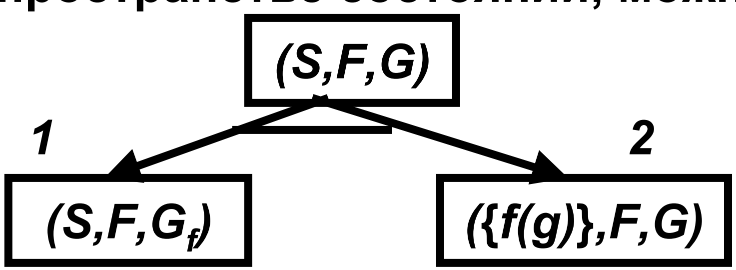
Если эти основные промежуточные состояния g_1, g_2, … ,g_N, определяются явно, то безразлично, в каком порядке решаются результирующие задачи. Если G_1, G_2, … , G_N : $\forall g_1 \in G_1$ может служить в качестве первого основного промежуточного состояния, $\forall g_2 \in G_2$ – в качестве второго и т.д., то описываемая тройкой (S, F, G_1) задача должна быть решена в первую очередь, чтобы можно было найти конкретное состояние $g_1 \in G_1$ прежде, чем будет сформулирована следующая задача ({g_1}, F, G_2), и т.д.

Использование ключевых операторов.

<u>Определение.</u> Ключевым мы будем в дальнейшем называть такой оператор $f \in F$, который будучи расположенным в решающей цепочке операторов, совершенно необходим для решения задачи (S,F,G) и его достаточно легко выделить при разбиении исходной задачи. Пример – “переложить диск *C* на колышек 3” в задаче о пирамидке.

Пусть $f \in F$ - ключевой оператор. Тогда первой задачей, следующей из (S,F,G), будет задача поиска пути к некоторому состоянию $g \in G_f$, G_f – множество состояний, к которым применим f, эта задача описывается тройкой (S,F,G_f). Как только такая подзадача решена и названо состояние $g \in G_f$, можно сформулировать элементарную задачу $(\{g\},F,\{f(g)\})$, где $f(g)$ – состояние, которое будет достигнуто после применения оператора f к состоянию g.

Таким образом, когда может быть найден ключевой оператор f в пространстве состояний, можно воспользоваться следующим способом:



При использовании стратегии сведения задач к подзадачам необходимо определить некоторый ключевой оператор для (S,F,G_f) и т.д. Проблема – в наличии формализуемого способа построения множества операторов-кандидатов в ключевые.

Вычисляемые различия.

Определение. Под различиями для задачи (S,F,G) понимается частичный список причин, в силу которых элементы множества S не удовлетворяют тем условиям, которым должны удовлетворять целевые состояния (элементы множества G). Если $\exists\ s \in S : s \in G$, то задача решена. Если $s \in S$ удовлетворяет части условий, то различием может служить перечень условий, которым элемент s не удовлетворяет.

Пример – “Обезьяна и бананы” (*Лекция 2*). Начальное состояние задачи (a,0,b,0) не удовлетворяет поставленной цели, последний элемент в нем не равен 1. За уменьшения этого различия отвечает оператор f_4 “схватить”. (a,0,b,0)$\notin G_{f4}$ в силу истинности утверждений : ящик не находится в точке *c*, обезьяна не находится в точке с и обезьяна не на ящике. Отсюда имеем множество из трех ключевых операторов : f_2 – передвинуть ящик в точку *c*, f_1 – переместиться (подойти) в точку *c* и f_3 – взобраться. С помощью ключевого оператора f_2 задача ({a,0,b,0},G_{f4}) сводится к паре задач : ({a,0,b,0},G_{f2}), обозначим ее (1.1), и ({$f_2(s_{11})$},G_{f4}), обозначим ее (1.2). $s_{11} \in G_{f2}$ есть результат решения задачи (1.1).

Различие для задачи (1.1) формулируется как “Обезьяны нет в точке b”, на основании чего получаем ключевой оператор f1 – подойти (b). Этот оператор используется для разбиения задачи (1.1) на две, из которой первая – элементарная. Процесс завершения построенных ранее решений продолжается до тех пор, пока не будет решена исходная задача. “И/ИЛИ” граф для решения этой задачи представлен на рис.7.

“И/ИЛИ” граф для задачи об обезьяне и бананах.

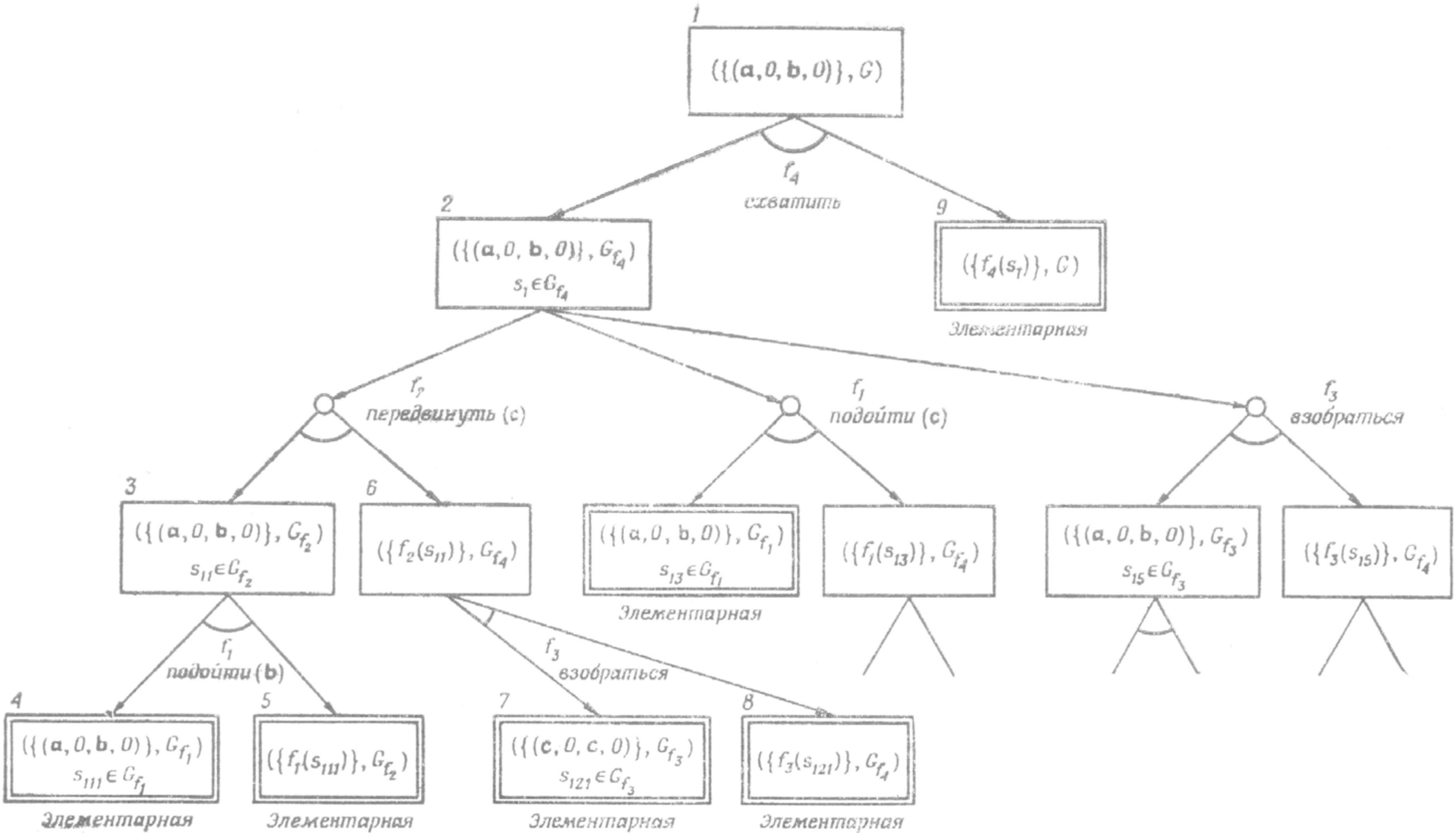


Рис.7

Выводы.

При использовании ключевых операторов задача расщепляется на части с последующим уменьшением объема необходимых поисковых усилий. Следует отметить, что в задаче об обезьяне и бананах пространство перебора достаточно мало и использование ключевых операторов не дало заметного увеличения эффективности.

Для того, чтобы воспользоваться ключевыми операторами, программе-решателю задач должна быть сообщена процедура вычисления различий и процедура, позволяющая связывать с ними ключевые операторы. Возможность установления хороших связей между различиями и операторами сильно зависит от конкретной задачи.

Литература.

Нильсон Н. Искусственный интеллект : Пер. с англ. - М.: Мир, 1973. С. 91-128.