Учебные вычислительные задачи: матеметическая теория.

Описываются усовершенствованный метод Э. Х. Тыугу [1] планирования решения задач на неопределённых системах элементарных уравнений, метод составления таких задач и метод преобразования задач и планов решений в графовом и табличном представлении. Ошибки планирования этим методом возможны лишь из-за недостаточной проработки модели проблемной области. Изложение материала ориентировано на интересующихся учителей, будущих учителей и старшеклассников. Я не жалел слов и рисунков. На мой взгляд, тема интересна хотя бы потому, что, разобравшись в ней, можно лучше представлять что-то из того, что регулярно делаешь.

Работа выполнена пенсионером инициативно, как хобби. К сожалению, кроме [1], [2] и материалов по программированию в ограничениях ([3] и др.), ориентированных на решение сложных задач численными методами, а не на теорию задач, решаемых в школе, найти первоисточники для опоры и ссылки на них мне не удалось. Некоторые идеи я в черне уже публиковал на сайте о черновиках [6].

Оглавление .

Указатель терминов и утверждений.

переопределимая подсистема и система уавнений; шаг структурирования; структурированная система уравнений.

Утверждения (7),(8). Предположение (п1в).

Разделение альтернатив.

Компонентно определёная система уравнений; компонентные ветви; совместно планируемая модель.

Утверждение (9). Предположение (п1г).

Представление графом.

Граф; вершины; рёбра; ориентированный граф; дуги; бидольный граф; путь; бикомпоненты (компоненты сильной связности);

внешние и внутренние вершины уравнений; разомкнутый и замкнутый граф решения задачи, его компоненты; формально, разомкнуто, замкнуто правильная ориентация; одинарная ориентация; инверсия (инвертирование) путей; непересекающиеся пути.

Утвержения (10а,б).

А1. Алгоритм планирования решения полных задач.

Контур; срабатывание вершин уравнений и компонент, готовность к срабатыванию; завершённые и незавершённые вершины уравнений.

Утверждение (11).

Композиция моделей. Естественный поток вычислений.

Экземпляр модели; композиция моделей; элементарный физический процесс, его физический закон; начальные условия; структурные связи; общие связи; естественный поток вычислений.

Предположение (п11а).

А2. Алгоритм планирования решения неполных задач.

Паросочетание, максимальное паросочетание; внутренний подграф решения; эквивалентные планы решения.

Утверждение (12).

Граф зависимостей.

Граф зависимостей; столбец учёта уравнений.

А3. Алгоритм ориентирования графа зависимостей при планировании решения задачи.

Составление задач.

А4. Алгоритм составления полной задачи.

Срабатывание вершины переменной и компоненты; незавершённые вершины переменных.

Составление неполной задачи.

Ветвящаяся модель.

Граф ветвящейся модели.

Планирование решения задачи.

Составление задачи на ветвящейся модели.

Пример 4.

Задачи с безальтернативной формулировкой.

Переменные общие, индивидуальные ветвей, ситуативные, переменные конструкции.

Другой метод.

Инверсия ветвящихся путей.

Табличное представление.

Таблица модели объекта; смежные вершины; матрица смежности; таблицы ветвей; подтаблица зависимостей;

Исходные определения.

Считаем, что *модель* объекта представлена системой уравнений, а *задача* на ней – списками заданных (*исходных*) и *искомых* переменных. модели, кроме того, указывается список зависимых подсистем, если такие есть, и условия действия альтернативных подсистем уравнений и зависимостей между уравнениями, если они есть. Уточнение — далее.

Сначала будем рассматривать задачи, в которых все переменные, кроме заданных, – искомые; назовём их *полными задачами.*

Задачи, в которых все переменные — искомые, будем называть *общими.*

Уравнения, содержащие производные или интегралы, рассматривать не будем.

Планирование решения неравенств рассматривать также не будем, но допускаем, что выбор уравнений, входящих в модель, и, соответственно, промежуточные результаты и ответ могут зависеть от условий, выраженных равнствами или неравенствами.

Далее выражение «числовой континуум» интерпретируем как «множество чисел, содержащее все числа из некоторого отрезка на вещественной оси, не вырожденного в точку». «Не более чем счётное множество чисел» – это одно число или конечное множество чисел или счётное множество чисел, обычно представляемое как множество всех значений функции натурального аргумента.

Считаем, что на вещественной оси более чем счётное множество чисел есть числовой континуум (континуум-гипотеза).

Системы уравнений и задачи, имеющие хотя бы одно решение, называем *разрешимыми.*

Далее вместо слов «A — уравнение, X — множество всех входящих в него переменных» – будем писать «A(X) — уравнение»; вместо слов «C — система уравнений, X — множество всех входящих в неё переменных» _будем писать «C(X) — система уравнений»; вместо слов «A — система уравнений задачи, X1 — множество всех её исходных переменных, X2 — множество всех её искомых переменных)» будем писать «A(X1/X2) — задача» или «A(X) — задача», где X — объединение X1 и X2.

Уравнения или системы уравнений или задачи A(X) и B(X) будем называть *эквивалентными*, если множества их решений совпадают. Записываем так: A(X1)$\Leftrightarrow$B(X2). Уравнение или система уравнений B(X) является *следствием* уравнения или системы уравнений A(X), если множество решений B(X) содержит множество решений A(X). Записываем: A(X)⇒B(X). При этом считаем, что множество переменных в A(X) и B(X) можно расширять: если X1 содержится в X, X1* является решением уравнения или системы уравнений A(X1) или B(X) и X* является любым значением X, в котором X1 имеет значение X1*, то X* является решением для A(X) или B(X) соответственно.

Разрешимые задачи называем также *задачами с непротиворечивыми данными*.

Разрешимые задачи, имеющие не более чем счётное число решений, будем называть *точечными,* имеющие ровно одно решение — определёнными, а имеющие континуум решений — *неопределёнными.*

Определённые, точечные или неопределённые задачи с множеством искомых переменных X будем называть соответственно *определёнными, точечными или неопределёнными относительно X*. Систему уравнений будем называть определённой, точечной, неопределённой, или определённой, точечной или неопределённой относительно X, если таковой является общая задача на этой системе.

(п1а). Будем считать, что система уравнений модели является неопределённой относительно любой из входящих в неё переменных (а следовательно и относительно любого множества входящих в неё переменных).

Точечные задачи, которые превращаются в неопределённые при исключении одного из исходных данных, будем называть задачами *с минимумом данных*. Задачу, разрешимую лишь тогда, когда исходные переменные удовлетворяют некоторому уравнению или системе уравнений, будем называть *переопределённой.* Другими словами, задача переопределена, если из выполнения её системы уравнений следует выполнение некоторого уравнения или системы уравнений, связывающих её исходные данные. Поэтому переопределёную задачу будем ещё называть задачей с *зависимыми данными*. Переопределённую неопределённую задачу будем называть *задачей с перекосом данных*.

Разрешимое уравнение, такое, что для каждой из входящих в него переменных все решения этого уравнения можно представить в виде не более чем счётного числа функций, выражающих значения этой переменной x через значения остальных переменных X\{x} (совокупность этих функций представляет многозначную функцию, которую будем называть *функцией решения* для x), будем называть *решаемым.*

Однорешаемость.

Уравнение A(X) *решаемо относительно подмножества переменных* X1 из X, если функция его решения для каждой переменной из X1 существует и не зависит от значений других переменных из X1. Если при этом X1 содержит более одной переменной, то уравнение A(X) будем называть *многоразрешимым.* Уравнение A(X), которое при условии P(X) разрешимо, но не решаемо относительно сразу двух переменных, назовём *одноразрешимыми при условии* P(X). Решаемые

одноразрешимые уравнения назовём *однорешаемыми*.
Уравнение может быть многоразрешимым, если, для того, чтобы ему удовлетворить, требуется одновременное достижение максимума или минимума двух и более аргументов функции, изменяющейся монотонно при изменении этих аргументов (практически — двух слагаемых или сомножителей).

Пример 1. Уравнение $((x-5)^2+2)((2-y)^2-1)=a+c$ не является одноразрешимым при $a+c=-2$, так как из него можно определить значения сразу двух переменных: $x=5$, $y=2$. Но при условии $a+c>-2$ оно однорешаемо.

(п1б). Считаем, что каждое уравнение, входящее в модель объекта, однорешаемо.

При этом всё равно возможно, что многорешаемое уравнение появится в процессе решения системы. Например:

$x=(y-a)^2$; $v=(w-b)^2$; $x+v+z=c$ (1*)
При $z=c$ из этих трёх уравнений можно получить четыре:
$y-a=0$, $w-b=0$, $x=0$, $v=0$ (2*)
так как при подстановке первого и второго уравнений в третье получается двурешаемое уравнение. Здесь условие минимума суммы (уравнение $x+v+z=c$) заменили на условия минимума каждого слагаемого и подставили минимальные значения в уравнения $x=(y-a)^2$; $v=(w-b)^2$. Это можно учесть при описании модели, сформулировав её эквивалентным образом так, чтобы многорешаемое уравнение не возникало:
«При $z=c$ (2*), при $z<c$ (1*), при $z>c$ модель не определена.»
В излагаемом здесь методе такие преобразования модели должны быть произведены до планирования решений и составления задач на ней. Если в процессе планирования появятся многорешаемые уравнения, возникнет ошибка планирования.

Систему уравнений будем называть *одноразрешимой*, если при её расширении при помощи применения любого числа операций решения одного из уравнений относительно одной из входящих в него переменных и подстановки такого решения в другое уравнение, - получается система, все уравнения которой одноразрешимы.

Зависимости уравнений.

Будем рассматривать зависимости уравнений без привлечения частных производных и якобианов.

Модель может содержать избыточные (*зависимые*) уравнения, выводимые из других её уравнений, например, законы сохранения в физических моделях. Наличие зависимости уравнений зависит от условий, при котрых они рассматриваются. Условия могут выражать некоторые ограничения или зависимости между исходными данными, или отделять одни корни от других. Для любой системы уравнений существуют условия, при которых она зависима. Например, это выполнение уравнения - суммы всех уравнений этой системы.
Практически зависимости уравнений устанавливаются путём преобразований, при

помощи которых выводится зависимое уравнение как следствие из других уравнений. При этом ситуация «вычисления» уравнений из системы зависимостей похожа на ситуацию вычисления значений из системы уравнений.
Пример 2 из кинематики:

(a) $y=v_0t-gt^2/2$;

(b) $v=v_0-gt$;

(c) $v_o^2-v^2=2gy$.

Выведем (a) из (b),(c). Обозначим такой вывод (b),(c)→(a), а уравнение (a) как его результат (a)←(b),(c).

(c)=($y=(v_0-v)(v_0+v)/2g$),

(b)=($v_0-v=gt$); подставим в предыдущее уравнение:

(c),(b)→(d)=($y=gt(v_0+v)/2g$)=($y=t(v_0+v)/2$); подставим сюда (b):

(d),(b)→($y=t(v0+v0-gt)/2$)=(a).

Обратимость: например, получим (c), (d)→(b), подставляя (d) в (c):
($t(v_0+v)/2= (v_0-v)(v_0+v)/2g$)=((b) или $v+v_0=0$).

Заменив в приведённом выводе (b),(c)→(a) преобразование (c),(b)→(d) на обратное (c),(d)→(b), получаем вывод (c),(d)→(a).
Обратное преобразование (a),(d)→(b) получим, приравнивая правые части уравнений, что можно рассматривать как подстановку одного из них в левую часть другого, или как результат вычитания уравнений.
Выведем (a),(c)→(b): подставляя, получим

($v^2=(v_0-gt)^2$)=($v=\pm(v_0-gt)$), то есть получили правильное и лишнее значения.

Рассмотрим снова это преобразование (a),(c)→(b). Имели до этого
(a),(d)→(b) и (c),(b)→(d), то есть
(a),((d)←(c),(b))→(b). Это аналог уравнения b=f(g(c,b),b) относительно b; то, что удалось построить преобразование (a),(c)→(b), - это аналог решаемости этого уравнения.
Вообще любому преобразованию соответствует функция, отображающая наборы уравнений в уравнения. Система преобразований уравнений поэтому, аналогично системе уравнений, может быть независимой или зависимой, в которой одна функциональная связь является следствием других. Например, обратное преобразование — следствие прямого.
Перейдём к формальному рассмотрению.

Уравнение в общем виде: f(X)=g(X), где X — набор переменных. Обозначив это уравнение через w, будем писать w=(f(X)=g(X))
Дадим определение функциональной зависимости уравнения y от системы уравнений C(X). Уравнение y должно получаться из уравнений C применением какой-либо функции h. Её аргументами могут быть левые и правые части уравнений C, а результатом — левая и правая части уравнения y. Но функция должна связывать

уравнения, взятые целиком, и давать в результате уравнение – следствие. Примем, что одна и та же функция h, применённая к левым частям уравнений C, даёт левую часть уравнения y, а применённая к правым частям C – правую часть y. Примеры показывают, что в число аргументов h нужно включить и переменные X. Сформулируем.

Пусть $w_1=(f_1(X)=g_1(X))$, …, $w_n=(f_n(X)=g_n(X))$ и h — некоторая *однозначная числовая* функция от n+p *числовых* аргументов, где p — количество переменных в их наборе X. *Функцией h от системы уравнений* $C=(f_1(X)=g_1(X))$, …, $(f_n(X)=g_n(X))$ будем называть следующее уравнение:

$h(C)=h(w_1, \ldots, w_n, X)=(h(f_1(X), \ldots, f_n(X), X)=h(g_1(X), \ldots, g_n(X), X))$.

Если функция h определена при условии P(X), то есть определены $h(f_1(X), \ldots, f_n(X), X)$ и $h(g_1(X), \ldots, g_n(X), X)$, то её будем называть *применимой* к данной системе уравнений C *при условии* P(X), а уравнение h (C) — *функционально зависимым* от C *при условии* P(X).

Очевидно, что если совокупность значений переменных X=X* является решением системы C и удовлетворяет условию применимости функции h, то X* удовлетворяет и уравнению h (C).

Систему уравнений, одно из которых можно представить как применимую функцию от остальных, называем *функционально зависимой*, а систему уравнений, не являющуюся функционально зависимой, – *функционально независимой* (при соответствующем условии, если оно указывается).

Попытаемся построить функцию h(C), дающую в результате $y=(f_n(X)=g_n(X))$, универсаного вида, воспользовавшись тем, что в число аргументов могут входить переменнные X, а вид функции можно выбирать произвольно.

Сначала преобразуем все уравнения $(f_i(X)=g_i(X))$, $i=1 \ldots n$ к виду $(f_i(X)-g_i(X)=0)$. Это можно сделать, отняв $g_i(X)$ от обеих частей: $w_i - g_i(X)$, $i=1 \ldots n$ (y в конце можно будет восстановить, прибавив $g_n(X)$ к обеим частям результата).

Запишем желаемый результат: $h_1(C)=f_n(X)-g_n(X)=0$, $h(C)=h_1(C)+g_n(X)$.

Для правых частей:

$h_1(f_1(X)-g_1(X),\ldots,f_{n-1}(X)-g_{n-1}(X),X)=f_n(X)-g_n(X)$, для левых частей — $h_1(0,\ldots,0,X)=0$.

Единая функция: $h_1(0,\ldots,0,X)=0$ и $h_1(f_1(X)-g_1(X),\ldots,f_{n-1}(X)-g_{n-1}(X),X)=f_n(X)-g_n(X)$, если хотя бы один из аргументов $f_1(X)-g_1(X),\ldots,f_{n-1}(X)-g_{n-1}(X)$ не равен нулю. Это означает, что $h_1(C)=y$ при любых значениях X тогда и только тогда, когда все решения системы C являются решениями уравнения y. Сформулируем.

(2). Уравнение w функционально зависит от системы уравнений C при условии P тогда и только тогда, когда, когда выполнение этого уравнения w при условии P является следствием выполнения C.

Доказательство.

Пусть $C(X)=(f_1(X)=g_1(X), \ldots, f_{n-1}(X)=g_{n-1}(X))$; $w_1=(f_1(X)=g_1(X))$, …, $w_n=(f_n(X)=g_n(X))$.

Тогда $w_i-g_i(X)=(f_i(X)-g_i(X)=0)$, $i=1...n$.

Пусть $(h(w_1, \ldots, w_{n-1},X)=(h_1(w_1(X)-g_1(X),\ldots,w_{n-1}(X)-g_{n-1}(X),X))+g_n(X))$,

где $h_1(z_1,\ldots,z_{n-1},X)=\{f_n(X)-g_n(X)$, если только не $z_1=\ldots=z_{n-1}=0$,

и 0, если $z_1=\ldots=z_{n-1}=0\}$ (#).

Эта функция h при условии P(X) даст в результате $w_n=(f_n(X)=g_n(X))$, если все корни системы C при условии P(X) являются корнями этого уравнения w_n, то есть если из выполнения системы C следует при условии P(X) выполнение уравнения w_n. Таким образом, уравнение — следствие из системы уравнений всегда функционально зависимо от этой системы.

Обратное утверждение очевидно, как уже говорилось.

Далее слово «*функционально*» в выражениях «функционально зависимое уравнение» или «функционально зависимая система» может опускаться.

Оговорим особо случай зависимости от одного уравнения. Выполнение уравнения A является следствием выполнения уравнения B тогда и только тогда, когда A можно получить из B функциональным преобразованием. A эквивалентно B тогда и только тогда, когда A и B связаны взаимно однозначным преобразованием (если a1(x)=a2(x), b1(x)=b2(x) – уравнения, b1(x)=f(a1(x)), b2(x)=f(a2(x)) и a1(x)=g(b1(x)), a2(x)=g(b2(x)), то f и g — взаимно обратные функции).

Эквивалентные преобразования уравнений, входящих в систему, не влияют на её свойство зависимости или независимости.

Приведём функциональные зависимости для *уравнений, выводимых на основании свойств равенства*, и введём для них сокращённые обозначения.

- Симметричность. Пусть w=(x=y). Тогда –(w–x–y)=(y=x).
  Введём обозначение: если w=(x=y), то w*=(y=x).
  Заметим, что $h(w_1^*, \ldots, w_n^*, X)=(h(w_1, \ldots, w_n, X))^*$.
- Транзитивность: Пусть w1=(x=y), w2=(y=z). Тогда w1–w2*+z=(x=z).
- Подстановка в левую часть уравнения: Пусть w1=(f(g(x))=h(x)), w2=(g(x)=q(x)).
  f(w2)=(f(g(x)=f(q(x)), f(w2)–w1=(0=f(q(x)–h(x)),
  (f(w2)–w1+h(x))*=(f(q(x))=h(x)).
  Подстановка в правую часть: Пусть w1=(h(x)=f(g(x))), w2=(g(x)=q(x)).
  (h(x)=f(q(x)))=(f(w2*)–w1+h(x))*.
  Введём обозначение: если w1=(f(g(x))=h(x)) или w1=(h(x)=f(g(x)),
  w2=(g(x)=q(x)), то w1%w2=(f (q(x))=h(x)) или, соответственно,
  w1%w2=(h(x)=f(q(x)).
  При записи композиций будем считать, что операция «*» имеет приоритет над операцией «%».
- Транзитивность с использованием введённых обозначений:
  если w1=(x=y), w2=(y=z), то w2%w1*=(x=z).

Для иллюстрации запишем зависимости в рассмотренном ранее примере 2, опуская промежуточные выражения (для составления планов решений выписывать формулы зависимостей не потребуется, достаточно знать состав зависимых подсистем).

$a=(y=v_0t-gt^2/2)$;

$b=(v=v_0-gt)$;

$c=(v_0{}^2-v^2=2gy)$;

$d=(y=t(v_0+v)/2)$;

$d=((c^*/2g)\%(v_0-b))^*$; $a=d\%b$; $c=2g((v_0+v)(v_0-b)/2-d+y)$.

Пусть $e=(v_0+v=0)$.

При условии ~e («~» обозначает «не») $b=v_0-(c\%d)/(v_0+v)$, так что b функционально зависит от c и d;

При условии e $d\%e=(y=0)$, $c\%e=(0=y)$, $((v_0-b\%(e-v_0))^*/g)=(t=2v_0/g)$, так что c и d функционально зависимы, но b не зависит от них.

$b=2(a\%d)/t-v_0$

Пусть в системе уравнений C имеется m зависимых подсистем C_i, $i=1...m$ $C_i=(f_{1(i)}(X)=g_{1(i)}(X)), ..., (f_{(n(i)+1)(i)}(X)=g_{(n(i)+1)(i)}(X))$,

$f_{(n(i)+1)(i)}(X)=(h_i(f_{1(i)}(X), ..., f_{n(i)(i)}(X), X))$,

$g_{(n(i)+1)(i)}(X)=h_i(g_{1(i)}(X), ...,g_{n(i)(i)}(X),X)$

Здесь, например, n(i) в индексе обозначает то же, что n_i (так, к сожалению, не записывается в моём текстовом редакторе).

Будем рассматривать h_i как функции своих числовых аргументов, а не как функции от X:

(**) $f_{(n(i)+1)(i)}=h_i(f_{1(i)}, ..., f_{n(i)(i)}, X)$, $i=1...m$,

где f_j понимаются как числовые переменные; множество этих переменных обозначим F. Область определения переменной f_j — пересечение областей изменения функций $f_j(X)$ и $g_j(X)$ – частей уравнения $f_j(X)= g_j(X)$, область определения функции h(F,X) включает область изменения её аргументов при изменении X в пределах области применимости зависимости h. .

(**) – система уравнений, и в ней тоже могут быть зависимые подсистемы. Назовём их *зависимостями второго уровня* (до этого рассматривали *зависимости первого уровня*).

Пусть одна из них h^1: $f_s=h^1(f_1,...,f_r)=h^1(h_1(F,X),...,h_r(F,X),F,X)$. Тогда, подставляя $f_j=f_j(X)$, $j=1...r,s$, получим левую часть зависимости первого уровня, а подставляя $f_j=g_j(X)$, $j=1...r,s$, получим её правую часть.

Могут быть и зависимости третьего уровня, и более высоких уровней.

Если выполнение уравнения $f_{(n(m)+1)(m)}=h_m(f_{1(m)}, ..., f_{n(m)(m)}, X)$ зависимости h_m является следствием выполнения системы уравнений зависимостей $h_1...h_{m-1}$, то будем говорить, что зависимость h_m является с*ледствием системы зависимостей*

$h_1 \ldots h_{m-1}$.

Систему независимых минимальных зависимостей H в системе уравнений C будем называть *базисом зависимостей* в C, если любая зависимость в C является её следствием. Здесь минимальность каждой зависимой подсистемы из базиса означает, что, если исключить из неё хотя бы одно любое уравнение, то она станет независимой.

Пусть h=(f=h'(F1)) — уравнение некоторой зависимости в C (h' должна быть однозначной функцией), H(F) – система уравнений базиса зависимостей в C. Если при значениях F=F* система уравнений H(F*) выполнена, то выполнено и h. h может быть execution выполено и при таких значениях переменных для частей уравнений F, когда H(F) не выполнено, но это не значит, что эти переменные связаны зависимостью, не вытекающей из H.

Базис зависимостей в C— это, другими словами, максимальное множество минимальных независимых зависимостей в C: добавление к нему любой зависимости делает это множество зависимостей зависимым.

Пусть задано минимальное множество исходных переменных Fи для задачи H(F); решение этой задачи — выражение Hи(Fи) остальных переменных F\Fи через Fи. Эти выражения представляют зависимости уравнений, соответствующих переменным F\Fи, от уравнений, соответствующих переменным Fи (если решение включает многозначную функцию, то предполагаем, что можно выделить соответствующую ветвь). В h'(F1) тоже подставим выражения из Hи(Fи) переменных F1 через Fи. Уравнение – результат этой подстановки h(Fи) должно выполняться, если выполняется Hи(Fи), так как h – следствие H, которое выполняется, если выполняется Hи(Fи): f(Fи)=h'(F1(Fи)), то есть h должно быть эквивалентно выражению f в решении Hи(Fи). Следовательно, Fи соответствует независимая подсистема уравнений, иначе Fи не было бы минимумом данных.

Заметим ещё, что уравнение y из C, не входящее ни в одну подсистему базиса зависимостей в C, не может быть зависимым в C, так как переменная для частей y не входит в F.

Таким образом, доказано следующее утверждение:

(3). Пусть H(F) — система уравнений базиса зависимостей в системе уравнений C, Fи – минимальное множество исходных переменных для задачи H(F), Φ — множество всех уравнений из C, входящих в базис зависимостей (их частям соответствуют переменные F), Cи – подмножество уравнений из Φ, частям которых соответствуют переменные Fи, Cн — множество уравнений из C, не входящих в Φ (Cн=C\Φ). Тогда

а) Cи – максимальное независимое подмножество уравнений в Φ, объединение Cи и Cн – максимальное независимое подмножество уравнений в C;

б) любая зависимость уравнений в C эквивалентна зависимости тех же переменных в решении задачи H(F).

(4). Пусть имеется некоторое множество Cз минимальных независимых зависимостей в С, Н(Fи,F0) — система уравнений зависимостей Сз, Fи — множество переменных для частей уравнений из Сз\С0, F0 — переменные для частей уравнений С0, Н(Fи/F0) — разрешимая не переопределённая задача и С\С0 — независимое подмножество уравнений в С. Тогда Сз — базис зависимостей в С.

Доказательство.

Поскольку задача Н(Fи/F0) разрешимая, F0 можно выразить через Fи. Если решение — многозначная функция, то прелполагаем, что можно выделить ветвь, в соответствии с которой Сз следует из Сз\С0.

Пусть С* — минимальная зависимая подсистема в С, уравнение y из С* принадлежит также С\С0, но не принадлежит Сз. Из уравнений (С\С0)\{y} следует Сз, значит, из них следует и С*∩Сз; уравнения (С*\Сз)\{y} содержатся в С\С0. Значит, из (С\С0)\{y} следует подсистема С*\{y} , а из .неё — y. Значит, С\С0 зависима, что противоречит условию. Если С* не содержит уравнений из С0, то есть принадлежит С\С0, то тоже С\С0 оказывается зависимой. Значит , все уравнения из С* принадлежат Сз, и среди них есть некоторые уравнения С0* из С0.

Пусть Fи — F0=Hи(Fи) — решение задачи Н(Fи/F0) относительно F0, h(F) — уравнение зависимости Сз. Подставим выражения F0=Hи(Fи) в h(F). Если при этом не получится тождество, то получится уравнение, связывающее Fи. Тогда задача Н(Fи/F0) переопределена, что противоречит условию. Значит, h(F) следует из Н(Fи,F0), зависимость С* следует из системы зависимостей Сз, Сз — базис завсимостей в С.

(4а). Следствие. Сохраним обозначения. Если Сз - множество минимальных зависимостей в С(X), для Сз существует формально не переопределённая задача Н(Fи/F0), имеющая одинарный план решения, и для С\С0(X) также существует задача, имеющая одинарный план решения, то Сз — базис зависимостей в С. Доказательство: см. далее утверждения (7а), (11).

(5). Рассмотрим некоторые типы вывода зависимостей - следствий из других зависимостей.

(а). Если в системе С одно уравнение функционально зависит от остальных уравнений при условии P(X), то и любое другое уравнение y из С функционально зависит от остальных уравнений при условии P(X) & Q(X) (где Q(X)—предикат, истинный, когда выражение зависимости опрелелено, & - логическая связка и). Q(X) будем называть *условием определённости зависимости уравнения* y в С. Случай ложности Q(X) нужно рассмотреть как отдельную альтернативу.

Соответствующую зависимость можно получить, решая уравнение зависимости относительно соответствующей переменной f_j. Если решение — многознаная функция, то нужная зависимость — одна из её ветвей (в силу непрерывности зависимостей и ветвей, но здесь понятие непрерывности считаем недоступным, поэтому просто предполагаем это как самостоятельное допущение). Для выделения ветви прелагаем использовать подходящее условие.

Условие одновременной определённости зависимостей каждого уравнения от остальных в зависимой системе С будем называть *условием определённости зависимости системы* С.

. Считаем, что для каждой альтернативы, для каждой зависимости при условии её определённости, для каждого уравнения, входящего в неё, имеется однозначная функция зависимости этого уравнения от остальных.

См. выше пример 2, условие определённости ~(e). О разделении альтернатив см. далее .

(б). Если в системе С имеются две независимых зависимости зависимых подсистем C_1,C_2, в каждую из которых входит общее уравнение y, то система С* - объединение систем C_1,C_2, из которой удалено уравнение y, зависима при ус (см. далее утверждениие (9) и предположение (п1г))ловии определённости зависимостей C_1 и C_2.

Действительно, из С1 без y выводим y, после чего С2, в которой появилось и y, становится снова зависимой.

(в). Если в системе С имеется независимая система зависимостей, состоящая из k зависимых подсистем $C_1,...,C_k$, в каждую из которых входят, кроме прочих, общие k уравнений $y_1,...,y_k$, то системы $(C^*,y_1),...,(C^*,y_k)$ — зависимые при условии определённости всех зависимостей. Здесь С* - объединение систем $C_1,...,C_k$, из которого удалены $y_1,...,y_k$.

Соответствующие зависимости можно получить, решая систему из k уравнений зависимостей для $C_1,...,C_k$ относительно k переменных $f_1...f_k$, соответствующих уравнениям $y_1...y_k$. Как и в утведении (а), предполагаем, однозначность решения.

О зависимости уравнений ещё см. далее утверждение (11).

Альтернативы.

Считаем, что альтернативные уравнения и подсистемы уравнений указываются в модели вместе с условиями их выбора. При этом могут не разделяться альтернативы, например, соответствующие различным корням уравнения, если их выбор не влияет на планирование. Такие альтернативы будем называть *совместимыми*. Решение в таких случаях удобно записывать при помощи связки «или соответственно».

Например:

Два шарика начинают двигаться одновременно в одном направлении из одной и той же точки: один — равномерно, второй - с начальной скоростью v_2=2м/с и постоянным ускорением a=-1м/с2 . С какой скоростью v_1 должен двигаться первый шарик, чтобы столкнуться со вторым на расстоянии s=1,5м ?

Время, через которое второй шарик будет на расстоянии s=1,5м

$t=(v_2\pm_1(v_2^2-2|a|s)^{1/2})/|a|=(2\pm_1 1)$сек $=(1\vee_1 3)$ сек ; требуемая скорость $v_1=s/t=(1{,}5\vee_1 0{,}5)$м/сек.

Здесь «$\vee_1$» – «или соответственно, группа 1». Во всех выражениях из того же примера, содержащих такую же связку, выбираются одновременно первые или одновременно вторые альтернативы. $x\pm_1 y$ обозначает то же, что $(x+y)\vee_1(x-y)$.

Другой логический знак «$\Rightarrow$» удобно применять для записи условий с соответствующими альтернативами. Пусть в предыдущем примере числовые значения не заданы.

$v_2^2-2|a|s<0 \Rightarrow\varnothing$(здесь «$\varnothing$» обозначает отсутствие решений), второй шарик не достигнет заданного расстояния;

$v_2^2-2|a|s=0 \Rightarrow t=v_2/|a|,\ v_1=s|a|/v_2;$

$v_2^2-2|a|s>0 \Rightarrow t=(v_2\pm(v_2^2-2\}a|s)^{1/2})/|a|,\ v_1=s/t.$

Отметим, что система последних трёх выражений интерпретируется не как их конъюнкция в логике с двумя значениями истинности (истинно и ложно), так как в случае $v_2^2+as<0$ последнее выражение не определено, следовательно, не определена и вся конъюнкция. Это выражение можно интерпретировать как условный оператор в программе вычислений, или, логически, — как выражение в трёхзначной логике со значениями истинности «истинно», «ложно», «не определено». Здесь не будем рассматривать эту логику.

Будем считать, что *ветвления модели* устроены иерархически: система уравнений (назовём её *главной ветвью*) состоит из двух подсистем, *ветвящейся* и *не ветвящейся* (*общей* для всех ветвей); любая из них может быть пустой. Ветвящаяся подсистема содержит несколько *ветвей*, каждая из которых — система уравнений, состоящая из ветвящейся и общей подсистем (это индуктивное определение). Ветвь представляет условие и систему, выполняющуюся при этом условии. Описание ветви в модели сопровождается указанием состава зависимостей, которые связывают некоторые уравнения её не ветвящейся части и, возможно, некоторые уравнения ветвей, в которые данная ветвь входит.

Ветви, не имеющие решений, в модель не включаются.

Не ветвящаяся система.

Рассмотрим сначала планирование решения и составление задач для не ветвящейся системы. Для иллюстрации будем использовать следующий пример:

Пример 3.

Объект модели: тело брошено под углом к горизонту и летит от некоторой высоты до земли; на тело действует сила тяжести, сопротивлением воздуха пренебрегаем.

Переменные модели:

h – высота над землёй начальной точки движения;

v_0 – модуль вектора начальной скорости;

A – угол между вектором начальной скорости и горизонтальной плоскостью;

v_{0x} – горизонтальная проекция вектора начальной скорости;

v_{0y} – вертикальная проекция этого вектора;

t – время полёта от начальной точки до земли;

x — длина горизонтальной проекции траектории;

v – модуль вектора конечной скорости;

v_x – горизонтальная проекция вектора конечной скорости;

v_y – вертикальная проекция этого вектора.

Поскольку $v_{0x}=v_x$, заменим в модели v_{0x} на v_x .

Система уравнений модели:

(1) $v_x=v_0\cos A$;

(2) $v_{oy}=v_0\sin A$;

(3) $v_y=v_{0y}-gt$;

(4) $h=-v_{0y}t+gt^2/2$;

(5) $v_y^2-v_{0y}^2=2gh$;

(6) $v^2=v_x^2+v_y^2$;

(7) $x=v_xt$.

Зависимая подсистема:

(3)(4)(5)

Задача:

Дано: A, v, x. Найти остальные переменные.

Пусть модель представлена независимой системой уравнений, и на ней задана здача с минимумом данных.

План решения задачи представим в общем виде как последовательность пар вида: (уравнение — переменная, вычисляемая из него), или, в более общем случае – (*минимальная подсистема уравнений* — переменные, вычисляемые из неё). На первом шаге «из ничего» вычисляются исходные переменные.

План решения примера 3:

( ) – A, v, x;

(1),(2),(6),(7),(3) — v_{0y}, v_x, v_y, t, v_o;

(5) – h.

Минимальность подсистемы уравнений означает, что никакую часть этой подсистемы при имеющихся *входных* данных (это исходные данные задачи и ранее вычисленные переменные) нельзя решить отдельно. Такой шаг плана решения будем называть его *множественной компонентой*, а план, содержащий хотя бы одну множественную компоненту — *множественным планом*. Если шаг решения представлен парой вида (уравнение — переменная), то его будем называть *одинарной компонентой* плана, а

план, состоящий только из одинарных компонент — *одинарным планом*.
Включение компоненты в план будем называть *срабатыванием* её системы уравнений. Одиночное уравнение готово к срабатыванию на некотором шаге построения плана, если к этому моменту уже спланировано вычисление всех входящих в него переменных, кроме одной. Тогда это уравнение с этой переменной можно выбрать и включить в план как очередную одинарную компоненту. Подсистема C из k уравнений готова к срабатыванию, если к этому моменту уже спланировано вычисление всех кроме k входящих в C переменных и никакая подсистема, входящая в C, к срабатыванию не готова. При выборе этого срабатывания в план включается множественная компонента — эта система C и эти k переменных.
Обычно при решении множественной компоненты какие-то искомые переменные принимают за *базовые* (назовём их так), остальные *выражают* через них и получают *базовую систему* с меньшим числом уравнений и неизвестных. Базовую систему составляют *базовые уравнения* минимальной подсистемы, которые после подстановок не содержат никаких искомых переменных кроме базовых. В школьных задачах по физике чаще всего достаточно выбрать одну базовую переменную, и тогда через неё можно выразить все остальные переменные минимальной подсистемы, получив одно базовое уравнение. План выражения одних переменных через другие из одних уравнений и подстановки в другие уравнения можно представить как одинарный план решения системы уравнений данной компоненты, в котором роль исходных переменных выполняют базовые переменные.
Пусть в примере 3 базовая переменная v_o;
(1) $v_{0y}=v_o\sin A$; (2) $v_x=v_0\cos A$;
из (6) $v_y=(v^2-v_0{}^2\cos^2 A)^{1/2}$;
из (7) $t=(x/(v_0\cos A))$
подставив все эти выражения в (3), получим после преобразований
базовое уравнение относительно v_o:

$$v_o{}^4-(2gx\cdot tgA+v^2)v_o{}^2+g^2x^2/\cos^2 A=0.$$

Это уравнение школьники могут решить. К сожалению, вообще говоря, нет гарантий решаемости школьными методами базовых систем, если эти системы не сводятся к линейным или другим решаемым в школе типам систем относительно базовых переменных (см., например, в справочнике [3], стр. 168 - 175). Но для получения численного решения можно воспользоваться калькулятором, например, unicalc.

Возможно, что при заданных значениях исходных данных задача не имеет решения. Это не означает, что план решения не применим: действуя по плану, на каком-то шаге получим неразрешимое уравнение.

Указанный метод решения представляет основу и при решении систем уравнений другими методами, в которых решаемая система получается при помощи предварительных эквивалентных преобразований: подстановок или/и замены одного из уравнений результатом обратимой операции над уравнениями.

(6). Если полная задача имеет несколько планов решения, полученных описанным ранее методом, и при реализации этих планов возникают только одноразрешимые уравнения, то все эти планы решения включают одинаковое число уравнений. Ведь из каждого уравнения вычисляется ровно одна переменная (сразу или сначала выражается, а позже вычисляется), а вычислить нужно все переменные, кроме исходных данных.

Зависимости исходных переменных.

Структурированность.

Рассмотрим пример. Система уравнений A(x,y,u,v):

$f1(g(x,y),u,v)=0$;

$f2(g(x,y),u,v)=0$.

Это два уравнения с четырьмя переменными, но если построить задачу с исходными переменными u, v (тогда формально система A готова к срабатыванию), то, в зависимости от того, удовлетворяют ли значения u, v уравнению y(u,v): $f1^{-}(u,v)=f2^{-}(u,v)$, она будет либо неразрешимой, либо одновременно переопределённой, неопределённой и зависимой при условии выполнения уравнения y(u,v).

Здесь $f1^{-}(u,v)$ и $f2^{-}(u,v)$ – решения первого и второго уравнений соответственно относительно g(x,y). При другом выборе пары исходных переменных система A(x,y,u,v) может быть точечной. Исходные данные u,v в такой ситуации называем перекошенными. Причина — зависимость исходных данных y(u,v).

Заменим систему A(x,y,u,v) на эквивалентную B(x,y,u,v):

$f1(g(x,y),u,v)=0$;

$f1^{-}(u,v)=f2^{-}(u,v)$.

Здесь зависимость y(u,v) явно выражена и при составлении задачи переменные (u,v) не могут быть выбраны исходными.

Перейдём к формальному рассмотрению.

Пусть задача с исходными переменными Хи на модели с системой уравнений С(Х) переопределена, из С(Х) следует некоторое уравнение y(Хп), где Хп из Хи. Вместо системы С(Х) можно взять любую её максимальную независимую подсистему Сн(Х), из неё тоже выводимо y. Возможны два случая.

1). При планировании решения в Сн описанным ранее методом появляется не сработавшее уравнение y*, в котором все переменные вычислены (если появляется подсистема с числом уравнений, большим числа не вычисленных переменных,то из неё можно удалять уравнения). Уравнение y* и задачу в этом случае назовём *формально переопределёнными*. Формально переопределённая задача является переопределённой, так как после подстановки в y* выражений входящих в него переменных через входные переменные получится уравнение зависимости исходных переменных.

2). В плане решения есть *внутренне переопределённая компонента*, то есть компонента C1(X1)= C1(X1вх,X1вых)–X1вых, где X1вх — множество переменных, входящих в C1 и вычисленных раньше формального срабатывания C1, X1вых=X1\X1вх, |C1|=|Xвых|, и из C1(X1) следует некоторое уравнение y1(X1вх).

Задача с внутренне переопределённой компонентой решения является переопределённой, зависимость между её исходными переменными получается при подстановке в зависимость между входными переменными переопределённой компоненты выражений этих входных переменных через исходные.

(7). Компонента формального плана решения может быть либо внутренне переопределённой, либо не разрешимой, либо зависимой, либо точечной.

Доказательство. Применим такой метод решения задачи C1(X1)–X1вых: выражаем одну из искомых переменных x_1 через остальные из одного уравнения y_1 из C1 и подставляем во все остальные уравнения; с получившейся системой C2 проделываем такую же операцию, выражая x_2 из y_2 из C2 и получая после подстановок C3, и т. д., пока после подстановки x_{k-1} не получим систему уравнений Cк без неизвестных или с одной неизвестной x_k. Если Cк содержит уравнение без неизвестных, содержащее входные переменные, то C1 – внутренне переопределённая система. Если Cк содержит тождество, то уравнение из системы C1, подстановкой в которое оно получено, удовлетворяется в следствиии остальных уравнений, C1 — зависимая система. Если Cк состоит из более чем одного уравнения с одной неизвестной переменной, то, по допущению (1), каждое из них одноразрешимо, получим чёткие решения. Если они не пересекаются, то C1 не разрешима. Если пересекаются, или если Cк состоит из одного уравнения с одной неизвестной, то получим чёткое решение для xк. Если число неизвестных |C1|=|Xвых|>к, то уравнения из C1, не вошедшие в последовательность y1,y2,...,yк, удовлетворяются в следствии удовлетворения уравнений y1,y2,...,yк, C1 – зависимая система. Если C1={ $y_1,y_2,...,y_k$}, то, подставляя решение x_k в y_{k-1}, получим x_{k-1}, и т. д., до x_1, получим чёткое решение C1. Конец доказательства.

(8). План решения разрешимой формально не переопределённой задачи на независимом множестве уравнений доставляет чёткое решение, если все его множественные компоненты внутренне не переопределены.

Пусть C(X) — система уравнений модели. Её независимую подсистему C1(X1) с выделенным множеством выходных переменных X1вых (входные переменные – X1вх=X1\Xвых) будем называть *возможной компонентой* системы C, если существует формально не переопределённая задача на максимальном независимом подмножестве уравнений системы С, для которой С1-Х1вых является компонентой плана решения.

Подсистему С1 системы С будем называть *переопределимой подсистемой*, если существует возможная внутренне переопределённая компонента С1-Х1вых . Систему, в которой есть переопределимые подсистемы, будем называть *переопределимой системой*.

Пусть С1(Х1)-Х1вых — возможная переопределённая компонента системы С(Х), Х1вх=Х1\Х1вых. Тогда существует уравнение у(Х1вх), выводимое из С1(Х1), система (С1, у) — зависимая. Пусть С11 из С1 — минимальная подсистема, для которой (С11,у) зависима, и уравнение у1 входит в С11. Заменим у1 на у в С (и, следовательно, в С1). Получим систему $С_1$ с подсистемой $С1_1$, эквивалентную С с подсистемой С1 при условии определённости зависимости (С1,у). Такую операцию будем называть *шагом структурирования* системы С.

В результате шага структурирования задача, приводившая к компонненте С1(Х1)-Х1вых, превращается в формально переструктурированную (у — формально переструктурированное уравнение), а внутренне переструктурированная компонента С1(Х1)-Х1вых исчезает.

Систему уравнений, в которой нет переопределимых подсистем, будем называть *структурированной*.

Доказать, что структурированную систему всегда можно получить из переопределимой последовательностью шагов структурирования, у меня не получилось.

(п1в). Будем предполагать, что система уравнений модели структурирована.

Разделение альтернатив.

Систему уравнений, в которой все возможные при каком-либо формально правильном плане решения компоненты — опрелённые (имеют единственное решение при любых допустмых значениях входных переменных), будем называть *компонентно определённой.* В частности, в компонентно определённой системе каждое уравнение имеет единственное решение относительно каждой из входящих в него переменных.

Компонентно определённая система уравнений является независимой. Любая полная задача на ней, имеющая формально правильный план решения, является определённой.

Если компонента плана решения задачи имеет несколько решений, то решение задач после неё может продолжаться для каждого решения этой компоненты отдельно. Такие продолжения будем называть компонентными ветвями.

(9). Если система уравнений базовых зависимостей в системе С при условии Р компонентно опрелелённая, то все компонентные ветви любой точечной задачи при условии Р совместимы.

Доказательство.

Разделение ветвей происходит тогда, когда входящее в план решения системы

С уравнение $y=(a1(X)=a2(X))$ (где X — набор переменных), из которого вычисляется или выражается переменная x, имеет несколько решений, например, $x1(X\backslash x)$ и $x2(X\backslash x)$. Тогда уравнение y эквивалентно дизъюнкции некоторых определённых уравнений, например: $y'=(a1'(X)=a2'(X))$, имеющего решение x1, и $y''=(a1''(X)=a2''(X))$, имеющего решение x2. Если формально правильный план решения подходит для y', но не подходит для y'', то на некотором шаге этого плана должна вычисляться или выражаться некоторая переменная x* из некоторого уравнения y*, в которое она фактически не входит после произведённых до этого подстановок и преобразований. Тогда возможны следующие случаи: y* противоречиво (тогда план считаем применимым, ветвь будет отброшена в процессе вычислений по плану); y* тождественно (тогда y* зависимо от некоторой подсистемы C*, включающей y, при $y=y''$, но не зависимо от той же подсистемы при $y=y'$); y* налагает ограничение – соотношение между входными переменными (тогда y* зависимо от C* при выполнении этого ограничения). Считаем, что в последних двух случаях требуется произвести разлеление альтернатив в модели (см. далее), так что они не должны встретиться при планировании решения отдельной ветви.

Но если все зависимости, содержащие y, действуют одновременно и для y', и для y'', то такой ситуации не возникнет. Так будет, если выполнение y — следствие выполнения остальных уравнений каждой зависимой подсистемы, в которую y входит, а это обеспечено, если функции этих зависимостей однозначны и обратимы относительно каждой переменной для частей уравнений.

Модель с альтернативами будем называть *совместно планируемой*, если для каждой альтернативы все компонентные ветви любой точечной задачи могут решаться по одному и тому же плану.

(п1г). Будем предполагать, что модели представлены в совместно планируемом виде.

Пример. Система уравнений

(a) $|x|=|y+z|$;

(b) $v=x+y-z$;

(c) $v=2x+2y$.

Обозначим переменные для частей уравнений (a), (b), (c) через a, b, c соответственно.

Из (b) и (c) следует (a): $a=|c-b-x|$. Но решение этого уравнения зависимости относительно b или c не однозначно: $b=c-x\pm a$, $c=b+x\pm a$. Причём эти формулы могут не выражать зависимости уравнений, так как правые и левые части уравнений могут быть связаны разными формулами (например, правые — с «+», а левые — с «–»).

Левые части: $c=b+x+a$ при $v=v+x+|x|$, $x=-|x|$, $x\leq 0$;

c=b+x-a при x=|x|, x≥0.

Правые части: c=b+x+a при 2x+2y=x+y-z+x+|y+z|,y+z=|y+z|, y+z≥0;

c=b+x-a, при y+z≤0.

Функции, связывающие левые и правые части, совпадают, если знаки x и y+z различны: x(y+z)≤0. Тогда c={b+x+a при x≤0 и y+z≥0, b+x-a при x≥0 и y+z≤0}. Если x(y+z)>0, то a=(x=y+z), система (a,b,c) независима и приводится к виду: x=0, y+z=0, v=2y.

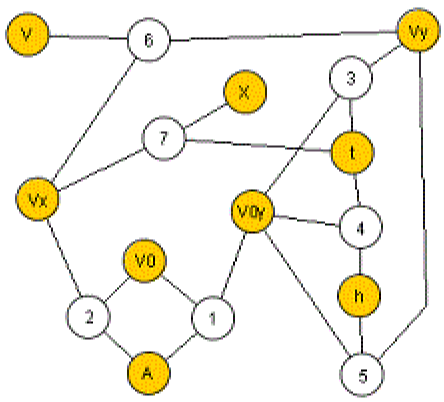
Другой пример: ранее рассмотренный пример 2. Там все зависимости можно выразить определёнными уравнениями, линейными относительно переменных для частей уравнений модели. Хотя некоторые уравнения модели имеют многозначные решения относительно некоторых переменных, различные решения модели могут вычисляться по одинаковому плану.

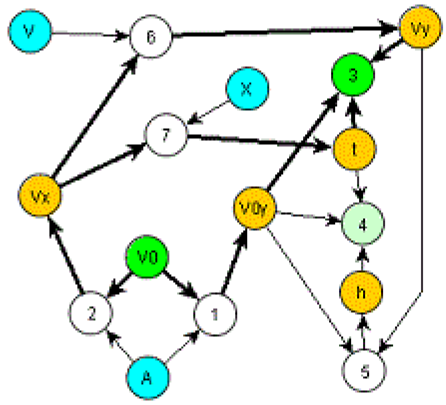
Представление графом.

Используемые здесь понятия см., например, в [4]. Здесь требуемые определения и доказательства также будут приводиться, поскольку теорию графов не проходят в школе. Доказательства утверждений, отмеченных «*», вынесены в приложение. Для рисования иллюстраций я использовал редакор yEd.

Граф — это множество *вершин* и множество неупорядоченных пар вершин, называемых *рёбрами*. *Ориентированный граф* – это множество вершин и множество упорядоченных пар вершин, называемых *дугами*. Таким образом, граф, неориентированный или ориентированный, — это математическая модель множества объектов с множеством связей между их парами.

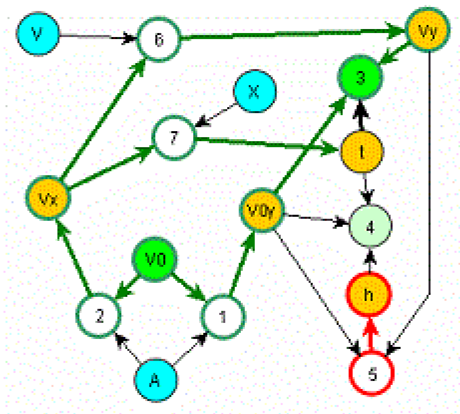
Модель объекта можно представить неориентированным *бидольным графом* (графом с двумя типами вершин, в котором рёбра связывают только разнотипные вершины), в котором вершины одного типа соответствуют переменным, а вершины другого типа — уравнениям; рёбра связывают каждую вершину уравнения с вершинами вхолящих в него переменных.





Задачу и план её решения можно представить ориентированным бидольным графом, получающимся при следующей ориентации описанного графа модели объекта (превращении рёбер в дуги приписыванием им направления): дуга идёт от вершины уравнения к вершине переменной, если эта переменная вычисляется или выражается из этого уравнения; все остальные рёбра ориентируются от вершин переменных к вершинам уравнений.

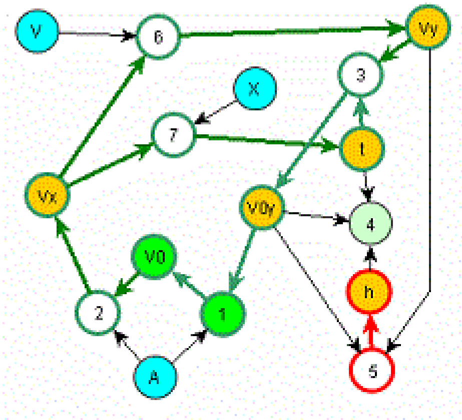
Таким образом, в каждую вершину *исходной переменной* (заданной по условию задачи; на рисунке выделены синим) не заходит ни одной дуги, также не заходит ни одной дуги в вершины базовых переменных минимальных подсистем (на рисунке — зелёная, v_0), а в каждую из остальных вершин переменных заходит ровно одна дуга, идущая из вершины уравнения, из которого она вычисляется или выражается. Вершины избыточных уравнений, присутствующих в модели, но не используемых в решении и удовлетворяющихся тождественно в следствии него , оказываются без исходящих дуг (на рисунке — (4), синяя); их будем называть *внешними,* а остальные вершины уравнений - *внутренними.* Также без исходящих дуг оказываются вершины базовых уравнений минимальных подсистем (на рисунке — (3), зелёная).



Полученный таким образом граф назовём *разомкнутым графом решения* задачи. Его подграфы (подграф графа Г — это граф, представляющий порождающее его подмножество вершин графа Г со всеми дугами, связывающими эти вершины в Г), порождённые вершинами уравнений минимальных подсистем и вершинами вычисляемых в них переменных, включая базовые, будем называть *компонентами* этого графа (в рассматриваемом примере две компоненты, выделенные на рисунке зелёными и красными линиями).

Возможно и другое изображение базовых уравнений.
Решение минимальной подсистемы всегда можно спланировать так, чтобы первая базовая переменная входила в первое базовое уравнение непосредственно, вторая — во второе непосредственно, и т. д.* (решаемость школьными методами, как уже отмечалось, не гарантирована).

На рисунке справа базовая переменная v0 входит в базовое уравнение (1). Этот план получен модификацией предыдущего плана: изменено направление дуг на пути из (1) в (3).
(*Путь* — это последовательность вершин и дуг, проходимых при движении по направлению дуг; путь начинается и заканчивается вершинами).



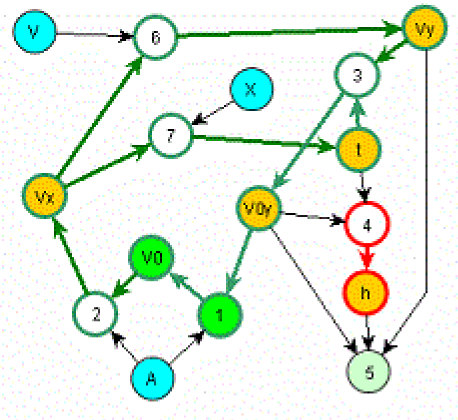
Пусть эту базовую систему планируем решать тоже выражением одних переменных через другие и подстановками. Выразим первую базовую переменную через остальные из первого базового уравнения и отразим это тем, что сориентируем ребро от вершины этого первого уравнения к смежной вершине этой первой переменной; подставим полученное выражение в остальные уравнения. Затем проделаем аналогичную операцию для второй базовой переменной во втором базовом уравнении, и т. д. Получим граф, в котором вершины базовых переменных имеют по одной заходящей дуге, а вершины базовых уравнений — по одной исходящей дуге, как и вершины других искомых переменных и уравнений соответствующей минимальной системы (вершины базовых переменных и базовых уравнений как бы нейтрализовали друг друга). Такой граф назовём *замкнутым графом решения* задачи.

(9). В замкнутом графе решения задачи вершины уравнений минимальных подсистем и вершины искомых переменных, определяемых из них, порождают подграфы, являющиеся его *бикомпонентами* (= компонентами сильной связности)* (*компонента сильной связности* — это максимальный подграф, в котором из любой вершины есть путь к любой другой вершине (следовательно, есть и обратный путь).

Ориентацию графа модели, в которой в каждую вершину переменной заходит не более одной дуги, а из каждой вершины уравнения исходит не более одной дуги, будем называть *формально правильной.* И замкнутый, и разомкнутый графы решения задачи представляют формально правильные ориентации. Формально правильно ориентированный граф с выделенными (отмеченными определённым образом) вершинами исходных и базовых переменных и внешних и базовых уравнений, для которого выполнены описанные для разомкнутого или замкнутого графа решения правила ориентации дуг, примыкающих к выделенным вершинам, будем называть *разомкнуто или замкнуто правильно ориентированным* (соответственно). Для замкнутого плана при этом вершины базовых уравнений и переменных выбирать не обязательно. *План решения системы уравнений С*, отображаемый правильно ориентированным графом, будем называть *правильным*, если уравнения, соответствующие внутренним вершинам, составляют максимальную независимую или максимальную независимую при некотором едином условии подсистему системы уравнений С.

Ориентацию графа будем называть *одинарной*, если ей соответствует одинарный план решения.

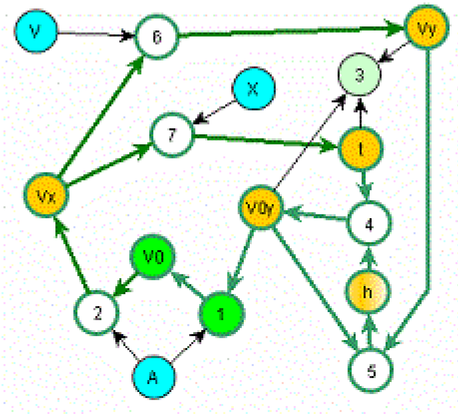
(10а). Любую замкнуто правильную ориентацию графа решения задачи с минимумом данных можно преобразовать в любую другую замкнуто правильную ориентацию решения этой же задачи инверсией попарно не пересекающихся циклов и/или путей из внутренних вершин уравнений во внешние.* При этом в полученный план не должна входить целиком ни одна зависимая подсистема. Как это обеспечить, см. далее.



(*Инверсией*, или *инвертированием* пути называем изменение направления всех входящих в него дуг. Два пути называем *не пересекающимися*, если они не содержат общих вершин (а следовательно и дуг).

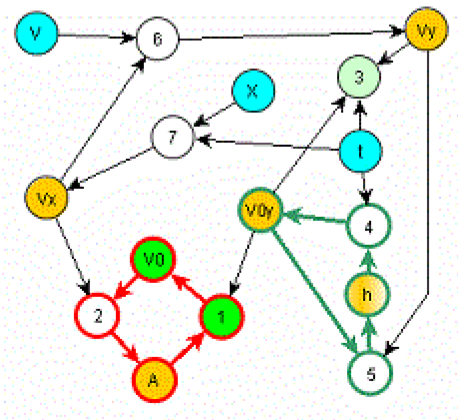
В рассматриваемом примере план *решения 2* получается из предыдущего плана решения инверсией пути (5) – h — (4).

План *решения 3* получается из плана первоначального решения инверсией пути (3) – v0y — (4). В этом плане – одна компонента: все внутренние вершины уравнений входят в циклы, имеющие между собой общие дуги.



(10б) Любую замкнуто правильную ориентацию графа решения полной задачи можно преобразовать в замкнуто правильную ориентацию решения любой другой полной задачи на этой же модели инверсией попарно не пересекающихся путей из исходных переменных в искомые переменные.*

План решения на рисунке справа получен из плана «решение 3» инверсией пути
A – (2) – vx – (7) – t.
Получилась задача: «даны t, x, v, найти остальные переменные», и её план решения:
(7) — v_x; (6) – v_y; (4),(5) – v_{0y},h; (1),(2) – v_0,A.
Более рациональное решение этой задачи можно получить инвертированием пути (4)– v_{0y} –(3).



A1. Алгоритм планирования решения полных задач.

Имеются: неориентированный граф структурированной модели, список списков уравнений зависимых подсистем, список исходных переменных задачи. Нужно построить граф решения полной точечной задачи.

1. Отмечаем на графе вершины исходных переменных. Ориентируем все примыкающие к ним рёбра в направлении от них к вершинам уравнений.

2. Ищем вершину уравнения, к которой примыкает только одно неориентированное ребро. Если нашли, то проводим её *срабатывание*: ориентируем это ребро в направлении от этой вершины к вершине переменной, а все другие рёбра, примыкающие к этой вершине переменной, ориентируем в направлении от них к вершинам уравнений.

Повторяем это, пока находятся вершины уравнений, готовые к срабатыванию.

3. Если не осталось неориентированных рёбер, то переходим к завершающей разметке, п. 4. Если ещё не ориентированные рёбра есть, но готовых к срабатыванию одиночных вершин нет, то ищем *готовую к срабатыванию компоненту* с несколькими вершинами уравнений. Для этого:

3.1 Ищем *контур* (контур — это замкнутая (конец совпадает с началом) последовательность примыкающих дуг к другу вершин и неориентированных рёбер, аналог цикла в ориентированном графе), обладающий следующим свойством:

1) к вершинам его *уравнений* не примыкают ещё не сориентированные рёбра, лежащие вне контура (такие вершины уравнений компоненты будем называть *завершёными*, а вершины уравнений заготовки компоненты, к которым примыкают неориениираваные рёбра, лежащие вне этой заготовки, – *незавершёнными*).;

2) при добавлении вершин уравнений контура к множеству уже сработавших вершин в нём не образуется подмножества вершин зависимых уравнений. При этом нужно учитывать, вообще говоря, не только зависимые системы, непосредственно указанные в описании модели, но и их следствия. Как это делать – опишем позже.

Если таких контуров несколько, то выбираем один из самых коротких, с более простыми уравнениями.
3.2. Если таких компонент не нашлось, то пытаемся добавить к выбранной части компоненты маршрут, начинающийся и заканчивающийся в её вершинах, один конец которого — незавершённая вершина уравнения, а другой — вершина переменной, такой, чтобы полученная компонента удовлетворяла требованиям п. 3.1. Если не удалось, повторяем этот п., выбирая другие начальные части компоненты или добавляя дополнительные маршруты (маршрут — это последовательность примыкающих друг к другу вершин и рёбер, аналог пути в ориентированном графе).

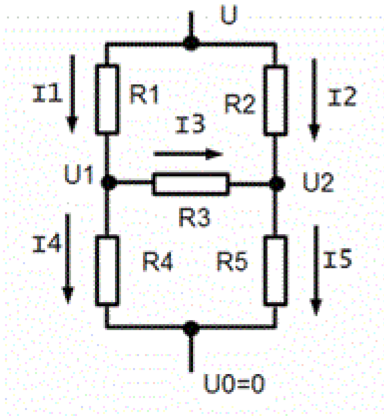
3.3. Если компоненту нашли, то проводим её *срабатывание*:
- ориентируем рёбра, лежащие внутри неё, превращая контуры в циклы, если строим замкнуто правильную ориентацию;
если строим разомкнуто правильную ориентацию, то нужно выбрать базовые переменные и затем сориентировать компоненту, пользуясь только срабатываниями одиночных уравнений, см. п. 2);
- ориентируем все рёбра, лежащие вне компоненты, но примыкающие к вершинам переменных компоненты, от вершин этих переменных к вершинам уравнений.
- возвращаемся к п.2.
4. Завершение. Отмечаем не срабатывавшие избыточные вершины уравнений (все примыкающие дуги направлены к ним) как внешние. Записываем полученный план решения.

Детально алгритм планирования решения полной задачи на неветвящейся модели будет описан ниже для табличного представления.

Заметим, что при срабатывании одного уравнения (п.2) проверять образование зависимых подсистем не приходится, так как срабатывающее уравнение содержит переменную, не встречающуюся в уравнениях, сработавших ранее. При срабатывании множественной компоненты тоже вычисляются переменные, не встречавшиеся ранее, но в процессе вычисления вводятся базовые переменные из их числа, и, если в компоненте есть зависимые уравнения, в результате подстановок получим базовые уравнения, не зависящие от них.
Но если базовое уравнение содержат исходную переменную, не встречавшуюся до этого и, поэтому, не сокращающуюся в преобразованиях, то его независимость от базовых переменных означает переопределённость, чего по предположению (п1г) не должно быть.

(11). Любая одинарно ориентируемая подсистема уравнений независима. Если в множественной ориентации подсистемы уравнений C каждое базовое уравнение z каждой множественной компоненты S содержит какую-либо исходную переменную, не входящую в уравнения, стоящие в данном плане решения и во всех эквивалентных ему планах раньше z (раньше вне S или раньше внутри S), то C — независимая подсистема.

Пример. Мост сопротивлений под постоянным напряжением. Обозначения переменных см. на рисунке.



Уравнения:
(1) U–U1=I1R1; (2) U–U2=I2R2;
(3) U1=I4R4; (4) U2= I5R5;
(5) U2–U2=I3R3;
(6) I1=I3+I4; (7) I3+I2=I5;
(8) I1+I2=I4+I5.

Зависимая подсистема (6),(7),(8).

Задача: дано U,R1,R2,R3,R4,R5. Найти: остальные переменные.

План вычислений: базовые переменные I4,I5.
I4(3)U1; I5(4)U2; U1,U2(5)I3; I1,I3(6)I4; I3,I5(7)I3; (I1,U1(1); I2U2) – базовые уравнения, содержащие исходную переменную U. Следовательно, подсистема (1),(2),(3),(4),(5),(6),(7) — независимая. Если бы выбрали в качестве базового уравнение (8), то полчили бы тождество, но в (8) нет исходных переменных.

Композиция моделей. Начальные условия. Естественный поток вычислений.

При математическом описании физических процессов они разбиваются на типовые элементарые части, связанные между собой общими переменными. Например, если описывается движение мячика, ударяющегося о потолок, то процесс разбивается на движение до удара, удар и движение после удара, скорость в момент перед ударом есть скорость движения до удара в момент достижения потолка и т. д. Движение после удара описывается такими же уравнениями, как и движение до удара, но при другом именовании переменных.

Модель M1, полученная из модели M переименованием переменных, будем называть *экземпляром модели* M.
Модель M, полученную объединением переменных X1, ..., Xк и уравнений C1, ..., Cк, моделей M1, ..., Mк, будем называть *композицией моделей* M1, ..., Mк. При этом считаем, что множества переменных X1, ..., Xк могут пересекаться между собой, а множества уравнений C1, ..., Cк попарно не пересекаются.

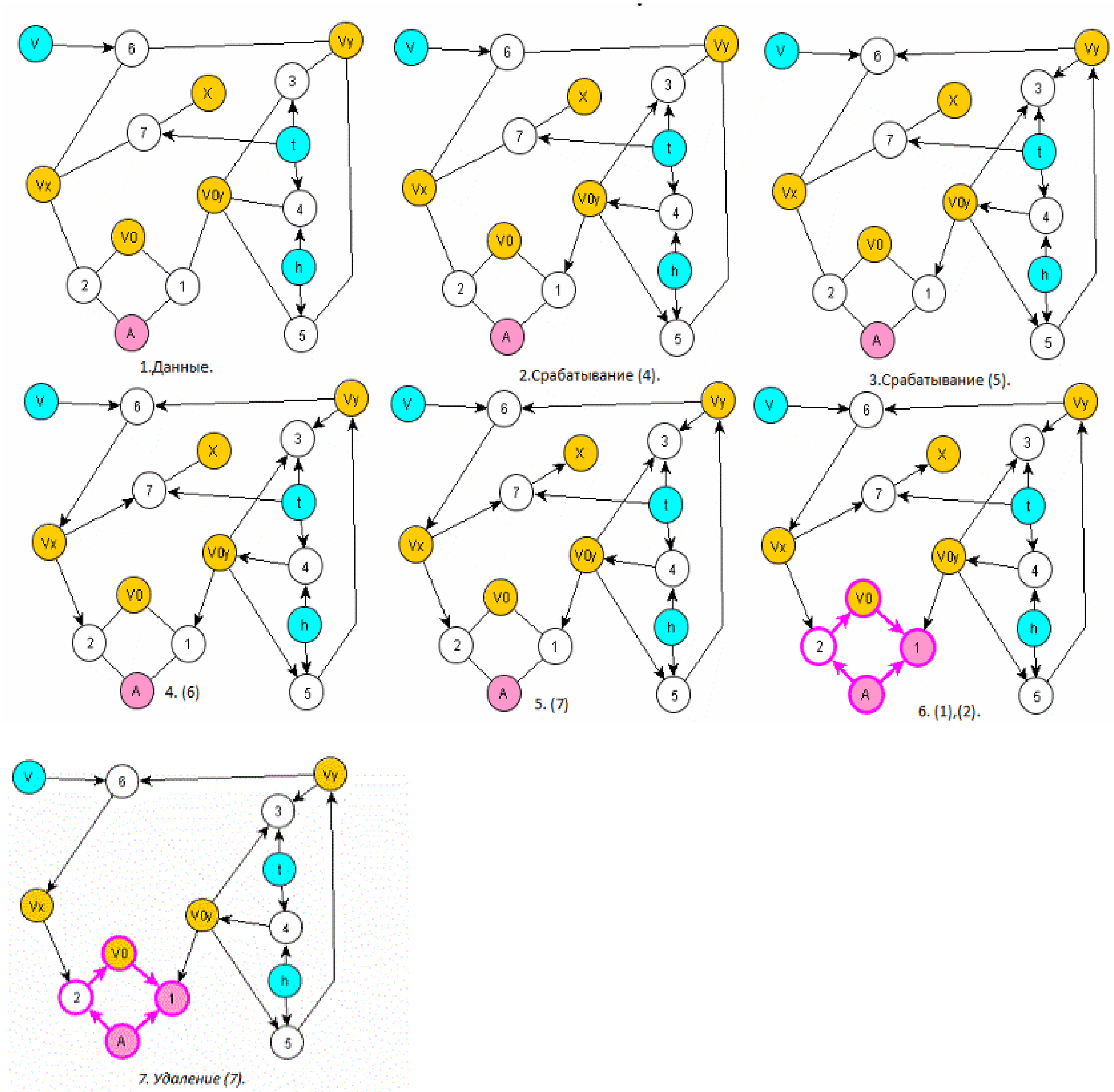
Для описания развёртывания во времени физического процесса, модель которого задана композицией экземпляров *физических законов* — моделей элементарных типовых физических процессов, – задают в качестве исходых переменных *начальные условия* — значения переменных в момент начала процесса, включая не изменяющиеся во время процесса переменные, описывающие конструкцию и другие внешние условия, – и строят план решения, соответствующий последовательности процесса во времени: если элементарный процесс A происходит после и в следствии элементарного процесса B, то уравнения A срабатывают по плану раньше уравнений

В. Такой план будем называть *естественным потоком вычислений*.
Композиция физических объектов при естественном потоке вычислений может развёртываться и *по структуре* невременных *связей*. При этом готовность к срабатыванию уравнений элементарного объекта опрелеляется только срабатыванием некоторых элементарных объектов из смежных с ним по структуре.
Например, в электрических схемах злементарные физические обекты — сопротивления, ёмкости, индуктивнсти, источники э.д.с., вентили, перекючатели и т. д., а также их соединения описываются экземплярами соответствующих физических законов. Внешнее воздействие может распределяться по структуре, расчёт этого распределения может соответствовать решению сложой компоненты плана вычислений; тогда развёртывание по структуре соответствует плану решения компоненты — выражению всех переменных через базовые переменные (см. предыдущий пример).
Кроме описания элементарных процессов и структурных связей, физические законы описывают *общие связи* физических переменных, которые обычно можно доказать как следствия законов элементаных процессов и связей. Это законы сохранения и общие структурные связи (см. (8) в предыдущем примере). Их использование может сократить решение неполных задач.
Иногда физические законы элементаного процесса представляются множественной компонентой. Пример: абсолютно упругое соударение шаров описывается уравнениями сохранения энергии и момента количества движения. Но добавление этих уравнений не приводит к ситуации сработавшей зависимой подсистемы.
(п11а). Етественный поток вычислений можно описать независимой системой уравнений, соответствующей условиям утверждения (11).

А2. Алгоритм планирования решения неполных задач.

Кроме исходной информации для полной задачи здесь задан список искомых переменных залачи.
Этот алгоритм отличается от описанного алгоритма планирования решения полных задач следующим.
- Срабатывания (п.2,3) выбираются из возможных не в произвольном порядке, а перебором вширь, по ярусам: сначала проводятся все готовые срабатывания текущего яруса; те срабатывания, что стали готовыми в результате них, образуют следующий ярус, и т. д.
- Окончание процесса срабатываний определяется не по окончанию ориентации всего графа модели, а по попаданию в ориентированный подграф вершин всех искомых переменных задачи.
- После окончания процесса срабатываний производится удаление из плана лишних вычислений: оставляется только подграф, образованный всеми путями из исходных переменных задачи к её искомым переменным.

Пример. На модели примера 3 задача: дано: h,t,v; найти: A.
Последовательные шаги 1..7 построения плана решения:



1.Данные. 2.Срабатывание (4). 3.Срабатывание (5). 4. (6) 5. (7) 6. (1),(2). *7. Удаление (7).*

Плану решения задачи на графе её решения взаимно однозначно соответствует *паросочетание* (множество дуг, попарно не примыкающих к общим вершинам), представленное множеством дуг, исходящих из вершин уравнений. Это паросочатание можно представить взаимно однозначной функцией из множества внутренних вершин небазовых уравнений на подмножество вершин переменных. *Подграф решения* задачи, порождённый всеми вершинами переменных и внутренними вершинами уравнений, будем называть *внутренним.* Внутреннему подграфу замкнутого графа решения соответствует *максимальное* (то есть такое, которое нельзя дополнить) паросочетание. Из теории графов известен алгоритм построения максимального паросочетания и теоремы о паросочетаниях. Но я здесь не ссылаюсь на эти материалы, а стараюсь привести нужные формулировки и доказательства в форме, более подходящей, на мой взгляд, для предполагаемой аудитории и конкретной темы.

Планы решения, отличающиеся только порядком расположения составляющих его действий (минимальных подсистем), но не самими действиями, считаем *эквивалентными*.

(12). Все планы решения задачи с минимумом данных на модели (или подмодели, то есть на подмножестве уравнений модели), уравнения которой независимы, эквивалентны.*

Граф зависимостей.

Для определения внешних уравнений, образующихся в процессе планирования решения с учётом зависимостей-следствий, можно использовать метод ориентации графа зависимостей, проводимой аналогично ориентации графа решения.

Графом зависимостей будем называть бидольный неориентированный граф, у которого вершины первого типа соответствуют зависимостям, указанным при описании модели, вершины второго типа — входящим в них уравнениям, и вершина зависимости связана ребром с вершиной уравнения тогда и только тогда, когда это уравнение входит в эту зависимость.

Кроме этого графа можно (но не обязательно) использовать *столбец учёта уравнений*, в котором каждому уравнению системы уравнений модели соответствует своя ячейка, в которую ставится знак «+», когда уравнение срабатывает, и знак «-», когда уравнение определяется как внешнее.

А3. Алгоритм ориентирования графа зависимостей при планировании решения задачи

Пример см. далее.

При срабатывании на графе решения уравнения или компоненты из нескольких уравнений проставляем в столбце учёта уравнений «+» в ячейках этих уравнений и на графе зависимостей ориентируем рёбра, примыкающие к вершинам этих уравнений (если они туда входят) от этих вершин уравнений к вершинам зависимостей. Если после этого некоторая вершина зависимости стала готова к срабатыванию, то есть если осталось только одно неориентированное ребро, примыкающее к ней, то проводим срабатывание: ориентируем это ребро от этой вершины зависимости к вершине уравнения и делаем это уравнение внешним:

1) ставим «-» в его ячейке в столбце учёта уравнений, и

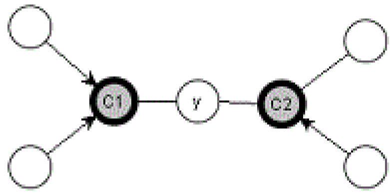
2) ориентируем и отмечаем его вершину на графе решения как внешнюю. Основание такой операции — утверждение (5а).

Если после какого-то шага планирования решения на графе зависимостей стало возможным выделить компоненту без незавершённых вершин зависимостей (определения и построения аналогичны приведённым для графа решения), то ориентируем эту компоненту (можно замкнуто, циклически) и проводим срабатывание всех уравнений, вершины которых входят в эту компоненту.

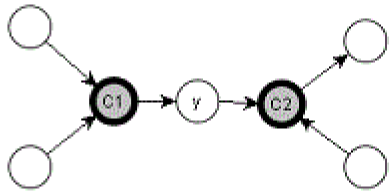
Основание: утверждение (5в).

Рассмотрим, как происходит ориениация в ситуации, соответствующей утверждению (5б), и в ситуации, сходной с ней.

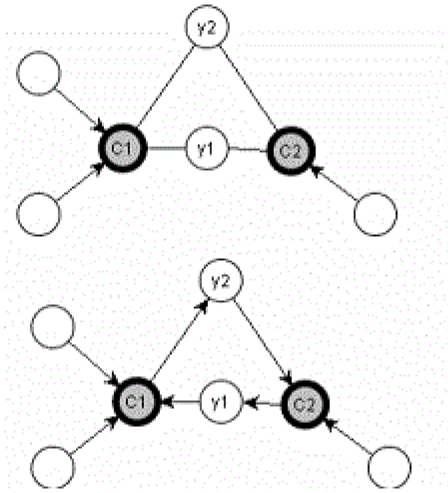
Исходная ситуация для срабатывания зависимой подсистемы, получаемой по (5б) из зависимых подсистем C1 и C2, включающих общее уравнение y.



Срабатывание по приведённому алгоритму: сначала срабатывает C1, затем C2.



Аналогичная ситуация, но C1 и C2 имеют два общих уравнения. По (5б) зависима система $C12=C1\cup C2\setminus\{y2\}$, но она сработать не может, так как ещё не сработало y2 на графе задачи. Но, оказываеися, оно и не должно срабатывать, это уравнение зависимо от уже сработавших, так как может сработать компонента. Соответствующая зависимая система $C3=C1\cup C2\setminus(\{y1,y2\})$ содержится в C12.



(13). Если граф зависимостей построен по системе зависимостей Ф=C1,...,Cк, представляющей базис зависимостей в системе C уравнений модели, то, применяя алгоритмы A1 и A3, получим граф правильного плана решения задачи.
Доказательство.
Правильность ориентации полученного графа очевидна по построению. Из утверждения (3) следует, что множеству его внутренних вершин соответствует максимальное подмножество независимых уравнений.

Составление задач.

Имеются: неориентированный граф одноразрешимой структурированной модели, в каждой подсистеме уравнений которой число уравнений меньше числа

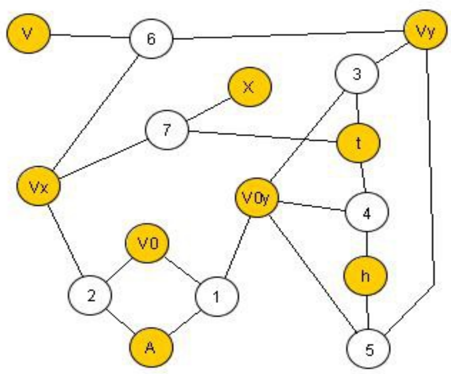
переменных; список списков уравнений её зависимых подсистем.
Требуется построить план решения, выбирая его, и, тем самым, задачу, к которой он относится, из возможных.

План решения удобно строить от конца к началу.
Пусть на каком-то шаге выбирается пара вида (уравнение — переменная), стоящая в плане непосредственно перед уже построенной его конечной частью. Удалим из рассмотрения уравнения построенной конечной части плана. Тогда переменная, вычисляемая в выбираемой паре, должна входить только в одно из оставшихся уравнений - в выбираемое. Ведь она не может быть ни исходным данным, ни переменной, вычисляемой по плану до неё. Таким образом, в качестве очередной пары можно выбрать любую переменную, входящую только в одно из оставшихся уравнений, и это уравнение, - но при условии, что после добавления этого уравнения к построенной части плана в неё не войдёт целиком какая-то зависимая подсистема (см. далее).
Пусть теперь на каком-то шаге выбирается компонента, состоящая из р уравнений У, из которых вычисляются р переменных Х. Опять будем считать, что построенная до этого шага часть плана удалена. Тогда (У,Х) должны удовлетворять следующим условиям:
(а) Переменные Х могут входить только в уравнения У, но не в другие из не удалённых уравнений.
Действительно, при решении задачи переменные Х не могут использоваться или вычисляться раньше срабатывания уравнений У.
(б) После добавления У к множеству уравнений построенной части плана в это множество не войдёт целиком никакая зависимая подсистема (см. далее).
В качестве очередной компоненты можно выбрать любые (У,Х), удовлетворяющие условиям (а) и (б). Это следует из структурированности системы и утверждения (8).

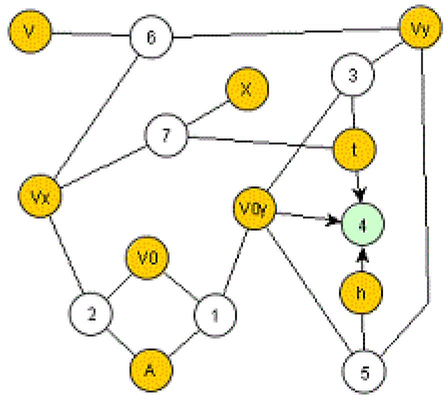
А4. Алгоритм составления полной задачи.

Для иллюстрации будем использовать ту же модель, пример 3.

Исходное состояние:
неориентированный граф модели.

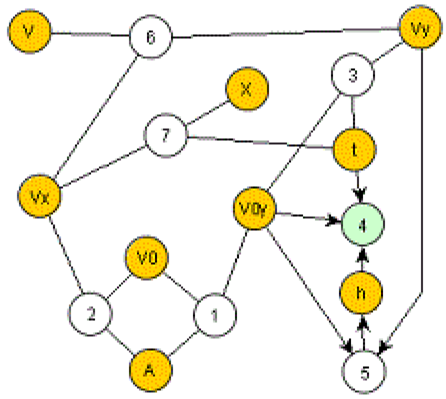


1. При помощи ориентирования графа зависимостей выбираем исключаемые (внешние) уравнения из зависимых подсистем так, чтобы осталась максимальная независимая подсистема (в рассматриваемом примере это тривиально, другой пример см. далее). Отмечаем выбранные внешние вершины уравнений на графе и ориентируем примыкающие к ним рёбра от вершин переменных к этим вершинам уравнений.

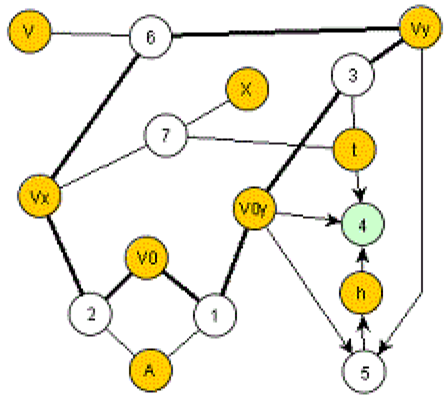


Затем производим в произвольном порядке описанные ниже шаги 2. и 3., выбирая компоненты, для которых выполнены требуемые условия, указанные при описании этих шагов, пока не будут сориентированы все дуги графа. После этого переходим к завершающему п. 4.

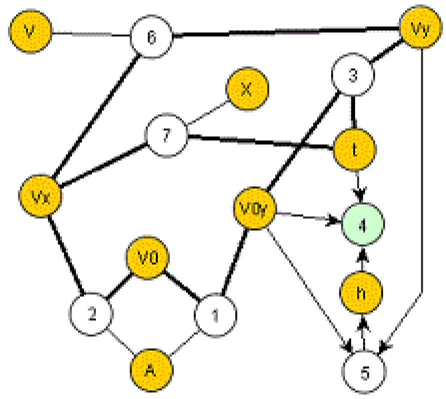
2. Выбираем вершину переменной, к которой примыкает только одно неориентированное ребро. Ориентируем его от вершины уравнения к вершине переменной, а остальные рёбра, примыкающие к этой вершине уравнения, ориентируем от вершин переменных к этой вершине уравнения. Такую операцию будем называть *срабатыванием вершины переменной*. На рисунке справа срабатывает вершина h.



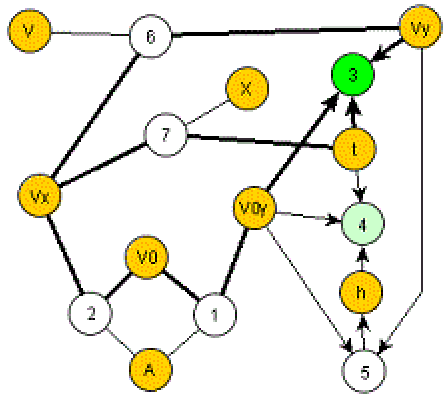
3. а). Отыскиваем контур, такой, чтобы в него не входило или входило как можно меньше вершин переменных, к которым примыкают неориентированные рёбра, лежащие вне этого контура. Назовём такие вершины переменных *незавершёнными*.
На рисунке справа выбранный контур выделен жирными рёбрами. В нём есть незавершённая вершина v_x.



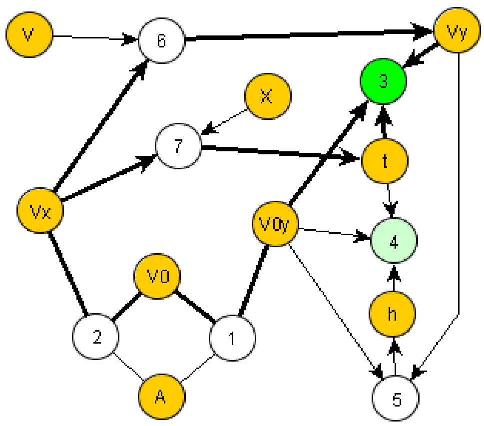
б). Если в построенной части компоненты есть незавершённые вершины, пытаемся добавить к ней маршруты, связывающие незавершённые вершины с вершинами уравнений этой построенной части компоненты так, чтобы компонента, расширенная этими маршрутами, не имела незавершённых вершин переменных.
На рисунке справа добавлен маршрут vx–(7)–t–(3). В полученной компоненте (она выделена жирными дугами) нет незавершённых вершин.



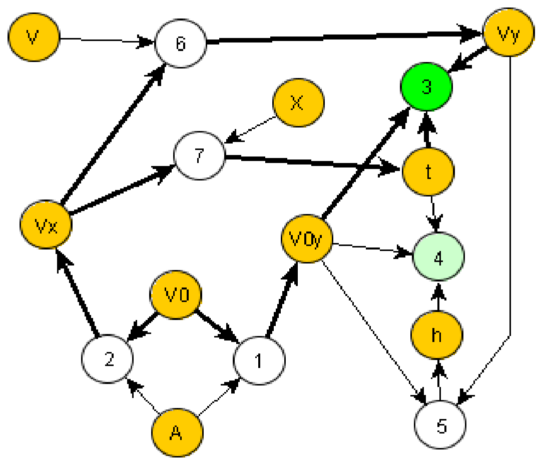
в). Планируем решение выбранной компоненты. Строим разомкнутый граф решения. Начинаем с выбора вершины базового уравнения. Выбираем её так, чтобы в компоненте появились готовые к срабатыванию вершины переменных (см. п.2). Если это не удаётся, добавляем ещё одну вершину базового уравнения, и т. д.
Ориентируем рёбра, примыкающие к выбранным вершинам, от вершин переменных к этим вершинам уравнений.
На рисунке справа выбрана вершина базового уравнения (3) (отмечена зелёным).



Производим срабатывания вершин переменных в выбранной компоненте.
В нашем примере сразу готовы к срабатыванию вершины vy и t.

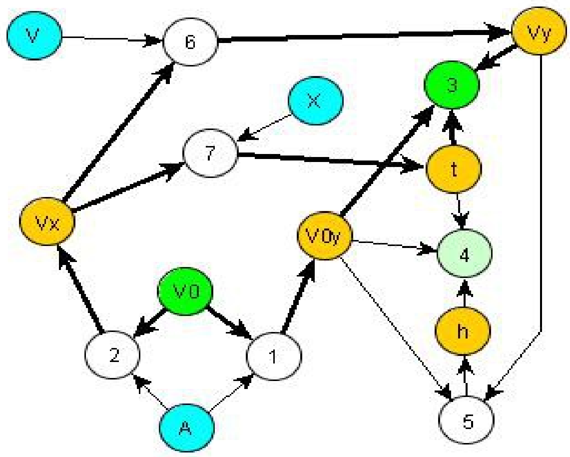


После этого срабатывают v_x и v_{0y}. Ориентация компоненты завершена, а с ней завершена и ориентация графа.



4. Отмечаем образовавшиеся вершины базовых переменных (вершины переменных в многоэлементных компонентах, к которым примыкают только исходящие дуги). В нашем примере это вершина v_0.

Отмечаем образовавшиеся вершины исходных переменных (вершины переменных вне многоэлементных компонент, к которым примыкают только исходящие дуги). В нашем примере это вершины v, x, A.



Записываем полученную задачу и её план решения.
В нашем примере задача: дано: v, x, A. Найти остальные переменные (эта задача уже рассматривалась). План решения можно взять из процесса составления задачи: (1),(2),(3),(6),(7) — v_o,v_x,v_{0y},v_y,t; (5) – h.

План решения первой компоненты:
– v_o, (1)–v_x, (2)–v_{0y}, (6)–v_y, (7)–t, (3)–v_o.

Затем можно построить другие планы решения этой задачи и выбрать наиболее рациональный план.
Затем нужно провести решение задачи по полученному плану, проверив её решаемость доступными средствами и оценив сложность решения, получить ответ и промежуточные результаты, сделать проверку по внешним уравнениям, отбросить лишние корни, если такие есть.

Составление неполной задачи.

Можно сначала составить полную, а затем выбрать искомые переменные и уточнить план решения, отбросив лишние компоненты, как это было описано ранее.

Ветвящаяся модель.

Конструкция ветвящейся модели была описана ранее в подразделе «альтернативы».

При решении ветвящейся задачи с заданными числовыми значениями исходных переменных желательно проверять, удовлетворяют ли значения вычисленных переменных условиям выбора альтернатив сразу же после вычисления этих значений, и, если не удовлетворяют, прервать дальнейшие вычисления, использующие эти значения в этой ветви. Но при планировании мы этого делать не можем, строим общий план, а не план для вычислений с конкретными значениями данных, и доводим каждую ветвь до конца, ведь каждая ветвь, входящая в модель, разрешима. Рассмотрим два метода, общий и удобный в некоторых случаях. Начнём с общего метода.

Граф ветвящейся модели.

Будут использоваться следующие соглашения.

Для каждой ветви, начиная с главной, изображаем отдельные графы моделей общей части и каждой подветви (можно было бы ввести специальные вершины ветвления и изображать всё на едином графе, но, на мой взгляд, большие графы плохо воспринимаются и их труднее рисовать).

Одна переменная, входящая в уравнения общей части и в уравнения ветвей, изображается на их графах одноимёнными вершинами, к каждой из которых примыкает отдельная безымянная (*немая*) вершина. Ребро между вершиной переменной и её немой вершиной ориентируется в направлении передачи информации (см. пример далее): если на графе общей части или ветви к вершине переменной примыкает какая-либо заходящая дуга или если переменная является исходной, то ребро на этом графе ориентируется из вершины переменной к немой вершине; если все дуги, примыкающие к вершине не исходной переменной на этом графе исходящие, то ребро на этом графе ориентируется из немой вершины к вершине переменной. Это направление должно быть одинаково на графах всех ветвей, куда входит эта переменная, и противоположно такому направлению на графе общей части.

Если часть уравненний зависимой системы входит в одну ветвь, а часть — общая для всех ветвей, то общие уравнения можно (но не обязательно) повторить в каждой ветви, не вынося их в общую часть, для того, чтобы в каждой ветви можно было выбирать исключаемые (внешние) уравнения не зависимо от их выбора в других ветвях. Но нужно учитывать все зависимости, в которые входят уравнения, дублируемые в ветвях. Если уравнение, перенесённое в ветви, входит в зависимости общей части и в зависимости ветвей, то оно должно быть либо всюду внешним, либо всюду внутренним. На графе зависимостей вершина такого уравненя аналогична вершине общей переменной на графе решения. Она должна быть либо исходной, либо в неё заходит дуга в общей части и из него исходят дуги в каждой ветви, либо в неё заходят дуги в каждой ветви и исходят дуги в общей части. В последнем случае, если рисуется единый граф зависимости, в вершину такого уравнения на нём заходят дуги из каждой ветви, то есть ориентация примыкающих к такой вершине дуг отличается от обычной (правильной).

Перенос части уравнений из общей части в ветви позволяет также включать эти уравнения в компоненты, часть уравнений которых расположена в ветвях. Перенос уравнений (а он изменяет исходное изображение модели) может производиться при построении модели или/и во время её ориентации.

Условие выбора ветви изображаем на графе ветви так же, как уравнение (вершиной условия, связанной рёбрами с вершинами входящих в него переменных), но с выделением (на рисунках примера далее — красный ромб). Примыкающие к вершине условия рёбра для условия, выраженного неравенством, при планировании не ориентируем. Выполнение условия ветви должно проверяться для каждого корня этой ветви. Если условие выбора ветви выражено равенством, то это равенство добавляется к системе уравнений ветви и ориентируется по общим правилам.

Планирование решения задачи.

Исходная информация:

1. Граф ветвящейся модели.
2. Списки состава зависимых подсистем для каждой ветви, сопровождающие графы не ветвящихся частей ветвей.
3. Список исходных переменных. Для неполной задачи ещё список искомых переменных.

Планирование решения производим по алгоритму, аналогичному описанному ранее алгоритму планирования решения задач на не ветвящейся модели, но с переходами с одного графа на другой. Сначала ориентируем, пока это возможно, общую часть и каждую ветвь по-отдельности, с переносом выходных переменных общей части на вход всех ветвей, а переменных, выходных для всех ветвей — на вход общей части. Если этот процесс останавливается до окончания ориентации всего графа, то не сработавшие уравнения общей части, для вершин которых остались примыкающие к ним не ориентированные рёбра, добавляем к каждой ветви (к графам ветвей добавляются вершины этих уравнений, вершины всех входящих в них переменных, и рёбра или дуги между ними). Затем продолжаем ориентирорать каждую ветвь.

Составление задачи на ветвящейся модели.

Составляемый план строим в такой же последовательности. Переход от ориентации общей части к ориентации ветвей с переносом оставшейся общей части в каждую ветвь можем проводить на любом шаге, если хотим продолжить ориентирование иначе, чем это можно сделать на ощей части.

Пример 4.

Как и в предыдущем примере 3 (см. в первой части статьи), шарик брошен под углом к горизонту с некоторой высоты и летит до земли без сопротивления воздуха, но при этом есть потолок, о который он абсолютно упруго ударяется, если долетает до него.

Сохраним все переменные модели примера 3 (см. их описание) и добавим следующие:

$h_п$ – высота потолка;

t_1 – время полёта до момента удара о потолок;

t_2 – время полёта от потолка до пола;

v_{1y} – вертикальная составляющая скорости непосредственно перед ударом о потолок.

Не ветвящаяся часть.

Уравнения:

(1) $v_x=v_0\cos A$;

(2) $v_{0y}=v_0\sin A$;

(3) $v_0^2=v_x^2+v_{0y}^2$;

(4) $v^2=v_x^2+v_y^2$;

(5) $x=v_xt$;

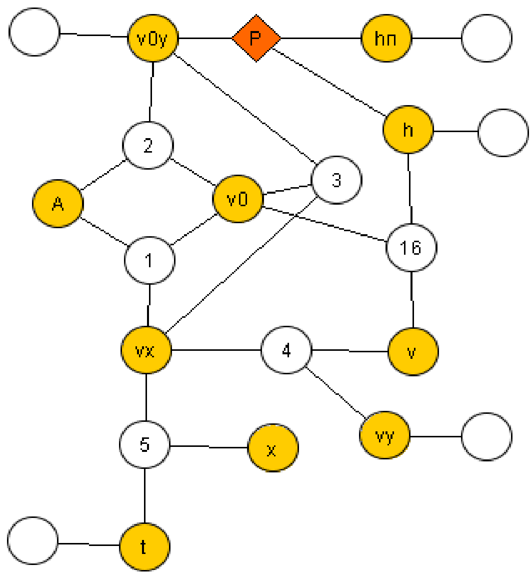
(16) $v^2-v_0^2=2gh$

(P) $v_{0y}^2\le 2g(h_п-h)$

Зависимости:

1,2,3;

3,4,8,16



Граф модели общей части.

Ветвь «без удара о потолок».

$v0y^2 \le 2g\ (hп-h)$

Уравнения:

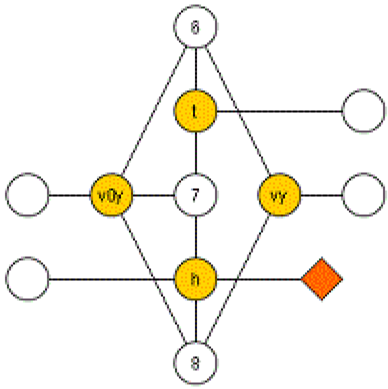
(6) $v_y = v_{0y} - gt$;

(7) $h = v_{0y}t - gt^2/2$;

(8) $v_y^2 - v_{0y}^2 = 2gh$

Зависимости:

6,7,8



Граф модели ветви.

Ветвь «с ударом о потолок»

$v0y^2 > 2g(hп-h)$

Уравнения:

(9) $v_{1y} = v_{0y} - gt_1$;

(10) $v_y = -v_{1y} - gt_2$;

(11) $h_п - h = v_{0y}t_1 - gt_1^2/2$;

(12) $h_п = v_{1y}t_2 + gt_2^2/2$;

(13) $v_{0y}^2 - v_{1y}^2 = 2g(h_п - h)$;

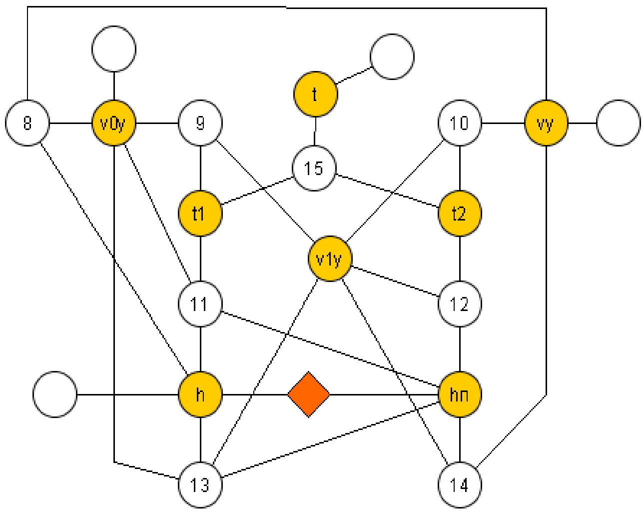
(14) $v_y^2 - v_{1y}^2 = 2gh_п$;

(15) $t = t_1 + t_2$;

(8) $v_y^2 - v_{0y}^2 = 2gh$.

Зависимости:

(9,11,13);(10,12,14);(13,14,8).

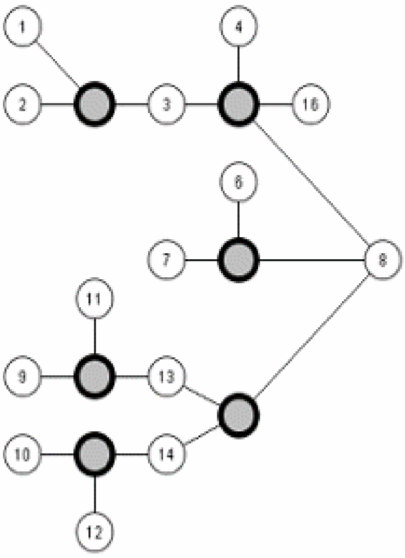


Граф модели ветви

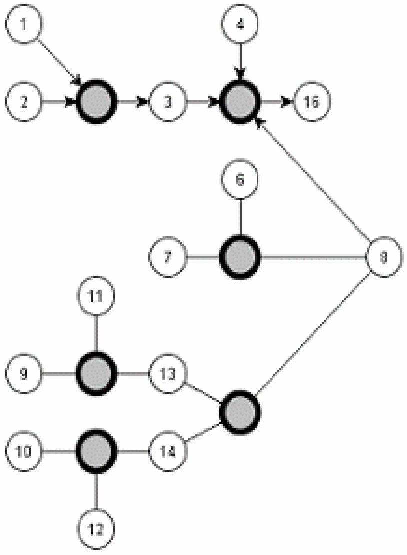
Если выбрать внешними уравнения 3,8,16,13,14 и исходные переменные A, v0, h, hп, то граф модели и граф зависимостей ориентируются одинарно. Следовательно, перечисленные выше минимальные зависимости образуют базис зависимостей для данной системы уравнений.

Граф зависимостей и его ориентирование.

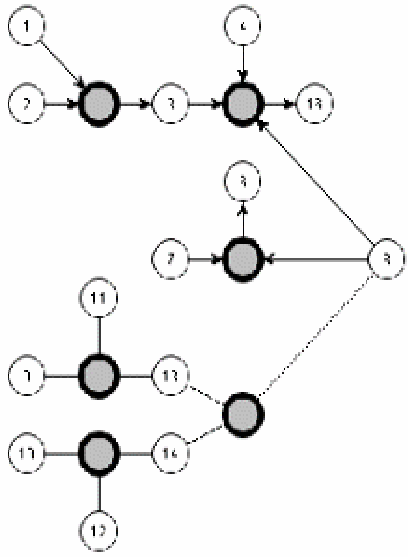
Уравнение (8) дублировано в ветвях, но оно также входит в зависимую подсистему уравнений общей части вместе с (3),(4),(16). Так как граф зависимостей небольшой, удобно изобразить его единым графом. Ориентируем его по тем же правилам, что были описаны для ориентации графа не ветвящейся модели при составлении задач, но общая вершина (8) ориентируется по особому правилу: она либо исходная, либо в неё заходят дуги из обеих ветвей и из неё исходит дуга в общей части (откуда уравнение (8) исключено).



Граф модели.

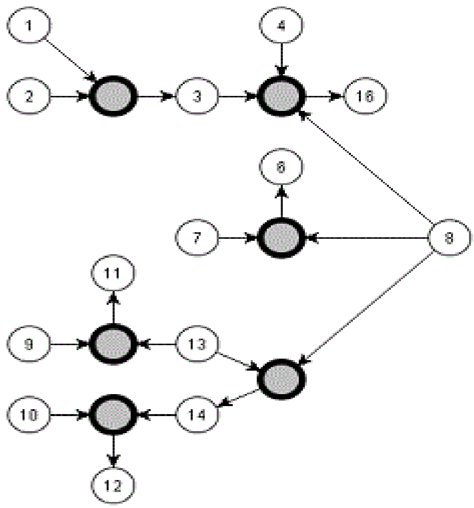


Ориентирование общей части.



Ориентирование ветви "без удара о потолок".

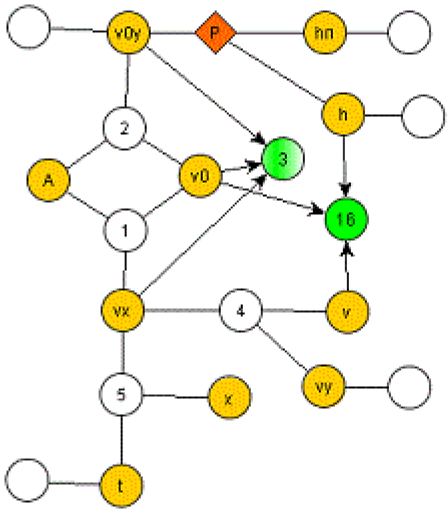
Получили множество внешних вершин уравнений, – тех, в которые заходят дуги на ориентированном графе зависимостей:
3, 16, 6, 11, 12, 14.



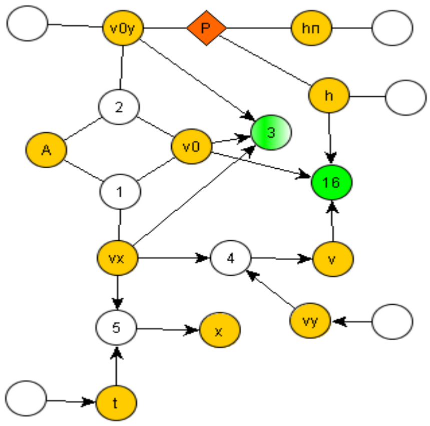
Ориентирование ветви с ударом о потолок.

Ориентирование не ветвящейся части графа модели

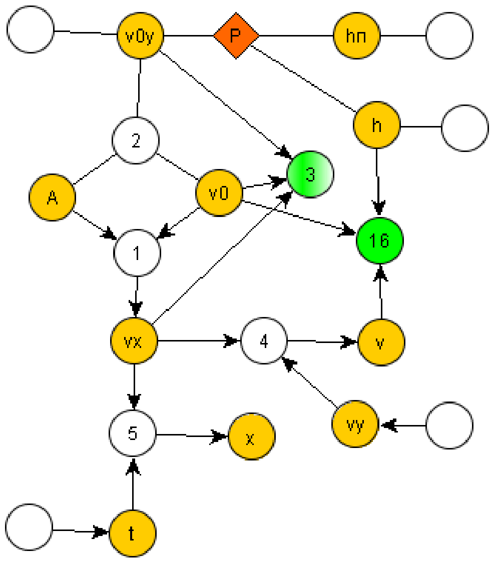
1. Выделяем внешние уравнения и ориентируем примыкающие к ним рёбра.



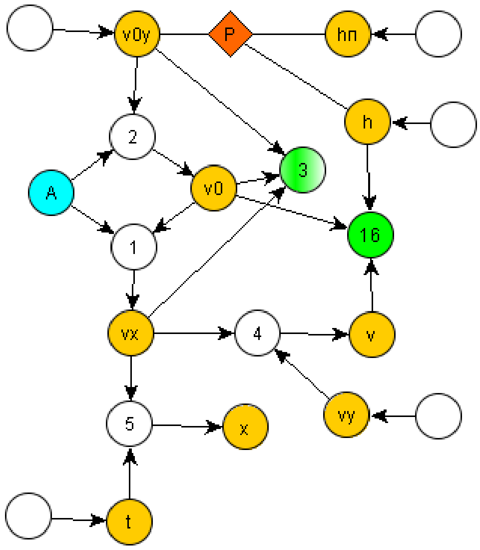
2. Готовы к срабатыванию вершины x и v. Выбираем обе и проводим срабатывание. Ориентируем рёбра, связывающие ориентированные вершины переменных с немыми вершинами.



3. Проводим срабатывание вершины v_x.

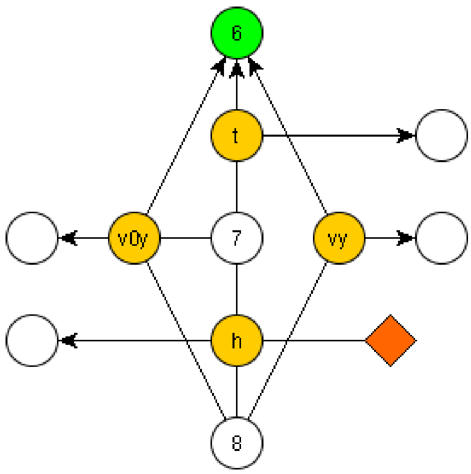


Готовы к срабатыванию вершины A и v_0. Выбираем v_0. Ориентируем немые вершины. Отмечаем образовавшуюся вершину исходной переменной A. Ориентация не ветвящейся части закончена.

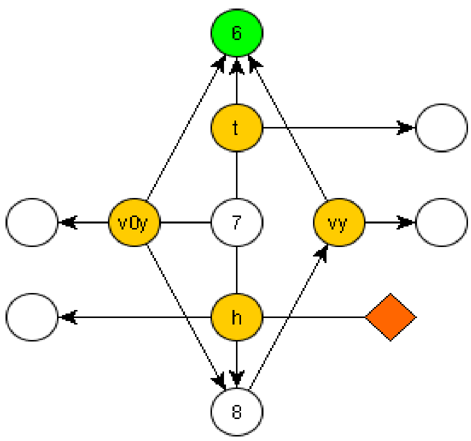


Ветвь без удара о потолок

Выделяем и ориентируем вершину исключаемого уравнения (6). Ориентируем также немые вершины в соответствии с их ориентацией на графе не ветвящейся части.

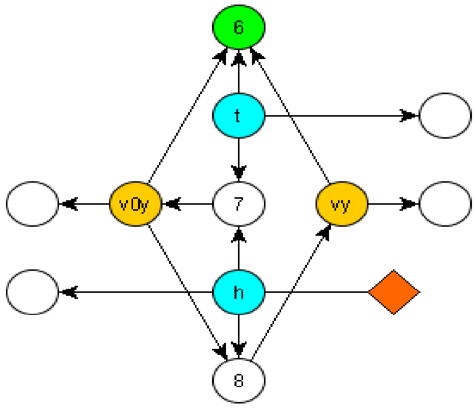


Проводим срабатываеие v_y



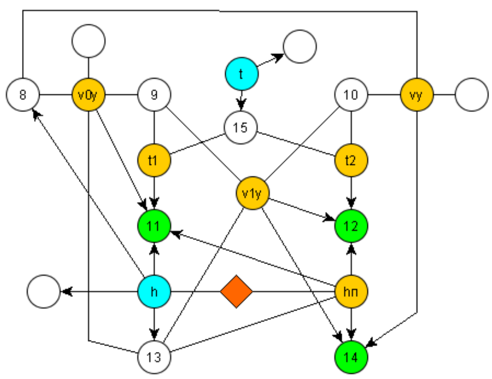
Проводим срабатывание V_{0y}.

Отмечаем образовавшиеся вершины исходных переменных t и h. Ориентация ветви завершена.

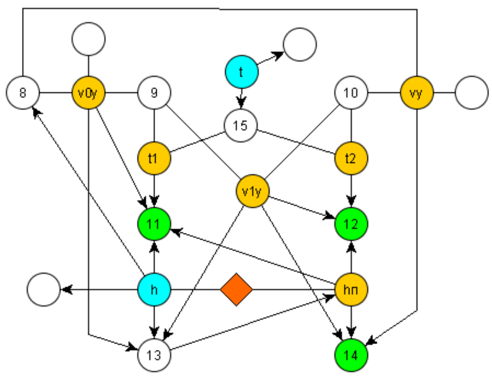


Ориентация ветви с ударом о потолок.

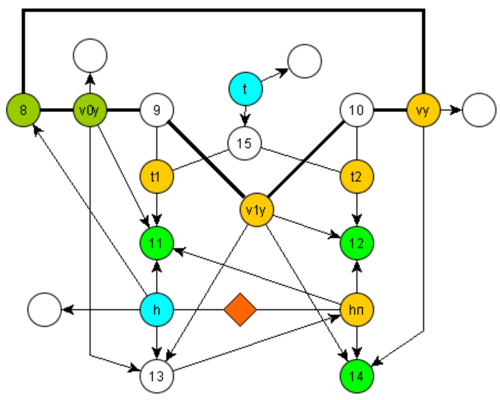
Выделяем и ориеннтируем вершины исключаемых уравнений 11,12,14.
Сохраняем исходные переменные, образовавшиеся при ориентации ветви без удара.



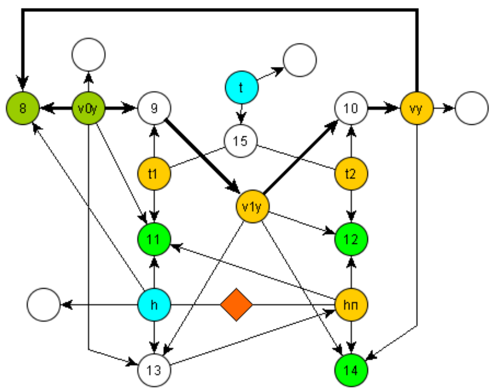
Проводим срабатывание $h_п$.



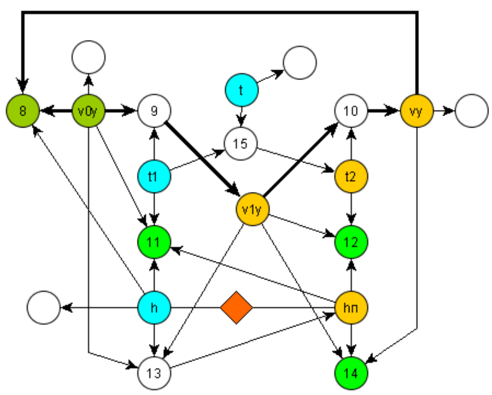
Выделенный на рисунке контур не содержит незавершённых вершин переменных, это компонента, готовая к срабатыванию. Выбираем базовую переменную v_{oy} и базовое уравнение (8).



Проводим ориентацию и срабатывание компоненты.



Из готовых к срабатыванию вершин t_1
и t_2 выбираем t_2. Ориентация ветви завершена.



Сформулируем полученную задачу:

Шарик брошен с высоты h над полом под углом A к нему и упал на пол через t секунд. Определить начальную и конечную скорость шарика и расстояние, которое он пролетел (в проекции на пол) в случае, если шарик ударился о потолок через t_1 секунд после броска и в случае, если он не долетел до потолка. Определить высоту потолка, если шарик до него долетел.
Сопротивлением воздуха пренебречь. Удар о потолок считать абсолютно упругим.

Задачи с безальтернативной формулировкой.

В приведённой формулировке условия задачи различны для случаев, соответствующих разным ветвям модели. Подобные задачи близки к набору отдельных задач для каждой ветви. Их будем называть *задачами с альтернативной формулировкой,* а задачи, в которых множества исходных и искомых переменных не зависят от ветви — *задачами с безальтернативной формулировкой*. Рассмотрим составление таких задач.

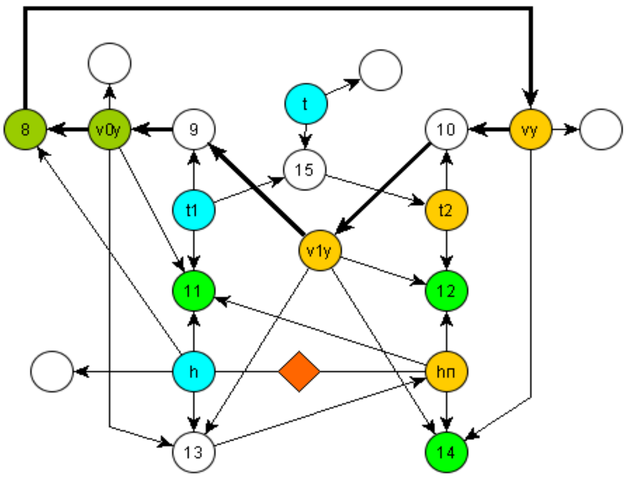
Множество переменных разделим на множество *общих переменных*, входящих в общую часть или/и в каждую ветвь модели, и множества *индивидуальных переменных ветвей*. Индивилуальные переменные ветвей разделим на *переменные конструкции*, описывающие единую физическую систему, в которой могут быть альтернативные ситуации или происходить альтернативные процессы, соответствующие ветвям модели, и *ситуативные переменные*, описывающие эти ситуации или процессы.

В рассмотренном примере индивидуальнные переменные есть в модели ветви с ударом о потолок: переменная конструкции – $h_п$, ситуативные — t_1, t_2, v_{y1}. В задаче с безальтернативной формулировкой ситуативные переменные не должны быть исходными. Конструктивные переменные могут быть исходными, так как их упоминание не требует выделения ветви. Искомые переменные неполной задачи нужно выбирать из числа общих переменных.

Для составления задачи с безальтернативной формулировкой (если это возможно для данной ветвящейся модели) можно составить задачу с альтернативной формулировкой, выбрать графы ветвей с ситуативными переменными, изобразить на них ориентацию компонент как замкнутую и проинвертировать пути от вершин ситуативных исходных переменных к вершинам общих или конструктивных переменных.

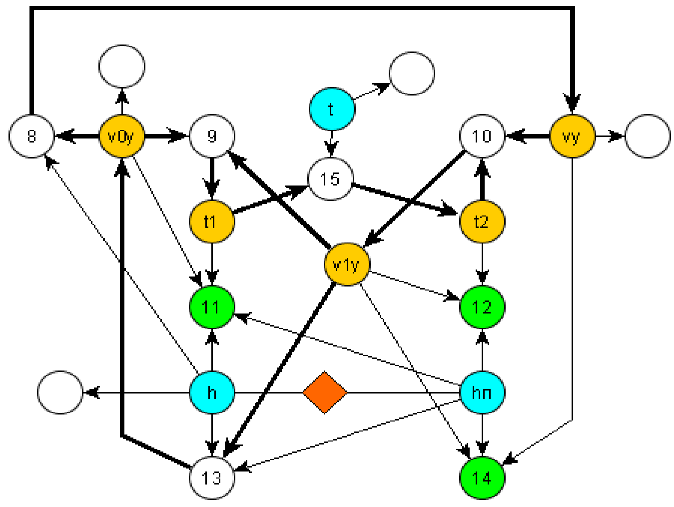
В рассмотренном примере:

Ветвь с ударом о потолок, замкнутая ориентация.



Инвертируем путь t_1–9–v_{0y}–13–$h_п$

Результат. Все уравнения ветви составляют одну компоненту.



Формулировка полученной задачи:

Шарик брошен с высоты h над полом под углом A к нему и упал на пол через t секунд. Высота потолка $h_п$. Определить начальную и конечную скорость шарика и расстояние, которое он пролетел (в проекции на пол).

Сопротивлением воздуха пренебречь. Удар о потолок считать абсолютно упругим.

Другой метод.

Пример 5. То же, что в предыдущем примере, но шарик может удариться не о потолок, а о вертикальную стенку, перпендикулярную плоскости его полёта.

Переменные v_0, v_{0x}, v_{0y}, A, v, v_x, v_y, t, h те же;

x — горизонтальная координа точки падения шарика на пол (ось x направлена к стенке; начало координат на полу под точкой броска);

L – координата стенки по оси x.

Условие P, при выполнении которого шарик ударится о стенку: $v_{ox}t > L$.

Уравнения при недолёте такие же как в примере 3; уравнение ветви: $v_x=v_{0x}$.

Уравнения ветви при ударе: $v_x=-v_{ox}$; $x=2L-v_{0x}t$.

Приём этого метода состоит в том, что ветвления переносятся внутрь уравнений, а сама модель становится не ветвящейся. Правда, ценой потери точечности таких уравнений.

В нашем примере можно записать:

(a1) $v_x=\{v_{0x}$ при $v_{0x}t<L$, $-v_{0x}$ при $v_{ox}t>L\}$;

(a2) $x=\{v_{0x}t$ при $v_{0x}t\le L$, $2L-v_{0x}t$ при $v_{0x}t>L\}$.

Обратные выражения:

(a3) $v_{0x}=\{v_x$ при $v_x>0$, $-v_x$ при $v_x<0\}$;

(a4) $v_{0x}t=\{x$ или $x+L$ при $0<x\le L$, $2L-x$ при $x\le 0$, не определено при $x>L\}$;

(a5) $L=\{x+v_{0x}t$ при $x<v_{0x}t$, любое$>v_{0x}t$ при $x=v_{0x}t$, не определено при $x>v_{0x}t\}$.

Вместо переменных x, L введём переменную $x^*=v_{0x}t$ и сориентируем граф

соответствующей модели Г. Затем, чтобы заменить x^* на x, L, воспользуемся (a2) или (a5), если вершина x^* ориентирована как вычисляемая, и (a4), если x^* ориентирована как исходная. В последнем случае, если выражение для искомой переменной по Г получилось $w=f(x^*,y)$, то в ответе оно будет: $w=f(x^*,y)$,
где $x^*=\{x$ или $x+L$ при $0<x\leq L$, $2L-x$ при $x\leq 0$, не определено при $x>L\}$.

Инверсия ветвящихся путей.

Для получения другого плана решения той же или другой ветвящейся задачи можно использовать преобразования ориентаций аналогично тому, как это было описано для не ветвящихся моделей. При этом можно инвертировать как пути, лежащие в пределах графа одной ветви или не ветвящейся части, так и ветвящиеся пути.

Ветвящимся путём будем называть связную часть ориентированного графа решения задачи на ветвящейся модели, составленную из путей, лежащих в графах различных ветвей, включая не ветвящуюся часть, не более чем по одному пути из каждой ветви, связанных между собой общими переменными. То есть это путь, который может разветвляться в вершинах общих переменных не ветвящейся части и ветвей.

Конкретнее: 1. Для перестройки в план другого решения той же задачи:

а). Переходим к замкнутой ориентации компонент. Далее можем менять направление ориентации циклов, как удобнее. Для примера 4, ветви с ударом о потолок, замкнутая ориентация была показана на рисунке при построении задачи с безальтернативной формулировкой.

б).Инвертируем один ветвящийся путь от вершины внутреннего уравнения не ветвящейся части к внешним вершинам ветвей.

Или б). Инвертируем сходящийся путь из внутренних уравнений ветвей во внешнее уравнение общей части.

Начинаем с инверсии пути в графе зависимостей, затем инвертируем соответствующий путь в графе решения.

В графе, полученном в примере 4, можно проинвертировать сходящийся путь от (8) в обеих ветвях через v_y к (4),v,(16) в общей части.

2. Для перестройки в план решения другой задачи с тем же выбором внешних вершин уравнений:

а). При инвертировании ветвящегося пути от вершины исходной переменной в общей части к вершинам искомых переменных в ветвях желательно сохранить единство условий задачи. Для этого инвертируемый путь должен заканчиваться в одноимённых переменных разных ветвей (в общей переменной ветвей).

б).Аналогично, желательно, чтобы инвертирцемый путь от вершин исходных переменных в ветвях к вершине искомой переменной в общей (не ветвящейся) части начинался с одноимённых исходных вершин.

Если выполнить условие одноимённости концов не удаётся, инверсия даёт задачу, при формулировке которой нужно оговаривать, при каких условиях что дано или/и

что найти.
В графе примера 4 можно проинвертировать путь от h,(8) в ветви без удара и от t_1, (9), v_{0y},(8) в ветви с ударом о потолок, через v_y к (4),v в общей части.

Табличное представление.

В школе не проходят теорию графов. Рисовать графы вручную не всегда удобно, приходится перерисовывать, чтобы было меньше пересечений дуг. Большие графы выглядят запутанно. Вместо графов можно использовать таблицы. Предлагаем следующие обозначения.
Таблица модели объекта содержит строки, соответствующие уравнениям, и столбцы, соответствующие переменным. В ячейке ставим метку «.», если переменная столбца входит в уравнение строки. Отмеченные ячейки соответствуют рёбрам в графовом представлении, а таблица такого типа называется *матрицей смежности* графа модели (вершины, соединённые ребром или дугой, называются *смежными*).
Усовершенствованная таблица модели объекта:
– в ячейке вместо «.» ставим метку «..», если решение соответствующего уравнения относительно соответствующей переменной не единственное (это в примере, приводимом далее, не используется);
– слева от столбцов переменных размещаем столбец уравнений, где записываются формулы уравнений и условий выбора ветвей или/и их номера;
– *единая таблица* может отражать все ветви. Тогда строки, соответствующие уравнениям одной ветви, группируются (располагаются подряд). Перед ними идёт строка условия выбора ветви, в ячейках которой отмечаются знаком «.» входящие в него переменные. Столбцы таблицы, таким образом, соответствуют всем переменным модели, входящим хотя бы в одну ветвь. Но для удобства редактирования таблицу можно разбить на отдельно располагаемые *таблицы ветвей* (в примере так и будет сделано).
– для каждой зависимой подсистемы, состав которой входит в описание модели (эти подсистемы должны составлять базис зависимостей), вводим свой столбец с левой стороны таблицы, в котором проставляем «.» в строках уравнений этой зависимой подсистемы. Это подтаблица зависимостей.
Таблицу плана решения задачи получаем из таблицы модели объекта добавлением снизу строки исходных и искомых переменных, добавлением слева столбца учёта сработавших и внешних уравнений, и расстановкой дополнительных знаков в уже отмеченные клетки.
Последовательность решения отображаем номерами вычислений.

А5. Табличный алгоритм планирования решения задач на не ветвящейся модели.

Пример см. далее.
Предполагаем, что задача разрешима.

0. В строке исходных данных ставим «0» в клетках столбцов, соответствующих исходным данным; во всех остальных строках в ячейки этих столбцов с меткой «.» вместо неё ставим метку «+» как знак того, что переменная вычислена (пустые ячейки остаются пустыми).
1. Поиск и срабатываеие одинарных компонент.
– Если не вычисленных переменных нет, конец процедуры; если есть, отыскиваем строки, в которых отмечены «+» все отмеченные ячейки кроме одной; если таких нет, переходим к п.3, полагая там k=2 (число уравнений в компоненте).
– Среди найденных строк выбираем строку уравнения, из которого не отмеченную переменную вычислить проще, и проводим её срабатывание, п.2. Возвращаемся к началу п.1.
2. Срабатывание строки уравнения.
– Ставим в клетке ещё не вычисленной переменной очередной номер срабатывания «h». Ставим «+» в ячейку столбца учёта уравнений и заменяем «.» на «+» в отмеченных ячейках столбцов зависимостей на этой строке (если такие есть).
– Проводим срабатывание зависимостей по уравнению, п.9, если в столбцы зависимостей внесены изменения.
– В столбце переменной данного срабатывания, в ячейке которого проставили его номер, заменяем все «.» на «+».
– Если в какой-то строке после этого не осталось ячеек переменных с меткой «.», а в ячейке учёта не стоит «-», то соответствующее уравнение и задача являются формально переопределёнными. Тогда ставимим «--» в ячейку этой строки в столбце учёта.
– Возврат.
3. Поиск и срабатывание множественной компоненты.
– Ищем k строк ровно с k ещё не вычисленными переменными («.», но не «+» в ячейках этих переменных). Если таких нет, увеличиваем k на 1 и повторяем поиск.
– Найденную таким образом компоненту обозначим Т. Проводим срабатывания зависимостей по компоненте Т, п.6. При «неудаче» повторяем поиск. При «удаче» присваиваем m=1 (число базовых переменных) и планируем вычисление компонненты, п.4.
– Отмечаем сработавшую компоненту: проставляем метки «+» вместо меток «.» во всех столбцах переменных, которые сработали при планировании компоненты (в которых появились ячейки с номерами срабатываний).
– Возвращаемся к п. 1.
4. Планирование вычисления компоненты Т.
Все метки в этом процессе (п.4, п.5) расставляются только в строках Т.
Выбираем и нумеруем очередным номером одну из не вычисленных переменных в Т – базовую, относительно которой намереваемся составить уравнение, и ставим этот номер в ячейку этой переменной в строке уравнения, выбираемого базовым. Ставим номер как пробный, после «.», а не вместо неё.

5. Проводим пробные срабатывания по пп. 1,2 в подсистеме T, пока это возможно. Если все переменные компоненты отмечены «+», компонента спланирована, снимаем статус пробности (стираем «.» в метках «.+»), возврат. При неудаче стираем срабатывания («+» из «.+») и возвращаемся к п.4 для другого выбора. Если варианты выбора исчерпаны, добавляем ещё одну пару «базовая переменная — базовое уравнение». Для этого возвращаемся к п.4, не стирая срабатывания, сделанные по одному из вариантов.
По ходу описанного процесса базовое уравнение компоненты будет составлено, когда все ячейки его переменных в его строке окажутся отмеченными «+» (без очередной нумерации).
6. Срабатывание зависимостей по компоненте T.
- Проводим пробное срабатывание компоненты T в столбцах зависимостей (п.7; для указания статуса «пробное» метки «.» не удаляем, а метки «+» или «-» проставляем после них).
- При удачном завершении снимаем статус пробного: удаляем «.» изо всех меток «.+» и «.-». Возврат «удача».
7. Пробное срабатывание компоненты T.
- В строках T в столбцах зависимостей и столбце учёта проставляем метки «+» после меток «.».
- Если в каком-то столбце зависимости все метки стали с «+», то компонента T выбрана неудачно. Тогда стираем все пробно проставленные метки и возвращаемся к п.3, «неудача».
- Если в каком-то столбце зависимости осталась только одна ячейка с меткой «.» (пусть её строка y), проводим срабатыванние y как строки зависимого уравнения, п. 8, и продолжаем поиск по данному п.
- Если в k столбцах зависимостей отмечены «.» только ячейки одних и те же k строк, то проводим срабатывание каждой из этих строк как строки зависимого уравнения, п. 8, и продолжаем поиск с начала данного п;
- если нет готовых к срабатыванию столбцов или групп столбцов, заканчиваем п., возврат «удача».
8. Пробное срабатывание строки y как строки зависимого уравнения:
-В ячейки строки y, отмеченные «.», добавляем (не удаляя «.») «+» в столбцах зависимостей и «-» в столбце учёта уравнений.
- Если добавленный «+» завершает зависимость (в столбце зависимости, куда он ставится, это последняя метка «.» без «+»), то компонента T выбрана неудачно. Тогда стираем все пробно проставленные метки и возвращаемся к п.3, «неудача».
- Возврат.
9. Срабатывание зависимостей по уравнению.
- Если в каком-либо столбце зависимостей осталась только одна метка «.», заменяем её на «+», проставляем «-» в ячейке этой строки столбца учёта и продолжаем поиск по данному п.

– Если в каких–либо k столбцах зависимостей метки «.» стоят только в одних и тех же k строках, то во всех этих строках заменяем их на «+», проставляем «–» в ячейки этих строк столбца учёта и продолжаем поиск по данному п.

– Если нет больше готовых к срабатывнию столбцов и групп столбцов, то возврат. Конец описания алгоритма.

Другие табличные алгоритмы здесь выписывать не будем. Табличные алгоритмы аналогичны графовым. Например, при составлении задачи срабатыванию вершины переменной соответствует следующая операция с таблицей: ищем столбец переменной, в котором все отмеченные ячейки строк, кроме одной в строке у, отмечены «+», и проставляем «+» во все ячейки с «.» в этой строке у.

Пример.

Проиллюстрируем табличное предатавление на примере планирования решения задачи на ветвящейся модели, составленной ренее (**пример 4**; систему уравнений, систему зависимосией и формулировку см. там).

Общая часть.

Таблица модели

«.» – переменные, входящие в уравнения,

«+» –исходные переменные среди них.

Ни одно уравнение не может сработать: в каждой строке не меньше двух «.».

Отложим ориентацию, сначала попробуем ориентировать ветви.

ур.	уч.	t	A	v0	vx	v0y	vy	v	x	h
1			+	.	.					
2			+	.		.				
3				.	.	.				
4					.		.	.		
5		+			.				.	
16				.				.		+
Вх/вых		0	0							0

Ветвь «без удара о потолок».

Таблица модели.

=	ур.	уч.	v_{0y}	v_y	h	$h_п$	t
	P		.		+	.	
.	6		.	.			+
.	7		.		+		+
.	8		.	.	+		
Вх/вых					0		0

В строке уравнения 7 одна «.», проведём срабатывание.
«1» – номер срабатывания.
Отмечены «+» сработавшее уравнение 7 и вхождения вычисленной переменной v_{oy}
Стали готовыми к срабатыванию строки P, 6 и 8.

=	ур.	уч.	v_{0y}	v_y	h	$h_п$	t
	P		+		+	.	
.	6		+	.			+
+	7	+	1		+		+
.	8		+	.	+		
	вх/ вых				0		0

Выбрали вторым срабатывание строки условия P. Это срабатывание обозначает решение неравенства
P относительно $h_п$, то есть ответ на вопрос: при каких $h_п$ шарик не ударится о потолок.
Третьим выбрано срабатывание строки уравнения 8. После него срабатывает столбец зависимости в строку уравнения 6, в столбец учёта там ставим «–».

=	ур.	уч.	v_{0y}	v_y	h	$h_п$	t
	P		+		+	2	
+	6	-	+	.			+
+	7	+	1		+		+
+	8	+	+	3	+		
	вх/ вых				0		0

После этого заменяем «.» на «+» в столбце переменной v_y, куда сработало уравнение 8. Все входящие в строку уравнения 6 переменные оказываются отмеченными «+», но так как зависимость 6 уже установлена («–» в столбце учёта), уравнение 6 не переопределённое, а зависимое.

=	ур.	уч.	v_{0y}	v_y	h	$h_п$	t
	P		+		+	2	
+	6	-	+	+			+
+	7	+	1		+		+
+	8	+	+	3	+		
	вх/ вых				0		0

Ветвь «с ударом о потолок».

Таблица модели.

зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	$h_п$
			P					.			+	.
.			9		+			.	.			
	.		10			.			.	.		
.			11		+			.			+	.
	.		12			.			.			.
.		.	13					.	.		+	.
	.	.	14						.	.		.
			15		+	.	+					
		.	8					.		.	+	
			min/max		0		0				0	

Готова к срабатыванию строка уравнения 15. После этого (срабатывание 4) строк, готовых к одинарному срабатыванию, нет. Ищем группу строк, готовую к множественному срабатыванию.
Двух строк с «.» в одних и тех же столбцах тоже нет.

зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	$h_п$
			P					.			+	.
.			9		+			.	.			
	.		10			+			.	.		
.			11		+			.			+	.
	.		12			+			.			.
.		.	13					.	.		+	.
	.	.	14						.	.		.
			15	+	+	4	+					
		.	8					.		.	+	
			min/max		0		0				0	

Начнём искать компоненту со строк 9 и 10 , в них «.» располагаются в трёх столбцах. Ищем строку, в которой все «.» лежат в этих же столбцах. Такая нашлась: 8. Получили компоненту для пробы: три строки — три столбца.

Отмечаем ячейки этой компоненты «.+» во всех столбцах, включая подтаблицу зависимостей. В подтаблице зависимостей строки и столбцы меняются ролями: срабатывать могут столбцы и группы столбцов.
В нашем примере столбцы зависимостей не срабатывают:

Зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	h_n
			P					.			+	.
.+			9	.+	+			.+	.+			
	.+		10	.+		+			.+	.+		
.			11		+			.			+	.
	.		12			+			.			.
.		.	13					.	.		+	.
	.	.	14						.	.		.
			15	+	+	4	+					
		.+	8	.+				.+		.+	+	
			max/ max		0		0				0	

столбцов с одной «.» нет, двух столбцов с точками только в двух строках нет, трёх столбцов с точками только в трёх строках нет. Следовательно, компонента не содержит зависимых уравнений.

Проводим пробное планирование решения внутри компоненты. Выбираем базовое уравнение 9, базовую переменную v_{1y}. Затем срабатывают 10–v_y, 8–v_{0y} . Решение компоненты успешно спланировано.

Зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	h_n
			P					.			+	.
.+			9	.+	+			+	.5			
	.+		10	.+		+			+	.6		
.			11		+			.			+	.
	.		12			+			.			.
.		.	13					.	.		+	.
	.	.	14						.	.		.
			15	+	+	4	+					
		.+	8	.+				.7		+	+	
			max/ max		0		0				0	

Снимаем статус пробности: стираем «.» в метках. Расставляем «+» в столбцах переменных, сработавших в компоненте. Продолжаем планирование: выбрано срабатывание 12–$h_п$. Тогда срабатывает средний столбец зависимостей в (14), в строке 14 в учёте

Зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	$h_п$
			$\overline{P}$					+			+	+
+			9	+	+			+	5			
	+		10	+		+			+	6		
+			11	-	+			+			+	+
	+		12	+		+			+			8
+		+	13	-				+	+		+	+
	+	+	14	-					+	+		+
			15	+	+	4	+					
		+	8	+				7		+	+	
			вх/ вых		0		0				0	

ставим «–», в зависимостях – «+». Тогда аналогично срабатывает правый столбец зависимосией в (13), после него — левый в (11). Затем расставляем «+» в столбцах вычисленных переменных. Уравнения 14, 13, 11 — зависимые, а не переопределённые.

Планирование компоненты завершено. Условие P получилось внешним. Это означает, что его нужно проверять после решения.

Возвращаемся к планированию общей части.

Отмечаем входные переменные, вычисленные в обеих ветвях: v_{oy}, v_y. – в строке вх/вых и в столбцах переменных.

Зависи-мости		ур.	уч.	t	A	v0	vx	v0y	vy	v	x	h
.		1			+	.	.					
.		2			+	.		+				
.	.	3				.	.	+				
	.	4					.		+	.		
		5		+			.				.	
	.	16				.				.		+
		Вх/ вых		0	0			⇑	⇑			0

Проводим последовательно одинарные срабатывания: 2-v_0, 1-v_x, срабатывание столбца зависимостей в (3), 5-x, 4-v, срабатывание столбца зависимостей в (16).

Зависимости		ур.	уч.	t	A	v0	vx	v0y	vy	v	x	h
+		1	+		+	+	9					
+		2	+		+	8		+				
+	+	3	-			+	+	+				
	+	4	+				+		+	11		
		5	+	+			+				10	
	+	16	-			+				+		+
		Вх/вых		0	0			⇑	⇑			0

Планирование решения завершено.

Замечание.

Если известно, что задача формально не переопределённая, то при одинарной ориентации можно не включать столбцы зависимостей в таблицу. Тогда определить, что уравнение зависимое (внешнее) можно по тому, что в его строке все метки становятся «+», а цифр (номеров срабатываний) нет. В рассмотренном примере это относится к ветви без удара о потолок и к общей ветви. В ветви с ударом о потолок работу с подтаблицей зависимостей можно прекратить после того, как убелились в выборе компоненты. Таблица с такой разметкой иcпользована далее.

Табличный вариант преобразования задач.

Рассмотрим на том же примере алгоритм, соответствующий в графовом представлении инвертированию пути из вершины исходной переменной к вершине искомой переменной. Проинвертируем в ветви «с ударом о потолок» путь от t_1 к $h_п$, чтобы получить задачу с безальтернативной формулировкой.

Из столбца t_1 можно попасть в строку 9 (в неё входит «+» в этом столбце); строка 9 срабатывала в v_{1y} (срабатывание 5) ,после инверсии будет срабатывать в t_1. Переносим метку «5» в этот столбец. Из v_{1y} можно попасть в строку 12, срабатывавшую в нужную нам переменную

Зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	$h_п$
			P					+			+	+
+			9	+	+			+	5			
	+		10	+		+			+	6		
.			11	-	+			+			+	+
	.		12	+		+			+			8
.		.	13	-				+	+		+	+
	.	.	14	-					+	+		+
			15	+	+	4	+					
		+	8	+				7		+	+	
			вх/вых		0		0				0	

$h_п$.

Перенесём метку «8». Перенесём метку «0» в строке «вх/вых» из столбца t_1 в столбец $h_п$. Преобразование закончено, но *нумерация срабатываний теперь может не соответствовать порядку вычислений.*

Зависимости			ур.	уч.	t_1	t_2	t	v_{0y}	v_{1y}	v_y	h	$h_п$
			P					+			+	+
+			9	+	5			+	+			
	+		10	+		+			+	6		
.			11	-	+			+			+	+
	.		12	+		+			8			+
.		.	13	-				+	+		+	+
	.	.	14	-					+	+		+
			15	+	+	4	+					
		+	8	+				7		+	+	
			вх/вых				0				0	0

Покажем, что полученный план соответствует одной множественной компоненте. Для этого убедимся, что из какой-либо строки можно попасть обратно в неё, двигаясь по срабатываниям, и в такие пути попадут все отмеченные срабатывания. Будем писать номера уравнений в скобках, а условные номера срабатываний без скобок. Пути от строки (9) к этой же строке:

(9)-5-(15)-4-(10),(12);

(10)-6-(8)-7-(9);

(12)-8-(9).

Вошли все номера срабатываний, соответствующие вычисляемым переменным, и все номера внутренних уравнений.

Приложение.
Доказательство утверждений о графах решения.

T1.

Любой разомкнуто правильно ориентированный граф Гр можно преобразовать в замкнуто правильно ориентированный граф Гз с теми же компонентами инверсией попарно не пересекающихся путей из вершин базовых переменных в вершины базовых уравнений, причём эти компоненты будут бикомпонентами Гз.

Проведём в рассматриваемой компоненте Гр инверсии путей из вершин базовых переменных в вершины базовых уравнений последовательно, один путь за другим, не заботясь о том, чтобы эти пути не пересекались. Пусть провели инверсию части путей. Докажем, что после этого существует хотя бы один путь из какой-либо оставшейся (не *нейтрализованной*) вершины базовой переменной в какую-либо оставшуюся вершину базового уравнения.

Рассмотрим подграф Г*, порождённый множеством вершин, из которых достижимы вершины оставшихся базовых уравнений, включая сами эти вершины (считая, что всякая вершина достижима из самой себя). Если Г* не содержит вершин базовых переменных, то в нём в каждую вершину переменной заходит дуга из вершины уравнения из Г*, так что число уравнений в Г* равно числу переменных в Г* плюс число оставшихся вершин базовых уравнений, то есть уравнений больше, чем переменных. Но если система уравнений содержит подсистему, в которой уравнений больше, чем переменных, то её граф не может быть правильно сориентирован (если нет внешних вершин (нет зависимых уравнений), то при разомкнутой правильной ориентации число базовых уравнений равно числу базовых переменных, а число остальных уравнений равно числу остальных переменных, так как из каждого уравнения выражается одна переменная). Но Гр был правильно ориентированным. Значит Г* содержит вершины базовых переменных, то есть из этих базовых переменных достижимы вершины базовых уравнений и можно продолжить нейтрализацию.

Пусть нейтрализовали все базовые вершины последовательностью инверсий путей и1,и2,...,ик. Покажем индукцией по к, что результат получается такой же, как при инверсии некоторой последовательности попарно не пересекающихся путей. Пусть это верно для к=р и после р нейтрализаций получили эквивалентные р попарно не пересекающихся путей П, изменивших направление. Проведём инверсию ещё одного пути п. Каждая дуга пересечения П и п проинвертировалась два раза и восстановила своё направление, так что изменили направление только не пересекающиеся части П и п, причём для каждого пересечения П с п можно движение, начатое в П, продолжить в п, а движение, начатое в п, продолжить в П. Проходя так через все встречающиеся пересечения, получим не пересекающиеся пути.

Докажем теперь, что после нейтрализации всех базовых вершин компонента становится сильно связанной. Пусть это не так и в Гз из вершины а1 не достижима вершина а2. Рассмотрим подграф Г1 компоненты Гз, порождённый всеми вершинами, достижимыми из а1, и подграф Г2, порождённый остальными вершинами Гз. Покажем, что они отображают подсистемы уравнений, которые можно решать отдельно, а следовательно множество уравнений, отражаемое Гз, не есть минимальная подсистема. Действительно, если в Гз есть дуга из вершины уравнения а в вершину переменной х, то, если в Г1 входит а (то есть а достижима из а1), то входит и х, а если входит х, то входит и а (иначе х не могла бы быть достижимой из а1), так что Г1 не имеет начальных и конечных вершин, все вершины входят в циклы (замкнутые пути). Перестроим Г1 в разомкнутый граф решения Гр1, проинвертировав нужное число дуг (инверсия дуги с→у означает, что у принимается вершиной базовой переменной (в вершину переменной у после инверсии не заходит никакой

дуги), а с становится вершиной базового уравнения (из неё теперь не исходит никакой дуги); после этой инверсии, если в Г1 остался хотя бы один цикл, инвертируем дугу из этого цикла, и т. д., пока не исчезнут все циклы). Для Г2 применимы аналогичные рассуждения и построения, его преобразуем в разомкнутый план решения Гр2. Таким образом, если Гз был бы не сильно связанным, то он отображал бы не минимальную подсистему.
Утверждение Т1 доказано.

Т2.

Планы решения, отличающиеся только порядком расположения составляющих его действий (минимальных подсистем), но не самими действиями, считаем *эквивалентными.*
Все планы решения задачи с минимумом данных на модели (или подмодели, то есть на подмножестве уравнений модели), уравнения которой независимы, эквивалентны.

Доказательство.
Граф решения независимой системы не содержит внешних вершин уравнений. Пусть некоторую переменную x можно вычислить двумя не эквивалентными способами. Пусть это можно сделать за один шаг. Тогда есть две различные минимальные подсистемы A и B, каждая из которых содержит только исходные переменные и вычисляемые в этой подсистеме переменные (включая x). Объединим B с A, взяв граф A без изменений и добавив вершины уравнений B, не входящих в A, сохранив ориентацию примыкающих к ним дуг, имевшуюся в B. Получим неправильную ориентацию C: какие-то переменные в A и B вычисляются по-разному; пусть среди них есть x1. Проинвертируем в C какой-либо путь p максимальной длины, лежащий в B и заканчивающийся в x1. Если проинвертированный p заканчивается в A, то удалим из C последнюю в p вершину уравнения. Таким же образом проинвертируем друг за другом и остальные пути в B, заканчивающиеся в A, если такие есть. Получим в результате правильно ориентированный граф, которому соответствует вычисление всех переменных, вычислявшихся в A и B, но в котором меньше уравнений или/и исходных данных. Значит, система C не была независима или/и набор исходныхданных был не минимальным.
Пусть теперь все переменные, вычисляемые за k шагов, можно вычислить только по эквивалентным планам, то есть соответствующий подграф решения единственный, и пусть на k+1 шаге переменную x* можно вычислить двумя различными способами. Тогда рассуждения, аналогичные предыдущим, приводят к аналогичному противоречию. По индукции Т2 доказана.

Т3.

Любую замкнуто правильную ориентацию графа решения задачи с минимумом данных можно преобразовать в любую другую замкнуто правильную ориентацию решения этой же задачи инверсией попарно не пересекающихся циклов и/или путей из внутренних вершин уравнений во внешние.

Действительно, пусть Ф1 и Ф2 — различные замкнуто правильные ориентации

графа решения некоторой задачи с минимумом данных. Тогда, по Т2, отличаются множества независимых уравнений, используемых в этих решениях, или/и отличается ориентация дуг внутри компонент. Будем перестраивать Ф1 в Ф2. Пусть у1 – вершина уравнения, внешняя в Ф2, но внутренняя в Ф1: в Ф1 из неё исходит дуга, направленная к вершине переменной x1. Проинвертируем эту дугу. В Ф2 к x1 направлена другая дуга — ведь это не исходная переменная. Пусть это дуга из у2. Проинвертируем её. Но, если у2 в Ф1 не внешняя, то в неё заходила другая дуга, пусть из x2; чтобы ориентация осталась правильной, а значит стала такой же, как в Ф2, её нужно проинвертировать. Продолжая аналогично, инверсию пути можем закончить только во внешней в Ф1 вершине уравнения. Тогда в Ф1 будет проинвертирован путь из внутренней вершины во внешнюю, правильность и замкнутость ориентации сохранится и этот путь станет ориентированным так же, как в Ф2. Если полученная ориентация ещё отличается от Ф2 множеством независимых уравнений, то, действуя аналогично, проинвертируем ещё один путь, и т. д.
Пусть получили ориентацию, отличающуюся от Ф2 только ориентацией бикомпонент. Тогда, начав переориентацию Ф1 в Ф2 с дуги, исходящей в Ф1 из вершины уравнения p и ведя её аналогично описанному, сможем остановиться только на дуге, заходящей в p в Ф1, но исходящей из p в Ф2. При этом проинвертируем цикл. Проинвертировав все отличающиеся циклы, получим Ф2. Как мы видели при доказательстве Т1, последовательность инверсий путей эквивалентна множеству попарно не пересекающихся инверсий.

Т4.

Любую замкнуто правильную ориентацию графа решения полной задачи с минимумом данных можно преобразовать в замкнуто правильную ориентацию решения любой другой полной задачи с минимумом данных на этой же модели инверсией попарно не пересекающихся путей из исходных переменных в искомые переменные.

Действительно, полные задачи с минимумом данных на заданной модели отличаются лишь разделением переменных на исходные и искомые, причём число вершин искомых переменных правильно ориентированного графа равно числу внутренних вершин уравнений и оно одинаково для всех полных задач (см. утверждение (6)). Пусть Ф1 и Ф2 — графы решения различных задач, использующие одно и то же максимальное множество независимых уравнений С*. Тогда хотя бы одна вершина x1 переменной, исходная в Ф1, не является исходной в Ф2. Будем перестраивать Ф1 в Ф2 аналогично тому, как это делали при доказательстве Т3. Пусть в Ф2 в x1 заходит дуга из y1. Проинвертируем и в Ф1 дугу из x1 в y1. Пусть в Ф1 из y1 вела дуга в x2. В Ф2 так быть не могло и так не может остаться после инверирования первой дуги, иначе нарушится правильность ориентации. Проинвертируем эту дугу. В изменённом Ф1 x2 стала исходной. Если в Ф2 это не так, продолжим инвертирование аналогично. Поскольку все вершины уравнений в С* участвуют в решении, процесс инвертирования может закончиться только вершиной переменной, исходной в Ф2. В результате него в Ф1 проинвертируется путь из вершины исходной переменной в вершину искомой переменной. Будем повторять такой процесс, пока множество исходных переменных перестроенного Ф1 не совпадёт с множеством исходных переменных Ф2. Последовательность произведённых инверсий эквивалентна множеству попарно не пересекающихся инверсий.

Литература.

1. Тыугу Э. Х. Концептуальное программирование.– М.: Наука. Главная редакция физико-матеметической литепатуры, 1984.

2. Карзанов А.В., Фараджев И.А. Планирование вычислений при решении задач на вычислительных моделях. - Программирование, №4, 1975.

3. Нариньяни А.С., Телерман В.В., Ушаков Д.М., Шевцов И.Е. Программирование в ограничениях и недоопределённые модели. (Российский НИИ Искусственного интеллекта). pandia.ru/text/79/301/5484.php , 13.03.2017.

4. Евстигнеев В. А. Применение теории графов в программировании.– М.: Наука. Главная редакция физико-матеметической литепатуры, 1985.

5. Кожухов И.Б., Прокофьев А.А. Математика. Школьникам и абитурьентам. - М.: Махаон, 2005.

6. Блуменау В.Л. Учебные задачи: черновики плюс ... URL: http://www.zdch.far.ru/ Дата обращения 28.03.2017.