

УДК 004.896

Р. А. КовалёвПетербургский государственный университет путей сообщения
Императора Александра I**ОПИСАНИЕ АЛГОРИТМА ПОИСКА МАРШРУТОВ ПРИ СИНТЕЗЕ ТАБЛИЦ
ВЗАИМОЗАВИСИМОСТЕЙ ПО СХЕМАТИЧЕСКОМУ ПЛАНУ СТАНЦИИ**

В статье рассмотрен основной алгоритм поиска обобщенных маршрутов по схематическому плану станции, используемый при синтезе поездных (маршрутов приема, передачи, отправления, пропуска), маневровых и составных маршрутов. Под обобщенным маршрутом понимается последовательность элементов станции, участвующих в передвижении поездного состава. Описан процесс преобразования входных данных в модель, используемую алгоритмом, и показан пример его работы. Описание алгоритма выполнено с помощью дополненной автором логической схемы алгоритмов (ЛСА) и может быть реализовано на любом Тьюринг-полном языке программирования.

таблица взаимозависимостей (ТВЗ), алгоритм синтеза, электронный документооборот, схематический план станции, ЛСА.

Введение

Под ТВЗ понимается таблица взаимозависимостей, в которую входят следующие данные:

- таблица маршрутов и показаний световых сигналов,
- таблица негабаритных секций и охранных стрелок,
- расчет переезда на станции,
- таблица стрелок, которые дополнительно замыкаются для предотвращения их взреза при угловых заездах,
- таблица стрелок, имеющих замедление на размыкание или оборудованных автовозвратом в исходное положение.

При составлении ТВЗ необходимо соблюдать определенную последовательность действий, которая описывается в алгоритме синтеза ТВЗ, где отдельные расчеты разделены по программным модулям, а основной алгоритм производит последовательный вызов интерфейсов этих модулей. Такая модульная система позволяет разрабатывать отдельные виды синтеза, реализующие заранее принятые соглашения (интерфейсы), независимо

от остальных, а также в некоторой степени упрощает тестирование алгоритмов.

1 Модель данных

Построение алгоритма стало возможным благодаря представлению данных схематического плана станции в виде типизированного документа в формате XML [1], в котором явно разделены различные виды составляющих элементов схематического плана. Таким образом, графическая часть является дополнительной составляющей к логической модели схематического плана и именно логическая модель используется в дальнейшем для работы алгоритма (рис. 1).

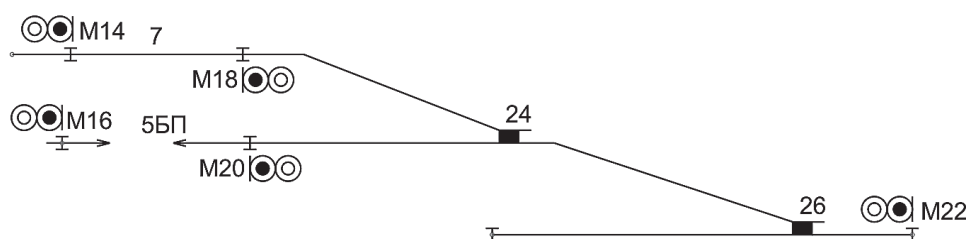
Графическая часть документа упускает часть информации для наглядности (рис. 2). При этом специализированные редакторы позволяют создавать и изменять документы таким образом, как будто пользователь работает с графическим изображением документа и набором атрибутов для каждого отдельного примитива или группы примитивов.

```

<SSP: Светофор id="39" Парк="1" ВводПараметров="с БДКП" МестоУстановки="справа"
  Марка="Св2-карл-Ч-Ж/Б Ф" ТипСветофора="линзовый" НазначениеСвет="маневровый"
  ТипМачты="отсутствует" КолОгней="2">
<SSP: Показание Номер="10" Показ="синий" Мигание="нет" ДвухнитеваяЛампа="нет" Символ="
  " ИспКакПригл="нет"/>
<SSP: Показание Номер="11" Показ="белый" Мигание="нет" ДвухнитеваяЛампа="нет" Символ="
  " ИспКакПригл="нет"/>
</SSP: Светофор>

```

Рис. 1. Фрагмент описания входных данных

Рис. 2. Фрагмент схематического плана станции.
Графическое представление

Для функционирования разрабатываемого алгоритма недостаточно описания только графической и логической составляющей каждого примитива. Необходимо представить исходные данные в виде строго формализованной модели с определенными правилами взаимодействия элементов внутри нее. В качестве такой модели была разработана иерархическая структура данных, представляющая собой иерархию графов (рис. 3).

Представленная на рис. 3 модель является графом первого уровня разработанной структуры данных. Вершины графа – модели, основанные на элементарных составляющих документа схематического плана станции, а дуги графа – логические связи между этими моделями. Граф является ориентированным.

При графическом отображении графа первого уровня опускаются пути, не имеющие наименования, а для вершин выводится наименование или – если наименование не указано – другая идентификационная информация.

Обозначив оргграф первого уровня G_1 , можно ввести следующие обозначения:

$$G_1 = (V, E);$$

V – множество вершин; (1)

E – множество пар вершин
 $(u, v) \in V, (u, v) \neq (v, u)$.

Граф G_1 служит основой для графа второго уровня G_2 (рис. 4) и состоит из вершин, описывающих секции однопутного плана станции и ребер, описывающих изолирующие стыки. Таким образом, каждая вершина графа G_2 представляет собой фрагмент графа G_1 .

Для наглядности $G_2 \ni G_1$ на рис. 4.

2 Описание алгоритма

Необходимо выделить несколько условий, которые должны соблюдаться в разрабатываемом алгоритме. Основное из них – соблюдение последовательности действий при построении ТВЗ. В качестве примера можно привести утверждение, что для определения показаний светофоров необходимо иметь по-

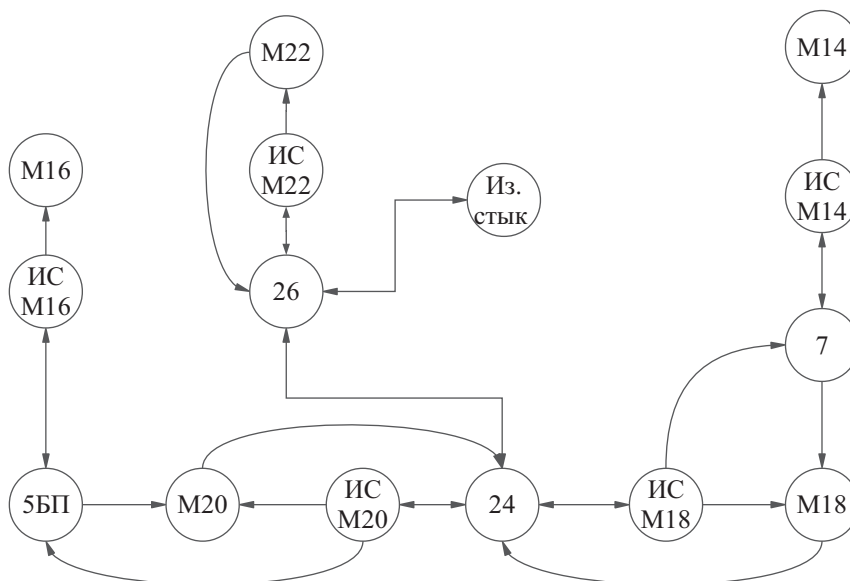
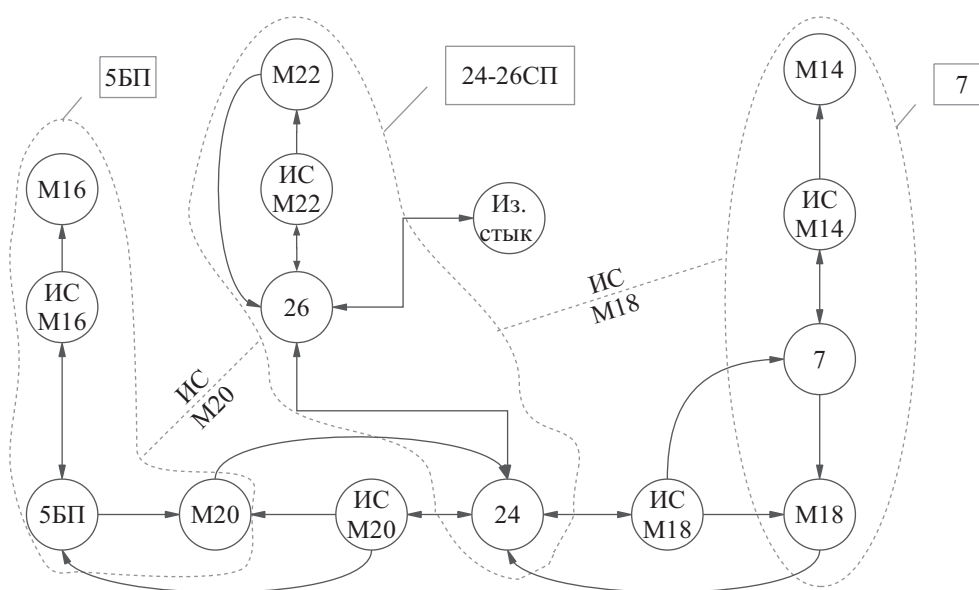


Рис. 3. Отображение фрагмента схематического плана станции в виде графа

Рис. 4. Граф второго уровня G_2

строенные маршруты. В результате можно составить следующую общую схему алгоритма:

1. Построение графов G_1 и G_2 ;
2. Поиск поездных и маневровых маршрутов;
3. Составление маршрутов безостановочного пропуска на основании поездных маршрутов;

4. Составление таблицы показаний светофоров в поездных и маневровых маршрутах;

5. Поиск случаев охранности и негабаритности;
6. Поиск переездов и расчет переездной сигнализации;
7. Поиск враждебных маршрутов;
8. Составление таблицы угловых заездов.

Для построения маршрутов реализован алгоритм поиска в глубину по графу [2]. Вначале выбираются первичные вершины поиска. Из каждой первичной вершины производится посекционный поиск вглубь. При прохождении секции, удовлетворяющей условиям поиска, пройденная цепочка секций запоминается, и при необходимости поиск может быть продолжен. Поиск заканчивается при достижении вершин графа, для которых нет исходящих ребер или все исходящие ребра ведут на вершины, уже пройденные в данной цепочке поиска.

Для описания алгоритма необходимо ввести следующие обозначения:

V_1 – множество вершин G_1 ;

V_2 – множество вершин G_2 .

Алгоритм описан с помощью ЛСА [3]. Для упрощения восприятия предлагается дополнить описание алгоритма специальными операторами, не входящими в классическое ЛСА:

- оператор привязки «*let [name, value]*». Этот оператор является эквивалентом математического равенства вида $f(value) = name$ и позволяет явно указать глобальные переменные и алгоритмы в описании взамен их упоминания в описаниях операторов ЛСА;

- «*map.a [b, value]*» – итеративный оператор накопления по пронумерованному множеству «*b*»; «*a*» – i -й элемент множества на i -й итерации. Накапливаемым результатом является множество $R = \{x\}$, формируемое на каждой итерации путем вызова оператора $A(x_i)$;

- «*mapc.a [b, value]*» – итеративный оператор накопления, идентичный оператору «*map*», но объединяющий результаты вызова $A(x_i)$ в одно множество.

$A(a)$ – передача аргумента «*a*» оператору накопления;

$B(G, v)$ – множество ребер графа G вида $u[v, k]$, где $v \in V, k \in V, V$ – множество вершин графа G ;

$C(u)$ – вершина v графа G_2 , в состав которой входит вершина v_2 графа G_1 при условии, что $u = (v_1, v_2)$;

$D(S)$ – проверка множества S на равенство пустому множеству;

$p_1(v)$ – проверка v по критериям начального элемента маршрута;

$p_2(v)$ – проверка v по критериям конечного элемента маршрута;

$p_3(v)$ – критерий продолжения поиска.

Алгоритм в терминах дополненного ЛСА приведен на рис. 5.

Пример работы алгоритма приведен на рис. 6.

Вычислительная сложность алгоритма $O = (V + E)$. Поиск сначала выполняется по графу G_2 , затем по полученным результатам строятся маршруты, содержащие вершины графа G_1 . Это позволяет значительно увеличить производительность поиска, а также дает возможность хранить информацию о секциях, участвующих в маршрутах, необходимую в других алгоритмах синтеза.

Заключение

С внедрением электронного документооборота появляется возможность автоматизировать многие технологические процессы, связанные с обработкой и созданием технической документации. Одним из наиболее сложных и ответственных процессов является

$$\begin{aligned} &let [V'_1, map.a[V_1, \bar{p}_1(a) \uparrow^1 A(a) \downarrow^1 .]] \\ &let [U_1, mapc.a[V'_1, B(G_1, a)]] \\ &let [S_1, map.a[U_1, C(a)]] \\ &let [Alg_1(S), D(S) \uparrow^2 map.a[S, p_2 \uparrow^3 \downarrow^4 Alg_1(B(G_2, a)) \downarrow^3 A(a) p_3 \uparrow^4 \downarrow^2 .]] \\ &Alg_1(S_1) \end{aligned}$$

Рис. 5. Алгоритм поиска маршрутов по графам G_2 и G_1

