

хранится в этом виде не только в базах данных брестского автобусного парка, но и в базах данных других городов, то данная модель способствует применению разрабатываемого приложения не только в пределах г. Бреста, но и в других городах Республики.

Основной задачей разрабатываемого в этой работа приложения конечно же является поиск оптимальных маршрутов между двумя остановками и вывод наиболее выгодных. Исходя из указанных требований был разработан следующий алгоритм поиска маршрутов без пересадок:

- 1) Выбор остановок и ввод времени
- 2) Поиск подходящих прямых маршрутов (Поиск осуществляется выбором общих маршрутов между двумя остановками)
- 3) Вычисление времени прибытия (Время же берется из класса маршрут)
- 4) Отбор наиболее выгодных маршрутов по временным затратам (В итоге будет выводиться три наиболее быстрых маршрута)

Т.к. мы не всегда можем добраться до нужной остановки без пересадок, то необходим алгоритм поиска нужных маршрутов. Что бы добраться из одной точки города в любую другую будет достаточно одной пересадки. Основываясь на этом мы разработали алгоритм поиска маршрутов с возможностью пересадки на другой маршрут в указанной программой остановке:

- 1) Поиск общей остановки у маршрутов, которые проходят через начальную и конечную остановки (Генерируем маршруты по алгоритму указанному выше)

- 2) Поиск наиболее выгодных маршрутов с начальной остановки до промежуточной

- 3) Поиск наиболее выгодных маршрутов с промежуточной остановки до конечной (Так же используем вышеуказанный алгоритм)

- 4) Отбор наиболее выгодных маршрутов по временным затратам

Так же существуют такие места, где иногда переход с одного остановочного пункта на другой может существенно сократить временные затраты на передвижение. Такие места мы объединяем в так называемые «узлы остановок» Остановки, объединенные в узел, зачастую они находятся по разные стороны одного перекрестка, и переход между остановками занимает не более двух минут.

В программе маршруты проходящие через «узел» будут рассматриваться как маршруты проходящие через одну остановку. Но если все же выбор маршрута потребует перехода с одной остановки на другую, то приложение покажет, как и куда следует перейти.

УДК 519.713

Гутников В.С.

Научный руководитель: ст. преподаватель Тузик И.В.

АВТОМАТИЗАЦИЯ ПРЕДСТАВЛЕНИЯ КОНЪЮНКЦИИ ДВУХ ПЕРЕМЕННЫХ В ОДНОЭЛЕМЕНТНОМ БАЗИСЕ

В данной работе рассматриваются возможности программного средства для поиска наилучшего представления конъюнкции в одноэлементном базисе в виде пошаговых действий.

Если булева функция образует базис, то любая логическая функция может быть представлена в нем. Но не каждая функция может служить базисом. По-

этому проводится предварительная проверка на возможность использования заданной функции в качестве базиса, и последующее нахождение представлений в этом базисе отрицания, конъюнкции и констант.

Основной целью работы было избавить пользователя от трудоемкого поиска подходящих вариантов представления конъюнкции в одноэлементном базисе путем перебора вручную. Рассматриваемая программа способна найти представления исходя из заданного выражения или значений булевой функции, что избавляет пользователя от всех промежуточных вычислений.

Пользователю разрешено вводить выражения, которые содержат от одной до 26 переменных, или вводить значения булевой функции. Поэтому для выполнения необходимых преобразований автором были разработаны методы, позволяющие увеличить скорость вычисления некоторых объемных данных. В результате работы программа выводит не только решение поставленной задачи, но и промежуточные результаты: 1) таблицу истинности заданной функции, 2) ее полином Жегалкина, 3) проверку критериев полноты системы, образованной заданной функцией.

Ниже приводится экранная копия (рис. 1), на которой показаны введенные данные (функция, заданная выражением), а также полученная таблица истинности и найденный полином Жегалкина.



Рисунок 1

После анализа выражения и вычисления промежуточных данных программа приступает к отысканию требуемых представлений. Для конъюнкций, в отличие от констант и отрицания, выводятся все возможные варианты пред-

ставления в заданном базисе. Пользователю предлагается выбрать один из вариантов, после чего выводится окончательный результат.

На рисунке 2 показаны полученные представления констант, отрицания и конъюнкции, и окончательные выражения для них, найденные исходя из выбранных пользователем вариантов.

Представления константы 1

#	Номер набора	Переменные набора
<input type="checkbox"/>	14	a b c ne
<input checked="" type="checkbox"/>	13	a b ce
<input type="checkbox"/>	11	a b ce
<input type="checkbox"/>	7	abec

Представление константы 0 будет представлено через константу 1: $0(x) = \neg e(x)$

Представление отрицания

#	Набор
a	0000

Представление конъюнкции

#	Замена переменных	c	β	γ	результат
<input type="checkbox"/>	c e = 0	false	false	true	(alb) \oplus 1
<input type="checkbox"/>	c e = 1	false	false	true	(alb) \oplus 1
<input type="checkbox"/>	b e = 0	true	false	true	a \oplus (alc) \oplus 1
<input checked="" type="checkbox"/>	b e = 1	false	false	true	(alc) \oplus 1
<input type="checkbox"/>	b c = 1	false	false	true	(ale) \oplus 1
<input type="checkbox"/>	a e = 0	true	false	true	b \oplus (bnc) \oplus 1
<input type="checkbox"/>	a e = 1	false	false	true	(bnc) \oplus 1
<input type="checkbox"/>	a c = 1	false	false	true	(ble) \oplus 1

Результат

$\neg x = f(x, x, x, x)$

$1 = e(x) = f(x, x, f(x, x, x, x), x)$

$0 = \neg e(x) = f(f(x, x, f(x, x, x, x), x), f(x, x, f(x, x, x, x), x), f(x, x, f(x, x, x, x), x), f(x, x, f(x, x, x, x), x))$

$alc = \neg((a, f(a, a, f(a, a, a, a), a), c, f(a, a, f(a, a, a, a), a)))$

Рисунок 2

Данная работа может быть развита в математический пакет булевой алгебры для решения большого количества задач, связанных с анализом логических функций.