

Федеральное агентство связи

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»

(СибГУТИ)

Е. П. Мачикина

**УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РГР
ПО ДИСЦИПЛИНЕ
«МАТЕМАТИЧЕСКАЯ ЛОГИКА
И ТЕОРИЯ АЛГОРИТМОВ»**

Учебно-методическое пособие

Новосибирск
2020

УДК [510.6+510.5](075.8)

Утверждено редакционно-издательским советом СибГУТИ

Рецензент: доцент каф. высшей математики СибГУТИ,
к.т.н., доцент Храмова Т.В.

Мачикина Е.П. Указания по выполнению РГР по дисциплине «Математическая логика и теория алгоритмов» : учебно-методическое пособие / Е.П. Мачикина ; Сибирский государственный университет телекоммуникаций и информатики ; каф. прикладной математики и кибернетики. – Новосибирск, 2020. – 44 с.

В пособии приведен необходимый теоретический материал, примеры решения задач РГР по дисциплине «Математическая логика и теория алгоритмов» для студентов второго курса заочного факультета СибГУТИ, направление 09.03.01 Информатика и вычислительная техника, профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем».

© Мачикина Е.П., 2020

© Сибирский государственный университет
телекоммуникаций и информатики, 2020

ОГЛАВЛЕНИЕ

Правила выполнения РГР	4
Задача 1	6
Задачи 2–5	9
Задача 6	16
Задача 7	26
Задача 8	33
Список литературы.....	42

Правила выполнения РГР

Освоение курса «Математическая логика и теория алгоритмов» невозможно без практики решения задач, при этом необходимо иметь знания по дискретной математике. При изучении курса «Математическая логика и теория алгоритмов» кроме конспекта лекций рекомендуется пользоваться учебниками, приведенными в списке литературы.

Выполнение РГР является обязательным результатом самостоятельной работы студентов, изучающих дисциплину «Математическая логика и теория алгоритмов». При выполнении РГР необходимо строго придерживаться указанных ниже правил. Работы, выполненные без соблюдения этих правил, не считаются и возвращаются студенту для доработки.

- РГР состоит из 8 задач, каждая задача имеет 25 вариантов. Номер варианта студента определяется по номеру в журнале группы.
- Решения задач необходимо располагать в порядке номеров, указанных в задании, сохраняя номера задач. Решение каждой задачи должно быть подробно и логично описано, отдельно выписан ответ задачи. Перед решением приводить условие задачи согласно варианту.
- Работа выполняется в редакторе Word (или аналог), файл с работой должен быть загружен в ЭИОС СибГУТИ.
- После получения прорецензированной работы с оценкой «незачет» студент должен исправить все отмеченные преподавателем ошибки и недочеты, выполнить все рекомендации и повторно загрузить в ЭИОС СибГУТИ (комментарии преподавателя не удалять). Без выполненной контрольной работы студент к экзамену не допускается.
- Каждая задача оценивается определенным количеством баллов. Критерии оценки приведены ниже. Максимальное количество баллов – 35, для зачета РГР необходимо получить не меньше 20 баллов.

Баллы за задачу			Критерии оценки задач
Задачи 8	Задачи 1,6,7	Задачи 2,3,4,5	
8	5	3	Задание выполнено полностью и абсолютно правильно. Приведено полное описание решения задачи.
6-7	4	2	Задание выполнено полностью и правильно, но решение содержит некоторые неточности и несущественные ошибки. Есть несущественные пробелы в оформлении решения задачи.
3-5	3-2	1	Задание выполнено не полностью, с существенными ошибками, но подход к решению,

			идея решения и метод – правильны. Описание решения задачи неполное и поверхностное.
1-2	1	0	Задание выполнено частично, имеет ошибки, осуществлена попытка решения на основе правильных методов и идей решения. Описание решения неполное.
0	0	0	Задание не выполнено, выполнено не самостоятельно.

Задача 1

Проверить полноту системы логических функций, используя критерий Поста. Заполнение таблицы Поста должно быть обоснованным. Если система не полна, то достроить до полной системы. (Запрещается дополнять функциями штрих Шеффера, стрелка Пирса, константами 0 и 1, отрицанием, конъюнкцией.)

Используя функции полученной полной системы, выразить константы 0 и 1, отрицание, конъюнкцию.

Булевы функции

1	$x_1 \rightarrow x_2 \& x_3$	6	$x_1 \rightarrow (x_2 \downarrow x_3)$	11	$x_1 x_2 \downarrow (x_2 x_3 \vee x_1 x_3)$	16	$x_1 \sim \neg x_2 \& x_3$
2	$\neg x_1 \sim x_2 \& x_3$	7	$x_1 \downarrow (x_2 \downarrow x_3)$	12	$(x_1 \rightarrow x_2) \rightarrow x_3$	17	(11010100)
3	$\neg x_1 \& (x_2 \oplus x_3)$	8	(01001000)	13	$x_1 \mid (x_2 \& x_3)$	18	$x_1 x_2 \oplus (x_2 x_3 \vee x_1 x_3)$
4	(01011100)	9	$x_1 x_2 \rightarrow x_2 x_3 \vee x_1 x_3$	14	$(x_1 \rightarrow x_2) \rightarrow \neg x_3$	19	$x_1 \& (x_2 \oplus \neg x_3)$
5	$x_1 x_2 \vee x_2 x_3 \vee x_1 x_3$	10	$x_1 \rightarrow \neg x_2 \& x_3$	15	$x_1 \& (x_2 \rightarrow x_3)$	20	$x_1 \downarrow (x_2 \rightarrow x_3)$

Варианты

№	функции	№	функции	№	функции	№	функции	№	функции
1	1,5,6	6	5,8,13	11	7,12,18	16	2,9,15	21	1,6,10
2	2,3,7	7	5,11,12	12	7,13,19	17	3,6,7	22	3,12,19
3	2,7,8	8	5,18,19	13	8,15,20	18	4,9,11	23	4,9,20
4	3,9,10	9	6,12,14	14	1,9,10	19	5,17,20	24	4,7,14
5	4,15,19	10	6,11,15	15	1,13,20	20	2,4,5	25	6,10,14

Необходимый теоретический материал

Система функций $F = \{f_1, f_2, \dots, f_k\}$ называется *полной*, если любая логическая функция может быть выражена через функции системы F с помощью подстановки (т.е. составления сложных функций).

Логическая функция f *сохраняет константу 0*, если $f(0,0,\dots,0) = 0$. Функцию f называют *функцией, сохраняющей константу 1*, если $f(1,1,\dots,1) = 1$. Множество всех функций, сохраняющих константу 0, обозначим T_0 , а множество всех функций, сохраняющих константу 1, — T_1 .

Двойственной к булевой функции $f(x_1, x_2, \dots, x_n)$ называется функция $f^*(x_1, x_2, \dots, x_n) = \neg f(\neg x_1, \neg x_2, \dots, \neg x_n)$. Если двойственная функция f^* совпадает с исходной функцией f , то такая функция f называется *самодвойственной*.

Из определения следует, что функция самодвойственна тогда и только тогда, когда на взаимно противоположных наборах значений переменных функция принимает противоположные значения. Множество всех самодвойственных функций обозначим S .

Логическая функция f *монотонная*, если для любых пар α, β наборов значений переменных таких, что $\alpha \leq \beta$, $f(\alpha) \leq f(\beta)$. Множество всех монотонных функций принято обозначать через M .

Если функция f не является монотонной, то найдутся два таких набора α, β и индекс i , что $\alpha = (\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n)$, $\beta = (\beta_1, \dots, \beta_{i-1}, 1, \beta_{i+1}, \dots, \beta_n)$, $f(\alpha) = 1$, $f(\beta) = 0$, т.е. эти два набора различаются значениями только в одной компоненте, а значение функции равно 0 на большем наборе и равно 1 на меньшем.

Булева функция $f(x_1, x_2, \dots, x_n)$ представлена в виде *полинома Жегалкина* от n переменных, если
$$f(x_1, \dots, x_n) = \bigoplus_{\{i_1, i_2, \dots, i_m\} \subseteq I} a_{i_1 i_2 \dots i_m} x_{i_1} x_{i_2} \dots x_{i_m},$$

где коэффициенты полинома $a_{i_1 i_2 \dots i_m} \in \{0, 1\}$ индексированы всеми возможными подмножествами множества $I = \{1, 2, \dots, n\}$. Между коэффициентами и переменными стоит конъюнкция, которая опущена для сокращения длины формулы.

Утверждение. Полином Жегалкина для любой булевой функции определен однозначно.

Пример. Пусть булева функция задана столбцом значений $f = (1, 1, 0, 0, 1, 0, 1, 1)$. Найдем полином Жегалкина, соответствующий заданной функции.

В общем виде полином от трех переменных имеет вид $f(x_1, x_2, x_3) = a_{123}x_1x_2x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_0$.

Определим вид полинома методом неопределенных коэффициентов. Количество коэффициентов совпадает с количеством значений логической функции, коэффициенты определяются однозначно из системы уравнений.

$$f(0, 0, 0) = a_0 = 1$$

$$f(0, 0, 1) = a_3 \oplus a_0 = a_3 \oplus 1 = 1, \quad a_3 = 0$$

$$f(0, 1, 0) = a_2 \oplus 1 = 0, \quad a_2 = 1$$

$$f(1, 0, 0) = a_1 \oplus 1 = 1, \quad a_1 = 0$$

$$f(1, 1, 0) = a_{12}x_1x_2 \oplus a_1x_1 \oplus a_2x_2 \oplus a_0 = a_{12} \oplus 1 \oplus 1 = 1, \quad a_{12} = 1$$

$$f(1, 0, 1) = a_{13}x_1x_3 \oplus a_1x_1 \oplus a_3x_3 \oplus a_0 = a_{13} \oplus 1 = 0, \quad a_{13} = 1$$

$$f(0, 1, 1) = a_{23}x_2x_3 \oplus a_2x_2 \oplus a_3x_3 \oplus a_0 = a_{23} \oplus 1 \oplus 1 = 0, \quad a_{23} = 0$$

$$f(1, 1, 1) = a_{123} \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_0 = 1, \quad a_{123} = 1$$

Таким образом, полином Жегалкина для функции f имеет вид $f = x_1x_2x_3 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2 \oplus 1$.

Функция f *линейная*, если f представима в виде полинома Жегалкина первой степени, т.е. $f(x_1, \dots, x_n) = a_1x_1 \oplus a_2x_2 \oplus \dots \oplus a_nx_n \oplus a_0$. Множество всех линейных функций обозначают через L . Логическая функция из предыдущего примера – нелинейная.

Множества функций T_0, T_1, S, M, L называются *классами Поста*.

Множество булевых функций F называют *замкнутым*, если любая подстановка (сложная функция) функций из F также принадлежит F .

Теорема. *Каждый класс Поста замкнут.*

Критерий Поста. *Множество булевых функций полно тогда и только тогда, когда оно не содержится целиком ни в одном из классов Поста.*

Пример. Проверить на полноту множество булевых функций $F = \{\sim, \vee, 0\}$.

Для исследования используют критериальную таблицу. Строки таблицы соответствуют функциям исследуемого множества, а столбцы – классам Поста.

Таблицы истинности функций из системы приведены ниже, по таблицам истинности можно определить принадлежность к классам T_0, T_1, S, M .

x	y	$x \sim y$	$x \vee y$	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0
1	1	1	1	0

Линейность определяется по виду многочленов Жегалкина для функций.

Найдем эти многочлены, используется равносильность $\neg x \equiv x \oplus 1$

$$x \vee y \equiv \neg(\neg x \& \neg y) \equiv (x \oplus 1) \& (y \oplus 1) \oplus 1 \equiv x \& y \oplus x \oplus y \oplus 1$$

$$x \sim y \equiv (\neg x \vee y) \& (x \vee \neg y) = ((x \oplus 1) \& y \oplus (x \oplus 1) \oplus y \oplus 1) \& (x \& (y \oplus 1) \oplus x \oplus (y \oplus 1) \oplus 1) \equiv x \oplus y \oplus 1$$

Если функция принадлежит классу Поста, то в таблицу ставится +, иначе –.

	T_0	T_1	S	M	L
$\sim$	-	+	-	-	+
$\vee$	+	+	-	+	-
0	+	-	-	+	+

По критерию Поста полная система функций не содержится ни в одном из классов Поста, т.е. для полной системы в каждом столбце критериальной таб-

лицы должен быть хотя бы один « $\leftrightarrow$ ». Таким образом, система функций $F = \{\sim, \vee, 0\}$ полная.

Выразим теперь константы, отрицание, конъюнкцию через функции системы $F = \{\sim, \vee, 0\}$. Константу 1 можно получить из эквиваленции, которая не сохраняет 0, но сохраняет 1, отождествлением переменных, т.е. $1 = x \sim x$. Отрицание можно получить из немонотонной функции ($\sim$) и констант, $\neg x = 0 \sim x$. Конъюнкция может быть получена из нелинейной функции (дизъюнкция), отрицания и констант, поскольку

$$x \& y = \neg(\neg x \vee \neg y) = (0 \sim (\neg x \vee \neg y)) = (0 \sim ((0 \sim x) \vee (0 \sim y))) .$$

Ответ: система функций $F = \{\sim, \vee, 0\}$ полная,

$$1 = x \sim x, \neg x = 0 \sim x, x \& y = (0 \sim ((0 \sim x) \vee (0 \sim y)))$$

Задачи 2–5

2. Записать рассуждение в логической символической форме, обосновать выбор логических связей.
3. Проверить правильность рассуждения методом Куайна.
4. Проверить правильность рассуждения методом редукции.
5. Проверить правильность рассуждения методом резолюций.

Варианты

- Вариант 1. Если бы он бегал по утрам, то у него было бы крепкое здоровье. Он бежит только в том случае, если на улице тепло. Сегодня было тепло, но он заболел. Значит, сегодня он не бежит.
- Вариант 2. Если у футболиста хорошее настроение, то он пинает мяч. Если футболист устал, то он не пинает мяч. У футболиста хорошее настроение, если он не устал. Значит, у футболиста хорошее настроение.
- Вариант 3. Если футболист не устал, то он пинает мяч. У футболиста хорошее настроение тогда и только тогда, когда футболист не устал. Футболист пинает мяч. Значит, у него хорошее настроение.
- Вариант 4. Франция выйдет на чемпионат мира по футболу, только если Германия не попадет на чемпионат мира. Бразилия выйдет на чемпионат мира, если Франция не попадет на чемпионат мира. Бразилия не попала на чемпионат мира. Значит, Германия не выйдет на чемпионат мира по футболу.
- Вариант 5. Когда у меня есть свободное время, я читаю книги. Если книга попадает скучная, я засыпаю. Сегодня у меня есть свободное время и мне посоветовали интересную книгу. Значит, сегодня я не сплю всю ночь.

- Вариант 6. Если ослик Иа пойдет гулять, то он обязательно потеряет хвост. Ослик Иа пойдет гулять, если Вини-Пух пойдет с ним. Вини-Пух решил остаться дома. Значит, ослик Иа не потеряет свой хвост.
- Вариант 7. Если есть масло, то я могу испечь капкейки. Я делаю чизкейк тогда и только тогда, когда дома есть творог. Дома всегда есть либо масло, либо творог. Значит, я не сделаю капкейки.
- Вариант 8. Хомяк добрый, если хозяин его вкусно кормит. Хомяка вкусно кормят, только если у хозяина есть деньги. У хозяина нет финансовых проблем. Значит, хомяк добрый.
- Вариант 9. Если Марии понравятся цветы, она купит семена и посадит цветы. Если погода будет плохой, Мария не станет сеять цветы. Цветы не посажены. Значит, либо Марии не понравились цветы, либо на улице была холодная погода.
- Вариант 10. Если герой помогает жителям городка, то герой положительный. Когда герой спасает принцессу, он не может помогать жителям городка. Герой отправился спасать принцессу. Значит, герой не положительный.
- Вариант 11. Если бы на стадо не напал волк, овечка Долли не сбежала бы. Если бы пастух был внимательным, на стадо не напал бы волк. Пастух был не внимательным и овечка Долли сбежала. Значит, на стадо напал волк.
- Вариант 12. Роутер раздает интернет, когда роутер включен в розетку и отсутствует задолженность по оплате трафика. Интернет не ловится, хотя роутер включен в розетку. Значит, интернет вовремя не оплачен.
- Вариант 13. Если щегол поет, то либо на улице солнечно, либо щегол сытый. На улице пасмурно и в кормушке нет корма. Значит, щегол молчит.
- Вариант 14. Если человек дальтоник, то ему запрещается водить автомобиль и он не может рисовать цветные картинки. Человек рисует цветные картинки, но ему запрещено водить автомобиль. Значит, у человека нет дальтонизма.
- Вариант 15. Если астероид столкнется с Землей, то произойдет катастрофа. Если произойдет катастрофа, то тираннозавры погибнут. Тираннозавры вымерли. Значит, произошло столкновение астероида с Землей.
- Вариант 16. Страус осознает, что ему грозит опасность, если видит хищное животное. Если страус осознает опасность, он прячет голову в песок. Страус не спрятал голову в песок. Значит, страус не видит хищное животное.
- Вариант 17. Если подозреваемый совершил кражу, то либо кража была тщательно подготовлена, либо имелся соучастник. Если бы кража была тщательно подготовлена, то был бы соучастник. Значит, подозреваемый не виновен в краже.
- Вариант 18. Если бы он хорошо знал английский язык или хотя бы она говорила помедленнее, то он бы ее понял. Но он ее не понял. Значит, она, как всегда, говорила слишком быстро.

- Вариант 19. Незнание правил дорожного движения не освобождает от ответственности в случае их несоблюдения. При нарушении правил водитель несет ответственность. Следовательно, знать правила нужно.
- Вариант 20. Если у меня хватит времени прочитать книгу, то я пойду погулять или встречу с друзьями. С друзьями я встречаюсь во время прогулки. Значит, я встречу с друзьями.
- Вариант 21. Если подозреваемый совершил кражу, то либо кража была тщательно подготовлена, либо имелся соучастник. Если бы кража была тщательно подготовлена, то был бы соучастник. Значит, подозреваемый виновен в краже.
- Вариант 22. Муравей поднимет соломинку, если ее вес не превышает собственный вес муравья более, чем в 10 раз. Муравей не будет поднимать соломинку, если она ему не нужна. Муравей не стал поднимать соломинку. Значит, соломинка слишком тяжелая.
- Вариант 23. Если человек обедает в кафе быстрого питания, то он голоден и куда-то торопится. Человек не обедает в кафе быстрого питания, хотя и очень торопится. Значит, он голоден.
- Вариант 24. Мне обязательно нужно сходить в магазин. Я хожу в магазин только тогда, когда я свободен. Когда я свободен, я предпочитаю отдыхать. Значит, я не пойду в магазин.
- Вариант 25. Если бы у нее было много денег, то она бы ездила в институт на такси и тогда бы никогда не опаздывала. У нее денег немного. Поэтому она постоянно опаздывает.

Необходимый теоретический материал

Под *высказыванием* принято понимать языковое предложение, о котором имеет смысл говорить, что оно истинно или ложно. Не всякое предложение может быть истинным или ложным, а значит, быть высказыванием. Так, вопросительное или повелительное предложение не является высказыванием.

В *логике высказываний* интересуются не содержанием, а истинностью или ложностью высказываний. Истинностное значение — истина или ложь — будем обозначать И и Л соответственно. Для соединения высказываний в более сложные высказывания используют следующие *логические операции (связки)*.

Отрицанием высказывания A называется высказывание, истинное тогда и только тогда, когда высказывание A ложно. Обозначается $\neg A$. В естественном языке отрицание $\neg A$ соответствует следующим конструкциям:

- не A (или то, что получится в результате вставки частицы "не" перед глаголом — основным или вспомогательным);
- A не имеет места;
- A не верно.

Конъюнкцией двух высказываний A и B называется высказывание, истинное тогда и только тогда, когда истинны оба высказывания. Обозначается $A \& B$. В естественном языке конъюнкция $A \& B$ соответствует следующим конструкциям:

- A и B ;
- не только A , но и B ;
- B , хотя и A ;
- B , несмотря на A ;
- как A , так и B ;
- A вместе с B ;
- A , в то время как B .

Дизъюнкцией двух высказываний A и B называется высказывание, ложное тогда и только тогда, когда оба высказывания ложны. Обозначается $A \vee B$. В естественном языке дизъюнкция $A \vee B$ соответствует следующим конструкциям:

- A или B , или оба;
- A или B ;
- A , если не B ;
- A и/или B (в юридических текстах).

Импликацией двух высказываний A и B называется высказывание, ложное тогда и только тогда, когда A истинно, а B ложно. Обозначается $A \rightarrow B$. В естественном языке импликация $A \rightarrow B$ соответствует следующим конструкциям:

- если A , то B ;
- коль скоро A , то B ;
- в случае A имеет место B ;
- для B достаточно A ;
- для A необходимо B ;
- B , если A ;
- A влечет B .

Эквиваленцией двух высказываний A и B называется высказывание, истинное тогда и только тогда, когда истинностные значения A и B совпадают. Обозначается $A \sim B$. В естественном языке эквиваленция $A \sim B$ соответствует следующим конструкциям:

- A , если и только если B ;
- если A , то B , и обратно;
- A , если B , и B , если A ;
- для A необходимо и достаточно B ;
- A равносильно B ;
- A тогда и только тогда, когда B .

Пример. Записать рассуждение в логической символике.

Я оплачу ремонт телевизора, только если после ремонта телевизор будет работать.

Телевизор не работает после ремонта. Значит, я не оплачиваю ремонт телевизора.

Данное рассуждение состоит из двух гипотез и вывода, которые необходимо записать в логической символике. Для этого выделим отдельные высказывания и обозначим их высказывательными переменными:

$A = \text{“Я оплачиваю ремонт”}$

$B = \text{“Телевизор работает”}$

В первой гипотезе используется конструкция «только если ...», что соответствует логической связке эквиваленция. Негативное высказывание соответствует логической связке отрицание. Поэтому рассуждение в логической символике выглядит так: $B \sim A, \neg B \vdash \neg A$.

Ответ: $B \sim A, \neg B \vdash \neg A$.

Следующая теорема, доказанная Постом, устанавливает взаимосвязь между тождественной истинностью и доказуемостью формулы.

Теорема (Пост, 1921). *Формула F является теоремой исчисления высказываний тогда и только тогда, когда F – тавтология.*

Таким образом, для проверки выводимости формулы в исчислении высказываний достаточно проверить тождественную истинность этой формулы, что можно сделать за конечное число шагов.

Метод Куайна представляет собой модификацию тривиального метода. Пусть $\langle X_1, X_2, \dots, X_k \rangle$ – множество высказывательных переменных в формуле F . Возьмем первую переменную X_1 и придадим ей, например, истинное значение (или ложное). Подставим это значение в формулу F и выполним логические операции, которые могут возникнуть при такой подстановке. После выполнения вычислений получим формулу F' с меньшим количеством переменных и далее опять применяем к ней описанную процедуру. Если на каком-то шаге получена формула, которая является тавтологией или противоречием независимо от значений высказывательных переменных, входящих в эту формулу, то алгоритм на этом шаге можно остановить. Таким образом, алгоритм Куайна приводит к рассмотрению меньшего количества интерпретаций, чем тривиальный алгоритм.

Пример. Проверить выводимость $B \sim A, \neg B \vdash \neg A$ методом Куайна.

Сначала применим теорему дедукции к данной выводимости. По теореме дедукции можно проверять выводимость

$$\vdash (B \sim A) \rightarrow (\neg B \rightarrow \neg A)$$

Пусть переменная B принимает ложное значение или $B = 0$. Тогда $(0 \sim A) \rightarrow (\neg 0 \rightarrow \neg A)$. Используя равносильные преобразования и таблицы истинности логических связок, $(0 \sim A) \rightarrow (\neg 0 \rightarrow \neg A) \equiv \neg A \rightarrow \neg A$. Можно продолжить, полагая $A = 0$. Тогда $\neg 0 \rightarrow \neg 0 \equiv 1$. При $A = 1$ $\neg 1 \rightarrow \neg 1 \equiv 1$.

Пусть теперь переменная B принимает истинное значение или $B = 1$. Тогда $(1 \sim A) \rightarrow (\neg 1 \rightarrow \neg A) \equiv 1$ при любом значении переменной A .

Таким образом, формула $(B \sim A) \rightarrow (\neg B \rightarrow \neg A)$ является тавтологией, а значит, выводима в исчислении высказываний.

Ответ: выводимость верная.

Рассмотрим теперь *метод редукции* распознавания тождественно истинных формул в исчислении высказываний. Он особенно удобен, когда в записи формул встречается много импликаций. Суть метода заключается в следующем. Пусть формула F имеет вид импликации, например, $F = A \rightarrow B$. Допустим, что в некоторой интерпретации формула F принимает ложное значение. Тогда в соответствии с таблицей истинности для импликации имеем $A = 1$, $B = 0$. Таким образом, проверка формулы F сводится к проверке формул A и B . После этого данный процесс применяется к формулам A и B и продолжается до тех пор, пока не будет получено противоречие с начальным предположением о ложности формулы. В этом случае доказано, что формула является тавтологией. В противном случае будет определен набор значений переменных, на котором формула принимает ложное значение, а значит, не является тавтологией.

Пример. Проверить выводимость формулы $B \sim A$, $\neg B \vdash \neg A$ методом редукции. По теореме дедукции можно проверять выводимость

$$\vdash (B \sim A) \rightarrow (\neg B \rightarrow \neg A)$$

Пусть в некоторой интерпретации формула имеет ложное значение. Тогда $B \sim A = 1$, $\neg B \rightarrow \neg A = 0$. Применим теперь редукцию к формуле $\neg B \rightarrow \neg A$. Если она в некоторой интерпретации имеет ложное значение, то $\neg B = 1$, $\neg A = 0$ или $B = 0$, $A = 1$. Подставляя полученные значения переменных в соотношение $B \sim A = 1$, получим противоречие ($0 \sim 1 = 0$) с первоначальным предположением о ложности исходной формулы. Таким образом, исходная формула не может быть ложной ни при какой интерпретации, т.е. формула является тавтологией и выводима в исчислении высказываний.

Ответ: выводимость верная.

Наиболее известный классический алгоритм проверки выводимости формул в ИВ называется *методом резолюций*. В основу метода резолюций положена следующая теорема (доказательство от противного).

Теорема (доказательство от противного). Если $\Gamma, \neg A \vdash_{\mathcal{L}} B$, где B – любое противоречие, то $\Gamma \vdash_{\mathcal{L}} A$.

В качестве формулы B при доказательстве от противного по методу резолюций принято использовать пустую формулу, которую будем обозначать $\emptyset$. Пустая формула не имеет никакого значения и не является истинной ни при какой интерпретации и по определению является противоречием. Метод резолюций работает с особой стандартной формой формул, которые называются *предложениями*. Предложение – это дизъюнкция литералов (переменных или

их отрицаний). Любая формула исчисления высказываний может быть преобразована в множество предложений следующим образом. Сначала формула приводится к КНФ, а затем конъюнкция дизъюнкций литералов разбивается на множество предложений.

Пример. Найдем множество предложений, соответствующих формуле

$$A \rightarrow \neg(B \rightarrow A).$$

Приведем формулу к КНФ.

$$A \rightarrow \neg(B \rightarrow A) \equiv \neg A \vee \neg(B \rightarrow A) \equiv \neg A \vee (B \& \neg A) \equiv (\neg A \vee B) \& \neg A.$$

Таким образом, множество предложений состоит из двух формул $\neg A \vee B$, $\neg A$.

Ответ: множество предложений $\{\neg A \vee B, \neg A\}$.

Рассмотрим теперь *правило резолюции*, т.е. правило вывода, которое используется в методе резолюций. Пусть A и B – два предложения, которые имеют следующий вид $A = P \vee A_1$ и $B = \neg P \vee B_1$, где P – переменная, а A_1 , B_1 – любые предложения (в частности, может быть пустые или состоящие только из одного литерала). Правило вывода $\frac{A, B}{A_1 \vee B_1} R$ называется *правилом резолюции*.

Предложения A и B называются *резольвируемыми* (или родительскими), предложение $A_1 \vee B_1$ – *резольвентой*, а формулы P , $\neg P$ – *контрарными* литералами.

Теорема. *Правило резолюции логично, т.е. резольвента является логическим следствием резольвируемых предложений.*

Метод резолюций для исчисления высказываний заключается в следующем. Пусть нужно установить выводимость $\Gamma \vdash_{\mathcal{L}} F$. Воспользуемся доказательством от противного и будем доказывать выводимость $\Gamma, \neg F \vdash_{\mathcal{L}} \emptyset$ с помощью метода резолюций. Для этого каждая формула множества гипотез Γ и формула $\neg F$ независимо преобразуются в множества предложений. В полученном совокупном множестве предложений S отыскиваются резольвируемые предложения, к ним применяется правило резолюции и полученная резольвента добавляется в множество предложений S . После чего процедура резольвирования повторяется. При этом возможны два случая:

- Среди предложений множества S нет резольвируемых или нельзя получить новую резольвенту. Это означает, что теорема опровергнута, т.е. формула F не выводима из множества формул Γ .
- В результате очередного применения правила резолюции получено пустое предложение. Это означает, что теорема доказана, т.е. $\Gamma \vdash_{\mathcal{L}} F$.

Пример. Докажем методом резолюций выводимость $B \sim A, \neg B \vdash \neg A$

Преобразуем во множество предложений отдельно гипотезы и отрицание вывода.

$$B \sim A \equiv (\neg B \vee A) \& (\neg A \vee B) \Rightarrow \{\neg B \vee A, \neg A \vee B\}$$

$$\neg B \Rightarrow \{\neg B\}$$

$$\neg(\neg A) \Rightarrow \{A\}$$

Таким образом, общее множество предложений $S = \{\neg B \vee A, \neg A \vee B, \neg B, A\}$. Далее проводим резольвирование полученных предложений, применяя правило резолюции

1	$\neg B \vee A$	
2	$\neg A \vee B$	
3	$\neg B$	
4	A	
5	$\neg A$	ПР из 2 и 3
6	$\emptyset$	ПР из 4 и 5

Поскольку получено пустое предложение, то исходная формула выводима в исчислении высказываний.

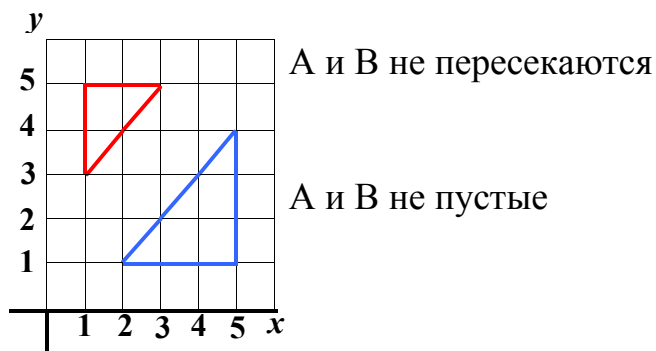
Ответ: выводимость верная.

Задача 6

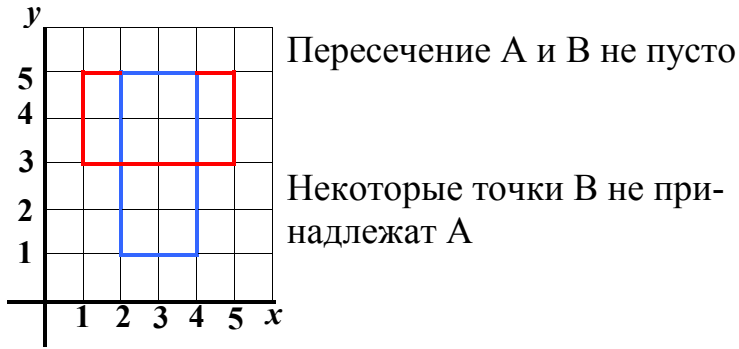
На координатной плоскости даны множества А и В (А ограничено красным цветом, В – синим). Описать предикаты, определенные на координатной плоскости $P_A(z)$ = "точка z принадлежит множеству А", $P_B(z)$ = "точка z принадлежит множеству В" (границы включены в множества, можно использовать логические операции и сравнение $\leq$). С использованием полученных предикатов записать в виде формул логики предикатов высказывания о множествах А и В.

Варианты

Вариант 1



Вариант 2



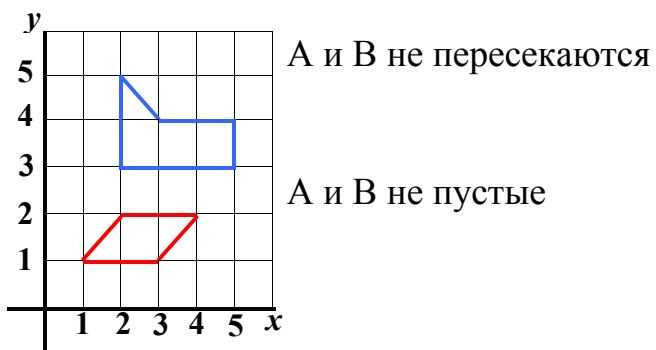
Вариант 3



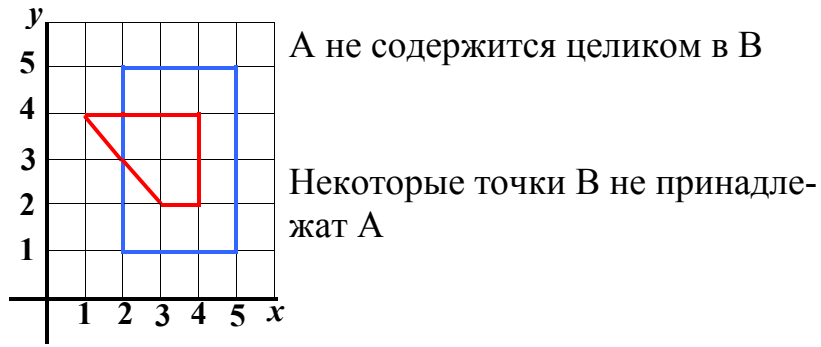
Вариант 4



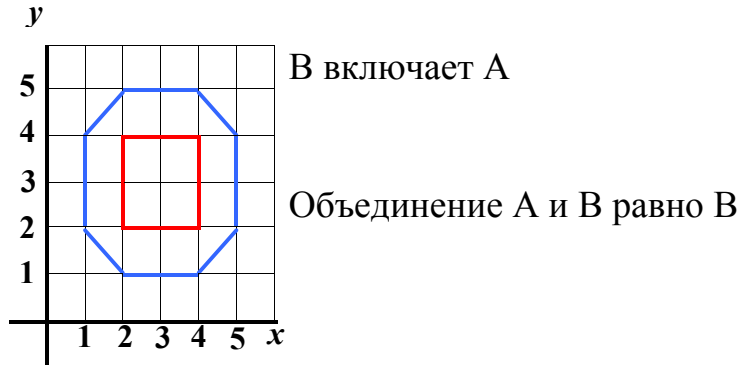
Вариант 5



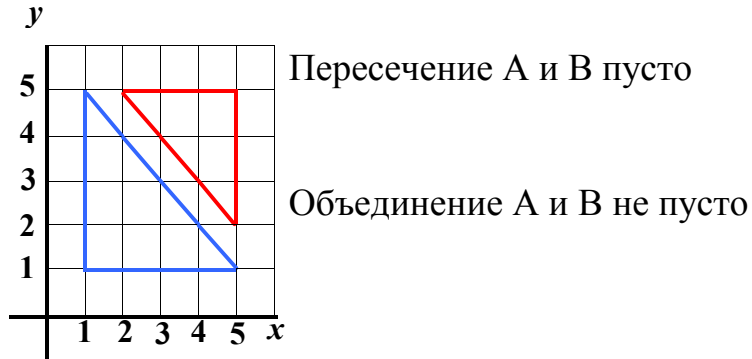
Вариант 6



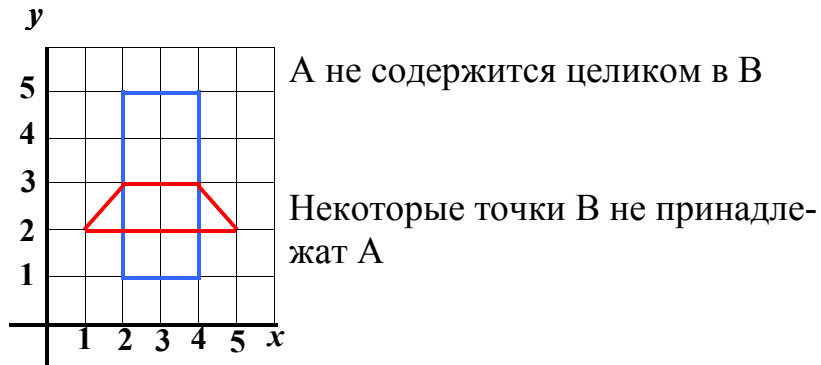
Вариант 7



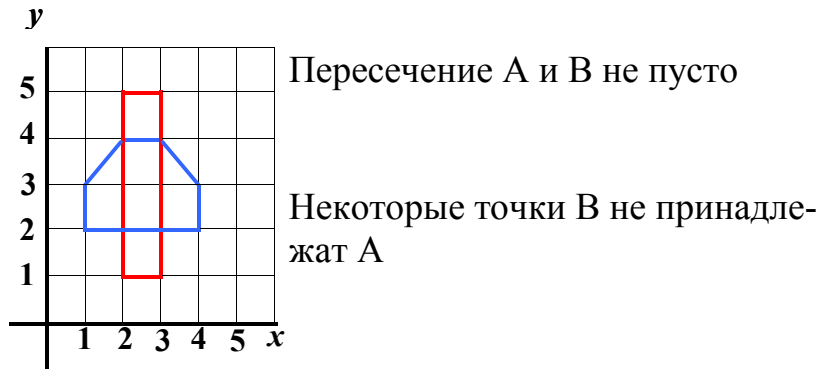
Вариант 8



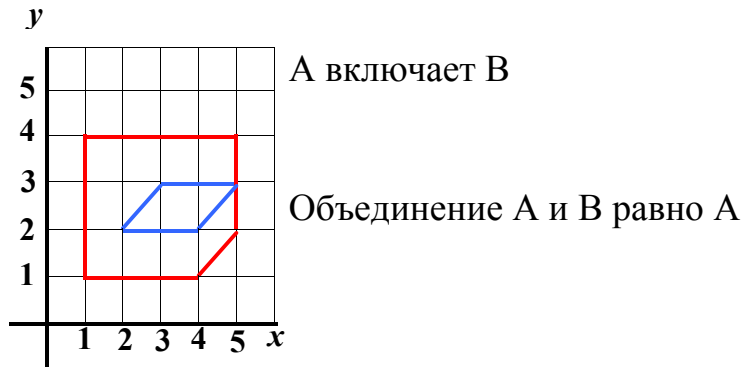
Вариант 9



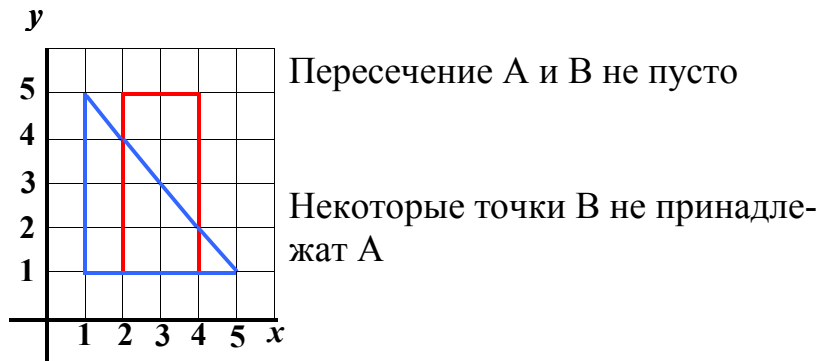
Вариант 10



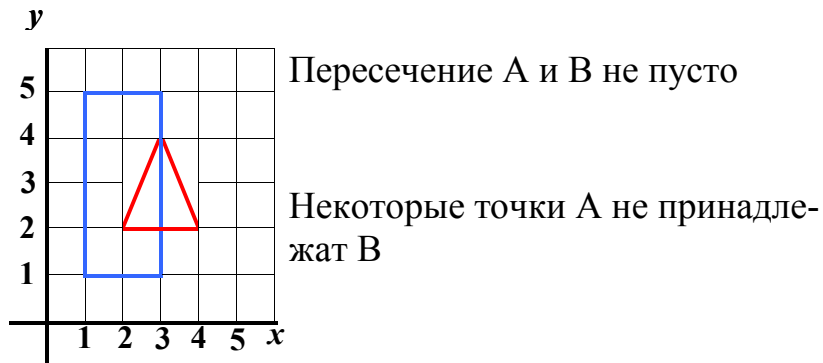
Вариант 11



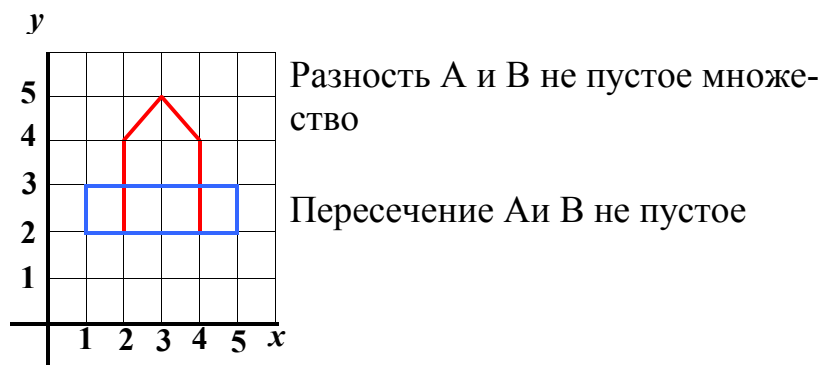
Вариант 12



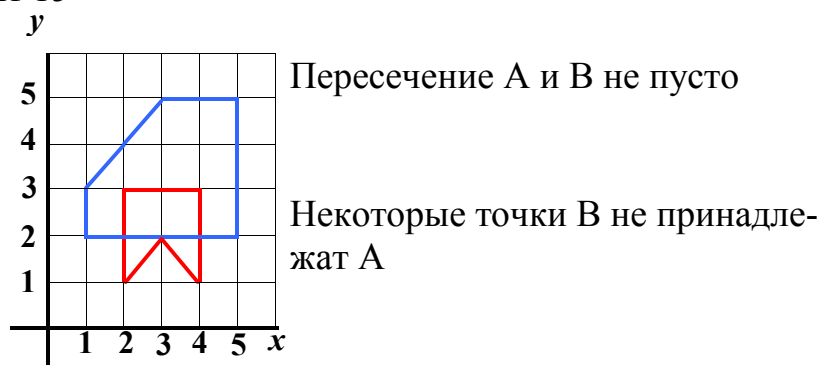
Вариант 13



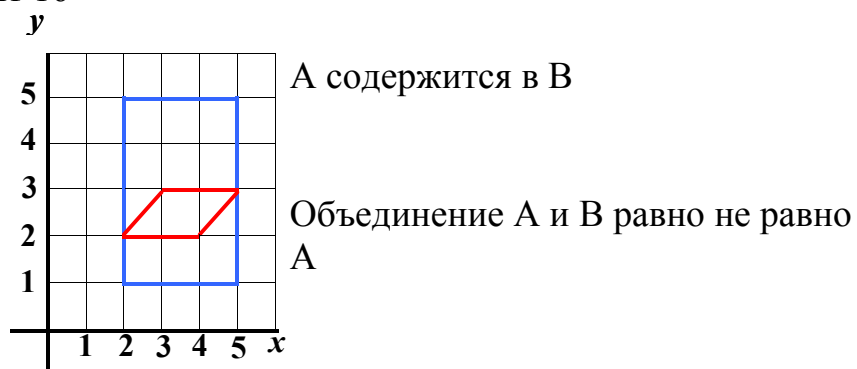
Вариант 14



Вариант 15



Вариант 16



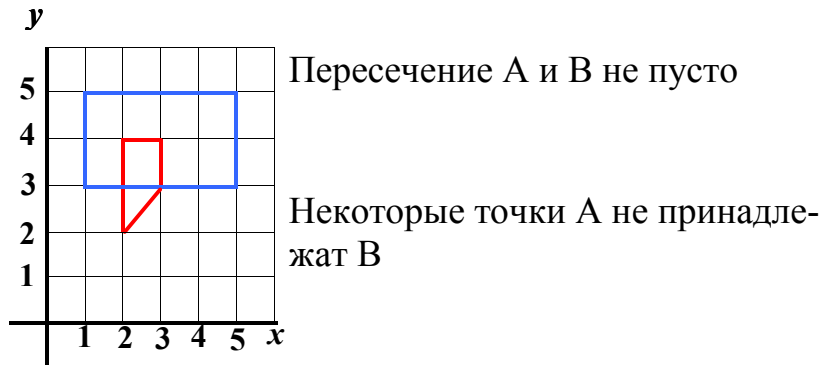
Вариант 17



Вариант 18



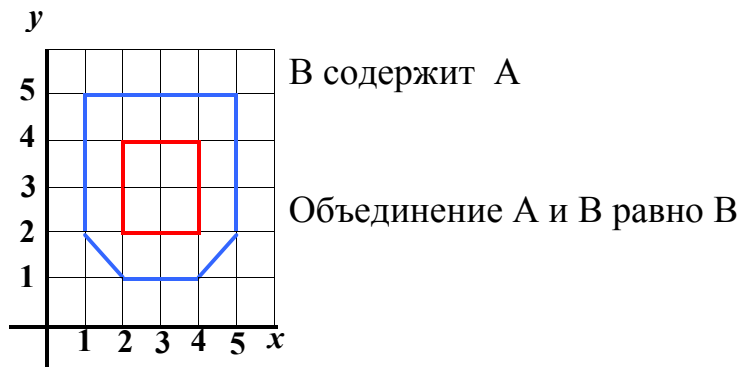
Вариант 19



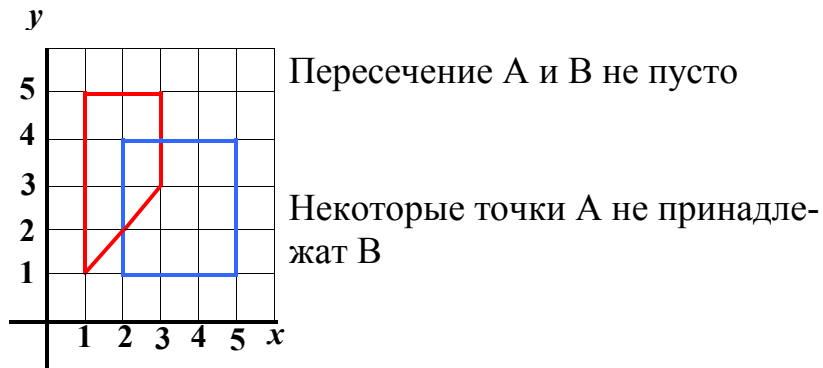
Вариант 20



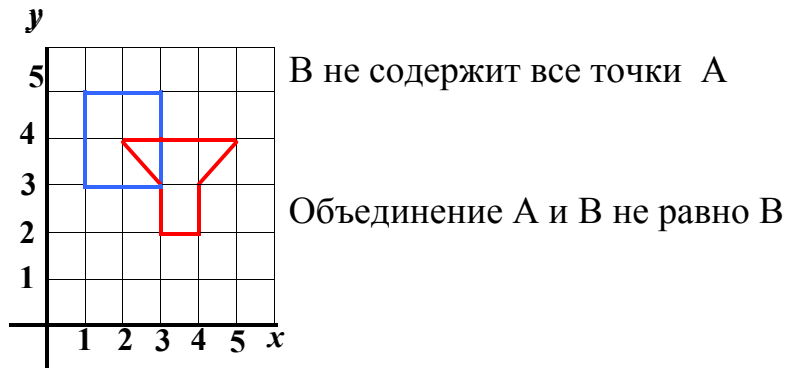
Вариант 21



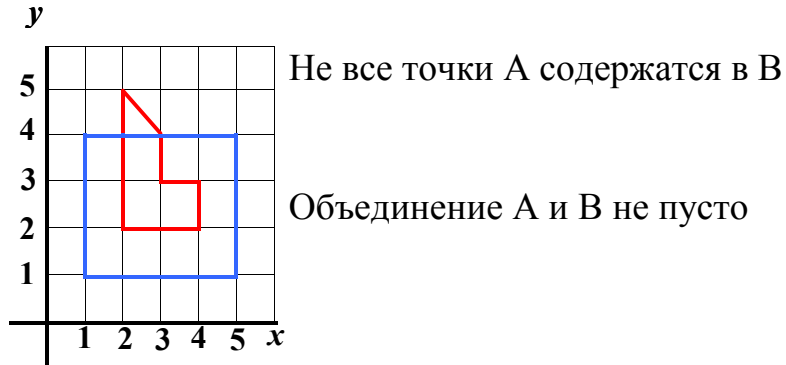
Вариант 22



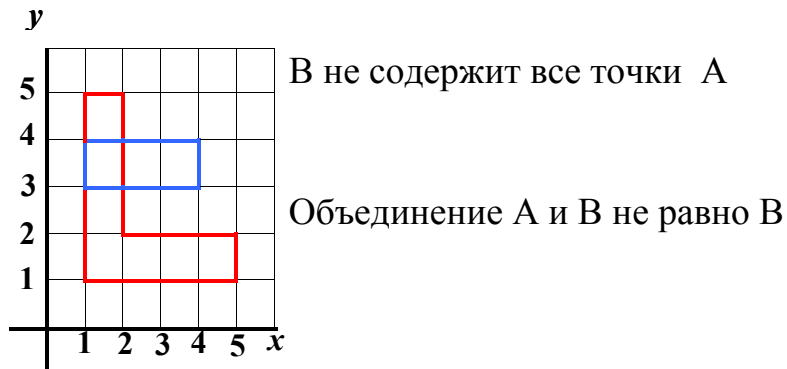
Вариант 23



Вариант 24



Вариант 25



Необходимый теоретический материал

Каждое высказывание представляет собой некоторое суждение о предмете высказывания (субъекте) или взаимосвязи нескольких субъектов. Предметы (субъекты), о которых делается суждение, могут быть самой различной природы. Множество субъектов, о которых делаются высказывания, называется *предметной областью* Ω . Для обозначения субъектов будем использовать *предметные переменные*. *Предикат* – это языковое выражение, обозначающее какое-то свойство субъекта или отношение между субъектами. В современной логике предикатами называются функции, значениями которых служат высказывания. *Предикатом мощности n (n -местным предикатом)* $P(x_1, \dots, x_i, \dots, x_n)$, $x_i \in \Omega$ определенным на предметной области Ω , называют отображение набора предметных переменных $x_1, \dots, x_n$ во множество высказываний.

Пример. Приведем примеры различных предметных областей и предикатов.

- $Q = \langle \langle 2+3=5 \rangle \rangle$ – нульместный предикат, определенный на множестве натуральных чисел $\mathbb{N}$.
- $P(x) = \langle \langle \text{Натуральное число } x \text{ четное} \rangle \rangle$ – одноместный предикат, определенный на множестве натуральных чисел $\mathbb{N}$.
- $D(x_1, x_2) = \langle \langle \text{Натуральное число } x_1 \text{ делится (без остатка) на натуральное число } x_2 \rangle \rangle$ – двуместный предикат, определенный на множестве пар натуральных чисел $\mathbb{N} \times \mathbb{N}$.
- $M(x) = \langle \langle x - \text{мужчина} \rangle \rangle$, одноместный предикат, определенный на множестве всех людей.

Придавая конкретные значения из предметной области предметным переменным, участвующим в предикатах, можно получить высказывания, которые принимают логическое значение истина или ложь в зависимости от значения переменных.

Поскольку предикаты представляют собой отображения со значениями во множестве высказываний, где введены логические операции, то эти операции естественным образом определяются и для предикатов. Пусть P, Q – предикаты мощности n , определенные на предметной области Ω . Тогда логические операции для предикатов вводятся следующим образом:

$$\begin{aligned}(\neg P)(x_1, \dots, x_n) &:= \neg(P(x_1, \dots, x_n)) \\(P \vee Q)(x_1, \dots, x_n) &:= (P(x_1, \dots, x_n) \vee Q(x_1, \dots, x_n)) \\(P \& Q)(x_1, \dots, x_n) &:= (P(x_1, \dots, x_n) \& Q(x_1, \dots, x_n)) \\(P \rightarrow Q)(x_1, \dots, x_n) &:= (P(x_1, \dots, x_n) \rightarrow Q(x_1, \dots, x_n))\end{aligned}$$

Пример. Пусть на множестве натуральных чисел $\mathbb{N}$ определены два предиката:

$$\begin{aligned}P(x) &= \langle \langle \text{Натуральное число } x \text{ делится на } 2 \rangle \rangle, \\Q(x) &= \langle \langle \text{Натуральное число } x \text{ делится на } 3 \rangle \rangle.\end{aligned}$$

Тогда

$(P \vee Q)(x) := (P(x) \vee Q(x)) =$ «Натуральное число x делится на 2 или на 3»,

$(P \& Q)(x) := (P(x) \& Q(x)) =$ «Натуральное число x делится на 6».

Таким образом,

$$(P \vee Q)(5) := (P(5) \vee Q(5)) = Л \vee Л = Л,$$

$$(P \& Q)(120) := (P(120) \& Q(120)) = И \& И = И.$$

Существуют такие виды высказываний, которые нельзя записать в виде формулы исчисления высказываний. Например,

- Все дети должны смеяться.
- Все люди смертны.
- Никто не забыт, и ничто не забыто.

Корректность таких высказываний определяется не только истинностью соответствующих логических связей, но и пониманием таких слов, как «все», «всякий» и т.д. В логике предикатов наряду с операциями логики высказываний основную роль играют операции, называемые *кванторами*. Именно использование кванторов делает логику предикатов более богатой и гибкой по сравнению с логикой высказываний.

Дадим определение операции кванторов. Пусть Q – одноместный предикат, определенный на предметной области Ω . Предположим сначала, что множество Ω имеет конечное число элементов, т.е. $\Omega = \{a_1, \dots, a_m\}$. Тогда высказывание $A =$ «Все элементы Ω обладают свойством Q » можно записать в виде конечной конъюнкции $A = Q(a_1) \& Q(a_2) \& \dots \& Q(a_m)$ и высказывание A будет истинно тогда и только тогда, когда предикат Q принимает истинное значение для любого $a_i \in \Omega$. Также можно рассмотреть высказывание $B =$ «Хотя бы один элемент Ω обладает свойством Q » и записать его в виде конечной дизъюнкции предикатов – $B = Q(a_1) \vee Q(a_2) \vee \dots \vee Q(a_m)$. Очевидно, что высказывание B будет истинным, если хотя бы для одного элемента из предметной области предикат принимает истинное значение. Однако в случае бесконечного множества Ω нельзя записать бесконечную конъюнкцию или дизъюнкцию предикатов и в этом случае используют операцию *квантор всеобщности* или *квантор существования*.

Пусть имеется n -местный предикат $P(x_1, \dots, x_n)$, определенный на предметной области Ω . Предикату P поставим в соответствие пару $(n-1)$ -местных предикатов:

$\forall x_i P(x_1, \dots, x_i, \dots, x_n)$ («для всякого $x_i P(x_1, \dots, x_i, \dots, x_n)$ ») и

$\exists x_i P(x_1, \dots, x_i, \dots, x_n)$ (существует x_i , для которого $P(x_1, \dots, x_i, \dots, x_n)$).

Эти $(n-1)$ -местные предикаты переменных $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ получены из исходного предиката навешиванием *квантора всеобщности* и *квантора существования*. При этом переменная x_i связана квантором всеобщности или существования.

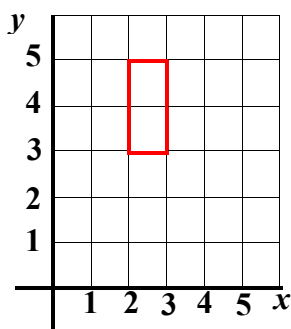
В естественном языке предикату $\forall x P(x)$ соответствуют фразы:

- Для любого x (имеет место) P .
- $P(x)$ при произвольном x .
- Для всех x (верно) $P(x)$.
- $P(x)$, каково бы ни было x .
- Для каждого x (верно) $P(x)$.
- Всегда имеет место $P(x)$.
- Каждый обладает свойством P .
- Свойство P присуще всем.
- Всё удовлетворяет P .
- Любой объект является P .
- Всякая вещь обладает свойством P .

В естественном языке предикату $\exists x P(x)$ соответствуют фразы:

- Для некоторых x (имеет место) $P(x)$.
- Для подходящего x (верно) $P(x)$.
- Существует x , для которого (такой, что) $P(x)$.
- Имеется x , для которого (такой, что) $P(x)$.
- Найдется x , для которого (такой, что) $P(x)$.
- У некоторых вещей есть признак P .
- Хотя бы для одного x (верно) $P(x)$.
- Кто-нибудь относится к (есть) P .
- По крайней мере, один объект есть P .

Пример. На координатной плоскости дано множество A (A ограничено красным цветом). Описать предикат, определенный на координатной плоскости, $P_A(z)$ = "точка z принадлежит множеству A ". (Границы фигуры включены в множество, можно использовать логические операции и сравнение $\leq$).



Для описания предиката необходимо определить предметную область Ω . В данном случае предметной областью будет координатная плоскость, которая содержит всевозможные точки $z = (x, y)$, $x, y \in R$. Определим для $z = (x, y) \in \Omega$ вспомогательный предикат $C(x, y)$ = " $x \leq y$ ".

Множеству A принадлежат такие точки $z = (x, y) \in \Omega$, что $2 \leq x \leq 3$ и $3 \leq y \leq 5$. Или с использованием обозначений и логических операций $P_A(z) = C(2, x) \& C(x, 3) \& C(3, y) \& C(y, 5)$.

Задача 7

Проверить общезначимость формулы методом резолюций.

Варианты

- Вариант 1. $(\exists x \forall y A(x, y)) \& (\exists x \forall y B(x, y))$
- Вариант 2. $(\exists x \forall y A(x, y)) \rightarrow (\exists x \forall y B(x, y))$
- Вариант 3. $(\exists x \forall y A(x, y)) \vee (\exists x \forall y B(x, y))$
- Вариант 4. $(\exists x \forall y A(x, y)) \rightarrow (\forall x \exists y B(x, y))$
- Вариант 5. $(\exists x \forall y Q(x, y)) \rightarrow (\forall y \exists x Q(x, y))$
- Вариант 6. $(\forall x \exists y Q(x, y)) \rightarrow (\exists y \forall x Q(x, y))$
- Вариант 7. $(\forall x \exists y Q(x, y)) \rightarrow ((\exists y \forall x Q(x, y)) \vee R(x, y))$
- Вариант 8. $(\forall x \exists y Q(x, y)) \rightarrow ((\exists y \forall x P(x, y)) \rightarrow Q(x, y))$
- Вариант 9. $\forall x Q(x, y) \rightarrow (\exists y Q(x, y) \vee \exists x R(x, y))$
- Вариант 10. $(\forall x \exists y Q(x, y)) \rightarrow (\exists y P(x, y) \rightarrow \forall x Q(x, y))$
- Вариант 11. $(\exists x \forall y A(x, y)) \vee \forall y B(x, y)$
- Вариант 12. $(\exists x \forall y A(x, y)) \rightarrow \neg(\forall x \exists y B(x, y))$
- Вариант 13. $\neg(\exists x \forall y Q(x, y)) \rightarrow (\forall y \exists x Q(x, y))$
- Вариант 14. $(\forall x \exists y Q(x, y)) \rightarrow \forall x Q(x, y)$
- Вариант 15. $\forall x Q(x, y) \rightarrow (\exists y Q(x, y) \vee \exists x R(x, y))$
- Вариант 16. $\exists y Q(x, y) \rightarrow (\exists y P(x, y) \rightarrow \forall x Q(x, y))$
- Вариант 17. $(\forall y A(x, y)) \vee (\exists x \forall y B(x, y))$
- Вариант 18. $(\exists x \forall y A(x, y)) \rightarrow \neg(\forall x B(x, y))$
- Вариант 19. $\neg(\exists x \forall y Q(x, y) \rightarrow \forall y \exists x Q(x, y))$
- Вариант 20. $(\forall x Q(x, y) \rightarrow \exists y Q(x, y)) \vee \exists x R(x, y)$
- Вариант 21. $(\exists y \exists x Q(x, y) \rightarrow \exists y P(x, y)) \rightarrow \forall x Q(x, y)$
- Вариант 22. $\forall y A(x, y) \vee \exists x \forall y B(x, y)$
- Вариант 23. $\neg(\exists x A(x, y)) \rightarrow (\forall x \exists y B(x, y))$
- Вариант 24. $\exists x \forall y A(x, y) \& B(x, y)$
- Вариант 25. $(\exists x \forall y A(x, y)) \rightarrow (\exists x \exists y B(x, y))$

Необходимый теоретический материал

Формула F находится в *предваренной форме*, если она имеет следующий вид $F = Q_1 x_1 Q_2 x_2 \dots Q_n x_n G(x_1, \dots, x_n)$, где Q_i – квантор всеобщности или существования или отсутствия квантора по переменной x_i , а G – бескванторная формула. Приставка $Q_1 x_1 Q_2 x_2 \dots Q_n x_n$ называется *префиксом*, а формула G – *матрицей*.

Пример. Формула $\exists x \neg P(x) \vee R(z)$ не является формулой в предваренной форме, поскольку квантор существования действует только на $\neg P(x)$, но не на $R(z)$.

Теорема. Для любой формулы исчисления предикатов существует логически эквивалентная ей формула в предваренной форме.

Доказательство. Для того чтобы привести формулу к предваренной форме, используют следующие шаги:

- Исключают импликации.
- Для внесения $\neg$ внутрь скобок применяют законы де Моргана и закон двойного отрицания.
- Для переноса кванторов применяют законы дистрибутивности, законы ограничения действия кванторов и равносильности с переименованием кванторов (Q_1, Q_2 – кванторы существования или всеобщности)

$$\begin{aligned}(Q_1 x)P(x) \& (Q_2 x)R(x) &\equiv (Q_1 x)(Q_2 z)(P(x) \& R(z)) \\ (Q_1 x)P(x) \vee (Q_2 x)R(x) &\equiv (Q_1 x)(Q_2 z)(P(x) \vee R(z)).\end{aligned}$$

Пример. Рассмотрим построение предваренной формы для формулы $\forall x P(x) \rightarrow \forall x R(x)$.

$$\begin{aligned}\forall x P(x) \rightarrow \forall x R(x) &\equiv \neg(\forall x P(x)) \vee \forall x R(x) \equiv \exists x \neg P(x) \vee \forall x R(x) \equiv \\ &\equiv \exists x \neg P(x) \vee \forall z R(z) \equiv \exists x \forall z (\neg P(x) \vee R(z))\end{aligned}$$

Пример. Привести формулу к предваренной форме. $(\forall x \exists y Q(x, y)) \rightarrow ((\exists y \forall x P(x, y)) \rightarrow Q(x, y))$.

Для построения предварённой нормальной формы выполним следующие действия:

Шаг 1. Переименуем связанные переменные, свободные переменные x, y не меняют своих имен.

$$(\forall u \exists v Q(u, v)) \rightarrow ((\exists t \forall s P(s, t)) \rightarrow Q(x, y))$$

Шаг 2. Исключим всюду логические операции импликации:

$$(\forall u \exists v Q(u, v)) \rightarrow ((\exists t \forall s P(s, t)) \rightarrow Q(x, y)) = \neg \forall u \exists v Q(u, v) \vee \neg \exists t \forall s P(s, t) \vee Q(x, y)$$

Шаг 3. Продвинем отрицания до элементарной формулы по законам де Моргана:

$$\neg \forall u \exists v Q(u, v) \vee \neg \exists t \forall s P(s, t) \vee Q(x, y) = \exists u \forall v \neg Q(u, v) \vee \forall t \exists s \neg P(s, t) \vee Q(x, y)$$

Шаг 4. Вынесем кванторы влево, используя несколько раз законы ограничения действия кванторов.

$$\begin{aligned}\exists u \forall v \neg Q(u, v) \vee \forall t \exists s \neg P(s, t) \vee Q(x, y) &= \exists u (\forall v \neg Q(u, v) \vee \forall t \exists s \neg P(s, t) \vee Q(x, y)) = \\ \exists u (\forall v (\neg Q(u, v) \vee \forall t (\exists s \neg P(s, t) \vee Q(x, y)))) &= \exists u (\forall v (\neg Q(u, v) \vee \forall t (\exists s \neg P(s, t) \vee Q(x, y)))) = \\ \exists u \forall v (\neg Q(u, v) \vee \forall t (\exists s \neg P(s, t) \vee Q(x, y))) &= \exists u \forall v \forall t (\neg Q(u, v) \vee \exists s \neg P(s, t) \vee Q(x, y)) = \\ \exists u \forall v \forall t \exists s (\neg Q(u, v) \vee \neg P(s, t) \vee Q(x, y))\end{aligned}$$

Получена равносильная формула, в которой все кванторы вынесены в начало формулы и действуют на бескванторную часть формулы.

Значение формулы определено лишь тогда, когда задана какая-нибудь интерпретация входящих в нее символов. Под *интерпретацией (моделью)* будем понимать предметную область Ω с определенными на ней n -арными предикатами.

Формула F *выполнима в данной модели*, если существует набор $(a_1, a_2, \dots, a_n)$, $a_i \in \Omega$, значений свободных переменных $x_1, x_2, \dots, x_n$ формулы F такой, что $F(a_1, a_2, \dots, a_n)$ принимает истинное значение.

Формула F *выполнима (в логике предикатов)*, если существует модель, в которой F выполнима.

Формула F *истинна в данной модели*, если она принимает истинное значение на любом наборе $(a_1, a_2, \dots, a_n)$, $a_i \in \Omega$, значений своих свободных переменных $x_1, x_2, \dots, x_n$.

Формула F *общезначима или (тождественно истинна в логике предикатов)*, если она истинна в любой интерпретации.

Пример. На множестве целых чисел определен предикат $P(x, y) = "x > y"$. Тогда поскольку при любом значении свободной переменной y эта формула истинна, то формула $\exists x P(x, y)$ выполнима в этой модели и выполнима вообще. Кроме того, формула $\exists x P(x, y)$ истинна в указанной модели, но не является общезначимой. Так как существует модель (на множестве целых чисел предикат $P(x, y)$ всюду ложный), в которой формула $\exists x P(x, y)$ не является истинной.

Пример. Очевидно, что формула $\exists x (P(x, y) \vee \neg P(x, y))$ истинна в любой интерпретации при любом наборе значений переменных, т.е. эта формула общезначима.

Теорема. (Теорема Гёделя о полноте исчисления предикатов.) *Формула доказуема в исчислении предикатов тогда и только тогда, когда формула общезначима.*

Метод резолюций для проверки выводимости формул в логических исчислениях, предложенный в 1965 г. А. Робинсоном, используется во многих системах логического программирования. Метод резолюций работает с особой стандартной формой формул, которые называются предложениями. Предложение – это бескванторная дизъюнкция предикатных символов и (или) их отрицаний (литералов). Любая формула ИП может быть преобразована во множество предложений, используя следующие шаги:

- Приведение к предваренной форме с использованием равносильных преобразований.
- Элиминация кванторов существования с использованием преобразований (неравносильных)

$$\begin{aligned} \exists x_1 Q_2 x_2 \dots Q_n x_n F(x_1, x_2, \dots, x_n) &\Rightarrow Q_2 x_2 \dots Q_n x_n F(a, x_2, \dots, x_n) \\ \forall x_1 \dots \forall x_{i-1} \exists x_i Q_{i+1} x_{i+1} \dots Q_n x_n F(x_1, x_2, \dots, x_i, \dots, x_n) &\Rightarrow \\ \Rightarrow \forall x_1 \dots \forall x_{i-1} Q_{i+1} x_{i+1} \dots Q_n x_n F(x_1, x_2, \dots, x_{i-1}, f(x_1, \dots, x_{i-1}), x_{i+1}, \dots, x_n) \end{aligned}$$

где a – новая предметная константа, f – новый функциональный символ, $Q_2, \dots, Q_n$ – кванторы существования или всеобщности (или отсутствие квантора) по соответствующим переменным. После этого этапа формула может содержать только кванторы всеобщности. Такой вид формул называется *скюлемовская стандартная форма*.

- Элиминация кванторов всеобщности с использованием преобразования (неравносильного)

$$\forall x A(x) \Rightarrow A(x)$$

После этого шага формула не содержит кванторов.

- Приведение к конъюнктивной нормальной форме и элиминация конъюнкций. После этого получается множество предложений.

Пример. Преобразовать во множество предложений формулу

$$\forall x (P(x) \rightarrow \exists y (E(y) \& D(x, y)))$$

$$\begin{aligned} \forall x (P(x) \rightarrow \exists y (E(y) \& D(x, y))) &\equiv \forall x (\neg P(x) \vee \exists y (E(y) \& D(x, y))) \equiv \\ &\equiv \forall x \exists y (\neg P(x) \vee E(y) \& D(x, y)) \Rightarrow \forall x (\neg P(x) \vee E(f(x)) \& D(x, f(x))) \Rightarrow \\ &\Rightarrow \neg P(x) \vee E(f(x)) \& D(x, f(x)) \Rightarrow (\neg P(x) \vee E(f(x))) \& (\neg P(x) \& D(x, f(x))) \Rightarrow \\ &\Rightarrow \{\neg P(x) \vee E(f(x)), \neg P(x) \vee D(x, f(x))\} \end{aligned}$$

В целом метод резолюций для предикатов похож на метод резолюций для исчисления высказываний, хотя и имеет свои особенности. При рассмотрении метода резолюций для исчисления высказываний особое внимание уделялось поиску контрарных пар во множестве предложений. В логике предикатов этот поиск заметно усложняется, поскольку в формулы входят предметные переменные.

Пример. Рассмотрим особенности поиска контрарных пар на примере. Пусть имеется два предложения C_1 и C_2 (P, R, Q – предикаты, определенные на некоторой предметной области)

$$C_1 = P(y) \vee R(y),$$

$$C_2 = \neg P(f(x)) \vee Q(x).$$

Не существует ни одного литерала в C_1 контрарного какому-либо литералу в C_2 . Однако если подставить терм $f(a)$ вместо y в предложении C_1 и терм a – вместо x в C_2 , то получим предложения C'_1 и C'_2 , в которых уже существуют контрарные литералы $\{P(f(a)), \neg P(f(a))\}$

$$C'_1 = P(f(a)) \vee R(f(a)),$$

$$C'_2 = \neg P(f(a)) \vee Q(a).$$

Новые предложения уже можно резольвировать и получить резольвенту $C'_3 = Q(a) \vee R(f(a))$. Таким образом, для получения резольвент из предложений в логике предикатов необходимо не только отыскать контрарные предикаты

катные символы, но и выполнить подстановки соответствующих термов вместо переменных для получения контрарных пар.

Пусть X – множество предметных переменных из Ω и T – множество термов над переменными из X . Подстановкой называется отображение $\sigma: X \rightarrow T$. Подстановка σ называется *конечной*, если $\sigma(x) \neq x$ лишь для конечного числа элементов $x \in X$. Подстановка, не имеющая элементов, называется *тождественной*. Подстановку будем обозначать как множество пар $\sigma = \{x_1 \leftarrow t_1, \dots, x_k \leftarrow t_k\}$, где $x_1, \dots, x_k$ – переменные, $t_1, \dots, t_k$ – термы.

Пусть имеются две подстановки $\sigma = \{x_1 \leftarrow t_1, \dots, x_k \leftarrow t_k\}$ и $\nu = \{y_1 \leftarrow p_1, \dots, y_n \leftarrow p_n\}$. Произведением подстановок σ и ν называется подстановка $\sigma * \nu$, полученная из множества $\{x_1 \leftarrow \nu(t_1), \dots, x_k \leftarrow \nu(t_k), y_1 \leftarrow p_1, \dots, y_n \leftarrow p_n\}$ отбрасыванием всех элементов, имеющих вид $x_j \leftarrow \nu(t_j)$, для которых $\nu(t_j) = x_j$ и всех элементов $y_j \leftarrow p_j$ таких, что $y_j \in \{x_1, \dots, x_k\}$.

Пусть имеется две конечные последовательности термов $(t_1, \dots, t_k)$ и $(p_1, \dots, p_k)$. Унификатором двух последовательностей термов называется подстановка σ такая, что $\sigma(t_1) = \sigma(p_1), \dots, \sigma(t_k) = \sigma(p_k)$.

Следующий рекурсивный алгоритм выясняет, унифицируемы ли два терма S и T :

- Если термы S и T – константы, то S и T унифицируемы тогда и только тогда, когда они являются одной и той же предметной константой.
- Если терм S является переменной, а T – произвольным термом, то термы S и T унифицируемы и $\sigma = \{S \leftarrow T\}$. Если терм T является переменной, а S – произвольным термом, то термы S и T унифицируемы и $\sigma = \{T \leftarrow S\}$.
- В остальных случаях термы S и T унифицируемы тогда и только тогда, когда S и T имеют одинаковый главный функциональный символ и все их соответствующие компоненты (подтермы) унифицируемы.

Пример. Рассмотрим теперь *правило резолюции*, т.е. правило вывода, которое используется в методе резолюций. Пусть A и B – два предложения в исчислении предикатов, причем $A = P_1 \vee A_1$ и $B = \neg P_2 \vee B_1$, где P_1 и P_2 унифицируемые с помощью унификатора σ контрарные литералы. Правило вывода

$$\frac{A, B}{(A_1 \vee B_1)\sigma} R$$
 называется *правилом резолюции* в исчислении предикатов. В этом случае резольвентой предложений A и B является предложение $(A_1 \vee B_1)\sigma$, полученное из предложения $A_1 \vee B_1$ применением унификатора σ .

Пусть нужно установить выводимость $\Gamma \vdash F$ в ИП методом резолюций. Воспользуемся доказательством от противного и будем доказывать выводимость $\Gamma, \neg F \vdash \emptyset$. Для этого каждая формула множества Γ и формула $\neg F$ независимо преобразуются во множества предложений. В полученном сово-

купном множестве предложений отыскиваются резольвируемые предложения, к ним применяется правило резолюции в ИП и резольвента добавляется во множество и т.д. При этом возможны три случая:

- Среди текущего множества предложений нет резольвируемых. Это означает, что теорема опровергнута, т.е. формула F не выводима из множества формул Γ .
- В результате очередного применения правила резолюции получено пустое предложение. Это означает, что теорема доказана, т.е. $\Gamma \vdash F$.
- Процесс не заканчивается, т.е. множество предложений пополняется все новыми резольвентами, среди которых нет пустых. Это ничего не означает.

Таким образом, ИП является полуразрешимой теорией, а метод резолюций является частичным алгоритмом автоматического доказательства теорем, т.е. метод резолюций позволяет однозначно определить общезначимость формулы, но остальных случаях метод резолюций не всегда дает определенный результат.

Пример. Покажем выводимость формулы $\forall x \forall y F(x, y) \rightarrow \forall y \forall x F(x, y)$ методом резолюций. Возьмем отрицание этой формулы и преобразуем во множество предложений. Сначала необходимо привести формулу к предваренной форме. Для этого заменяются импликации и внешние отрицания переносятся к предикатным символам.

$$\begin{aligned} \neg(\forall x \forall y F(x, y) \rightarrow \forall y \forall x F(x, y)) &\equiv \neg(\neg(\forall x \forall y F(x, y)) \vee \forall y \forall x F(x, y)) \equiv \\ &\equiv \forall x \forall y F(x, y) \& \neg(\forall y \forall x F(x, y)) \equiv \forall x \forall y F(x, y) \& (\exists y \exists x \neg F(x, y)) \end{aligned}$$

Далее одноименные связанные переменные переименовываются и кванторы переносятся в начало формулы

$$\forall x \forall y F(x, y) \& (\exists y \exists x \neg F(x, y)) \equiv \forall y \exists y_1 \forall x \exists x_1 (F(x, y) \& \neg F(x_1, y_1))$$

После получения предваренной формы из формулы необходимо удалить кванторы. Сначала слева направо элиминируются кванторы существования, затем кванторы всеобщности. Самый левый квантор существования в формуле по переменной y_1 , и поскольку левее находится квантор всеобщности по переменной y , то при удалении из формулы квантора существования переменная y_1 заменяется на терм от переменной y (преобразования теперь не равносильные).

$$\forall y \exists y_1 \forall x \exists x_1 (F(x, y) \& \neg F(x_1, y_1)) \Rightarrow \forall y \forall x \exists x_1 (F(x, y) \& \neg F(x_1, f(y)))$$

Далее двигаясь по префиксу слева направо, удаляются все имеющиеся кванторы существования. Следующий квантор существования навешан по переменной x_1 и левее его находятся кванторы всеобщности по переменным x, y , поэтому при удалении из формулы квантора существования по переменной x_1 , переменная x_1 заменяется на терм от переменных x, y

$$\forall y \forall x \exists x_1 (F(x, y) \& \neg F(x_1, f(y))) \Rightarrow \forall y \forall x (F(x, y) \& \neg F(g(x, y), f(y)))$$

Кванторы всеобщности из формулы удаляются без дополнительных преобразований.

$$\forall y \forall x (F(x, y) \& \neg F(g(x, y), f(y))) \Rightarrow F(x, y) \& \neg F(g(x, y), f(y))$$

Полученную бескванторную формулу необходимо представить как конъюнкцию дизъюнкций предикатных символов. После удаления конъюнкций формула распадается на предложения

$$F(x, y) \& \neg F(g(x, y), f(y)) \Rightarrow \{F(x, y), \neg F(g(x, y), f(y))\}$$

Резольвируем полученное множество предложений, в правой колонке указаны унификаторы для правила резолюции

1 $F(x, y)$	$x \leftarrow g(a, b), y \leftarrow f(b)$
2 $\neg F(g(x, y), f(y))$	$x \leftarrow a, y \leftarrow b$
3 $\emptyset$	ПР из 1,2

Ответ: выводимость установлена.

Пример. Проверить правильность рассуждения:

Всякий, кто может решить эту задачу, – математик.

Ни один математик не может решить эту задачу.

Значит, она неразрешима.

Определим на множестве людей следующие предикаты:

$R(x)$ = « x может решить эту задачу»

$M(x)$ = « x математик».

Тогда фразы можно записать в логической символике следующим образом:

$$\forall x (R(x) \rightarrow M(x))$$

$$\neg (\exists x (M(x) \rightarrow R(x)))$$

$$\neg (\exists x R(x))$$

или

$$\forall x (R(x) \rightarrow M(x)), \neg (\exists x (M(x) \rightarrow R(x))) \vdash \neg (\exists x R(x)).$$

Проверим данную выводимость методом резолюций. Преобразуем во множество предложений гипотезы и отрицание целевой формулы. Полученное множество предложений $\{\neg R(x) \vee M(x), \neg R(x), M(x), R(a)\}$ резольвируется до получения пустой формулы.

Ответ: рассуждение правильное.

Задача 8

Построить машину Тьюринга для вычисления арифметической функции $f(x, y)$. Внешний алфавит состоит только из 0 и 1, 0 – пустой символ. Пояснения по построению программы для МТ обязательны. Проверить работу машины Тьюринга для конкретных значений x , y и нарисовать граф, соответствующий построенной МТ.

Варианты

Вариант 1. $f(x, y) = 2x + 6y$

Вариант 2. $f(x, y) = (x \div y) + y$ (используется усеченная разность)

Вариант 3. $f(x, y) = 2x + |y - x|$

Вариант 4. $f(x, y) = \begin{cases} x, & y < 5, \\ x + 1, & y \geq 5 \end{cases}$

Вариант 5. $f(x, y) = \begin{cases} y, & y < x, \\ x + 4, & y \geq x \end{cases}$

Вариант 6. $f(x, y) = \begin{cases} 3, & y = 6, \\ 4, & y = 7 \\ x + 1, & \text{иначе} \end{cases}$

Вариант 7. $f(x, y) = \begin{cases} 3, & 2 \leq y \leq 6 \\ x + 1, & \text{иначе} \end{cases}$

Вариант 8. $f(x, y) = (x + y) \bmod 2$

Вариант 9. $f(x, y) = (y \div x) + y$ (используется усеченная разность)

Вариант 10. $f(x, y) = x + |y - 3x|$

Вариант 11. $f(x, y) = \begin{cases} 3, & y = 6, \\ 2, & y \leq 7 \\ x + 1, & \text{иначе} \end{cases}$

Вариант 12. $f(x, y) = \begin{cases} 5, & 3 \leq x \leq 8 \\ x \div y, & \text{иначе} \end{cases}$ (используется усеченная разность)

Вариант 13. $f(x, y) = \begin{cases} 2x, & y < 5, \\ x + 1, & y \geq 5 \end{cases}$

Вариант 14. $f(x, y) = \begin{cases} 3x, & y < x, \\ 3y, & y \geq x \end{cases}$

Вариант 15. $f(x, y) = 2y + |5y - x|$

Вариант 16. $f(x, y) = \begin{cases} x, & y < 5, \\ 2x, & y \geq 5 \end{cases}$

Вариант 17. $f(x, y) = \begin{cases} 6x, & y < x, \\ x + 1, & y \geq x \end{cases}$

- Вариант 18. $f(x, y) = \begin{cases} x + y, y = 3, \\ 5, y = 7 \\ x + 1, \text{ иначе} \end{cases}$
- Вариант 19. $f(x, y) = x + 2 |y - x|$
- Вариант 20. $f(x, y) = \begin{cases} x, y = 6, \\ 4, y = 7 \\ 3x + 1, \text{ иначе} \end{cases}$
- Вариант 21. $f(x, y) = \begin{cases} 5, 3 \leq y \leq 8 \\ 3x \div 1, \text{ иначе} \end{cases}$ (используется усеченная разность)
- Вариант 22. $f(x, y) = \begin{cases} x + y, y < 5, \\ x + 1, y \geq 5 \end{cases}$
- Вариант 23. $f(x, y) = \begin{cases} 5x, 4 \leq y \leq 8 \\ x \div 1, \text{ иначе} \end{cases}$ (используется усеченная разность)
- Вариант 24. $f(x, y) = \begin{cases} x, y < x, \\ x + y, y \geq x \end{cases}$
- Вариант 25. $f(x, y) = \begin{cases} x, y < 5, \\ 5x + 1, y \geq 5 \end{cases}$

Необходимый теоретический материал

Под одноленточной машиной Тьюринга понимают такое кибернетическое устройство, которое состоит из следующих элементов:

- *бесконечной ленты*, разделенной на ячейки, которая используется для ввода и вывода данных, а также для записи промежуточных результатов. На ленте могут записываться буквы некоторого алфавита $A = \{a_0, a_1, \dots, a_m\}$ (внешний алфавит машины Тьюринга). Удобно считать, что пустые ячейки на самом деле содержат некоторую букву алфавита A (будем считать для определенности, что пустой – символ это буква a_0);
- *управляющей головки*, способной читать символы, содержащиеся в ячейках ленты, писать символы в эти ячейки и оставаться на месте или передвигаться на одну ячейку вправо или влево;
- *выделенной ячейки памяти*, содержащей символ конечного алфавита, задающий состояние машины Тьюринга. Головка может находиться в одном из состояний $Q = \{q_0, q_1, \dots, q_n\}$ (внутренний алфавит машины Тьюринга);
- *механического устройства управления*, обеспечивающего перемещение головки относительно ленты;
- *функциональной схемы* – области памяти, содержащей команды (программу) машины Тьюринга, доступной только для чтения.

Обычно машина Тьюринга схематично изображается так:

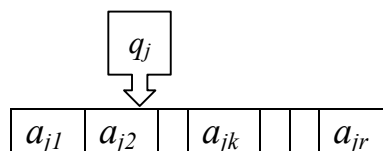


Рис. 1. Схематичное представление машины Тьюринга

Поскольку бесконечную ленту физически представить затруднительно, то обычно предполагается, что она конечная и разбита на конечное число ячеек. В процессе работы к существующим ячейкам машина может пристраивать новые ячейки, так что лента может считаться потенциально неограниченной в обе стороны. Все вновь пристраиваемые ячейки пристраиваются пустыми.

Управляющая головка – это некоторое устройство, которое может перемещаться вдоль ленты так, что в каждый рассматриваемый момент времени оно находится в определенной ячейке ленты. Если какая-нибудь ячейка находится в управляющей головке, то говорят также, что машина в данный момент «воспринимает» или «обозревает» эту ячейку.

Внутренняя память машины – это некоторое устройство, которое в каждый рассматриваемый момент находится в некотором «состоянии». Предполагается, что число возможных состояний внутренней памяти конечное и для каждой машины фиксированное. Состояние внутренней памяти мы будем обозначать символами алфавита $Q = \{q_0, q_1, \dots, q_n\}$ или любыми другими символами, не входящими во внешний алфавит машины. Совокупность символов, обозначающих состояния внутренней памяти, называется внутренним алфавитом машины. Состояния внутренней памяти часто называют *внутренними состояниями* машины. Одно из этих состояний называется *начальным*, с него начинается работу любая машина. Обозначим это состояние q_1 . В *заключительном* состоянии q_0 машина Тьюринга завершает работу, т.е. если в какой-то момент времени внутренняя память машины приходит в заключительное состояние q_0 , то дальнейших изменений в машине не происходит и машина называется остановившейся. Может случиться, что в машине не будет происходить никаких изменений и при каком-то другом внутреннем состоянии q_i . Однако в этом случае мы будем говорить, что машина продолжает работать «вечно».

Предполагается, что машина снабжена особым механизмом, который в зависимости от состояния воспринимаемой ячейки и состояния внутренней памяти может изменить состояние внутренней памяти и одновременно изменить состояние воспринимаемой ячейки и сдвинуть управляющую головку в соседнюю ячейку. В частном случае состояние воспринимаемой ячейки и/или состояние внутренней памяти могут оставаться неизменными, а управляющая головка – неподвижной. Если управляющая головка находится в самой правой ячейке и по ходу работы машина должна сдвинуть управляющую головку в соседнюю справа (отсутствующую) ячейку, то предполагается, что, сдвигая головку, машина одновременно пристроит недостающую ячейку в пустом состоянии. Аналогично работает машина и в случае, когда головка воспринимает

самую левую ячейку и по ходу выполнения программы машине надо сдвинуть головку еще левее.

Работа машины состоит в том, что она из текущего состояния по истечении одного такта работы механического устройства переходит в следующее состояние, затем от нового состояния по истечении такта работы переходит к новому состоянию и т.д. Таким образом, если машина, имея внутреннее состояние q_i и воспринимая ячейку ленты с символом a_j , переводит внутреннюю память в состояние q_r и одновременно содержимое воспринимаемой ячейки заменяет символом a_s , а управляющая головка остается на месте (H), сдвинется на одну ячейку вправо (R) или влево (L), то говорят, что машина выполняет одну из команд:

$$q_i a_j \rightarrow a_s H q_r,$$

$$q_i a_j \rightarrow a_s R q_r,$$

$$q_i a_j \rightarrow a_s L q_r.$$

Совокупность всех команд, которые может выполнять машина, называется ее *программой*. Так как работа машины по условию целиком определяется состоянием в данный момент ее внутренней памяти q_i и состоянием воспринимаемой ячейки a_j , то для каждой пары $q_i a_j$ ($i = 1, \dots, n$, $j = 1, \dots, m$), программа машины должна содержать одну и только одну команду, начинающуюся словом $q_i a_j$. Таким образом, программа машины с внешними символами $\{a_0, a_1, \dots, a_m\}$ и внутренними состояниями $\{q_0, q_1, \dots, q_n\}$ может содержать максимум $(n + 1)(m + 1)$ команд.

Текущее состояние машины Тьюринга, или *конфигурация*, – это полная информация о внутреннем состоянии машины, содержимом ячеек ленты и ячейке, которую обозревает головка машины. Конфигурация машины может быть записана в виде слова xq_iy , где x и y – слова, записанные на ленте, причем:

- левее слова x и правее слова y все ячейки содержат пустой символ;
- головка машины обозревает ячейку, которая находится сразу после слова x ;
- первая буква слова y записана в ячейке ленты, обозреваемой головкой машины;
- q_i – состояние машины на данном шаге.

На рисунке определение конфигурации можно продемонстрировать следующим образом:

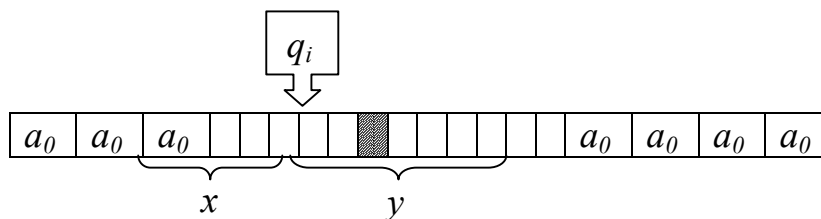


Рис. 2. Текущая конфигурация машины Тьюринга

Конфигурация с начальным состоянием головки называется *начальной*, с заключительным состоянием – *заключительной*. Переход за один такт работы МТ из конфигурации $x_1q_iy_1$ в конфигурацию $x_2q_jy_2$ будем обозначать так:

$$x_1q_iy_1 \rightarrow x_2q_jy_2.$$

Если переход из одной конфигурации в другую осуществляется более чем за один шаг, то это будем обозначать таким образом:

$$x_1q_iy_1 \Rightarrow x_2q_jy_2.$$

Действия МТ весьма разнообразны и зависят не только от программы, но и от начальной записи на ленте и начального положения головки. Любые действия машины можно назвать вычислениями.

Вычисления, производимые МТ в алфавите A , называются *правильными*, если выполняются следующие условия

- В начальный момент машина находится в состоянии q_1 , а на ленте записано некоторое слово W в алфавите A и все ячейки ленты, не содержащие слово W , содержат пустой символ. Головка машины обозревает ячейку с первой буквой слова W .

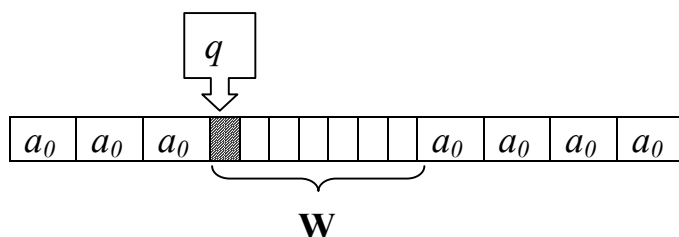


Рис. 3. Головка машины установлена на первом символе слова W

- При переходе МТ в заключительное состояние q_0 все ячейки левее головки содержат пустой символ. Головка находится над самой левой среди ячеек, не содержащей пустой символ (если такие имеются).

Результатом правильных вычислений МТ считается:

- Пустое слово, если ячейки ленты не содержат непустых символов.
- Слово, первая буква которого находится в ячейке, обозреваемой головкой, а последняя отлична от пустого символа и такая, что все буквы на ленте являются пустыми символами.

Пример. Построить МТ, которая осуществляет переход:

$$0q_11^n \Rightarrow 01^nq_00.$$

Нетрудно понять, что внешний алфавит для такой МТ достаточно взять двухсимвольный $A=\{0,1\}$, причем пустым символом будет 0.

Действия МТ в данном случае сводятся к переходу к первому нулю после последовательности единиц и остановке.

Программа МТ для перехода:

$$\begin{aligned}
q_1 0 &\rightarrow 0 H q_0 \\
q_1 1 &\rightarrow 1 R q_2 \\
q_2 1 &\rightarrow 1 R q_2 \\
q_2 0 &\rightarrow 0 H q_0.
\end{aligned}$$

Удобно представлять программу МТ в виде таблицы, строки которой помечены символами внешнего алфавита МТ, а столбцы – символами внутреннего алфавита МТ. Если МТ имеет команду $q_i a_i \rightarrow a_s H q_b$, то в ячейке на пересечении строки a_i и столбца q_i находится выражение $a_s H q_b$. Заполним таблицу, которая соответствует данной МТ.

	q_1	q_2
0	$0 H q_0$	$0 H q_0$
1	$1 R q_2$	$1 R q_2$

Приведем пошаговое исполнение программы для начальной конфигурации при $n=3$:

$$0 q_1 1 1 1 0 \rightarrow 0 1 q_2 1 1 0 \rightarrow 0 1 1 q_2 1 0 \rightarrow 0 1 1 1 q_2 0 \rightarrow 0 1 1 1 q_0 0$$

Пример. Построим МТ, которая работает следующим образом ($x, y \geq 1$):

$$q_1 1^x 0 1^y \Rightarrow \begin{cases} 0 q_0 1 0, & \text{если } x \geq y \\ 0 q_0 0, & \text{если } x < y \end{cases}$$

Внешний алфавит для такой МТ достаточно взять двухсимвольный $A = \{0, 1\}$, причем пустым символом будет 0. Будем удалять из каждой группы единиц по одной единице до тех пор, пока одна из групп не станет пустой, причем из первой группы будем удалять из начала, а из второй – из конца. Тогда если пустой стала первая группа единиц, то $x < y$ и необходимо обнулить все оставшиеся единицы и остановиться в заключительном состоянии. Если пустой стала вторая группа единиц, то $x \geq y$ и необходимо обнулить все оставшиеся единицы, поставить одну единицу и остановиться в заключительном состоянии. Программа такой МТ имеет следующий вид:

$q_1 1 \rightarrow 0 R q_2$	Удаляем 1 из начала первой группы.
$q_2 1 \rightarrow 1 q_2 R$	Проходим через первую группу единиц вправо.
$q_2 0 \rightarrow 0 q_3 R$	Проходим через разделяющий 0 вправо.
$q_3 1 \rightarrow 1 q_3 R$	Проходим через вторую группу единиц вправо.
$q_3 0 \rightarrow 0 q_4 L$	Устанавливаем головку на конец второй группы единиц.
$q_4 1 \rightarrow 0 q_5 L$	Удаляем 1 из конца второй группы.
$q_5 1 \rightarrow 1 q_5 L$	Проходим через вторую группу единиц влево.
$q_5 0 \rightarrow 0 q_6 L$	Проходим через разделяющий 0 влево.
$q_6 1 \rightarrow 1 q_6 L$	Проходим через первую группу единиц влево.
$q_6 0 \rightarrow 0 q_1 R$	Устанавливаем головку на начало первой группы единиц

и повторяем все действия.

$q_1 0 \rightarrow 0 q_8 R$	Если первая группа закончилась, удаляем оставшиеся единицы.
$q_8 1 \rightarrow 0 q_8 R$	Проходим через вторую группу единиц вправо, заменяя 1 на 0.
$q_8 0 \rightarrow 0 q_8 H$	Остановка.
$q_4 0 \rightarrow 0 q_7 L$	Если вторая группа закончилась, удаляем оставшиеся единицы.
$q_7 1 \rightarrow 0 q_7 L$	Проходим через первую группу единиц влево, заменяя 1 на 0.
$q_7 0 \rightarrow 1 q_0 H$	Остановка на ячейке с единицей.

Приведем пошаговое исполнение программы для начальной конфигурации при $x=3, y=2$:

$0 q_1 1 1 1 0 1 1 0 \rightarrow 0 0 q_2 1 1 0 1 1 0 \rightarrow 0 0 1 q_2 1 0 1 1 0 \rightarrow 0 0 1 1 q_2 0 1 1 0 \rightarrow$
 $\rightarrow 0 0 1 1 0 q_3 1 1 0 \rightarrow 0 0 1 1 0 1 q_3 1 0 \rightarrow 0 0 1 1 0 1 1 q_3 0 \rightarrow 0 0 1 1 0 1 q_4 1 0 \rightarrow$
 $\rightarrow 0 0 1 1 0 q_5 1 0 \rightarrow 0 0 1 1 q_5 0 1 0 \rightarrow 0 0 1 q_6 1 0 1 0 \rightarrow 0 0 q_6 1 1 0 1 0 \rightarrow$
 $\rightarrow 0 q_6 0 1 1 0 1 0 \rightarrow 0 0 q_1 1 1 0 1 0 \rightarrow 0 q_2 1 0 1 0 \rightarrow 0 1 q_2 0 1 0 \rightarrow$
 $\rightarrow 0 1 0 q_3 1 0 \rightarrow 0 1 0 1 q_3 0 \rightarrow 0 1 0 q_4 1 0 \rightarrow 0 1 q_5 0 0 0 \rightarrow 0 q_6 1 0 0 0 \rightarrow$
 $\rightarrow 0 q_6 1 0 0 0 \rightarrow 0 q_6 0 1 0 0 0 \rightarrow 0 q_1 1 0 0 0 \rightarrow 0 q_2 0 0 0 \rightarrow$
 $\rightarrow 0 q_3 0 0 0 \rightarrow 0 q_4 0 0 0 \rightarrow 0 q_7 0 0 0 \rightarrow 0 q_0 1 0 0$

Далее будем применять МТ для правильного вычисления арифметических функций $f: \bar{N} \rightarrow \bar{N}$, т.е. функций, определенных на множестве неотрицательных целых чисел, со значениями во множестве неотрицательных целых чисел. Для вычисления арифметических функций на машине Тьюринге с двоичным внешним алфавитом аргумент функции x представлен на ленте в виде последовательности $x+1$ единиц. Если функция зависит от нескольких аргументов, то аргументы последовательно записаны группами единиц, разделенные 0.

Правильным вычислением арифметической функции $f(x)$ на МТ будем называть правильные вычисления, производимые МТ в алфавите $A = \{0, 1\}$ при переходе от конфигурации $0 q_1 1^{x+1} 0$ к конфигурации $0 q_0 1^{f(x)+1} 0$.

Пример. Вычислим функцию $O(x) = 0$. Чтобы реализовать вычисление этой функции на МТ, необходимо осуществить переход $0 q_1 1^{x+1} 0 \Rightarrow 0 q_0 1 0$. Очевидно, что действия МТ сводятся к последовательной замене всех единиц на ленте нулями.

Программа вычислений:

	q_1	q_2
0		$1 H q_0$
1	$0 R q_2$	$0 R q_2$

Пример. Вычислим функцию $S(x) = x + 1$. Чтобы реализовать вычисление этой функции на МТ, необходимо осуществить переход $0 q_1 1^{x+1} 0 \Rightarrow 0 q_0 1^{(x+1)+1} 0$.

Для вычисления этой функции достаточно приписать слева одну единицу к последовательности единиц на ленте. Программа вычислений приведена в таблице:

	q_1	q_2
0		$1Hq_0$
1	$1Lq_2$	

Пример. Построить машину Тьюринга для вычисления функции $C(x) = S(O(x))$. Искомую машину будем строить как суперпозицию машин, вычисляющих функции $O(x) = 0$ и $S(x) = x + 1$.

Конструкция суперпозиции машин Тьюринга состоит в следующем. Пусть имеются две машины T_1 и T_2 , которые вычисляют функции $f_1(x)$ и $f_2(x)$ соответственно в одном и том же алфавите. Построим новую машину Тьюринга T следующим образом. Состояния машины T_2 переобозначим (с помощью верхнего индекса) так, чтобы они отличались от состояний T_1 . Начальное состояние q_1^1 машины T_1 объявляем начальным состоянием q_1 новой машины T , а заключительное состояние q_0^2 машины T_2 — заключительным состоянием q_0 для машины T . Заключительное состояние q_0^1 машины T_1 и начальное состояние q_1^2 машины T_2 отождествляются. Полученные команды для обеих машин объединяются в одну программу новой машины. Построенная машина T вычисляет суперпозицию функций $f(x) = f_2(f_1(x))$ и называется суперпозицией машин T_1 и T_2 .

Применим эту конструкцию для вычисления функции $C(x) = S(O(x))$. Объединим множества внутренних состояний и программы машин Тьюринга, которые были ранее построены для вычисления функций $O(x) = 0$ и $S(x) = x + 1$. В результате из двух таблиц

	q_1	q_2
0		$1Hq_0$
1	$0Rq_2$	$0Rq_2$

	q_1	q_2
0		$1Hq_0$
1	$1Lq_2$	

получим одну

	q_1	q_2^1	q_0^1	q_2^2
0		$1H q_0^1$		$1Hq_0$
1	$0R q_2^1$	$0R q_2^1$	$1L q_2^2$	

Таким образом, построенная машина будет сначала вычислять $O(x)$, по окончании вычислений головка МТ остановится на самой левой единице результата. А затем продолжит вычисление функции $S(x)$.

Пример. Построить МТ для вычисления функции $f(x) = \begin{cases} 0, x > 1 \\ x + 1, x \leq 1 \end{cases}$

Ранее были построены МТ для вычисления функций $O(x) = 0$ и $S(x) = x + 1$. Нетрудно видеть, что исходная функция $f(x)$ – это функция $O(x)$, если $x > 1$ и функция $S(x)$ в противном случае. Таким образом, сначала МТ необходимо определить, сколько единиц находится на ленте, и в зависимости от этого переходить к вычислению функций $O(x)$ или $S(x)$. Для определения количества единиц на ленте будем сдвигаться по ленте вправо и на каждом сдвиге менять состояние МТ. Если обнаружили больше двух единиц, необходимо вернуться назад, на начало последовательности единиц, и применить МТ для вычисления функции $O(x)$. Если обнаружили две или одну единицу, то устанавливаем головку на начале последовательности единиц и применяем МТ для вычисления функции $S(x)$. Внутренние состояния машин для вычисления функций $O(x)$ и $S(x)$ пометим буквами O и S соответственно. Заключительные состояния этих машин отождествим:

$q_1 1 \rightarrow 1 R q_2$

$q_2 1 \rightarrow 1 R q_3$ *На ленте больше одной единицы*

$q_3 1 \rightarrow 1 L q_4$ *На ленте больше двух единиц*

$q_4 1 \rightarrow 1 L q_4$

Сдвигаемся обратно на начало последовательности

$q_4 0 \rightarrow 0 R q_{O1}$ *Начинаем вычисление функции $O(x)$*

$q_2 0 \rightarrow 0 L q_{S1}$ *На ленте одна единица, начинаем вычисление $S(x)$*

$q_3 0 \rightarrow 0 L q_5$

На ленте две единицы, переходим на начало последовательности

$q_5 1 \rightarrow 1 L q_5$

$q_5 0 \rightarrow 0 R q_{S1}$ *Начинаем вычисление $S(x)$*

Пример. Построить МТ для вычисления функции $g(x, y) = x + y$.

Чтобы правильно вычислить данную функцию, необходимо построить МТ, которая переводит конфигурацию $0 q_1 1^{x+1} 0 1^{y+1}$ в конфигурацию $0 q_0 1^{x+y+1} 0$. Порядок действий для МТ можно выбрать такой. Сначала происходит движение головки вправо до появления на ленте нуля. Этот нуль разделяет аргументы на ленте. Поставим в эту ячейку вместо нуля единицу и вернемся к началу последовательности единиц. Заменяем первые две единицы нулями. Тогда на ленте останется $x + y + 1$ единиц. Набор команд для такой МТ:

$q_1 1 \rightarrow 1 R q_2$

$q_2 1 \rightarrow 1 R q_2$

$q_2 0 \rightarrow 1 L q_3$

$$\begin{aligned} q_3 1 &\rightarrow 1 L q_3 \\ q_3 0 &\rightarrow 0 R q_4 \\ q_4 1 &\rightarrow 0 R q_5 \\ q_5 1 &\rightarrow 0 R q_0 \end{aligned}$$

Программу МТ можно изобразить в виде ориентированного графа, вершинами которого являются внутренние состояния МТ, а дуги помечены тройками из символов внешнего алфавита и символов из множества $\{L, R, H\}$.

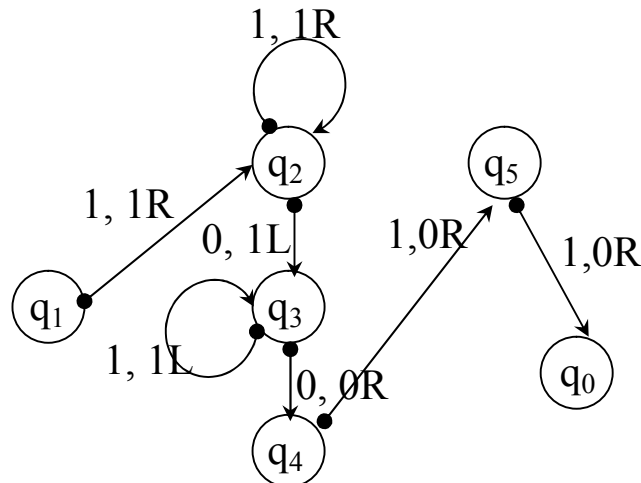


Рис. 4. Граф машины Тьюринга

Проверим работу МТ для конкретных значений переменных $x = 3, y = 2$

Последовательно применим команды МТ для данной конфигурации:

$$\begin{aligned} 0 q_1 11110111 &\rightarrow 0 q_2 11110111 \rightarrow 011 q_2 110111 \rightarrow 0111 q_2 10111 \rightarrow 01111 q_2 0111 \rightarrow \\ &\rightarrow 01111 q_3 11111 \rightarrow 011 q_3 111111 \rightarrow 01 q_3 1111111 \rightarrow 0 q_3 11111111 \rightarrow \\ &\rightarrow 0 q_3 011111111 \rightarrow 0 q_4 111111111 \rightarrow 00 q_5 11111111 \rightarrow 000 q_0 111111 \end{aligned}$$

Таким образом, головка МТ остановилась в заключительном состоянии на самой левой единице последовательности из 6 $(3+2+1)$ единиц. Функция вычислена правильно.

Список литературы

1. Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. – М.: Наука, 1975.
2. Нефедов В.Н., Осипова В.А. Курс дискретной математики: Учеб. пособие. – М.: Изд-во МАИ, 1992.
3. Капитонова Ю.В. и др. Лекции по дискретной математике. – СПб.: БЧВ-Петербург, 2004.
4. Новиков Ф.А. Дискретная математика для программистов. – СПб.: Питер, 2001.
5. Унучек, С. А. Математическая логика [Электронный ресурс] : учебное пособие / С. А. Унучек. – Электрон. текстовые данные. – Саратов : Ай Пи Эр Медиа, 2018. – 239 с. – 978-5-4486-0086-9. – Режим доступа:

- <http://www.iprbookshop.ru/69312.html>. Лицензия: весь срок охраны авторского права.
6. Макоха, А. Н. Математическая логика и теория алгоритмов [Электронный ресурс] : учебное пособие / А. Н. Макоха, А. В. Шапошников, В. В. Бережной. – Электрон. текстовые данные. – Ставрополь : Северо-Кавказский федеральный университет, 2017. – 418 с. – 2227-8397. – Режим доступа: <http://www.iprbookshop.ru/69397.html>. Лицензия: весь срок охраны авторского права.
 7. Математическая логика и теория алгоритмов [Электронный ресурс] : методические указания к самостоятельной работе / – Электрон. текстовые данные. – Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2014. – 25 с. – Режим доступа: <http://www.iprbookshop.ru/55106>. – ЭБС «IPRbooks» Лицензия: весь срок охраны авторского права.
 8. Зюзьков, В. М. Математическая логика и теория алгоритмов [Электронный ресурс] : учебное пособие / В. М. Зюзьков. – Электрон. текстовые данные. – Томск : Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2015. – 236с. – 978-5-4332-0197-2. – Режим доступа: <http://www.iprbookshop.ru/72122.html>. Лицензия: весь срок охраны авторского права.
 9. Балюкевич, Э.Л. Математическая логика и теория алгоритмов [Электронный ресурс] : учебное пособие / Балюкевич Э.Л., Ковалева Л.Ф. – Электрон. текстовые данные. – М.: Евразийский открытый институт, 2009. – 188 с. – Режим доступа: <http://www.iprbookshop.ru/10772>. – ЭБС «IPRbooks». Лицензия до 17.02.2020.
 10. Маньшин, М.Е. Математическая логика и теория алгоритмов [Электронный ресурс] : учебное пособие / Маньшин М.Е. – Электрон. текстовые данные. – Волгоград : Волгоградский институт бизнеса, Вузовское образование, 2009. – 106 с. – Режим доступа: <http://www.iprbookshop.ru/11334>. – ЭБС «IPRbooks», по паролю. Лицензия 09.12.2023.
 11. Зарипова, Э.Р. Лекции по дискретной математике. Математическая логика [Электронный ресурс] : учебное пособие / Зарипова Э.Р., Кокотчикова М.Г., Севастьянов Л.А. – Электрон. текстовые данные. – М.: Российский университет дружбы народов, 2014. – 120 с. – Режим доступа: <http://www.iprbookshop.ru/22190>. ЭБС «IPRbooks», по паролю. Лицензия: весь срок охраны авторского права.
 12. Алябьева, В.Г. Теория алгоритмов [Электронный ресурс] : учебное пособие для специальности 050201.65 – «Математика с дополнительной специальностью “Информатика”», направление подготовки 050100 – «Педагогическое образование» / Алябьева В.Г., Пастухова Г.В. – Электрон. текстовые данные. – Пермь : Пермский государственный гуманитарно-педагогический университет, 2013. – 125 с. – Режим доступа: <http://www.iprbookshop.ru/32100>. – ЭБС «IPRbooks». Лицензия: весь срок охраны авторского права.

Учебное издание

Елена Павловна Мачикина

Указания по выполнению РГР по дисциплине
«Математическая логика и теория алгоритмов»

Корректор: А.Р. Шерстнёва

Принято редакционно-издательским советом № 21 от 25.03.2020,
подписано в печать 31.03.2020.

п. л. – 2,3, заказ № 21, 44 стр., электронная версия – 675 Кб.

Редакционно-издательский отдел СибГУТИ
630102, г. Новосибирск, ул. Кирова, 86, офис 105
тел. (383) 269-83-56