

Дискретная математика (2 часть)

Судаков Владимир Анатольевич
2020

Литература

- Клини С.К. Математическая логика. – М.: Изд-во Мир, 1973.
- Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. М.: Наука. 1983.
- Пенроуз Р. Новый ум короля. — М.: Едиториал УРСС, 2003.
- Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. - М.: Мир, 1982.

Некоторые из применений

- SQL и MDX
- Формализация требований к АБС
- Искусственный интеллект (машинное обучение и инженерия знаний)
- Построение и верификация моделей

Разделы курса

- Логика высказываний
- Логика предикатов
- Автоматическое доказательство
- Теория вычислимости

Математическая логика

- Математическая логика - логика развиваемая с помощью математических методов
- Математическая логика занимается обоснованием математических утверждений с помощью законов логики.
- Логика – наука о рассуждении - наука о формах, методах и законах интеллектуальной познавательной деятельности, формализуемых с помощью логического языка
- Логика изучает, что из чего следует

Высказывания

- Высказывание – повествовательное предложение. Может быть ложным или истинным

Различают:

- Атомарные высказывания
- Неатомарные высказывания – состоят из атомарных высказывания и связок: и, или, не, если, эквиваленция.

Некоторые задачи

Определение характера высказывания:

- Тавтология (общезначимое высказывание)— высказывание которое всегда истинно.
Например, $(X \& Y) \rightarrow Y$
- Выполнимое высказывание – есть хотя бы одна 1 в таблице истинности
- Нейтральное высказывание – есть хотя бы одна 1 и хотя бы один 0 в таблице истинности
- Логических следствий – если истинно высказывание X, то истинно высказывание Y. Для всех строк таблицы истинности где $X=1$ там и $Y=1$. Из X следует Y если $X \rightarrow Y$ - тавтология

Высказывания

- *Высказывание* – это повествовательное предложение, о котором можно сказать истинно оно или ложно. Что является высказыванием и что из этого атомарно:
 - А. «Число $\sqrt{2}$ является иррациональным».
 - Б. «Неверно, что число $\sqrt{2}$ является иррациональным».
 - В. «Если число $\sqrt{2}$ является иррациональным, то число $\sqrt{2}+1$ также является иррациональным».
 - Г. «Который час?»
 - Д. «Идите решать задачу к доске»
 - Е. «Число $\sqrt{2}+1$ является иррациональным»

Интерпретация

- Пусть G – данная формула логики высказываний и $A_1, A_2, \dots, A_n$ её атомы. Тогда интерпретацией формулы G называется такое приписывание истинностных значений атомам $A_1, A_2, \dots, A_n$ при котором каждому из A_i приписано либо 1, либо 0 (но не оба вместе).
- Интерпретация – строка таблицы истинности.
- Говорят, что формула G истина при некоторой интерпретации, тогда и только тогда, когда G получает значение истина(1) в этой интерпретации; в противном случае говорят, что G ложна(0) при этой интерпретации.

Перевод с естественного языка на язык ЛОГИКИ ВЫСКАЗЫВАНИЙ

Если я поеду автобусом, а автобус опоздает, то я пропущу назначенное свидание. Если я пропущу назначенное свидание и начну огорчаться, то мне не следует ехать домой. Если я не получу работу, то я начну огорчаться и мне следует поехать домой. Следовательно, если я поеду автобусом, а автобус опоздает, то я получу эту работу.

1) Обозначим высказывания

X - “я поеду автобусом”, Y - “автобус опоздает”,
 Z - “я пропущу свидание”, U - “я начну огорчаться”,
 V - “мне следует ехать домой”, W - “я получу работу”.

2) Запишем высказывания в виде формул:

$X \& Y \rightarrow Z, Z \& U \rightarrow \neg V, \neg W \rightarrow U \& V, X \& Y \rightarrow W.$

Тавтология

- **Обозначение тавтологии:** $\models$
- **Теорема.** Пусть формула E – формула, в которую входят только атомы $A_1, A_2, \dots, A_n$, а E^* – формула, которая получена из E подстановкой формул $F_1, F_2, \dots, F_n$ вместо $A_1, A_2, \dots, A_n$ соответственно. Если $\models E$, то $\models E^*$.

Логическое следствие

- Одна из основных целей логики состоит в разработке формального аппарата для доказательства того, является ли данное утверждение следствием других.
- **Определение.** Формула G называется *логическим следствием* формул $F_1, F_2, \dots, F_k$, тогда и только тогда когда для всякой интерпретации, в которой $F_1 = F_2 = \dots = F_k = 1$, G так же истина.
- **Алгоритм для проверки логического следствия:** По таблице истинности нужно последовательно просматривать строки пытаясь найти такую строку, в которой $F_1 = F_2 = \dots = F_k = 1$, $G = 0$. Если такую строку найти удастся, то G не есть логическое следствие $F_1, F_2, \dots, F_k$, в противном случае G есть логическое следствие $F_1, F_2, \dots, F_k$.
- **Замечание:** если нет ни одной строки, где $F_1 = F_2 = \dots = F_k = 1$ не найдено, то G есть логическое следствие $F_1, F_2, \dots, F_k$.
- **Теорема.** Формула G является логическим следствием формул $F_1, F_2, \dots, F_k$ тогда и только тогда, когда

$$\models (F_1 \& F_2 \dots \& F_k) \rightarrow G$$

Проверка логического следствия

- Таблица истинности – для большого числа высказываний долго.
- Можно использовать теорему:
- **Теорема.** Формула G является логическим следствием формул $F_1, F_2, \dots, F_k$ тогда и только тогда, когда множество формул

$$L = \{F_1, F_2, \dots, F_k, \neg G\}$$

невыполнимо

Тавтологии. Общезначимые.

$\models$

Если $\models E$ то $\models E^*$

Обратное не верно!

Основные тавтологии

Если $\models A$ и $\models A \rightarrow B$ то $\models B$

Следствие

	h	i	j	k	l
P Q R	$(P \rightarrow Q) \& (P \text{ or } R)$	$Q \text{ or } R$	$P \rightarrow R$	$P \text{ or } \sim Q \rightarrow R$	$P \& \sim Q$

$l, j \models k$ $h, l \models i$ $h \models i$

Знаем что то и хотим узнать ещё

Теорема $A \models B$ тогда и только тогда когда $\models A \rightarrow B$

$A_1, A_2, \dots \models B$ тогда и только тогда когда $\models A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow B)$

Продолжение задачи

- Логично ли рассуждение в задаче приведенной выше?

$$F_1 = X \& Y \rightarrow Z, F_2 = Z \& U \rightarrow \neg V, F_3 = \neg W \rightarrow U \& V, F_4 = X \& Y \rightarrow W.$$

Выяснить, будет ли формула F_4 логическим следствием формул F_1, F_2, F_3 ?

Решение:

- Предположим, что при некоторой интерпретации формула F_4 принимает значение 0, а формулы F_1, F_2, F_3 – значение 1.
- Если $F_4 = 0$, то $X = Y = 1$ и $W = 0$.
- Из того, что $F_1 = F_3 = 1$, следуют равенства $Z = U = V = 1$.
- Но это *противоречит* тому, что $F_2 = 1$.
- Полученное противоречие показывает, что если $F_1 = F_2 = F_3 = 1$, то $F_4 = 1$, т.е. что формула F_4 является логическим следствием формул F_1, F_2, F_3 и рассуждение логично.

Логические связи

- $A \leftrightarrow B$ - если А то и В и обратно, А если В и В если А, для А необходимо и достаточно В, А эквивалентно В, А равносильно В
- $A \rightarrow B$ - если А то В, А только если В, коль скоро А то В, В если А, в случае А имеет место В, А имплицитует В, для В достаточно А, для А необходимо В, А влечет В
- $A \& B$ - А и В, как А так и В, не только А но и В, А вместе с В, В хотя и А, А в то время как В, В несмотря на А
- $A \vee B$ - или А или В или оба, А и/или В, А если не В
- $A \oplus B$ - или А или В, но не оба, либо А либо В

Проблемы при переводе на язык логики

- Потеря причинно-следственной связи:
 - «у Джейн родился ребенок, и она вышла замуж»
 - «Джейн вышла замуж, и у нее родился ребенок»
$$A \& B = B \& A$$
- Двусмысленность терминов, когда их надо переводить связками:
 - В меню ресторана указано: «Чай или кофе бесплатно»,
 - Объявление гласит, что «книжные пожертвования принимаются в церкви или в школе»

Расстановка скобок

- Если Джонс присутствует [Д] или если Уильямс выскажется за наше предложение [У] и Старк не станет возражать [С], то наше предложение будет принято [П].

Как надо переводить?

- $(D \vee U) \wedge \neg C \supset P$

или так:

- $D \vee (U \wedge \neg C) \supset P$

Примеры логических рассуждений

- Я заплатил бы за работу по ремонту телевизора только если бы он стал работать. Он не работает. Поэтому я платить не буду

А - я заплатил

В - он стал работать

$$A \rightarrow B, \neg B \models \neg A \text{ (modus tollens)}$$

Доказательство: $A=1$ (следствие), тогда по первой посылке $B=1$, но $B=0$ по второй посылке. Получено противоречие.

- Если бы он не сказал, она ни за что не узнала бы. А не спроси она его, он бы и не сказал ей. Но она узнала. Значит, она спросила.

С – он сказал

У – она узнала

В – она спросила его

$$\neg C \rightarrow \neg U^{(1)}, \neg V \rightarrow \neg C^{(2)}, U^{(3)} \models V^{(4)}$$

- Доказательство: $V=0(4), C=0(2), U=0(1)$, но $U=1(3)$. Получено противоречие.

Примеры нелогичных и неполных задач

Утверждение нелогично:

- Он сказал, что придет, если не будет дождя. Идет дождь. Значит, он не придет.

П – он придёт, Д-будет дождь,

$$\neg D \rightarrow P, D \models \neg P$$

Опровержение: $P=1, D=1$, из $0 \rightarrow 1, 1 \models 0$ что неверно

Недосказанности (неполные суждения, энтимемы)

- Если я выпью кофе, я не смогу заснуть. Поэтому, с вашего позволения, я не стану пить кофе.

К - я выпью кофе, З – я засну

$$K \rightarrow \neg Z \models \neg K \text{ – нелогично,}$$

- пропущена посылка я хочу заснуть : $K \rightarrow \neg Z, Z \models \neg K$

Теория доказательств

Аксиомы

1. $A \rightarrow (B \rightarrow A)$
2. $(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$
3. $A \rightarrow (B \rightarrow (A \& B))$
4. $(A \& B) \rightarrow A$ $(A \& B) \rightarrow B$
5. $A \rightarrow (A \vee B)$ $B \rightarrow (B \vee A)$
6. $A \rightarrow C \rightarrow ((B \rightarrow C) \rightarrow (A \vee B \rightarrow C))$
7. $(A \rightarrow B) \rightarrow ((A \rightarrow \neg B) \rightarrow \neg A)$
8. $(\neg \neg A) \rightarrow A$
9. $(A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow (A \Leftrightarrow B))$
10. $(A \Leftrightarrow B) \rightarrow (B \rightarrow A)$ $(A \Leftrightarrow B) \rightarrow (A \rightarrow B)$

Правило вывода (modus ponens)

Если $\vdash A$ и $\vdash A \rightarrow B$ то $\vdash B$

Пример задачи: $\vdash A \rightarrow A$

Решение:

1. $A \rightarrow (A \rightarrow A)$ (по акс. 1)
2. $A \rightarrow (A \rightarrow A) \rightarrow ((A \rightarrow (A \rightarrow A) \rightarrow A)) \rightarrow (A \rightarrow A)$
(по акс. 2)
3. $((A \rightarrow (A \rightarrow A) \rightarrow A)) \rightarrow (A \rightarrow A)$ (*m.p.* 1,2)
4. $A \rightarrow ((A \rightarrow A) \rightarrow A)$ (по акс. 1)
5. $A \rightarrow A$ (*m.p.* 4,3)

Пример задачи (2)

Дано: $A \rightarrow (B \rightarrow C)$, $A \& B \vdash C$

1. $A \rightarrow (B \rightarrow C)$ (дано)
2. $A \& B$ (дано)
3. $A \& B \rightarrow A$ (по акс.4)
4. A (т.р. 2,3)
5. $B \rightarrow C$ (т.р. 4,1)
6. $A \& B \rightarrow B$ (по акс.4)
7. B (т.р. 2,6)
8. C (т.р. 7,5)

Теорема о дедукции (Теорема Эрбрана)

- Если $A \vdash B$, то $\vdash A \rightarrow B$
- Если $A_1, A_2 \dots A_m \vdash B$ то $A_1, A_2 \dots A_{m-1} \vdash A_m \rightarrow B$

Следствия

$A_1, A_2 \dots A_m \vdash B$ тогда и только тогда

- $\vdash A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow (A_m \rightarrow B) \dots)$
- $\vdash A_1 \& A_2 \& \dots \& A_m \rightarrow B$

Связь теории доказательств и теории моделей

- Теорема: Всякая доказуемая формула общезначима (является тавтологией)

Если $\vdash E$ то $\models E$

Следствие(непротиворечивость)

- Не существует такой V что доказуемы V и $\neg V$
- Ни для какой V не выполняется одновременно

$\vdash V$ и $\vdash \neg V$

- Если $\not\models E$ то $\nvdash E$

Проблемы логики высказываний

- Все люди смертны (А), Сократ - человек (В), следовательно Сократ смертен (С)
- $A, B \vdash C$?
- Нельзя доказать пользуясь инструментом логики высказываний
- Логика высказываний – логика нулевого порядка

Логика первого порядка (логика предикатов)

- Пропозициональная функция — это функция областью значений которой являются высказывания.
- Предикатом будем называть пропозициональную функцию $P(x_1, x_2, \dots, x_n)$ с любым числом независимых переменных $n \geq 0$
- x_i — объект
- Сократ является человеком.
- _____ является человеком
- x является человеком
- Предикат — сказуемое. Сказуемое — это главный член предложения, обозначающий действие, состояние или признак подлежащего и отвечающий на вопросы: что делает предмет? что с ним происходит? каков предмет? что он такое? кто он таков?

Связь логики первого порядка и теории множеств

- $n=0$ P – высказывание
- $n=1$ $P(x)$ – свойство, аналог принадлежности множеству: $x \in A$ или (исключающее) $x \notin A$
- $n=2$ $P(x,y)$ – бинарное отношение $R \subseteq X \times Y$

Примеры предикатов

- « x – простое число»,
- « x – четное число»,
- « x больше 10»,
- « x больше y »,
- « x делит y нацело»,
- « x плюс y равно 10»

Квантор существования

- Пусть дано выражение:
«существует x такой, что $x+y=10$ ».
- На множестве натуральных чисел это предложение определяет одноместный предикат $P(y)$, так $P(2)$ и $P(9)$ – истинные высказывания, $P(11)$ – ложное.
- Обозначим предикат « $x+y=10$ » через $S(x,y)$
- $P(y)$ можно записать так:
«существует x такой, что $S(x,y)$ ».
- $P(y)$ получен из предиката $S(x,y)$ навешиванием *квантора существования* на x :
$$P(y)=\exists x S(x,y).$$

Квантор общности

- $Q(y)$: «для всех x справедливо, что $y^3 \geq x^2$ »
- Если предикат « $y^3 \geq x^2$ » обозначить через $T(x, y)$, то $Q(y)$ можно записать так:

«для всех x справедливо $T(x, y)$ ».

- Предикат $Q(y)$ получен из предиката $T(x, y)$ навешиванием квантора общности на x :

$$Q(y) = \forall x T(x, y).$$

Определения кванторов

- **Определение.** Пусть $P(x_1, \dots, x_n)$ – предикат, заданный на множестве M , y – переменная. Тогда выражение «для всякого y выполняется $P(x_1, \dots, x_n)$ » – предикат, полученный из P навешиванием квантора общности на переменную y , а выражение «существует y такой, что выполняется $P(x_1, \dots, x_n)$ » – предикат, полученный из P навешиванием квантора существования на переменную y .
- В определении не требуется, чтобы y была одна из переменных $x_1, \dots, x_n$, хотя в содержательных примерах, квантор навешивается на одну из переменных $x_1, \dots, x_n$. Указанное требование не накладывается, чтобы избежать усложнения определения формулы логики предикатов. Если y – одна из переменных $x_1, \dots, x_n$, то после навешивания квантора на y новый предикат является $(n-1)$ -местным, если $y \notin \{x_1, \dots, x_n\}$, то местность нового предиката равна n .

Связь предиката с функциями и отношениями

- n -местная функция может рассматриваться как $(n+1)$ -местный предикат:
 - функции $y=f(x_1, \dots, x_n)$, заданной на множестве M можно поставить в соответствие выражение « y равно $f(x_1, \dots, x_n)$ »: $P(x_1, \dots, x_n, y)$;
 - если элемент b есть значение функции в точке $(a_1, \dots, a_n)$, то высказывание $P(a_1, \dots, a_n, b)$ истинно.
- Предикат можно представить отношением:
 - Пусть $P(x_1, \dots, x_n)$ задан на множестве M : $x_i \in M$
 - $M^n = M \times M \times \dots \times M$ и подмножество D_p множества M^n , определяемое равенством:
 - $D_p = \{(a_1, \dots, a_n) \in M^n \mid \text{высказывание } P(a_1, \dots, a_n) \text{ истинно}\}$.

Примеры предикатов

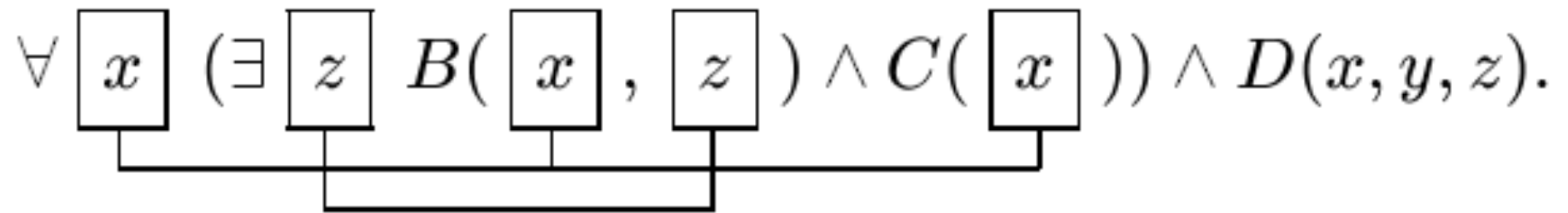
$L(x,y)$ – x любит y

- | | |
|---|--|
| • Кто-то любит Машу | $\exists x L(x, \text{Маша})$ |
| • Есть некто кто любит Машу | $\exists x L(x, \text{Маша})$ |
| • Все любят Машу | $\forall x L(x, \text{Маша})$ |
| • Никто не любит Машу | $\forall x \neg L(x, \text{Маша})$
$\neg \exists x L(x, \text{Маша})$ |
| • Каждый кого-нибудь любит | $\forall x \exists y L(x, y)$ |
| • Кого-то любят все | $\exists y \forall x L(x, y)$ |
| • Всяк любит себя | $\forall x L(x, x)$ |
| • Не существует никого кто бы не любил себя | $\neg \exists x \neg L(x, x)$ |

Область действия квантора

- Область действия квантора — наивысший приоритет
$$\forall x A(x) \rightarrow B(x) = (\forall x A(x)) \rightarrow B(x)$$
- Первая переменная x — связанная, вторая — переменная x — свободная.
- Свободная переменная может принимать любое значение. Значением связанной переменной руководит квантор.
- Квантор связывает ту переменную, которая идёт за ним
- Связанное вхождение - всякое вхождение переменной, связанной квантором.
- Все вхождения переменной, лежащие вне области действия квантора, связывающего переменную называются свободными

Область действия квантора (2)



Пример про Сократа

- Все люди смертны

$$\forall x (\text{Человек}(x) \rightarrow \text{Смертен}(x))$$

- Сократ - человек

$$\text{Человек}(\text{Сократ})$$

- следовательно Сократ смертен

$$\text{Смертен}(\text{Сократ})$$

Интерпретация в логике первого порядка

- Интерпретация формулы F логики первого порядка состоит из непустой (предметной) области D и указания «оценки» (значения) всех констант, функциональных символов и предикатных символов, встречающихся в F :
- Пример интерпретации:
 $D=\{1,2\}, P(1)=\text{true}, P(2)=\text{false}$
- $\forall x P(x) = ? \exists x \neg P(x) = ?$

Пример интерпретации

x	y	P(x,y)
1	1	true
1	2	false
2	1	false
2	2	true

$$\forall x \exists y P(x,y) = ?$$

$$\exists y \forall x P(x,y) = ?$$

Пример интерпретации (2)

x	f(x)
1	2
2	1

x	P(x)
1	false
2	true

x	y	Q(x,y)
1	1	true
1	2	true
2	1	false
2	2	true

a=1

$\forall x(P(x) \rightarrow Q(f(x),a))=?$

$\forall x\exists y(P(x)\&Q(x,y))=?$

Свойства формул

- Формула G непротиворечива (выполнима) тогда и только тогда когда существует такая интерпретация I , что G имеет значение истина в I .
- Если формула G есть истина в интерпретации I , то мы говорим, что I есть модель формулы G и I удовлетворяет G .
- Формула G противоречива (невыполнима) тогда и только тогда когда не существует интерпретации которая удовлетворяет G .
- Формула G общезначима (тавтология) тогда и только тогда когда не существует такой интерпретации которая не удовлетворяет формуле G .
- Формулы $F(x_1, \dots, x_n)$ и $G(x_1, \dots, x_n)$ называются *равносильными(равными)*, если для любой интерпретации I с областью M высказывания $F(a_1, \dots, a_n)$ и $G(a_1, \dots, a_n)$ при любых $a_1, \dots, a_n$ из M одновременно истинны или одновременно ложны.

Законы логики предикатов

1. $(\forall x)(F(x) \& G(x)) = (\forall x)F(x) \& (\forall x)G(x)$
2. $(\exists x)(F(x) \vee G(x)) = (\exists x)F(x) \vee (\exists x)G(x)$
3. $(\forall x)(\forall y)F(x, y) = (\forall y)(\forall x)F(x, y)$
4. $(\exists x)(\exists y)F(x, y) = (\exists y)(\exists x)F(x, y)$
5. $\neg(\forall x)F(x) = (\exists x)\neg F(x)$
6. $\neg(\exists x)F(x) = (\forall x)\neg F(x)$
7. $(Q_1 x)(F(x) \vee G) = (Q_1 x)F(x) \vee G$
8. $(Q_1 x)(F(x) \& G) = (Q_1 x)F(x) \& G$, где G не содержит x
9. $(Q_1 x)(Q_2 u)(F(x) \vee G(u)) = (Q_1 x)F(x) \vee (Q_2 u)G(u)$
10. $(Q_1 x)(Q_2 u)(F(x) \& G(u)) = (Q_1 x)F(x) \& (Q_2 u)G(u)$

Логическое следствие в логике предикатов

- Формула $G(x_1, \dots, x_n)$ называется логическим следствием формул $F_1(x_1, \dots, x_n), \dots, F_k(x_1, \dots, x_n)$, если для любой интерпретации φ с областью M и любых $a_1, \dots, a_n \in M$ из истинности высказываний $(\varphi F_1)(a_1, \dots, a_n), \dots, (\varphi F_k)(a_1, \dots, a_n)$ следует истинность высказывания $(\varphi G)(a_1, \dots, a_n)$.
- **Теорема.** Формула $G(x_1, \dots, x_n)$ является логическим следствием формул $F_1(x_1, \dots, x_n), \dots, F_k(x_1, \dots, x_n)$ тогда и только тогда, когда множество формул $\{F_1(x_1, \dots, x_n), \dots, F_k(x_1, \dots, x_n), \neg G(x_1, \dots, x_n)\}$ невыполнимо.

Связь логики предикатов с естественным языком

- «Выгул кошек *и* собак воспрещен» —
$$\forall x((K(x) \vee C(x)) \rightarrow \neg B(x))$$
- Все ангелы имеют крылья.
- Некоторые ангелы имеют крылья.
- Некоторые ангелы крыльев не имеют.
- Некоторые политики мошенники. Неверно, что все политики, но некоторые политики мошенники.

Силлогизмы

Простой категорический силлогизм — рассуждение, состоящее из трёх простых атрибутивных высказываний: двух посылок и одного заключения.

- Общеутвердительным называется высказывание вида «Все предметы класса P суть предметы класса Q ». Обозначается буквой A .
- Общеотрицательным называется высказывание вида «Никакие предметы класса P не суть предметы класса Q ». Обозначается буквой E .
- Частноутвердительным называется высказывание вида «Некоторые предметы класса P суть предметы класса Q ». Обозначается буквой I .
- Частноотрицательным называется высказывание вида «Некоторые предметы класса P не суть предметы класса Q ». Обозначается буквой O .

Barbara

- Все животные смертны.
- Все люди — животные.
- Все люди смертны.

Cesare

- Ни одна здоровая еда не полнит.
- Все торты полнят.
- Ни один торт не здоровая еда.

Предварённая нормальная форма

- Формула G имеет *предваренную нормальную форму* (сокращенно: ПНФ), если

$$G = (Q_1, x_1) \dots (Q_n, x_n) H,$$

где $Q_1, \dots, Q_n$ кванторы, а формула H не содержит кванторов.

Для всякой формулы F существует формула G , равносильная F и имеющая предваренную нормальную форму.

Алгоритм приведения к предваренной нормальной форме

Шаг 1. Используя законы логики исключить эквиваленцию и импликацию.

Шаг 2. Занести отрицание к атомарным формулам.

Шаг 3. С помощью законов 1-2, 7-10 вынести кванторы вперед, используя при необходимости переименование связанных переменных.

Пример:

$$F = (\forall x)P(x) \rightarrow (\exists y)(\forall z)(P(y) \& Q(y, z)).$$

Выполнив шаг 1, получим формулу

$$F_1 = \neg(\forall x)P(x) \vee (\exists y)(\forall z)(P(y) \& Q(y, z)).$$

С помощью закона 5 перейдем к формуле

$$F_2 = (\exists x)\neg P(x) \vee (\exists y)(\forall z)(P(y) \& Q(y, z)).$$

Тем самым, шаг 2 также выполнен. Применим закон 9 слева направо $Q_1 = \exists$, $Q_2 = \exists$, $u = y$, получим формулу

$$F_3 = (\exists x)(\exists y)[\neg P(x) \vee (\forall z)(P(y) \& Q(y, z))].$$

(Пользуемся тем, что $\neg P(x)$ не содержит y , а $(\forall z)(P(y) \& Q(y, z))$ не содержит x . Так как формула $\neg P(x)$ не содержит z , то применение закона 7 слева направо дает формулу

$$F_4 = (\exists x)(\exists y)(\forall z)[\neg P(x) \vee (P(y) \& Q(y, z))].$$

Сколемовская нормальная форма

Определение. Формула G имеет *сколемовскую нормальную форму* (сокращенно: СНФ), если

$$G = (\forall x) \dots (\forall x_n) H,$$

где формула H не содержит кванторов и имеет конъюнктивную нормальную форму (КНФ).

Например, формула $(\forall x)[P(x) \vee (P(y) \& Q(x, y))]$ имеет сколемовскую нормальную форму, а формулы $(\forall x)(\exists y)Q(x, y)$, $(\forall x)[P(x) \& (P(y) \vee Q(x, y))]$ не имеют.

Алгоритм приведения к сколемовской нормальной форме

Шаг 1. Привести к ПНФ.

Шаг 2. Бескванторную часть привести к конъюнктивной нормальной форме.

Шаг 3. Исключить кванторы существования.

Исключение кванторов

Пусть дана формула

$$F=(\exists x)(\forall y)(\exists z)(\forall u)(\exists v)H(x,y,z,u,v),$$

где H – не содержит кванторов. Предположим, что формула не содержит константы c , символов одноместной функции f и двухместной функции g . Тогда в формуле H заменим x на c , z – на $f(y)$, v заменим на $g(y,u)$. Все кванторы существования вычеркнем. Получим формулу

$$G=(\forall y)(\forall u)H(c,y,f(y),u,g(y,u)).$$

Это и есть результат выполнения шага 3.

Функции $f(y)$, $g(y,u)$ называют сколемовскими функциями.

В общем виде на шаге 3 переменная с квантором существования заменяется функцией аргументы которой – это все переменные расположенные *слева от исходной переменной с кванторами общности*. Если таких переменных нет, то вместо функции замена происходит на константу.

Пример приведения к СНФ

Пусть

$$F = (\exists x)(\forall y)[P(x,y) \rightarrow (\exists z)(Q(x,z) \& R(y))].$$

Применяя закон 2, получаем формулу

$$F_1 = (\exists x)(\forall y)(\exists z)[\neg P(x,y) \vee (Q(x,z) \& R(y))].$$

Она имеет предваренную нормальную форму. Приводим бескванторную часть к КНФ:

$$F_2 = (\exists x)(\forall y)(\exists z)[(\neg P(x,y) \vee Q(x,z)) \& (\neg P(x,y) \vee R(y))].$$

Сделаем подстановку $x=a$, $z=f(y)$, получим искомую формулу

$$G = (\forall y)[(\neg P(a,y) \vee Q(a, f(y))) \& (\neg P(a,y) \vee R(y))].$$

Эрбрановский универсум

Определение. Пусть H_0 — множество констант, встречающихся в S . Если никакая константа не встречается в S , то H_0 состоит из одной константы, скажем $H_0 = \{a\}$. Для $i = 0, 1, 2, \dots$ пусть H_{i+1} есть объединение H_i и множества всех термов вида $f^n(t_1, \dots, t_n)$ (при всех n) для всех функций f^n , встречающихся в S , где t_j ($j = 1, \dots, n$) принадлежит H_j . Тогда каждое H_i называется *множеством констант i -го уровня* для S и H_∞ называется *эрбрановским универсумом S* .

Пример 4.4. Пусть $S = \{P(a), \sim P(x) \vee P(f(x))\}$: Тогда

$$H_0 = \{a\};$$

$$H_1 = \{a, f(a)\};$$

$$H_2 = \{a, f(a), f(f(a))\};$$

$$\dots$$

$$H_\infty = \{a, f(a), f(f(a)), f(f(f(a))), \dots\}.$$

Эрбрановский универсум (пример)

Пример 4.5. Пусть $S = \{P(x) \vee Q(x), R(z), T(y) \vee \sim W(y)\}$. Так как не существует никаких констант в S , положим $H_0 = \{a\}$. Не существует никаких функциональных символов в S , следовательно, $H = H_0 = H_1 = \dots = \{a\}$.

Пример 4.6. Пусть $S = \{P(f(x), a, g(y), b)\}$. Тогда

$$H_0 = \{a, b\};$$

$$H_1 = \{a, b, f(a), f(b), g(a), g(b)\};$$

$$H_2 = \{a, b, f(a), f(b), g(a), g(b), f(f(a)), f(f(b)), f(g(a)), f(g(b)), g(f(a)), g(f(b)), g(g(a)), g(g(b))\};$$

.....

Основной пример

Определение. Пусть S есть множество дизъюнктов. Тогда множество основных атомов вида $P^n(t_1, \dots, t_n)$ для всех n -местных предикатов P^n , встречающихся в S , где $t_1, \dots, t_n$ — элементы эрбрановского универсума S , называется *множеством атомов* множества S или *эрбрановским базисом* S .

Определение. Основной пример дизъюнкта C множества дизъюнктов S есть дизъюнкт, полученный заменой переменных в C на члены эрбрановского универсума S .

Пример 4.7. Пусть $S = \{P(x), Q(f(y)) \vee R(y)\}$. $C = P(x)$ — дизъюнкт в S и $H = \{a, f(a), f(f(a)), \dots\}$ — эрбрановский универсум S . Тогда $P(a)$ и $P(f(f(a)))$ суть основные примеры C .

Н интерпретация

...многозначная интерпретация множества S .

Определение. Пусть S —множество дизъюнктов, H —эрбрановский универсум S и I —интерпретация S над H . Говорим, что I есть H -интерпретация множества S , если она удовлетворяет следующим условиям:

1. I отображает все константы из S в самих себя.
2. Пусть f есть n -местный функциональный символ и $h_1, \dots, h_n$ —элементы H . В I через f обозначается функция, которая отображает $(h_1, \dots, h_n)$ (элемент из H^n) в $f(h_1, \dots, h_n)$ (элемент из H).

При этом не возникает никаких ограничений при придании значения любому n -местному предикатному символу в S . Пусть $A = \{A_1, A_2, \dots, A_n, \dots\}$ —эрбрановский базис множества S . H -интерпретацию I удобно представлять в виде

$$I = \{m_1, m_2, \dots, m_n, \dots\},$$

где m_j есть или A_j или $\sim A_j$ для $j = 1, 2, \dots$. Смысл этого множества в том, что если m_j есть A_j , то атому A_j присвоено значение «истинно», в противном случае—значение «ложно».

Пример H интерпретации

Пример 4.8. Рассмотрим множество $S = \{P(x) \vee Q(x), R(f(y))\}$. Эрбрановский универсум H для S есть $H = \{a, f(a), f(f(a)), \dots\}$. В S входят три предикатных символа: P , Q и R . Следовательно, эрбрановский базис S есть

$$A = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \dots\}.$$

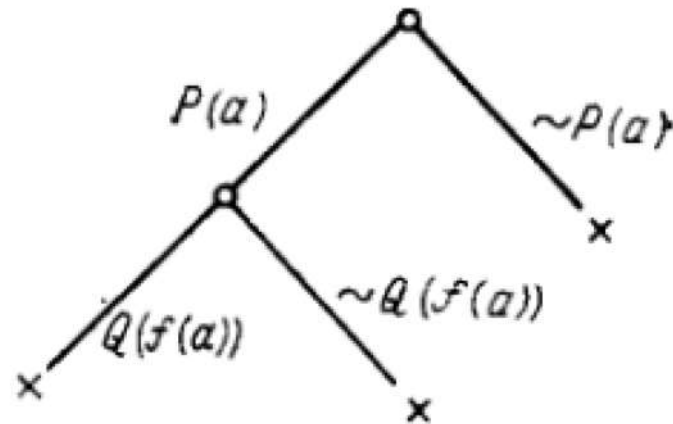
Некоторые H -интерпретации множества S суть

$$\begin{aligned} I_1 &= \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \dots\}, \\ I_2^* &= \{\sim P(a), \sim Q(a), \sim R(a), \sim P(f(a)), \sim Q(f(a)), \\ &\quad \sim R(f(a)), \dots\}, \\ I_3^* &= \{P(a), Q(a), \sim R(a), P(f(a)), Q(f(a)), \sim R(f(a)), \dots\}. \end{aligned}$$

Семантическое дерево

Пример 4.14. Рассмотрим $S = \{P(x), \sim P(x) \vee Q(f(x)), \sim Q(f(a))\}$. Эрбрановский базис множества S есть

$$A = \{P(a), Q(a), P(f(a)), Q(f(a)), \dots\}.$$



Теорема Эрбрана

Теорема 4.3 (теорема Эрбрана, вариант I). Множество дизъюнктов S невыполнимо тогда и только тогда, когда любому полному семантическому дереву S соответствует конечное замкнутое семантическое дерево¹).

Теорема 4.4 (теорема Эрбрана, вариант II). Множество дизъюнктов S невыполнимо тогда и только тогда, когда существует конечное невыполнимое множество S' основных примеров дизъюнктов S .

Пример применения теоремы Эрбрана

$$S = \{P(a), \sim P(x) \vee Q(f(x)), \sim Q(f(a))\};$$

$$H_0 = \{a\};$$

$$S'_0 = P(a) \wedge (\sim P(a) \vee Q(f(a))) \wedge \sim Q(f(a))$$

$$= (P(a) \wedge \sim P(a) \wedge \sim Q(f(a))) \vee (P(a) \wedge Q(f(a)) \wedge \sim Q(f(a)))$$

$$= \square \vee \square = \square.$$

Метод резолюций

Для любых двух дизъюнктов C_1 и C_2 , если существует литера L_1 в C_1 , которая контрарна литере L_2 в C_2 , то, вычеркнув L_1 и L_2 из C_1 и C_2 соответственно, построим дизъюнкцию оставшихся дизъюнктов. Построенный дизъюнкт есть резольвента C_1 и C_2 .

Пример 5.1. Рассмотрим следующие дизъюнкты:

$$C_1: P \vee R,$$

$$C_2: \sim P \vee Q.$$

Дизъюнкт C_1 имеет литеру P , которая контрарна литере $\sim P$ в C_2 . Следовательно, вычеркивая P и $\sim P$ из C_1 и C_2 соответственно и составляя дизъюнкцию оставшихся дизъюнктов R и Q , мы получим резольвенту $R \vee Q$.

Метод резолюций (2)

Важным свойством резольвенты является то, что любая резольвента двух дизъюнктов C_1 и C_2 есть логическое следствие C_1 и C_2 . Это устанавливается в следующей теореме.

Теорема 5.1. Пусть даны два дизъюнкта C_1 и C_2 . Тогда резольвента C дизъюнктов C_1 и C_2 есть логическое следствие C_1 и C_2 .

Определение. Пусть S — множество дизъюнктов. Резолютивный вывод C из S есть такая конечная последовательность $C_1, C_2, \dots, C_k$ дизъюнктов, что каждый C_i или принадлежит S или является резольвентой дизъюнктов, предшествующих C_i , и $C_k = C$. Вывод $\square$ из S называется опровержением (или доказательством невыполнимости) S .

Унификация

$$\begin{array}{ll} C_1: & P(x) \vee Q(x), & C'_1: & P(f(a)) \vee Q(f(a)), \\ C_2: & \sim P(f(x)) \vee R(x). & C'_2: & \sim P(f(a)) \vee R(a). \\ & & C'_3: & Q(f(a)) \vee R(a). \end{array}$$

$$C_1^*: P(f(x)) \vee Q(f(x)).$$

$$C_3: Q(f(x)) \vee R(x).$$

Определение. *Подстановка* — это конечное множество вида $\{t_1/v_1, \dots, t_n/v_n\}$, где каждая v_i — переменная, каждый t_i — терм, отличный от v_i , все v_i различны.

Подстановка

Определение. Пусть $\theta = \{t_1/v_1, \dots, t_n/v_n\}$ — подстановка и E — выражение. Тогда $E\theta$ — выражение, полученное из E заменой одновременно всех вхождений переменной v_i ($1 \leq i \leq n$) в E на терм t_i . $E\theta$ называется *примером* E . Заметим, что определение примера совместимо с определением основного примера дизъюнкта, данным в главе 4.

Пример 5.7. Пусть $\theta = \{a/x, f(b)/y, c/z\}$ и $E = P(x, y, z)$. Тогда $E\theta = P(a, f(b), c)$.

Определение. Пусть $\theta = \{t_1/x_1, \dots, t_n/x_n\}$ и $\lambda = \{u_1/y_1, \dots, u_m/y_m\}$ — две подстановки. Тогда *композиция* θ и λ есть подстановка (обозначим $\theta \circ \lambda$), которая получается из множества

$$\{t_1\lambda/x_1, \dots, t_n\lambda/x_n, u_1/y_1, \dots, u_m/y_m\}$$

вычеркиванием всех элементов $t_j\lambda/x_j$, для которых $t_j\lambda = x_j$, и всех элементов u_i/y_i таких, что $y_i \in \{x_1, x_2, \dots, x_n\}$.

Пример композиции

Пример 5.8. Пусть $\mathcal{A} = \langle A, \{f\} \rangle$ и $\mathcal{B} = \langle B, \{g\} \rangle$ — алгебры.

$$\theta = \{t_1/x_1, t_2/x_2\} = \{f(y)/x, z/y\},$$

$$\lambda = \{u_1/y_1, u_2/y_2, u_3/y_3\} = \{a/x, b/y, y/z\}.$$

Тогда

$$\{t_1\lambda/x_1, t_2\lambda/x_2, u_1/y_1, u_2/y_2, u_3/y_3\} = \{f(b)/x, y/y, a/x, b/y, y/z\}.$$

Унификатор

О п р е д е л е н и е. Подстановка θ называется *унификатором* для множества $\{E_1, \dots, E_k\}$ тогда и только тогда, когда $E_1\theta = E_2\theta = \dots = E_k\theta$. Говорят, что множество $\{E_1, \dots, E_k\}$ *унифицируемо*, если для него существует унификатор.

О п р е д е л е н и е. Унификатор σ для множества выражений $\{E_1, \dots, E_k\}$ будет *наиболее общим* унификатором тогда и только тогда, когда для каждого унификатора θ для этого множества существует такая подстановка λ , что $\theta = \sigma \circ \lambda$.

Алгоритм унификации

О п р е д е л е н и е. Множество рассогласований непустого множества выражений W получается выявлением первой (слева) позиции, на которой не для всех выражений из W стоит один и тот же символ, и затем выписыванием из каждого выражения в W подвыражения, которое начинается с символа, занимающего эту позицию. Множество этих подвыражений и есть *множество рассогласований* в W .

Алгоритм унификации

Шаг 1. Множество $k=0$, $W_k=W$ и $\sigma_k=\varepsilon$.

Шаг 2. Если W_k — единичный дизъюнкт, то остановка: σ_k — наиболее общий унификатор для W . В противном случае найдем множество D_k рассогласований для W_k .

Шаг 3. Если существуют такие элементы v_k и t_k в D_k , что v_k — переменная, не входящая в t_k , то перейти к шагу 4. В противном случае остановка: W не унифицируемо.

Шаг 4. Пусть $\sigma_{k+1} = \sigma_k \{t_k/v_k\}$ и $W_{k+1} = W_k \{t_k/v_k\}$. (Заметим, что $W_{k+1} = W \sigma_{k+1}$.)

Шаг 5. Присвоить значение $k+1$ и перейти к шагу 2.

Пример унификации

Пример 5.11. Найти наиболее общий унификатор для

$$W = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}.$$

1. $\sigma_0 = \varepsilon$ и $W_0 = W$. Так как W_0 — не единичный дизъюнкт, то σ_0 не является наиболее общим унификатором для W .

2. Множество рассогласований $D_0 = \{a, z\}$. В D_0 существует переменная $v_0 = z$, которая не встречается в $t_0 = a$.

3. Пусть

$$\begin{aligned}\sigma_1 &= \sigma_0 \circ \{t_0/v_0\} = \varepsilon \circ \{a/z\} = \{a/z\}, \\ W_1 &= W_0 \{t_0/v_0\} \\ &= \{P(a, x, f(g(y))), P(z, f(z), f(u))\} \{a/z\} \\ &= \{P(a, x, f(g(y))), P(a, f(a), f(u))\}.\end{aligned}$$

4. W_1 — не единичный дизъюнкт, так как нашли множество рассогласований D_1 для W_1 :

$$D_1 = \{x, f(a)\}.$$

Пример (2)

5. Из D_1 мы найдем, что $v_1 = x$ и $t_1 = f(a)$.

6. Пусть

$$\begin{aligned}\sigma_2 &= \sigma_1 \circ \{t_1/v_1\} = \{a/z\} \circ \{f(a)/x\} = \{a/z, f(a)/x\}, \\ W_2 &= W_1 \{t_1/v_1\} \\ &= \{P(a, x, f(g(y))), P(a, f(a), f(u))\} \{f(a)/x\} \\ &= \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}.\end{aligned}$$

7. W_2 — не единичный дизъюнкт, так как мы нашли множество рассогласований D_2 для W_2 : $D_2 = \{g(y), u\}$. Из D_2 мы найдем, что $v_2 = u$ и $t_2 = g(y)$.

8. Пусть

$$\begin{aligned}\sigma_3 &= \sigma_2 \circ \{t_2/v_2\} = \{a/z, f(a)/x\} \circ \{g(y)/u\} = \\ &= \{a/z, f(a)/x, g(y)/u\}, \\ W_3 &= W_2 \{t_2/v_2\} \\ &= \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\} \{g(y)/u\} \\ &= \{P(a, f(a), f(g(y))), P(a, f(a), f(g(y)))\} \\ &= \{P(a, f(a), f(g(y)))\}.\end{aligned}$$

9. Так как W_3 — единичный дизъюнкт, то $\sigma_3 = \{a/z, f(a)/x, g(y)/u\}$ есть наиболее общий унификатор для W .

Метод резолюций в логике предикатов

Определение. Пусть C_1 и C_2 — два дизъюнкта (называемые *дизъюнктами-посылками*), которые не имеют никаких общих переменных. Пусть L_1 и L_2 — две литеры в C_1 и C_2 соответственно. Если L_1 и $\sim L_2$ имеют наиболее общий унификатор σ , то дизъюнкт

$$(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$$

называется (бинарной) *резольвентой* C_1 и C_2 . Литеры L_1 и L_2 называются *отрезаемыми литеры*.

Пример 5.14. Пусть $C_1 = P(x) \vee Q(x)$ и $C_2 = \sim P(a) \vee R(x)$. Так как x входит в C_1 и C_2 , то мы заменим переменную в C_2 , и пусть $C_2 = \sim P(a) \vee R(y)$. Выбираем $L_1 = P(x)$ и $L_2 = \sim P(a)$. Так как $\sim L_2 = P(a)$, то L_1 и $\sim L_2$ имеют наиболее общий унификатор $\sigma = \{a/x\}$. Следовательно,

$$\begin{aligned} & (C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma) \\ &= (\{P(a), Q(a)\} - \{P(a)\}) \cup (\{\sim P(a), R(y)\} - \{\sim P(a)\}) \\ &= \{Q(a)\} \cup \{R(y)\} = \{Q^*(a), R(y)\} = Q(a) \vee R(y). \end{aligned}$$

Таким образом, $Q(a) \vee R(y)$ — бинарная резольвента C_1 и C_2 . $P(x)$ и $\sim P(a)$ — отрезаемые литеры.

Пример доказательства

Пример 5.16. Показать, что внутренние накрест лежащие углы, образованные диагональю трапеции, равны.

Чтобы доказать эту теорему, мы сперва аксиоматизируем ее. Пусть $T(x, y, u, v)$ обозначает, что $xuyv$ — трапеция с верхней левой вершиной x , верхней правой вершиной y , нижней правой вершиной u и нижней левой вершиной v . Пусть $P(x, y, u, v)$ обозначает, что отрезок xu параллелен отрезку yv , и пусть

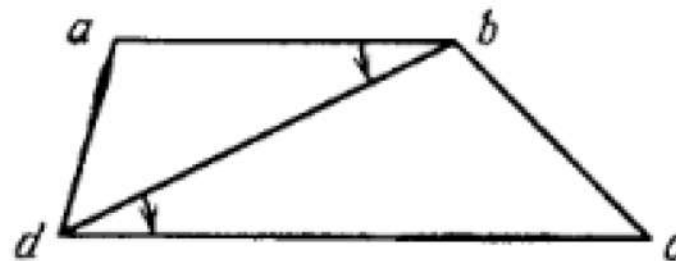


Рис. 7.

$E(x, y, z, u, v, w)$ означает, что угол xuz равен углу uvw . Тогда мы имеем следующие аксиомы:

Пример (2)

$A_1: (\forall x) (\forall y) (\forall u) (\forall v) [T(x, y, u, v) \rightarrow$ $\rightarrow P(x, y, u, v)]$	определение трапеции,
$A_2: (\forall x) (\forall y) (\forall u) (\forall v) [P(x, y, u, v) \rightarrow$ $\rightarrow E(x, y, v, u, v, y)]$	внутренние накрест лежащие углы для параллельных линий равны,
$A_3: T(a, b, c, d)$	дана на рис. 7.

Из этих аксиом мы должны уметь вывести, что $E(a, b, d, c, d, b)$ истинно, т. е.

$$A_1 \wedge A_2 \wedge A_3 \rightarrow E(a, b, d, c, d, b)$$

Пример (3)

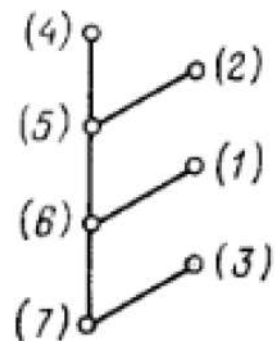
$$S = \{ \sim T(x, y, u, v) \vee P(x, y, u, v), \sim P(x, y, u, v) \vee \\ \vee E(x, y, v, u, v, y), T(a, b, c, d), \sim E(a, b, d, c, d, b) \}.$$

Эта стандартная форма S есть множество из четырех дизъюнктов. Теперь методом резолюций докажем, что множество S невыполнимо:

- | | |
|---|------------------------|
| (1) $\sim T(x, y, u, v) \vee P(x, y, u, v)$ | дизъюнкт в S , |
| (2) $\sim P(x, y, u, v) \vee E(x, y, v, u, v, y)$ | дизъюнкт в S , |
| (3) $T(a, b, c, d)$ | дизъюнкт в S , |
| (4) $\sim E(a, b, d, c, d, b)$ | дизъюнкт в S , |
| (5) $\sim P(a, b, c, d)$ | резольвента (2) и (4), |
| (6) $\sim T(a, b, c, d)$ | резольвента (1) и (5), |
| (7) $\square$ | резольвента (3) и (6). |

Пример (4)

- | | |
|---|------------------------|
| (1) $\sim T(x, y, u, v) \vee P(x, y, u, v)$ | дизъюнкт в S , |
| (2) $\sim P(x, y, u, v) \vee E(x, y, v, u, v, y)$ | дизъюнкт в S , |
| (3) $T(a, b, c, d)$ | дизъюнкт в S , |
| (4) $\sim E(a, b, d, c, d, b)$ | дизъюнкт в S , |
| (5) $\sim P(a, b, c, d)$ | резольвента (2) и (4), |
| (6) $\sim T(a, b, c, d)$ | резольвента (1) и (5), |
| (7) $\square$ | резольвента (3) и (6). |



Пример (5)

Пример 5.20. Рассмотрим следующее множество формул:

$$F_1: (\forall x)(C(x) \rightarrow (W(x) \wedge R(x))),$$

$$F_2: (\exists x)(C(x) \wedge O(x)),$$

$$G: (\exists x)(O(x) \wedge R(x)).$$

Наша задача — доказать, что G является логическим следствием F_1 и F_2 . Преобразуем F_1 , F_2 и $\sim G$ в стандартную форму и получим следующие пять дизъюнктов:

- $$\begin{array}{ll} (1) \sim C(x) \vee W(x), & \{ \\ (2) \sim C(x) \vee R(x) & \} \text{ из } F_1, \\ (3) C(a), & \{ \\ (4) O(a) & \} \text{ из } F_2, \\ (5) \sim O(x) \vee \sim R(x) & \text{из } \sim G. \end{array}$$

Это множество дизъюнктов невыполнимо. Это можно доказать при помощи метода резолюций следующим образом:

- $$\begin{array}{ll} (6) R(a) & \text{резольвента (3) и (2),} \\ (7) \sim R(a) & \text{резольвента (5) и (4),} \\ (8) \square & \text{резольвента (7) и (6).} \end{array}$$

Таким образом, G — логическое следствие F_1 и F_2 .

Теорема Черча

- Не существует алгоритма, который для любой формулы логики предикатов устанавливает, общезначима она или нет.

Алгоритм

- Алгоритм — это конечный набор правил, который определяет последовательность операций для решения конкретного множества задач и обладает пятью важными характеристиками:
 - конечность,
 - определённости,
 - ввод
 - вывод,
 - эффективность.

Другие определения

- Алгоритм — это всякая система вычислений, выполняемых по строго определённым правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи». (А.Н. Колмогоров)
- Алгоритм — это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к искомому результату (А.А. Марков)

Все ли задачи можно решить с помощью компьютера?

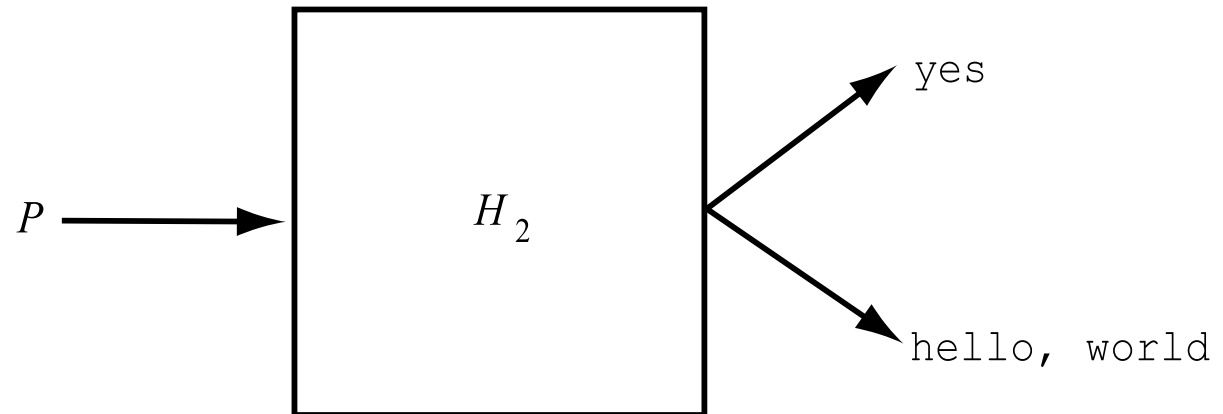
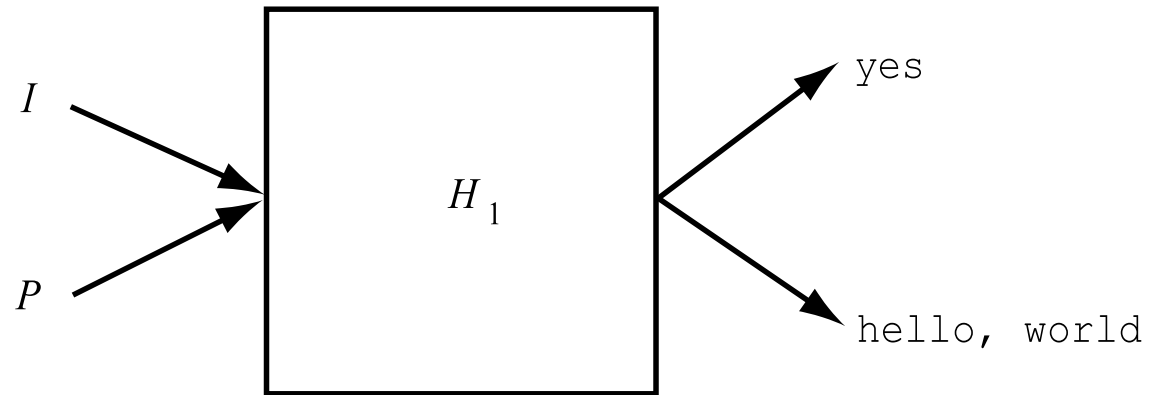
```
n=gets.to_i
t = 3
loop do
  for x in (1..t-2)
    for y in (1..t-x-1)
      z = t-x-y
      if x**n + y**n == z**n
        puts 'hello, world'
        exit
      end
    end
  end
  t += 1
end
```

- Напечатает ли данная программа hello, word?

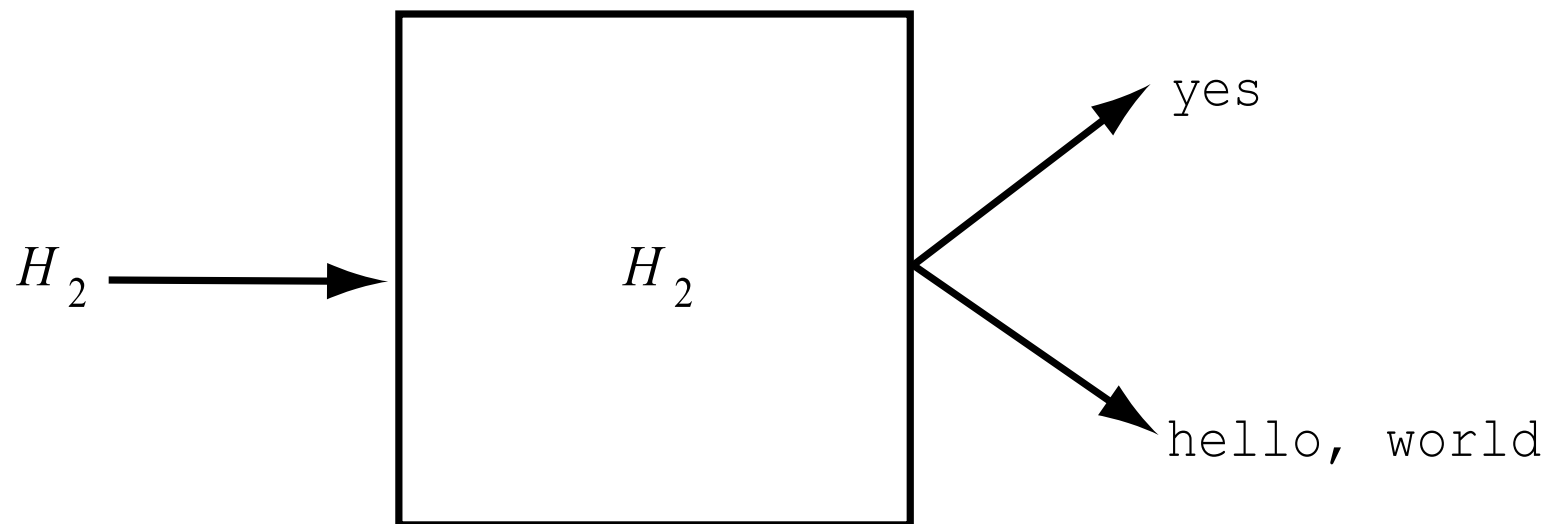
Доказательство Hello, world



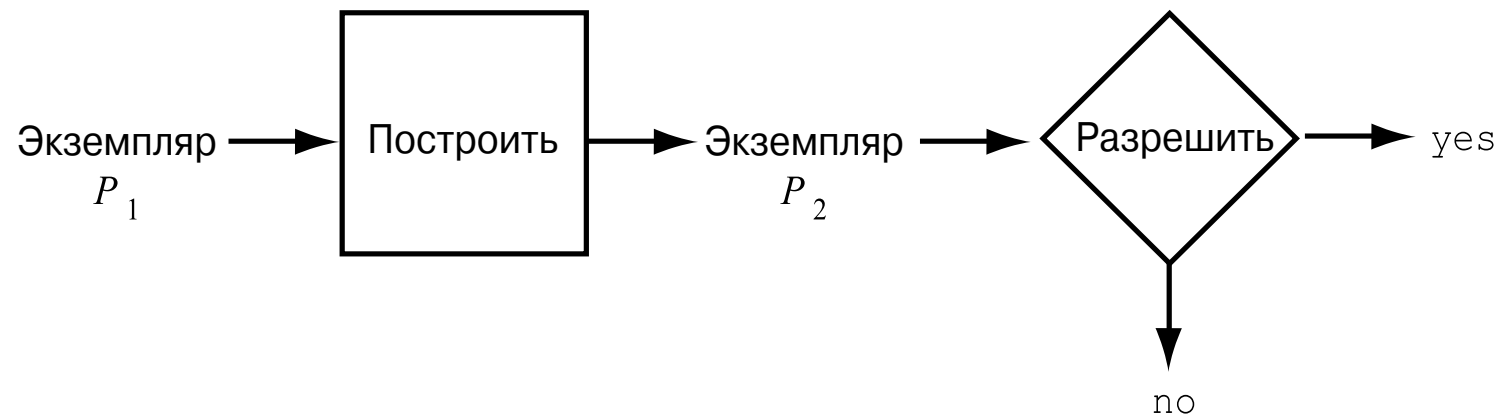
Hello, world (2)



Hello, world (3)



Сведение программы



Программа вызова функции foo()

Дана программа Q и ее вход u , необходимо построить программу R и ее вход z так, чтобы R со входом z вызывала foo тогда и только тогда, когда Q со входом u печатает hello, world:

1. Если Q содержит foo(), переименовать ее и все ее вызовы. Очевидно, новая программа Q_1 выполняет то же самое, что и Q .
2. Добавить в Q_1 функцию foo(). Эта функция не делает ничего и не вызывается. Полученная программа называется Q_2 .
3. Изменить Q_2 так, чтобы первые 12 символов ее печати запомнились в глобальном массиве A . Пусть полученная программа называется Q_3 .
4. Изменить Q_3 так, чтобы после каждого выполнения инструкции печати с использованием массива A проверялось, не образуют ли первые 12 напечатанных символов текст hello, world. В этом случае вызвать новую функцию foo, добавленную в п. 2. Полученная программа есть R , а ее вход z совпадает с u .

Проблема математического доказательства

- В начале XX века Д. Гильберт поставил вопрос о поисках алгоритма, который позволял бы определить истинность или ложность любого математического утверждения.
- В 1931 г. К. Гедель опубликовал свою знаменитую теорему о неполноте. Он доказал, что существует истинная формула первого порядка с целыми, которую нельзя ни доказать, ни опровергнуть в исчислении предикатов первого порядка над целыми.

Теорема о неполноте

- Перенумеруем все утверждения арифметики в лексикографическом порядке
- Пусть n -я (из пронумерованных выбранным способом строк символов) такая функция от аргумента ω обозначается:

$$P_n(\omega)$$

- Запишем предикат:

$$\neg \exists x [\Pi_x \text{ доказывает } P_\omega(\omega)] = P_k(\omega)$$

- Подставим $\omega=k$

$$\neg \exists x [\Pi_x \text{ доказывает } P_k(k)] = P_k(k)$$

Парадокс Рассела

- Назовем множество «обычным», если оно не является своим собственным элементом.
- Примером «необычного» множества является множество всех множеств, так как оно само является множеством, а следовательно, само является собственным элементом.
- Рассмотрим множество, состоящее только из всех «обычных» множеств – **«*расселовское* множество»**.

Содержит ли расселовское множество себя?

- Пусть в некой деревне живёт брадобрей, который бреет всех жителей деревни, которые не бреются сами, и только их. Бреет ли брадобрей сам себя?

Неразрешимые проблемы

- Проблема останова
- Не существует алгоритма, который узнавал бы по произвольному диофантову уравнению, имеет ли оно решения в целых числах или нет.
- Проблема соответствий Поста (ПСП):

Дано 2 списка

$$A = w_1, w_2, \dots, w_k \quad \text{и} \quad B = x_1, x_2, \dots, x_k$$

Существует ли последовательность индексов

$i_1, i_2, \dots, i_m$ такая, что

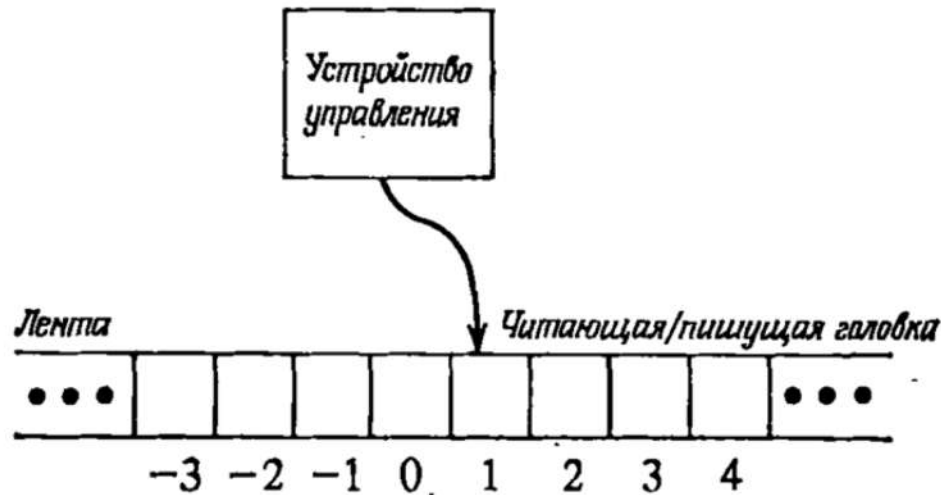
$$w_{i_1} w_{i_2} \dots w_{i_m} = x_{i_1} x_{i_2} \dots x_{i_m}$$

Пример ПСП

	Список A	Список B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

$$w_2 w_1 w_1 w_3 = x_2 x_1 x_1 x_3 = 101111110$$

Машина Тьюринга (ДМТ)



- Γ – множество ленточных символов,
- $\Sigma \subset \Gamma$ – входные символы,
- $b \in \Gamma \setminus \Sigma$ – пустой символ,
- Q – множество состояний. q_0 – начальное состояние, $\{q_y, q_n\}$ – конечные состояния
- Функция перехода:

$$\delta: (Q \setminus \{q_y, q_n\}) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 1\}$$

Пример машины Тьюринга

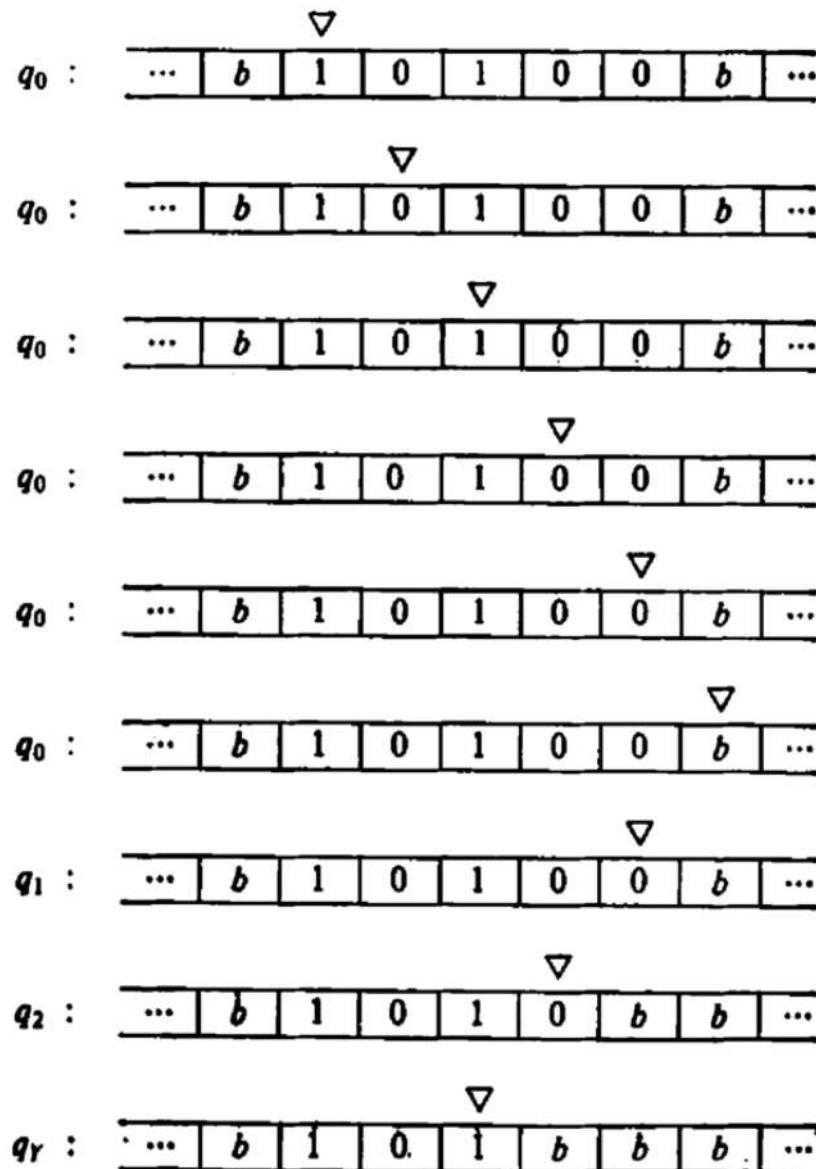
Делимость на 4

$$\Gamma = \{0, 1, b\}, \Sigma = \{0, 1\}$$

$$Q = \{q_0, q_1, q_2, q_3, q_Y, q_N\}$$

q	0	1	b
q_0	$(q_0, 0, +1)$	$(q_0, 1, +1)$	$(q_1, b, -1)$
q_1	$(q_2, b, -1)$	$(q_3, b, -1)$	$(q_N, b, -1)$
q_2	$(q_Y, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$
q_3	$(q_N, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$

$$\delta(q, s)$$



Рекурсивные функции

К числу базовых примитивно рекурсивных функций относятся функции следующих трёх видов:

- Нулевая функция O — функция без аргументов, всегда возвращающая 0 .
- Функция следования S одного переменного, сопоставляющая любому натуральному числу x непосредственно следующее за ним натуральное число $x + 1$.
- Функции I_n^m , где $0 < m \leq n$, от n переменных, сопоставляющие любому упорядоченному набору $x_1, \dots, x_n$ натуральных чисел число x_m из этого набора.

Операторы подстановки и примитивной рекурсии определяются следующим образом:

- **Оператор суперпозиции** (иногда — *оператор подстановки*). Пусть f — функция от m переменных, а $g_1, \dots, g_m$ — упорядоченный набор функций от n переменных каждая. Тогда результатом суперпозиции функций g_k в функцию f называется функция h от n переменных, сопоставляющая любому упорядоченному набору $x_1, \dots, x_n$ натуральных чисел число

$$h(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)) .$$

- **Оператор примитивной рекурсии**. Пусть f — функция от n переменных, а g — функция от $n + 2$ переменных. Тогда результатом применения оператора примитивной рекурсии к паре функций f и g называется функция h от $n + 1$ переменных вида

$$\begin{aligned} h(x_1, \dots, x_n, 0) &= f(x_1, \dots, x_n) ; \\ h(x_1, \dots, x_n, y + 1) &= g(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)) . \end{aligned}$$

Рекурсивные функции (2)

$$Sum(x, 0) = I_1^1(x);$$

$$Sum(x, y + 1) = F(x, y, Sum(x, y)) ;$$

$$F(x, y, z) = S(I_3^3(x, y, z)).$$

$$Mul(x, 0) = O(x) ;$$

$$Mul(x, y + 1) = G(x, y, Mul(x, y)) ;$$

$$G(x, y, z) = Sum(I_3^1(x, y, z), I_3^3(x, y, z)) .$$

Рекурсивные функции (3)

Частично рекурсивная функция определяется аналогично примитивно рекурсивной, только к двум операторам суперпозиции и примитивной рекурсии добавляется ещё третий оператор — минимизации аргумента.

- **Оператор минимизации аргумента.** Пусть f — функция от n натуральных переменных. Тогда результатом применения оператора минимума аргумента к функции f называется функция h от $n - 1$ переменных, задаваемая следующим определением:

$$h(x_1, \dots, x_{n-1}) = \min y, \text{ при условии } f(x_1, \dots, x_{n-1}, y) = 0$$

То есть функция h возвращает минимальное значение последнего аргумента функции f , при котором её значение равно 0.

Частично рекурсивные функции для некоторых значений аргумента могут быть не определены, так как оператор минимизации аргумента не всегда корректно определён, поскольку функция f может быть не равной нулю ни при каких значениях аргументов. С точки зрения императивного программирования, результатом частично рекурсивной функции может быть не только число, но и исключение или уход в бесконечный цикл, соответствующие неопределённому значению.

Общерекурсивная функция

Общерекурсивная функция — частично рекурсивная функция, определённая для всех значений аргументов. Задача определения того, является ли частично рекурсивная функция с данным описанием общерекурсивной или нет, алгоритмически неразрешима.

Тезис Черча - Тьюринга

Эвристическое утверждение:

- интуитивное понятие алгоритмической вычислимости

и строго формализованные понятия:

- частично рекурсивной функции,
- функции, вычислимой на машине Тьюринга

эквивалентны.

- Любая функция, которая может быть вычислена физическим устройством, может быть вычислена машиной Тьюринга
- Функции, вычисляемые посредством алгоритмов, являются частично рекурсивными

Временная сложность алгоритма

- n - объем входных данных. Размер цепочки символов в заданной схеме кодирования.
- $f(n)$ –временная сложность – затраты времени. Максимальное (во всем индивидуальным задачам) время, затрачиваемое алгоритмом на решение индивидуальной задачи длины n .
- Функция не будет определена полностью, пока не будет выбрана схема кодирования и вычислительное устройство. Но эти детали оказывают несущественное влияние на класс труднорешаемых задач.
- Будем считать число шагов машины Тьюринга выполненных до момента остановки

Полиномиальная и экспоненциальная сложность

- $f(n)$ есть $O(g(n))$, если $\exists c : |f(n)| \leq c|g(n)|$
- Полиномиальный алгоритм (алгоритм полиномиальной временной сложности) – это алгоритм для которого временная сложность есть:

$O(p(n))$, где $p(n)$ – некоторая полиномиальная функция, а n – входная длина.

- Алгоритмы не поддающиеся такой оценке будем называть *экспоненциальными* (трудноразрешимыми).

Сравнение сложности

Функция временной сложности	Размер n					
	10	20	30	40	50	60
n	0,00001 сек	0,00002 сек	0,00003 сек	0,00004 сек	0,00005 сек	0,00006 сек
n^2	0,0001 сек	0,0004 сек	0,0009 сек	0,0016 сек	0,0025 сек	0,0036 сек
n^3	0,001 сек	0,008 сек	0,027 сек	0,064 сек	0,125 сек	0,216 сек
n^5	0,1 сек	3,2 сек	24,3 сек	1,7 мин	5,2 мин	13,0 мин
2^n	0,001 сек	1,0 сек	17,9 мин	12,7 дней	35,7 лет	366 столетий
3^n	0,059 сек	58 мин	6,5 лет	3855 столетий	2×10^8 столетий	$1,3 \times 10^{13}$ столетий

Влияние быстродействия ЭВМ

*Размеры наибольшей задачи,
разрешимой за один час*

<i>Функция временной сложности</i>	<i>На современных ЭВМ</i>	<i>На ЭВМ, в 100 раз более быстрых</i>	<i>На ЭВМ, в 1000 раз более быстрых</i>
n	N_1	$100 N_1$	$1000 N_1$
n^2	N_2	$10 N_2$	$31,6 N_2$
n^3	N_3	$4,64 N_3$	$10 N_3$
n^5	N_4	$2,5 N_4$	$3,98 N_4$
2^n	N_5	$N_5 + 6,64$	$N_5 + 9,97$
3^n	N_6	$N_6 + 4,19$	$N_6 + 6,29$

Экспоненциальная сложность на практике

- Некоторые задачи имеют экспоненциальную сложность, но это сложность для *наихудшего* случая
- Во многих практических задачах эта сложность не проявляется
- Проблема – нет способа спрогнозировать как поведет себя экспоненциальный алгоритм на тех или иных данных

Теория NP-полных задач

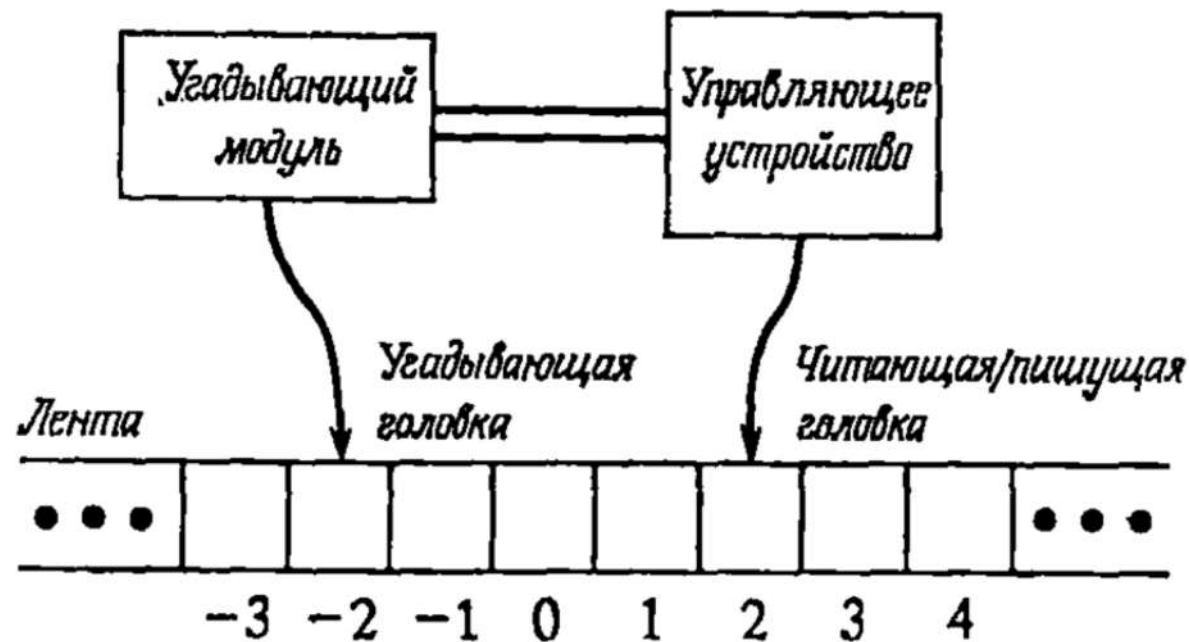
- Рассматриваются задачи распознавания:
 - Принадлежит ли цепочка символов заданному языку?
- Примеры задач распознавания:
 - Изоморфизм подграфу: Верно ли что граф G_1 содержит подграф изоморфный графу G_2 ?
 - Коммивояжер: Существует ли в графе путь через все города длина которого не превосходит B ?
- Задачи оптимизации не проще соответствующих задач распознавания.
- Многие задачи распознавания не проще соответствующих задач оптимизации

Задачи и языки

- Задача (массовая задача) P – общий вопрос, на который следует дать ответ. Задача содержит:
 - список параметров;
 - формулировка тех свойств, которым должен удовлетворять ответ.
- Индивидуальная задача I – получается из P путем приписывания всем параметрам конкретных значений.
- Задача распознавания свойств – P состоит из индивидуальных задач D_P и $Y_P \subseteq D_P$.
- Σ – алфавит. Σ^* – множество конкретных цепочек составленных из символа алфавита.
- Язык в алфавите Σ – это любое подмножество $L \subseteq \Sigma^*$.
- Задача P и схема кодирования e разбивают Σ^* на 3 класса:
 - слова не являющиеся кодами индивидуальных задач из P ;
 - слова являющиеся кодами индивидуальных задач из P с отрицательным ответом на вопрос;
 - слова являющиеся кодами индивидуальных задач из P с положительным ответом на вопрос (это язык L):

$$L[P, e] = \left\{ x \in \Sigma^*: \begin{array}{l} \text{есть алфавит схемы кодирования } e, \\ \text{а } x \text{ — код индивидуальной задачи} \\ I \in Y_P \text{ при схеме кодирования } e \end{array} \right\}.$$

Недетерминированная машина Тьюринга (НДМТ)

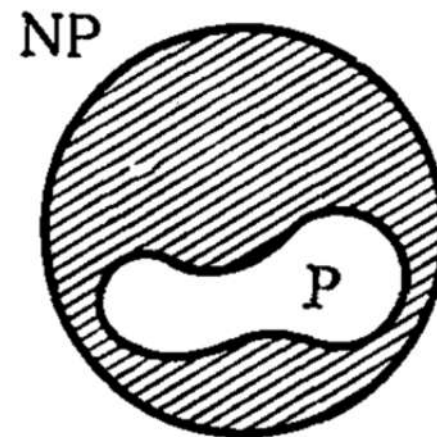


2 стадии:

- Угадывание. Угадывающий модуль записывает догадку
- Проверка. Проверяется догадка так же как и для детерминированной машины Тьюринга

Взаимоотношения классов P и NP

- Класс P – задачи, которые могут быть решены ДМТ за полиномиальное время
- Класс NP - задачи, которые могут быть решены НДМТ за полиномиальное время
- $P \subseteq NP$. Любая задача класса P может быть решена за полиномиальное время на НДМТ.
- $P \neq NP$?



Сводимость задач

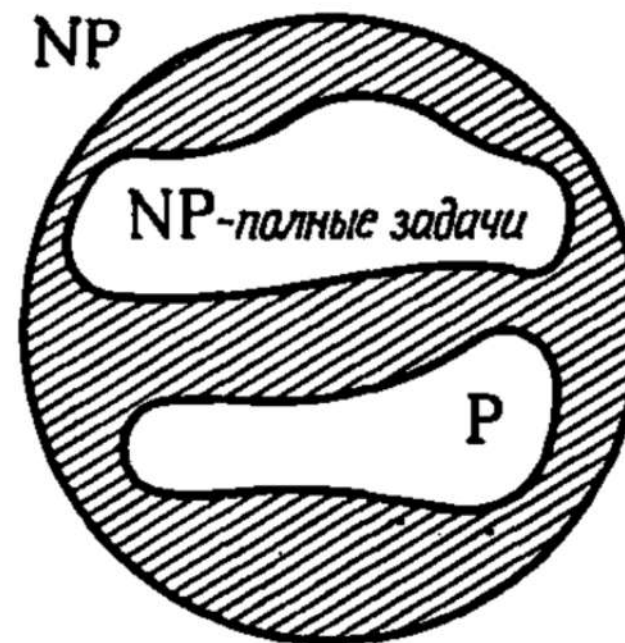
- Имеет место полиномиальная сводимость языка $L_1 \subseteq \Sigma_1^*$ к языку $L_2 \subseteq \Sigma_2^*$, если существует функция $f: \Sigma_1^* \rightarrow \Sigma_2^*$ удовлетворяющая условиям:
 - существует ДМТ вычисляющая f с полиномиальной временной сложностью;
 - $(\forall x \in \Sigma_1^*) x \in L_1 \iff f(x) \in L_2$
- $L_1 \propto L_2$
- Пример: Задача поиска гамильтонова цикла (ГЦ) сводится к задаче Коммивояжёр (КМ):
 - $G=(V,E)$, $|V|=m$,
 - $d(v_i, v_j)=1$ если $(v_i, v_j) \in E$, иначе $d(v_i, v_j)=2$
 - Существует ли путь длиной $\leq m$?
- $\text{ГЦ} \propto \text{КМ}$

NP-полные задачи

- Язык L называется NP-полным, если $L \in NP$ и любой другой язык $L' \in NP$ сводится к L :

$$L' \leq L$$

- Если хотя бы одна NP-полная задача может быть решена за полиномиальное время, то и все NP задачи могут быть решены за полиномиальное время.
- Если L_1 NP-полный язык, $L_2 \in NP$ и $L_1 \leq L_2$, то L_2 NP-полный язык
- Чтобы доказать, что P является NP-полной задачей достаточно доказать:
 - $P \in NP$
 - Какая-то одна известная NP-полная задача $P' \leq P$



Теорема Кука

- Задача выполнимости (ВЫП):
- Дано множество булевых переменных:
$$U = \{u_1, u_2, \dots, u_m\}$$
- Дана ФАЛ $f(u_1, u_2, \dots, u_m)$ в форме КНФ
- Вопрос: выполняема ли f ?
- Теорема Кука: ВЫП есть NP-полная задача

Доказательство теоремы Кука

- ВВП – принадлежит классу NP. Так как НДМТ достаточно указать набор значений $u_1, u_2, \dots, u_m$ и далее проверить $f(u_1, u_2, \dots, u_m) = 1$ за полиномиальное время
- Далее нужно показать $\forall L \in NP \propto L_{\text{ввп}}$:
- НДМТ M : $\{\Gamma, \Sigma, b, Q, q_0, q_y, q_n, \delta\}$
- n – размер входа
- $p(n)$ полином ограничивающий временную сложность $T_M(n)$
- Нужно найти ФАЛ f_L , реализующую полиномиальную сводимость

Переменные f_L

Переменная	Пределы изменения индексов	Придаваемый смысл
$Q[i, k]$	$0 \leq i \leq p(n)$ $0 \leq k \leq r$	В момент времени i программа M находится в состоянии q_k
$H[i, j]$	$0 \leq i \leq p(n)$ $-p(n) \leq j \leq p(n) + 1$	В момент времени i читающая/пишущая головка просматривает ячейку с номером j
$S[i, j, k]$	$0 \leq i \leq p(n)$ $-p(n) \leq j \leq p(n) + 1$ $0 \leq k \leq v$	В момент времени i в j -й ячейке записан символ s_k

Состав f_L

Группа дизъюнкций	Налагаемое ограничение
G_1	В любой момент времени i программа M находится ровно в одном состоянии.
G_2	В любой момент времени i читающая/пишущая головка просматривает ровно одну ячейку.
G_3	В любой момент времени i каждая ячейка содержит ровно один символ из Γ .
G_4	В момент времени 0 вычисление находится в исходной конфигурации стадии проверки при входе x .
G_5	Не позднее чем через $p(n)$ шагов M переходит в состояние q_v и, следовательно, принимает x .
G_6	Для любого момента времени i , $0 \leq i \leq p(n)$, конфигурация программы M в момент времени $i+1$ получается из конфигурации в момент времени i одноразовым применением функции перехода δ .

Дизъюнкции для G_1 - G_5

- G_1 $\{Q[i, 0], Q[i, 1], \dots, Q[i, r]\}, \quad 0 \leq i \leq p(n),$
 $\{\overline{Q[i, j]}, \overline{Q[i, j']}\}, \quad 0 \leq i \leq p(n), \quad 0 \leq j < j' \leq r,$
- G_2 $\{H[i, -p(n)], H[i, -p(n) + 1], \dots, H[i, p(n) + 1]\}, \quad 0 \leq i \leq p(n),$
 $\{\overline{H[i, j]}, \overline{H[i, j']}\}, \quad 0 \leq i \leq p(n), \quad -p(n) \leq j < j' \leq p(n) + 1$
- G_3 $\{S[i, j, 0], S[i, j, 1], \dots, S[i, j, v]\}, \quad 0 \leq i \leq p(n),$
 $\quad \quad \quad -p(n) \leq j \leq p(n) + 1,$
 $\{\overline{S[i, j, k]}, \overline{S[i, j, k']}\}, \quad 0 \leq i \leq p(n), \quad -p(n) \leq j \leq p(n) + 1,$
 $\quad \quad \quad 0 \leq k < k' \leq v,$
- G_4 $\{Q[0, 0]\}, \{H[0, 1]\}, \{S[0, 0, 0]\},$
 $\{S[0, 1, k_1]\}, \{S[0, 2, k_2]\}, \dots, \{S[0, n, k_n]\},$
 $\{S[0, n + 1, 0]\}, \{S[0, n + 2, 0]\}, \dots, \{S[0, p(n) + 1, 0]\},$
 где $x = s_{k_1} s_{k_2} \dots s_{k_n}$,
- G_5 $\{Q[p(n), 1]\}.$

G_6

- Если головка не просматривает ячейку j в момент i , то значение в ячейке j не изменится к моменту $i+1$

$$\{\overline{S[i, j, l]}, \overline{H[i, j]}, \overline{S[i+1, j, l]}\}, \quad 0 \leq i \leq p(n), \\ -p(n) \leq j \leq p(n)+1, \quad 0 \leq l \leq v.$$

- Перестройка конфигурации происходит согласно функции δ :

$$(i, j, k, l), \quad 0 \leq i < p(n), \quad -p(n) \leq j \leq p(n)+1, \\ 0 \leq k \leq r \quad \text{и} \quad 0 \leq l \leq v,$$

$$\{\overline{H[i, j]}, \overline{Q[i, k]}, \overline{S[i, j, l]}, \overline{H[i+1, j+\Delta]}\},$$

$$\{\overline{H[i, j]}, \overline{Q[i, k]}, \overline{S[i, j, l]}, \overline{Q[i+1, k']}\},$$

$$\{\overline{H[i, j]}, \overline{Q[i, k]}, \overline{S[i, j, l]}, \overline{S[i+1, j, l']}\},$$

если $q_k \in Q \setminus \{q_Y, q_N\}$, то $\delta(q_k, s_l) = (q_{k'}, s_{l'}, \Delta)$,

а если $q_k \in \{q_Y, q_N\}$, то $\Delta = 0$, $k' = k$ и $l' = l$.