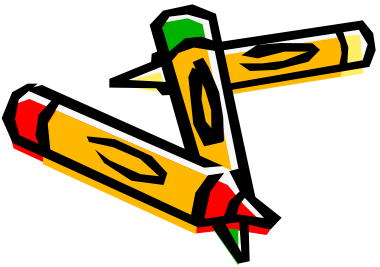
Алгоритм индуцирования знаний из БД

Алгоритм генерирует продукционные правила.

В алгоритме используется представление знаний в виде деревьев решений.

Рассмотрим пример.

Пусть необходимо построить базу знаний для получения ответа: «Как поступить, чтобы прибыль росла?».



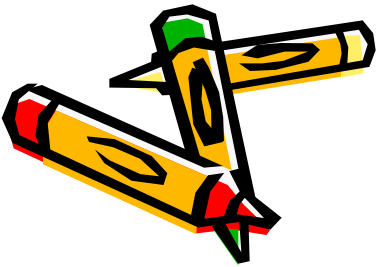
Исходная база данных, из которой извлекаются знания

ПРИБЫЛЬ	ВОЗРАСТ	КОНКУ-РЕНЦИЯ	ТИП
падает	старый	нет	ПО
падает	средний	есть	ПО
растёт	средний	нет	ЭВМ
падает	старый	нет	ЭВМ
растёт	новый	нет	ЭВМ
растёт	новый	нет	ПО

Окончание на следующем слайде…

(окончание)

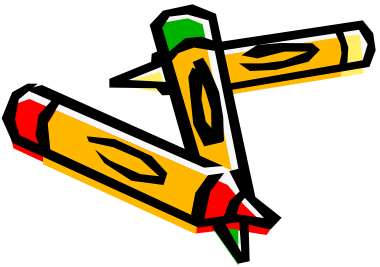
ПРИБЫЛЬ	ВОЗРАСТ	КОНКУ-РЕНЦИЯ	ТИП
растёт	средний	нет	ПО
растёт	новый	есть	ПО
падает	средний	есть	ЭВМ
падает	старый	есть	ПО

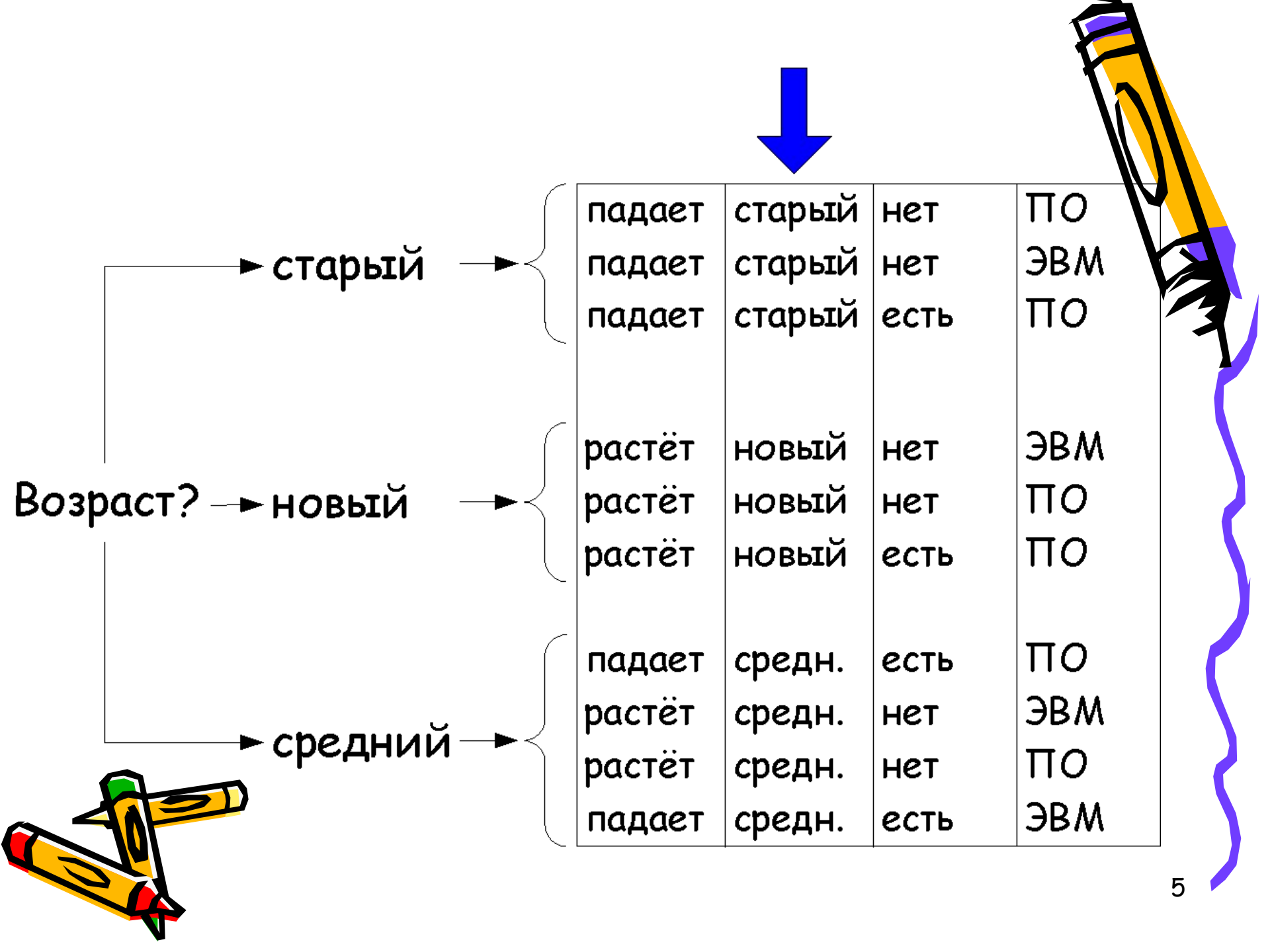


Искомый атрибут «Прибыль» будем называть *атрибутом класса*.

Для построения дерева решений нужно взять один из атрибутов таблицы в качестве *основного (корневого) атрибута*. Пусть это будет «Возраст».

Преобразуем исходную таблицу к следующему виду (сортируем по графе Возраст):

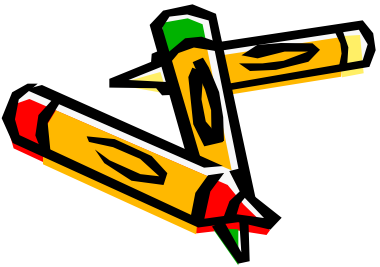



Возраст?
старый
новый
средний
падает старый нет ПО
падает старый нет ЭВМ
падает старый есть ПО
растёт новый нет ЭВМ
растёт новый нет ПО
растёт новый есть ПО
падает средн. есть ПО
растёт средн. нет ЭВМ
растёт средн. нет ПО
падает средн. есть ЭВМ

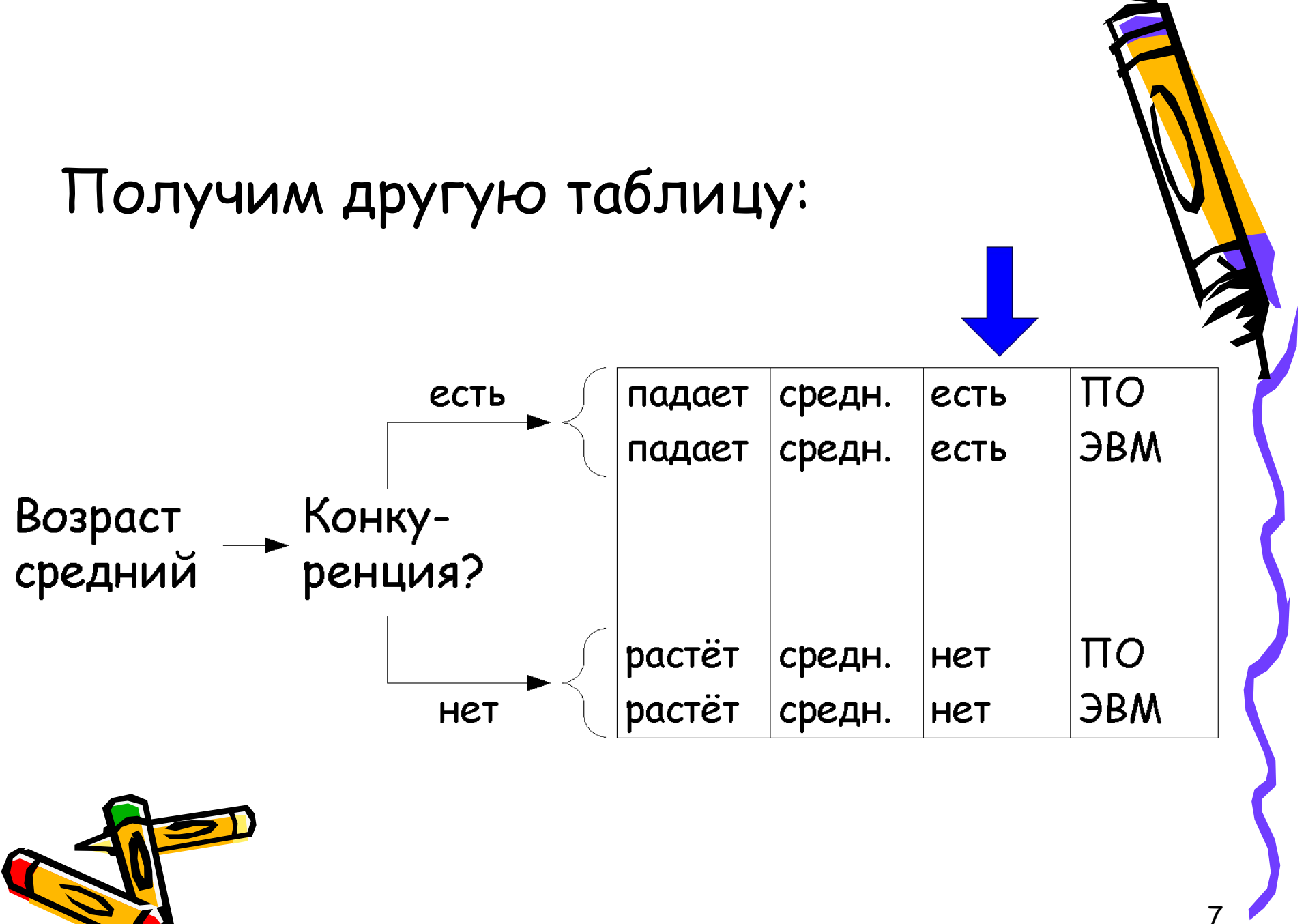
Из таблицы видно, что при значении атрибута *«Возраст»*, равном *«новый»*, прибыль всегда *растёт*, а при значении *«старый»* – *падает*.

В случае же значения *«средний»* такого определённого вывода сделать нельзя.

Поэтому продолжим разбивку нижней подтаблицы по атрибуту *Конкуренция»*.



Получим другую таблицу:



падает	средн.	есть	ПО
падает	средн.	есть	ЭВМ
растёт	средн.	нет	ПО
растёт	средн.	нет	ЭВМ

Поскольку теперь для атрибута класса наше дерево решений выводит однозначный ответ, то дерево решений построено.

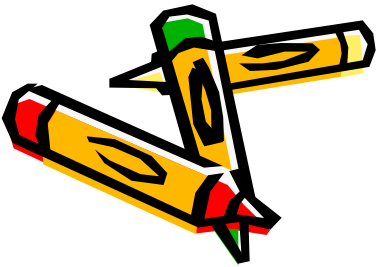
Порождаем правила:

1. **ЕСЛИ** Возраст = новый
   ТО Прибыль = растёт.

2. **ЕСЛИ** Возраст = старый
   ТО Прибыль = падает.

3. **ЕСЛИ** Возраст = средний
И Конкуренция = нет
ТО Прибыль = растёт.

4. **ЕСЛИ** Возраст = средний
И Конкуренция = есть
ТО Прибыль = падает.



Алгоритм C4.5

Улучшает базовый алгоритм индуцирования знаний.

Основное отличие: следующий условный атрибут, по которому проводится разбиение, определяется по критерию минимизации энтропии.

Теперь алгоритм не зависит от порядка следования атрибутов таблицы данных.

Общее описание алгоритма C4.5

Алгоритм работает для таких таблиц данных, в которых атрибут класса (целевой атрибут) может иметь конечное множество значений.

Обозначения

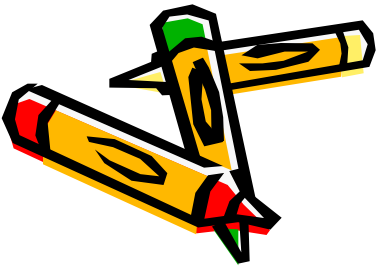
T — множество примеров (таблица или подтаблица данных);

m — количество условных атрибутов (столбцов таблицы)

$|T|$ — мощность множества примеров (количество строк в таблице или подтаблице данных);

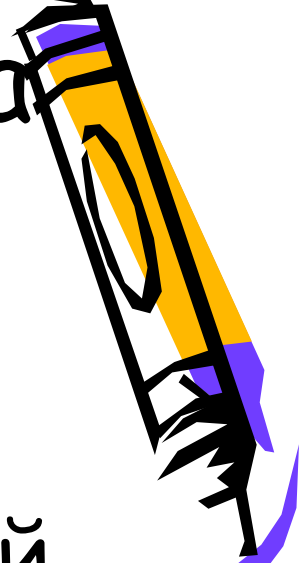
$C_1, C_2, \ldots, C_k$ — значения, принимаемые атрибутом класса;

X — текущий условный атрибут, по которому мы хотим провести разбиение



A_1 , A_2 , …, A_n — значения, принимаемые текущим условным атрибутом;

Выбор условного атрибута для разбиения



Пусть рассматриваем условный атрибут X, принимающий n значений $A_1, A_2 \ldots A_n$. Тогда разбиение множества (таблицы) T по атрибуту X даст нам подмножества (подтаблицы) $T_1, T_2 \ldots T_n$.

Пусть $freq(C_j, T)$ — количество примеров из множества T, в которых атрибут класса равен C_j

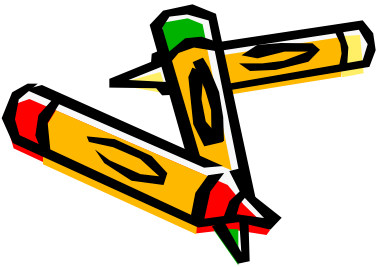
Тогда вероятность того, что случайно выбранная строка из таблицы T будет принадлежать классу C_j, равна

$$P = \frac{\text{freq}(C_j, T)}{|T|}$$

Например, вероятность того, что прибыль будет расти, составляет $P = 5 / 10 = 0{,}5$

Согласно теории информации, количество содержащейся в сообщении информации зависит от её вероятности $\log_2(1/P) = -\log_2(P)$.

В качестве единицы энтропии принят бит, что соответствует логарифмам при основании 2.



Энтропия таблицы T, то есть среднее количество информации, необходимое для определения класса, к которому относится строка из таблицы *T*:

$$\mathrm{Info}(T) = -\sum_{j=1}^{k} \frac{\mathrm{freq}(C_j, T)}{|T|} \cdot \log_2\left(\frac{\mathrm{freq}(C_j, T)}{|T|}\right)$$

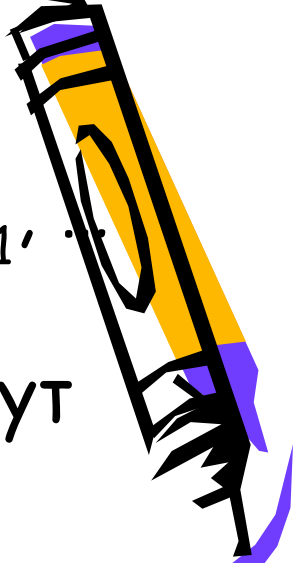
Энтропия таблицы T после её разбиения по атрибуту X на n подтаблиц:

$$\mathrm{Info}_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot \mathrm{Info}(T_i)$$

Критерий для выбора атрибута X – следующего атрибута для разбиения:

$$\text{Gain}(X) = \text{Info}(T) - \text{Info}_X(T)$$

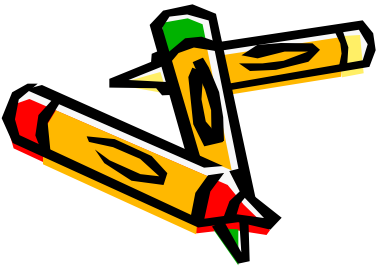
Шаги алгоритма C4.5



Шаг 1. Для всех условных атрибутов $X_1, \ldots X_m$ таблицы T вычисляем критерий разбиения $Gain(X_i)$. Выбираем такой атрибут X, для которого $Gain(X_i)$ максимально.

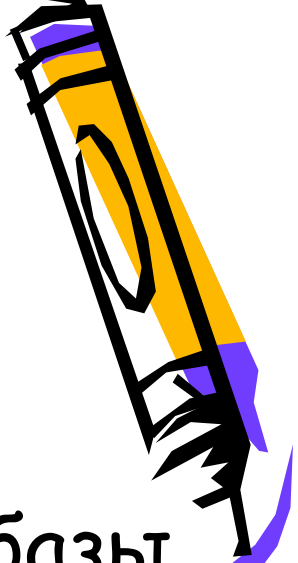
Шаг 2. Разбиваем таблицу по выбранному атрибуту на N подтаблиц. Проверяем каждую подтаблицу следующим образом.

2.1. Если подтаблица монотонна (все строки относятся к одному классу), то порождаем правило.



2.2. В противном случае рекурсивно применяем алгоритм C4.5 к полученной подтаблице

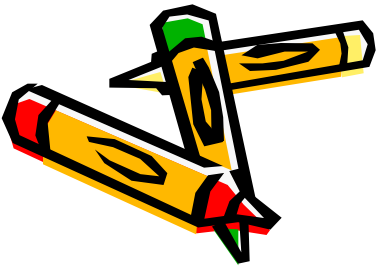
Пример работы алгоритма C4.5



В качестве примера возьмём уже известную нам задачу о построении базы знаний для получения ответа: «Как поступить, чтобы прибыль росла?».

Рассмотрим поведение алгоритма C4.5.

1. Рассчитаем Gain(X) для всех условных атрибутов исходной таблицы.



$$Info(T) = -(0{,}5 \cdot \log_2(0.5) + 0{,}5 \cdot \log_2(0.5)) = -(-0{,}5-0{,}5) = 1$$

Расчёт критерия разбиения для атрибута «**ВОЗРАСТ**»

$Info(T_1) = -(3/3 \cdot \log_2(3/3)) = 0.$

$Info(T_2) = -(3/3 \cdot \log_2(3/3)) = 0.$

$Info(T_3) = -(2/4 \cdot \log_2(2/4) + 2/4 \cdot \log_2(2/4)) = 1.$

$Info_{ВОЗРАСТ}(T) = 3/10 \cdot 0 + 3/10 \cdot 0 + 4/10 \cdot 1 = 0,4;$

Gain(ВОЗРАСТ) = 1 – 0,4 = 0,6.

Расчёт критерия разбиения для атрибута **«КОНКУРЕЦИЯ»**

$Info(T_1) = -(1/4 \cdot \log_2(1/4) + 3/4 \cdot \log_2(3/4)) = 1.$

$Info(T_2) = -(2/6 \cdot \log_2(2/6) + 4/6 \cdot \log_2(4/6)) = 1.59.$

$Info_{КОНКУРЕНЦИЯ}(T) = 4/10 \cdot 1 + 6/10 \cdot 1{,}59 = 1.354$

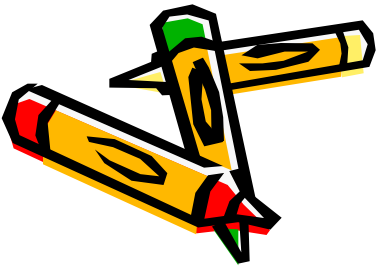
Gain(КОНКУРЕНЦИЯ) = 1 – 1,354 = = -0,354.

Расчёт критерия разбиения для атрибута «**ТИП**»

$Info(T_1) = -(2/4 \cdot \log_2(2/4) + 2/4 \cdot \log_2(2/4)) = 1.$

$Info(T_2) = -(3/6 \cdot \log_2(3/6) + 3/6 \cdot \log_2(3/6)) = 1.$

$Info_{ТИП}(T) = 4/10 \cdot 1 + 6/10 \cdot 1 = 1$



Gain(ТИП) = 1 – 1 = 0.