

## Лекция 6. Рекурсия (продолжение)

### Некоторые неформальные определения

1.	Рекурсия	<p><b>А.</b> Введение в <i>определение</i> объекта ссылки на сам объект.</p> <p><b>Б.</b> Прием сведения решения некоторой задачи к решению серии задач, подобных исходной.</p> <p><b>В.</b> <i>Свойство</i> алгоритмической системы на промежуточных этапах своего функционирования создавать другие системы, включая идентичные себе самой, и использовать результаты их функционирования в дальнейшей работе. При достаточно широкой трактовке понятия алгоритмической системы концепция рекурсивности отражает основные формы развития материи и является одним из важнейших методов познания.</p> <p><b>Г.</b> Результат отчуждения (отвлечения, <i>абстрагирования</i>) свойств рекурсивности от совокупности рекурсивных объектов.</p>
2.	Рекурсивный алгоритм (процедура, функция)	<p><b>А.</b> Алгоритм называется рекурсивным, если в его определении содержится прямой или косвенный вызов этого же алгоритма.</p> <p><b>Б.</b> Рекурсивная функция - одно из математических уточнений интуитивного понятия вычислимой функции.</p>
3.	Рекурсивные вычисления	Вычисления, проводимые с помощью рекурсивных алгоритмов (процедур, функций).
4.	Прямая рекурсия	Непосредственный вызов алгоритма (функции, процедуры) $F$ из текста самого алгоритма $F$ .
5.	Косвенная (взаимная) рекурсия	Циклическая последовательность вызовов нескольких <i>алгоритмов</i> $F_1, F_2, \dots, F_k$ (функций, процедур) друг друга: $F_1$ вызывает $F_2, F_2$ вызывает $F_3, \dots, F_k$ вызывает $F_1$ ( $k > 1$ ).
6.	Рекурсивные обращения	Рекурсивные вызовы в прямой или косвенной рекурсии.
7.	Пространство параметров	Пусть $t_k$ ( $k=1, 2, \dots, n$ ) - параметры задачи (алгоритма, процедуры, функции), принимающие значения из некоторых множеств объектов $M_k$ ( $k=1, 2, \dots, n$ ). Декартово произведение $M$ множеств $M_k$ называется пространством параметров задачи. Таким образом, элементами $M$ являются наборы (упорядоченные множества) объектов $m_1, m_2, \dots, m_n$ , где $m_k \in M_k$ ( $k=1, 2, \dots, n$ ) вида: $(m_1, m_2, \dots, m_n)$ . Областью определения параметризованной задачи является совокупность элементов пространства параметров, при которых она имеет решение.
8.	Рекурсивная база	Совокупность наборов значений параметров и соответствующих им решений задачи или простых правил для получения этих решений. <u>Выделение базы - один из основных этапов решения задачи с помощью рекурсии. База может быть динамической, то есть меняться в процессе вычислений.</u>

9.	Индикаторы завершения рекурсивных вызовов	Элементы постоянной или динамической <b>рекурсивной базы</b> .
10.	Полное дерево рекурсивных вызовов	Граф, вершинами которого являются наборы фактических параметров <b>при всех рекурсивных вызовах функции</b> , начиная с обращения к ней, а ребрами - пары таких наборов, соответствующих соседним вызовам. Заметим, что вершины этого дерева удобно называть просто рекурсивными вызовами. Полное дерево рекурсивных вызовов однозначно определяется алгоритмом (функцией, процедурой) и набором её фактических параметров. Заметим, что одни и те же наборы параметров могут соответствовать разным вершинам дерева.
11.	Корень полного дерева рекурсивных вызовов	Вершина полного дерева рекурсивных вызовов, соответствующая начальному обращению к функции (процедуре).
12.	Дерево рекурсивных вызовов	Любое поддерево полного дерева рекурсивных вызовов.
13.	Полная рекурсивная траектория	Пусть $X=(x_1, x_2, \dots, x_n)$ и $F(X)$ - рекурсивная функция, которую требуется вычислить в некоторой точке $X_0$ из области определения $F(X)$ . Конечная последовательность аргументов $F(X)$ вида $X_0, X_1, \dots, X_n$ называется рекурсивной траекторией, если элементы $X_k$ ( $k=1, 2, \dots, n$ ) - суть наборы параметров при последовательных рекурсивных вызовах, а $X_n$ принадлежит базе рекурсии. Полная рекурсивная траектория указывает последовательность обхода вершин полного рекурсивного дерева.
14.	Рекурсивная траектория	Любая начальная подпоследовательность полной рекурсивной траектории.
15.	Объем рекурсии	Количество вершин полного рекурсивного дерева (полной рекурсивной траектории) без единицы.
16.	Рекурсивный стек (магазин)	Область памяти, в которую заносятся значения всех локальных переменных алгоритма (программы) в момент рекурсивного обращения. Каждое такое обращение формирует один слой данных стека. При завершении вычислений по конкретному обращению $a$ из стека считывается соответствующий ему слой данных, и локальные переменные восстанавливаются, снова принимая значения, которые они имели в момент обращения $a$ .
17.	Глубина рекурсивных вызовов	Максимальное количество слоев рекурсивного стека, заполняемых при конкретном вычислении значения рекурсивной функции (процедуры, алгоритма). Количество элементов полной рекурсивной траектории всегда не меньше глубины рекурсивных вызовов. Эта величина не должна превосходить максимального размера стека используемой вычислительной среды.

18.	<b>Декомпозиция</b>	Процесс последовательного разложения задачи на серию подзадач двух типов: тех, которые мы решать умеем и тех, которые в чем-то аналогичны исходной задаче. В последнем случае каждая из полученных подзадач должна быть упрощенным вариантом предыдущей задачи. При этом декомпозицию следует осуществлять так, чтобы можно было доказать, что при любом допустимом наборе значений параметров рано или поздно она приведет нас к одному из выделенных тривиальных случаев, т.е. к задаче с набором параметров, являющимся индикатором завершения рекурсивных вызовов.
19.	<b>Параметризация задачи</b>	Выявление совокупности исходных величин, определяющих постановку и решение задачи. Значения этих параметров или некоторых из них влияют на трудоемкость решения задачи.
20.	<b>Вспомогательные параметры рекурсии</b>	Искусственно вводимые параметры, напрямую с постановкой задачи не связанные, но помогающие изменить тип рекурсии или перейти к обобщенной задаче, где рекурсия проглядывается явно.
21.	<b>Рекурсивная триада</b>	Три основных этапа решения задач с помощью рекурсии: параметризация, выделение базы (или выделение начальной базы и правил её изменения), декомпозиция.
22.	<b>Предвычисления (предварительные вычисления)</b>	Декомпозиция предполагает наличие некоторых вычислений, предшествующих и способствующих переходам к более простым подзадачам. Их называют предварительными вычислениями или предвычислениями. Именно предвычисления и подготавливают переходы к следующим уровням рекурсивных обращений.
23.	<b>Отложенные вычисления</b>	Вычисления, проводимые после того, как рекурсивная траектория попала в базу, то есть стала полной. Возможно, что отложенные вычисления состоят лишь из серии передач значений и управления в порядке, обратном рекурсивным вызовам. В этом случае говорят об отсутствии отложенных вычислений.
24.	<b>Терминальная ветвь рекурсивного алгоритма</b>	Часть алгоритма (процедуры, функции), соответствующая проверке получения какой-либо задачи из рекурсивной базы и тем предписаниям, которые должны быть выполнены, если действительно такая задача при вычислениях будет получена. В алгоритме может быть несколько терминальных ветвей.
25.	<b>Рабочая ветвь рекурсивного алгоритма</b>	Дополнительная к совокупности терминальных ветвей часть алгоритма (процедуры, функции). Рабочая ветвь реализует предвычисления, рекурсивные обращения и отложенные вычисления.
26.	<b>Управляющие параметры рекурсии</b>	Один или несколько параметров задачи, с помощью которых организуется её декомпозиция и обеспечиваются правила выполнения рекурсивных вызовов. Значения управляющих параметров рекурсии могут участвовать в предварительных и отложенных вычислениях.
27.	<b>Прямой и обратный ход вычислений</b>	Рекурсивные вычисления состоят из двух стадий, называемых прямым ходом и обратным ходом. Первая из них соответствует совокупности всех предвычислений, реализуемых до входа

		рекурсивной траектории в базу, а вторая - совокупности отложенных вычислений, производимых после встречи с индикатором завершения рекурсивных вызовов.
28.	<b>Срез рекурсивных вычислений</b>	При решении задачи каждое рекурсивное обращение, в том числе и начальный запуск вычислений, инициируют работу как бы со своим экземпляром исходного алгоритма. Последовательность вычислений значений локальных и глобальных переменных, соответствующая одному конкретному “виртуальному” экземпляру алгоритма и не включающая в себя вычисления по вызовам из данного экземпляра (но использующая их результаты!), называется срезом рекурсивных вычислений.
29.	<b>Линейная рекурсия</b>	Тип рекурсии, при которой рекурсивные вызовы на любом рекурсивном срезе, инициируют не более одного последующего рекурсивного вызова. Для линейной рекурсии дерево рекурсивных вызовов всегда состоит из одной ветви.
30.	<b>Повторительная рекурсия</b>	Частный случай линейной рекурсии с отсутствующими предварительными или отложенными вычислениями. От повторительной рекурсии достаточно просто избавиться переходом к циклу.
31.	<b>Каскадная (древовидная) рекурсия</b>	Рекурсия, не являющаяся линейной. При каскадной рекурсии, рекурсивные обращения, как правило, приводят к необходимости многократно решать одни и те же подзадачи.
32.	<b>Возвратная рекурсия</b>	Рекурсивная реализация метода перебора с возвратом (backtracking).
33.	<b>Динамическая рекурсивная база</b>	Рекурсивная база, меняющаяся в процессе вычислений. Как правило, она расширяется за счет получения решений промежуточных задач и облегчает выполнение процесса отложенных вычислений. Возможно и сужение рекурсивной базы. Альтернативой динамической базы могут быть организуемые и поддерживаемые в процессе вычислений специальные списки решений промежуточных задач (динамическое программирование).
34.	<b>Удаленная рекурсия</b>	Рекурсивная функция, в теле которой при рекурсивных вызовах, в выражениях, являющихся фактическими параметрами, снова встречаются рекурсивные вызовы этой функции. Удаленная рекурсия трудна в понимании и, как правило, приводит к достаточно громоздким вычислениям.
35.	<b>Рекурсивная тавтология</b>	<b>А.</b> Прямое или косвенное обращение рекурсивной функции (алгоритма, программы) к самой себе с тем же самым набором значений параметров, с которого начиналось вычисление этой функции. <b>В.</b> Рекурсивная функция (алгоритм, программа), допускающая рекурсивную тавтологию.
36.	<b>Рекурсивные определения</b>	Описание совокупности объектов через порождающие правила, содержащие ссылки на сам определяемый объект. Такое описание предполагает наличие некой базы неопределяемых первичных

		объектов, заданных простым перечислением. Примерами подобных определений являются синтаксические диаграммы Вирта, нотация Бэкуса-Наура. Рекурсивные определения могут быть как прямыми, так и косвенными.
37.	<b>Рекурсивное мышление</b>	<p><b>А.</b> Способ решения прикладных задач с преимущественной опорой на рекурсивные алгоритмы.</p> <p><b>В.</b> Разновидность математического (диалектического, продуктивного) мышления, позволяющая видеть рекурсию там, где она на первый взгляд не просматривается.</p>

Еще примеры рекурсии:

Сказка «Репка», Песня колобка из сказки «Колобок»

Стихотворение «Дом, который построил Джек» в переводе С.Я.Маршака и оригинал Сепульки

В романе С. Лемма «Звёздные дневники Ийона Тихого», в «Путешествии четырнадцатом» герой последовательно переходит от статьи о сепульках к статье о сепуляции, оттуда к статье о сепулькарнях, в которой снова стоит отсылка к статье «сепульки»:

*Нашёл следующие краткие сведения:*

*«СЕПУЛЬКИ — важный элемент цивилизации ардритов (см.) с планеты Энтерофия (см.). См. СЕПУЛЬКАРИИ».*

*Я последовал этому совету и прочёл:*

*«СЕПУЛЬКАРИИ — устройства для сепуления (см.)».*

*Я поискал «Сепуление»; там значилось:*

*«СЕПУЛЕНИЕ — занятие ардритов (см.) с планеты Энтерофия (см.). См. СЕПУЛЬКИ».*

Еще

У Лема же есть серия рассказов «Кибериада», и в одном из них фигурирует разумная машина, которая обладает достаточным умом и ленью, чтобы для решения поставленной задачи построить себе подобную, и поручить решение ей. Итогом стала бесконечная рекурсия, когда каждая новая машина строила себе подобную и передавала задание ей.

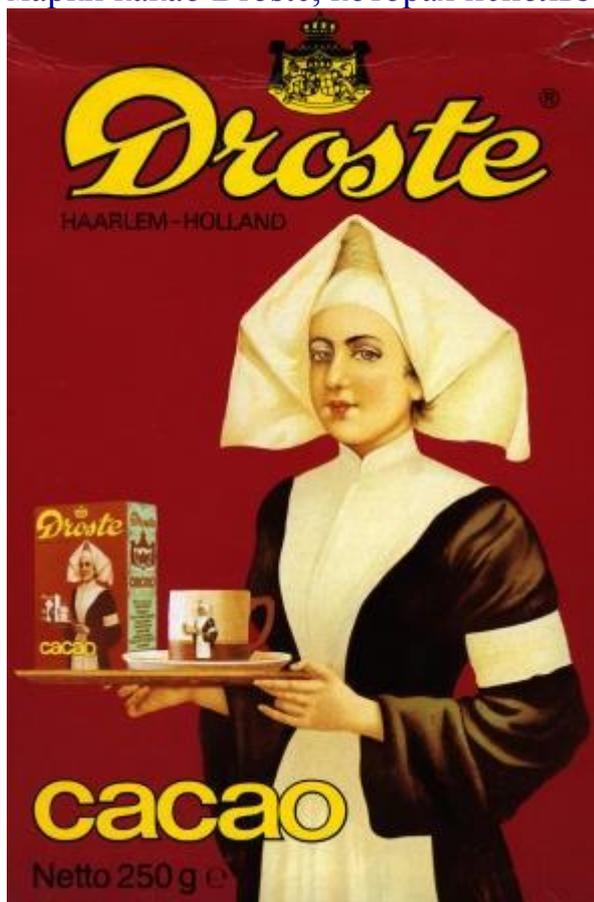
## В искусстве

М. Эшер часто эксплуатировал в творчестве рекурсии.



## Эффект Дросте

Одна из визуальных форм рекурсии называется «Эффект Дросте» — по названию марки какао Droste, которая использовала этот эффект в рекламе.

**Рекурсивная тавтология:**

У попа была собака, он ее любил...

*У попа была собака, он её любил,  
Она съела кусок мяса, он её убил,  
В землю закопал,  
Надпись написал:  
У попа была собака, он её любил,  
Она съела кусок мяса, он её убил,  
В землю закопал,  
Надпись написал...*



*Салат «рекурсивный»: помидоры, огурцы, салат...*