

Модели вычислений

1. Краткая характеристика
2. Тип языка программирования
3. Характер вычислений
4. Механизмы управления и передачи данных

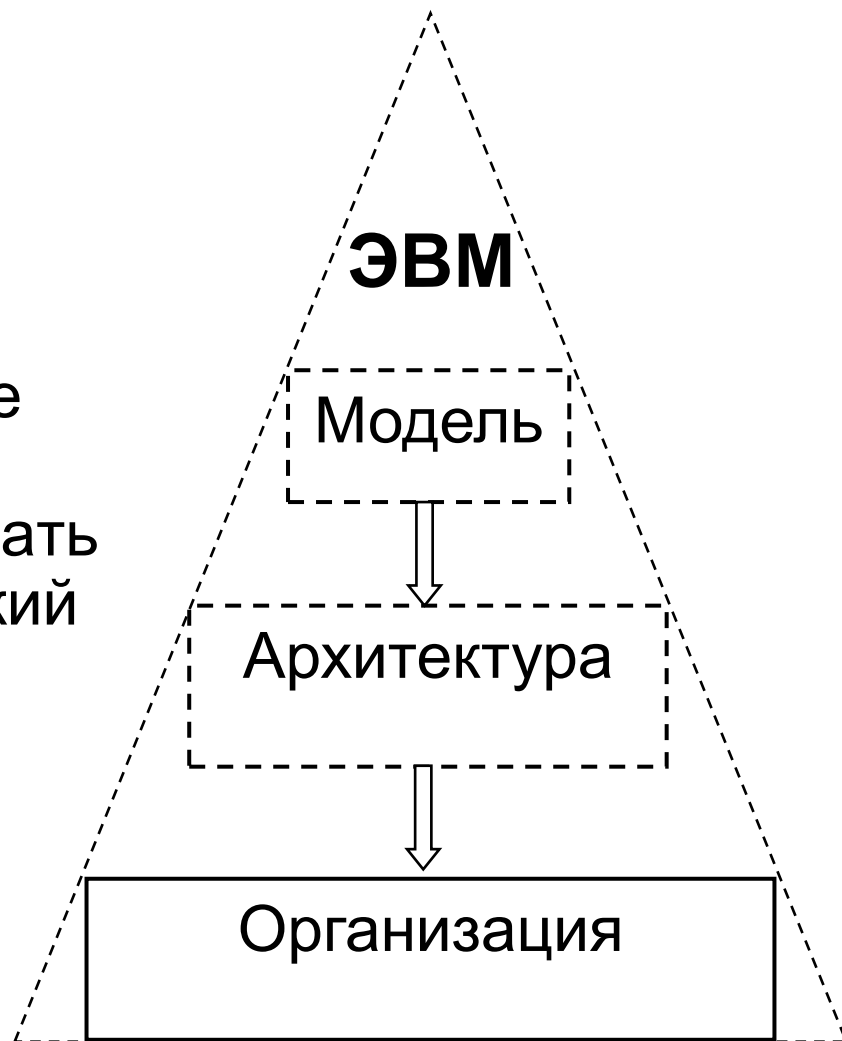
Знать: понятие модели вычислений и влияние на нее типа языка программирования; особенности моделей вычислений в зависимости от характера вычислений; три класса механизмов управления и два класса механизмов передачи данных, применяемых в моделях вычислений.

Модели вычислений

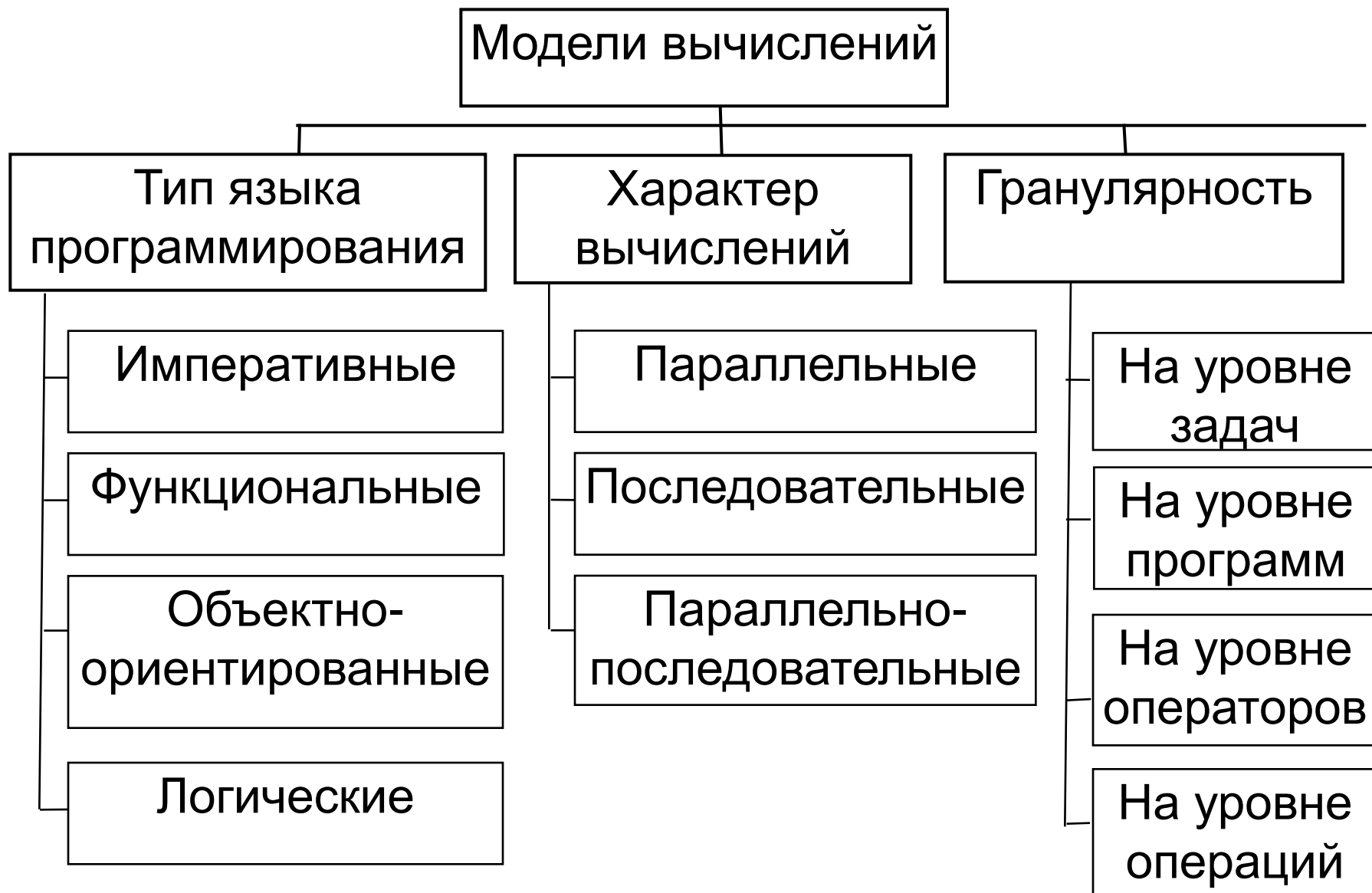
- **Помнить:** о семантическом разрыве между языками программирования, на которые опираются модели вычислений, и внутренними языками вычислительных машин.
- **Литература:**
- Страбыкин Д.А. Организация машин параллельного логического вывода: Учебное пособие для вузов. – Киров, изд-во ВятГТУ, 1999 с.85-96.

1. Краткая характеристика

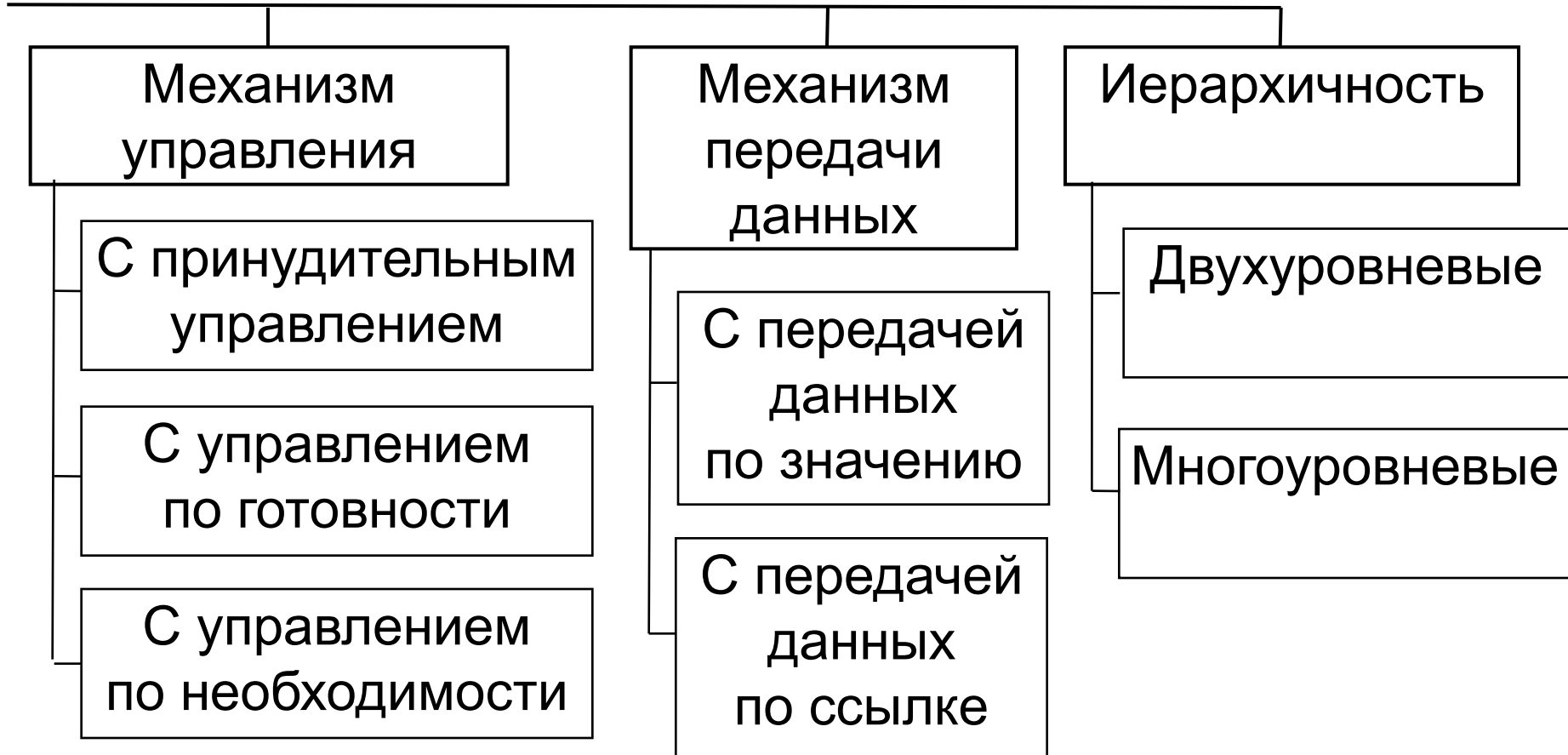
- Можно выделить три важнейших аспекта рассмотрения ЭВМ: модель вычислений, архитектура и организация (рисунок 1).
- Модели вычислений представляют собой описание некоторой гипотетической машины, способной реализовать определенный алгоритмический процесс.
- Понятие модели вычислений тесно связано с понятиями алгоритма решения задачи, языка программирования и архитектуры машины.



Характеристика моделей вычислений



Характеристика моделей вычислений (продолжение)



2. Тип языка программирования

- Существуют четыре основных типа моделей вычислений: императивные, функциональные, объектно-ориентированные и логические.
- Алгоритмы, используемые в этих моделях, реализуются с помощью языков программирования четырех типов:
 - процедурных (императивных),
 - функциональных,
 - объектно-ориентированных,
 - логических.

2.1. Императивная модель вычислений

- Процессор последовательно по одной считывает команды из памяти и в соответствии с имеющимися в них указаниями считывает данные, хранящиеся по некоторому адресу в памяти, выполняет заданные операции и записывает результаты операции в память по определенному адресу.
- Содержимое памяти соответствует состоянию машины, а вычисления выполняются путем последовательного изменения этого состояния. Таким образом, выполнение вычислений сопровождается побочным эффектом.
- В процедурных языках высокого уровня побочный эффект представлен оператором присваивания, который указывает подстановку некоторого значения в переменную.
- Примерами языков, основанных на императивной модели вычислений, могут служить Фортран, Паскаль и др.

2.2. Функциональная модель вычислений

- Вычислительная модель, в которой программа рассматривается как множество определений функций, называется функциональной моделью.
- Функциональные программы строятся из “чистых” математических функций, которые по сравнению с функциями многих императивных программ свободны от побочных эффектов, т.е. их выполнение не может изменить среду вычислений.
- Не существует назначаемого состояния программы. Из-за этого нет возможности больше программировать “с помощью эффекта”, так что величина, которая должна быть вычислена программой, и сама программа редуцируются к одному и тому же результату.
- Функциональная модель вычислений основана на простой и четкой математической модели, называемой “лямбда-исчислением”.
- К функциональным языкам относятся “чистый” Лисп (предложенный Джоном Маккарти), Hope, Miranda и др.

2.3. Объектно-ориентированная модель вычислений

- Объект операции принимается за субъект и строится такая вычислительная модель, в которой применение операции рассматривается как передача запроса объекту. При этом интерпретация запроса и действия, предусмотренные в нем выполняются самим объектом, принявшим запрос.
- Вычисления моделируются с помощью множества объектов, передающих друг другу запросы.
- *Объект обладает следующими особенностями:*
 - имеет внутреннее состояние, которое при приеме запроса становится активным;
 - управлять внутренним состоянием объекта извне можно только посредством передачи ему запросов;
 - в ходе вычислений может передавать запросы другим объектам;
 - объекты одного типа определяются как класс, и из этого класса можно генерировать другие объекты (классы).
- Одним из первых языков, основанных на объектно-ориентированной модели, был Смолток.

2.4. Логическая модель вычислений

- Модель, в которой вычисления представляются с помощью логики предикатов, называется логической, а язык программирования, используемый для описания вычислений, называется логическим языком.
- Основное отличие логического программирования от процедурного состоит в том, что программа является не четко определенной последовательностью операторов, а представляет собой формализованное описание предметной области. Причем, это описание может носить декларативный характер, а не предписывать ЭВМ каким образом она должна решать задачу.
- Типичным примером языка логического программирования является Пролог, основанный на хорновской логике, представляющей собой подмножество логики предикатов первого порядка.

Программа на Прологе

- Программа на языке Пролог состоит из предложений.
- Предложения бывают трех типов: факты, правила и запросы (цели). Факты содержат утверждения, которые являются всегда, безусловно верными.
- Правила содержат утверждения, истинность которых зависит от некоторых условий.
- С помощью запросов пользователь может устанавливать какие утверждения являются истинными.
- Предложения Пролога состоят из головы и тела.
- Тело – это список целей, разделенных запятыми. Запятая понимается как конъюнкция.
- Факты – это предложения, имеющие пустое тело.
- Запросы имеют только тело, а правила имеют голову и (непустое) тело.
- По ходу вычислений вместо переменной может быть подставлен другой объект. В этом случае говорят, что переменная конкретизирована.

Пример программы на Прологе

- Для иллюстрации того, что и как вычисляют логические программы, рассмотрим следующий пример.

нравится(джилл,вино).	% Факт (1)
нравится(джон,еда).	% Факт (2)
нравится(джон,вино).	% Факт (3)
мужчина(джон).	% Факт (4)
проводит(Х,У):- нравится(У,вино),мужчина(У).	% Правило (5)
:-проводит(мэри,Z).	% Цель (запрос): “есть ли % проводит у Мэри, кто он ?”

Выполнение программы на Прологе

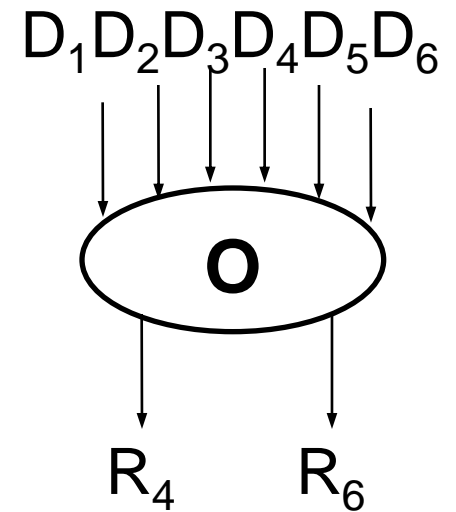
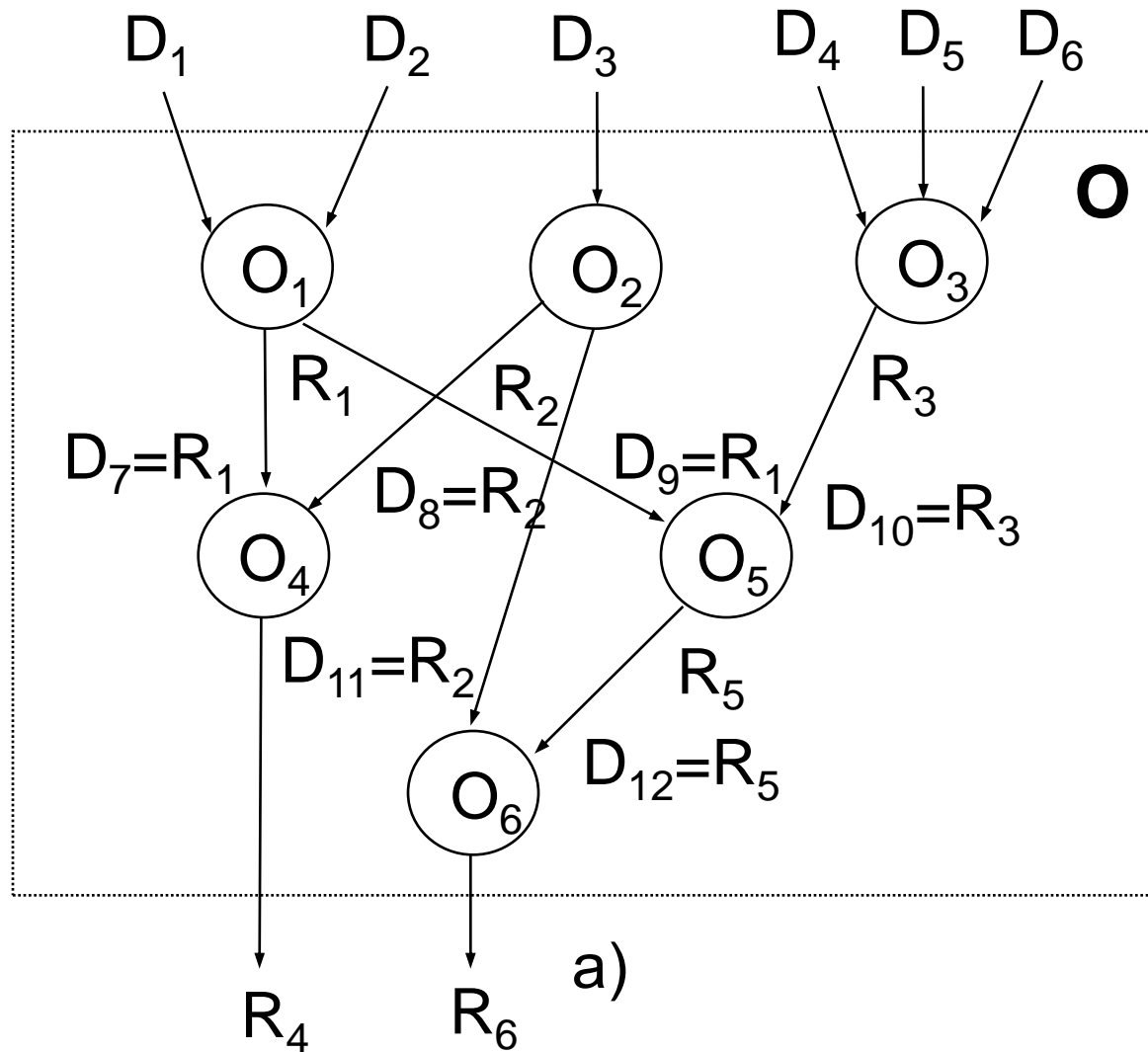
- При сопоставлении цели и предложения с “проводатым” (5) переменная X принимает значение “мэри”.
- Затем справа в этом предложении из условий выбирается новая цель – “нравится(Y, вино)”. Она сопоставима с утверждением “нравится(джилл, вино)” (1), что обеспечивает переход ко второму условию правила (5), связанному с первым условием связкой И. В результате порождается новый запрос – “мужчина(джилл)”.
- Поскольку таких фактов нет запрос не подтверждается. Далее происходит возврат для поиска альтернативного значения переменной Y. Оно будет обнаружено в факте (3), содержащем константу “джон”.
- Теперь порождается новая цель – “мужчина(джон)”, которая удовлетворяется, причем, переменная (Y1) принимает значение “джон”.
- После этого запрос в целом удовлетворяется, а переменная Z, как и переменная Y1, принимает значение “джон”.

3. Характер вычислений

3.1. Параллельные модели

- Алгоритм может быть представлен в виде информационного графа, являющегося однонаправленной сетью или ациклическим графом, узлы которого поставлены в соответствие единицам обработки данных (операторам), а однонаправленные дуги – путям передачи данных между ними, т.е. информационным связям.
- Пример информационного графа с соответствующей ему обобщенной единицей обработки информации (примитивной операции) **О** приведен на рисунке 3.

Простейший информационный граф (а) и
соответствующая ему единица обработки информации (б)



б)

Рисунок 3

Отношения между операторами

- В графе отсутствует указание способа перехода от одного оператора к следующему, однако он позволяет выявить отношения предшествования и безразличия между отдельными операторами.
- Оператор i **предшествует** оператору j , если результат оператора i является исходным значением для оператора j или, рекуррентно, для любого ему предшествующего. Иными словами, оператор j не может быть выполнен раньше или одновременно с оператором i из-за отсутствия входного операнда.
- Операторы i и j находятся в отношении **безразличия**, если исходные операнды для каждого из них могут быть получены вне зависимости от результатов выполнения другого.
- Отношения предшествования и безразличия между операторами позволяют выделить параллельные ветви вычислений.

Реализация параллельной модели вычислений

- Для реализации параллельной модели вычислений можно представить машину, в которой для каждого оператора информационного графа предусмотрен собственный обрабатывающий элемент; все обрабатывающие элементы связаны между собой линиями связи, по которым в пространстве передаются результаты “промежуточных” вычислений.
- Процесс развивается в пространстве и времени вычислительной среды, представляющей собой сеть обрабатывающих элементов. Решение задачи сводится к “настройке” сети, т.е. установлению логических связей между отдельными обрабатывающими элементами, и возбуждению этих элементов по мере готовности подлежащих обработке операндов.
- Программа такой машины должна представлять собой формализованное описание информационного графа. В этом случае отсутствует представление о программе как о последовательности команд.

3.2. Последовательные модели

- При последовательном выполнении всех операторов информационного графа можно воспользоваться единственным универсальным устройством обработки. Совокупность всех операций, которые могут быть реализованы в устройстве обработки, образуют систему псевдокоманд машины.
- Для передачи промежуточных результатов от оператора к оператору и сохранения всего информационного графа в форме некоторой программы необходима память.
- Программа такой машины традиционна и представляет собой жесткую последовательность псевдокоманд, каждая из которых соответствует одному оператору.
- Если процессор может выполнять широкий набор операций, то последовательная машина является универсальной. Параллельная машина должна соответствовать особенностям структуры параллельного алгоритма и, в общем случае, не универсальна.

4. Механизмы управления и передачи данных

- Если рассматривать модели с точки зрения механизма вычислений, то последний можно представить состоящим из двух компонентов: *механизма управления и механизма передачи данных.*
- Механизм управления определяет, каким образом каждая единица обработки (например, оператор) запускает выполнение другой единицы обработки.
- Механизм передачи данных определяет, как передаются данные от одной единицы обработки к другой.

4.1. Механизмы управления

- Механизмы управления можно разделить на три класса: с принудительным управлением (с управлением с помощью формирователя последовательности), с управлением по готовности и с управлением по необходимости.
- При управлении с помощью формирователя последовательности заранее определяется последовательность единиц обработки, и они выполняются в этой последовательности с помощью формирователя (программного счетчика).
- При управлении по готовности выполнение единицы обработки начинается сразу же после того, как будут определены все ее входные параметры.
- При выполнении по необходимости, когда запускается некоторая единица обработки данных, то инициируется выполнение единиц обработки данных, которые вычисляют значения ее входных параметров. Только после завершения выполнения этих единиц обработки данных возможно выполнение запущенной единицы обработки данных.

4.2. Механизмы передачи данных

- Механизмы передачи данных можно разделить на два типа:
 - передача данных по значению и
 - передача данных по ссылке.
- В первом случае все данные копируются для каждой единицы обработки данных, для которой они являются входными, и в каждую единицу обработки данных передается эта копия.
- Во втором случае значения данных накапливаются в одном месте и в единицу обработки данных, для которой они являются входными, передается только информация, указывающая место хранения этих значений.

4.3. Механизмы вычислений

- На основании классификации механизмов управления и передачи данных механизмы вычислений можно классифицировать, как показано в таблице 1.
- Модели вычислений можно также классифицировать в зависимости от используемых механизмов вычислений.
- Механизм вычислений с потоком управления (директивным управлением) применяется в модели вычислений неймановской машины.
- Механизмы вычислений с потоком данных являются основой, так называемых, потоковых моделей вычислений.
- А механизмы вычислений, использующие редукцию – редукционных моделей вычислений.

Механизмы вычислений в зависимости от механизмов управления и механизмов передачи данных

Таблица 1

Классификация механизмов вычислений

Механизм управления	Механизм передачи данных	
	С передачей данных по значению	С передачей данных по ссылке
С принудительным управлением	—	С потоком управления
С управлением по готовности	С потоком данных и системой копирования	С потоком данных и системой ссылок
С управлением по необходимости	Строчной редукции	Графовой редукции