

Императивные и декларативные модели вычислений

21 октября 2021

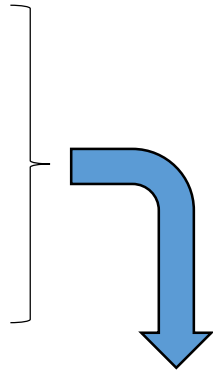
Скопин Игорь Николаевич

Обсуждается проблематика нетрадиционных моделей вычислений: функциональные, сентенциальные и ассоциативные модели. Более подробно рассматривается модель В.С. Бурцева как один из наиболее развитых подходов к реализации эффективных ассоциативных вычислений. Особое внимание уделено вопросу осуществимости совместного использования различных моделей при разработке единой программной системе — можно или нет избежать эклектики и известных ее проблем при таком совмещении?

Повелительное и изъявительное наклонения

- Изъявительное наклонение и развитость языка
- Идеи внедрения изъявительного наклонения

- Системы продукций
- Системы функций
- Коммутационные системы
- Ассоциативные системы
- Аксиоматические системы



- Различные стили программирования
- То, как программист думает, когда разрабатывает алгоритм
- Использование модели фон-Неймана связано с особым стилем, называемым *операционным* и/или *императивным*
- Повелительное наклонение \equiv Императивный язык
- Изъявительное наклонение \equiv Декларативный язык

Зачем нужны новые термины, когда есть и уже давно существуют адекватные слова в родном языке?

Вычислительная машина на человеческой элементной базе

- *Задача* из истории: во Франции в связи с переходом на метрическую систему измерений требовалось точно и как можно быстрее пересчитать артиллерийские поправки прицеливания на дальность, ветер и др. по известным формулам.
Условие: Много формул, типов оружия, результаты нужны как можно скорее
- Карно предложил использовать две роты солдат, каждому из которых поручено выполнять одну арифметическую операцию с аргументами, получаемыми от соседей, и передавать результат одному из них.
- В каждой роте было выделено по **три группы солдат**:
 - 1) для приема входных данных, 2) для счета, 3) для вывода

Зачем потребовались две роты?

Это *действительно* вычислительная машина (не арифмометр!), у которой в ПАМЯТИ записана программа, каждый элемент ПАМЯТИ активен, когда появляются входные данные:

- распределенные вычисления,
- управление потоками данных,
- высокая степень параллелизма,
- защита от сбоев

Системы продукций как основа новой модели вычислений

- Соотношения записываются как *правила вывода*:
Левая Часть => Правая Часть
- Обрабатываемые данные сопоставляются с *шаблонами* (части правила для распознавания, когда правило применимо: Левая Часть — лишь пример, но один из наиболее употребительных).
 - Соответствие шаблону → разрешение *применить правило*:
 - замена выделенного при сопоставлении фрагмента данных на что-то другое, соответствующее Правой Части;
 - однократное выполнение замены — атомарный акт вычисления.
 - Особенность — иерархическая структура программ-предложений
- Примеры языков:
 - Пролог — языковая часть японского проекта компьютеров нового поколения — неудача.
 - Рефал — среди прочего замечателен продуманностью переключения на вычисления других моделей
 - Lisp и CLOS (?),
 - SNOBOL (?)
 -

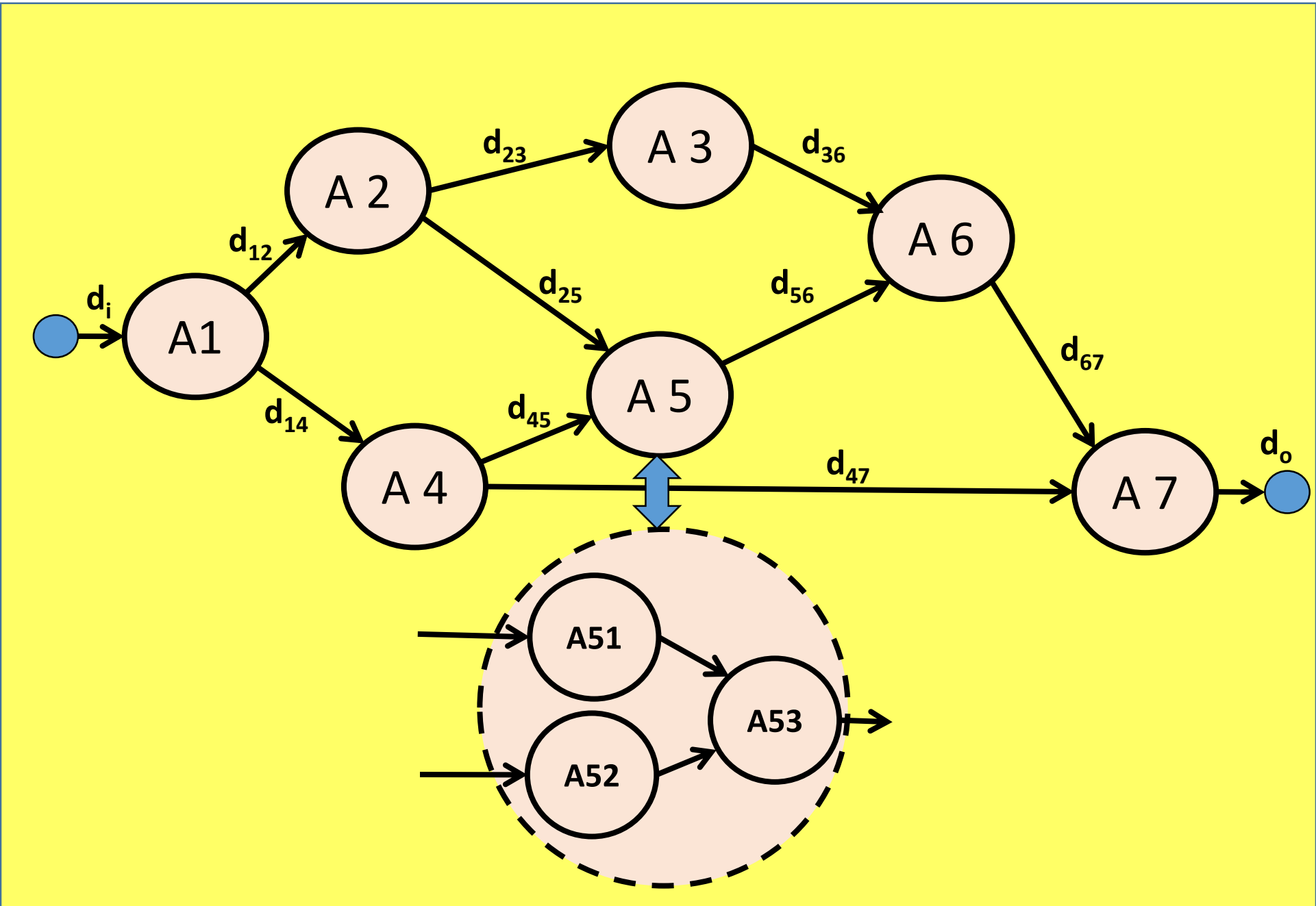
Системы функций как основа новой модели вычислений

- Программа – соотношения между функциями, связанных между собой аргументами, которые в свою очередь могут быть функциями.
- Атомарный акт вычислений — это подготовка аргументов для использующей функции.
- Готовность аргументов (не обязательно всех!) — разрешение вычисления функции
- Однократное вычисление всех значений, которые возникают в динамике вычислений ⇒ мемоизация
- Следствия:
 - Пассивная память не нужна!
 - Возможность оперирования функциями высоких порядков (не только с простыми функционалами ⇒
 - Гибкость
 - Параллельные вычисления
- Но практичная элементная база, реально конкурирующая с фоннеймановской моделью, так и не получилась
- Может, недостаточно интенсивно исследовалась функциональная модель (пример — мемоизация)?

Коммутационные системы как варианты функциональной модели

- **Элемент системы** – вершина графа (входные и выходные места)
- **Дуги** – каналы передачи значений
- **Программа** – граф с вершиной без входных мест (*генератор перерабатываемых данных*) и вершина без выходных мест (*получатель результата*)
- **Вычисление** – действие, связанное вершиной
- **Активизация вычислений** – поступление данных на входные места
- Возможна **вложенность** (структурные вершины)

В императивном программировании часто интерпретируются как автоматные системы, построенные на графах состояний — конечные автоматы Мура и Мили (см. далее)



Коммутационные системы (продолжение)

Если **граф без циклов**, то коммутационная система – форма представления нерекурсивной системы функций

Если **граф с циклами**, то его можно проинтерпретировать как рекурсивную систему функций

- Но это не единственная вычислительная модель коммутационной системы (другой пример – сети Петри)
- **Статическая коммутация vs. динамическая коммутация**
- В императивном программировании могут интерпретироваться как автоматные системы, построенные на графах состояний (конечные автоматы Мура и Мили):
 - Вершина — состояние
 - Дуга — **условие срабатывания перехода** (часто значение из управляющего потока: символ строки и др.) + **действие**, которое связывается с
 - выходом из состояния, т.е. зависит только от состояния — **автомат Мура**
 - входом в состояние, т.е. зависит как от состояния, так и от условия перехода — **автомат Мили**

Ассоциативные модели вычислений

- Это модели, предусматривающие активность элементов данных.
 - Пример — ассоциативные базы данных: поиск информации по заданным значениям ключевых атрибутов хранимых данных
- Активность данных:
 - Среда хранения тегированных элементов
 - Вычислитель ассоциированных данных
- Акт вычислений — два шага:
 - Нахождение подходящих аргументов для этого акта (некоторой функции)
 - Активизация процесса вычисления
 - Размещение результатов в среду хранения элементов данных.

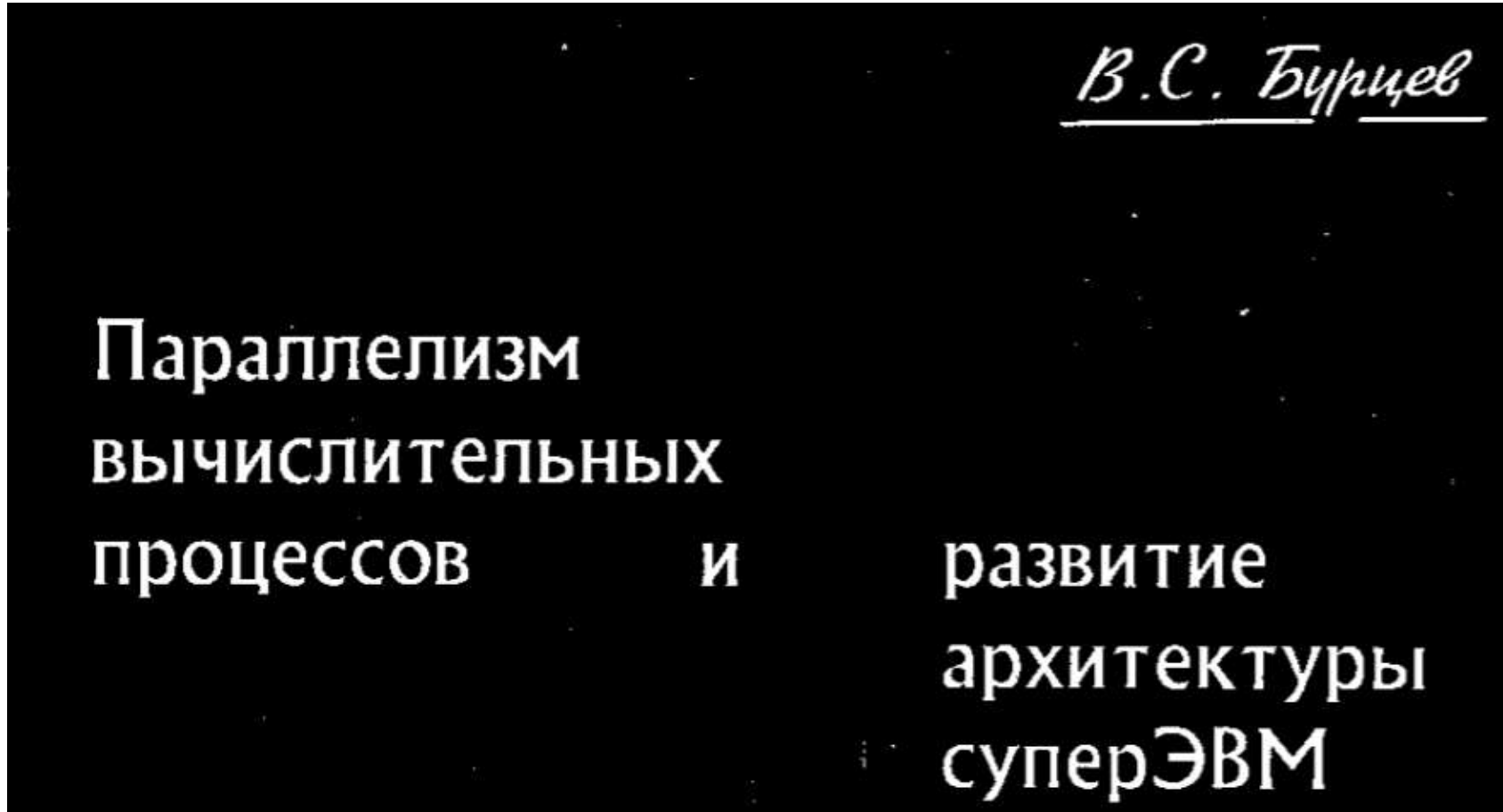
Далее эти положения рассматриваются на примере процессора В.С. Бурцева

Бурцев — разработчик основ параллельных вычислительных систем

Всёволод Сергеевич Бурцев ([11 февраля 1927](#) — [14 июня 2005](#)) — советский и российский учёный в области систем управления и теории конструирования универсальных [ЭВМ](#), академик [РАН](#). Основоположник создания первых многопроцессорных вычислительных комплексов «[Эльбрус](#)». Руководил разработкой [ЭВМ](#) для систем [противоракетной обороны СССР](#) «[А](#)», [А-35](#), [А-135](#) и [ПВО С-300](#)^[1]. Лауреат Ленинской премии и двух Государственных премий СССР



Юбилейный выпуск трудов В.С. Бурцева



Содержание см. на следующем слайде

В книге представлена часть трудов академика Всеволода Сергеевича Бурцева, написанных им за период с 1993 по 1996 годы. Часть работ публикуется впервые. В работах прослеживается основное направление исследований — создание нетрадиционных высокопараллельных архитектур вычислительных машин и комплексов с использованием новых физических принципов, в частности, оптических и оптоэлектронных устройств.

Юбилейный выпуск трудов В.С. Бурцева

Содержание

<i>Предисловие</i>	3
<i>В.С.Бурцев. Значение создания ENIAC в развитии информационно-вычислительных и управляющих систем России</i>	5
<i>В.С.Бурцев. О необходимости создания суперЭВМ в России</i>	18
<i>В.С.Бурцев. Новые подходы к оценке качества вычислительных средств</i>	28
<i>В.С.Бурцев. Выбор новой системы организации выполнения высокопараллельных вычислительных процессов, примеры возможных архитектурных решений построения суперЭВМ</i>	41
<i>В.С.Бурцев. Использование оптических методов обработки информации в архитектуре суперЭВМ</i>	79
<i>В.С.Бурцев, В.Б.Федоров. Ассоциативная память на принципах оптической обработки информации для суперЭВМ нового поколения</i>	105
<i>В.С.Бурцев, Л.Г.Тарасенко. Использование микропроцессоров традиционной архитектуры в системе потока данных</i>	121
<i>В.С.Бурцев. Особенности проектирования векторного исполнительного устройства в системе массового параллелизма с автоматическим распределением ресурсов</i>	140
<i>Основные этапы научной деятельности В.С.Бурцева</i>	148

С В.Бурцев, И. В. Шахнович, «СуперЭВМ в России, история и перспективы», журнал Электроника: НТБ № 4 2000 г.

СуперЭВМ в России

ИСТОРИЯ и ПЕРСПЕКТИВЫ



Рассказывает
академик РАН
В.С. БУРЦЕВ

Всеволод Сергеевич, каковы, на ваш взгляд, основные вехи развития отечественной высокопроизводительной вычислительной техники?

Первопроходцем вычислительной техники в Советском Союзе безусловно был Сергей Алексеевич Лебедев. Начинал он на Украине, где под его руководством в Институте электротехники АН Украины создали Малую электронную счетную машину (МЭСМ). В 1950 году мы пришли к нему на дипломную работу в Институт точной механики и

Высокопроизводительные вычислительные системы, суперЭВМ не зря называют форпостом компьютерной техники. Но они не только превосходят и определяют направления развития этой отрасли: суперкомпьютеры – один из краеугольных камней в экономической независимости и национальной безопасности государства. Однако в нашей стране прочность этой цитадели вызывает опасение. Мы встретились с одним из ведущих специалистов в области суперЭВМ, учеником и соратником С.А. Лебедева, создателем вычислительных комплексов для системы ПРО, ЗРК С-300 и многопроцессорного комплекса “Эльбрус” академиком Всеволодом Сергеевичем Бурцевым. Полувековой опыт успешной рабо-

Шахнович И.В. Работы по архитектуре сверхвысокопроизводительного компьютера продолжаются. ЭЛЕКТРОНИКА: Наука, Технология, Бизнес 2/2002. http://vbakanov.ru/dataflow/doc/2002_2_1.pdf

КОМПЬЮТЕРНАЯ И ИНФОРМАЦИОННАЯ ТЕХНИКА

РАБОТЫ ПО АРХИТЕКТУРЕ СВЕРХПРОИЗВОДИТЕЛЬНОГО КОМПЬЮТЕРА ПРОДОЛЖАЮТСЯ

Новая архитектура микропроцессоров рождается в России. “Опять утопия!” – скептически заметит умудренный проектами последних лет читатель. Однако не будем спешить со скоропалительными выводами.

И.Шахнович

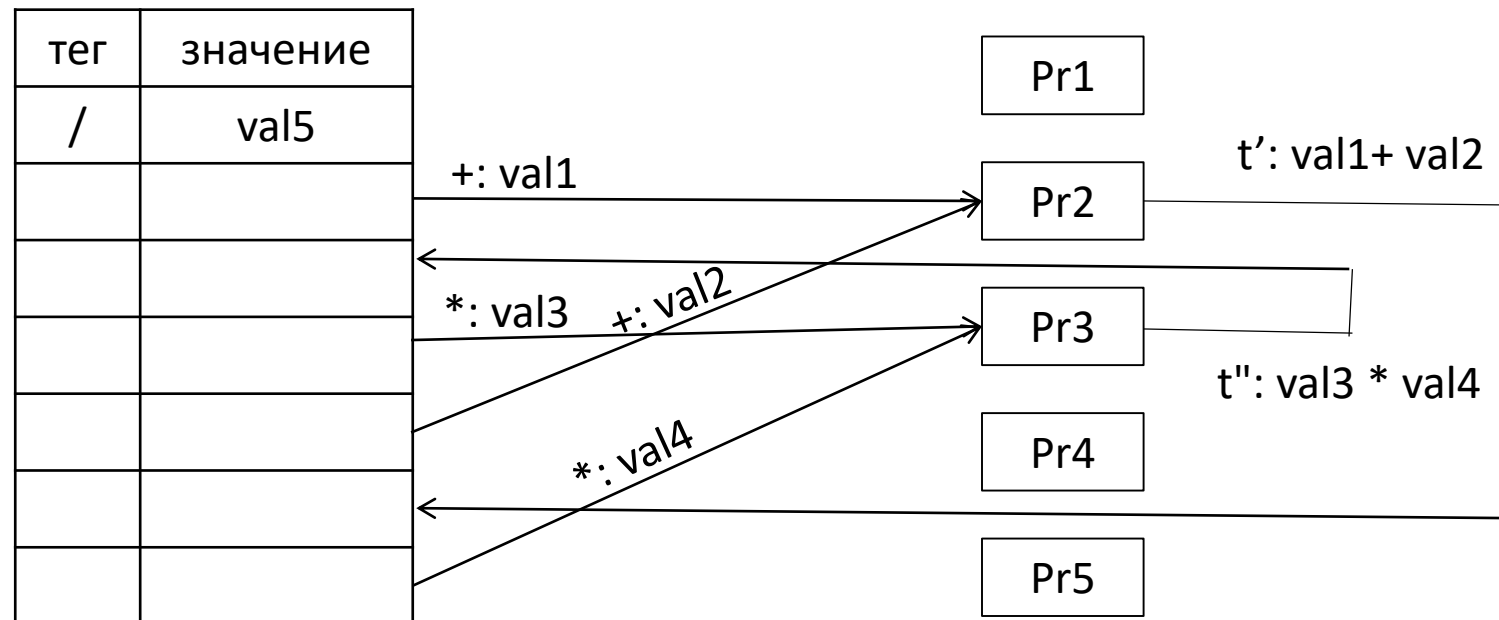
Начнем с истории. В середине 80-х годов в СССР начались работы по разработке новой архитектуры ЭВМ, основанной на новых физических принципах – программа “Основные направления фундаментальных исследований и разработок по созданию оптической сверхвысокопроизводительной машины Академии наук” (ОСВМ РАН). Организатором работ был Г.И. Марчук, тогда – председатель ГНТК, позднее – президент АН СССР. Научным руководителем работ выступил В.С. Бурцев – главный конструктор, в том числе многопроцессорных вычислительных комплексов “Эльбрус-1” и “Эльбрус-2”. Были привлечены специалисты многих институтов со всего СССР. До 1998 года работы велись сначала в ВЦКП РАН, затем – в Институте высокопроизводительных

центрах – государственных и частных – по всему миру, например в Массачусетском технологическом институте (процессор Tagget Token), лабораториях корпорации Texas Instruments (США), в Манчестерском университете (Англия). Однако дальше экспериментальных машин дело пока не пошло. Из наиболее известных data flow-систем следует отметить проекты Monsoon и Epsilon (США), CSRO-2 (Австралия). В чем суть метода?

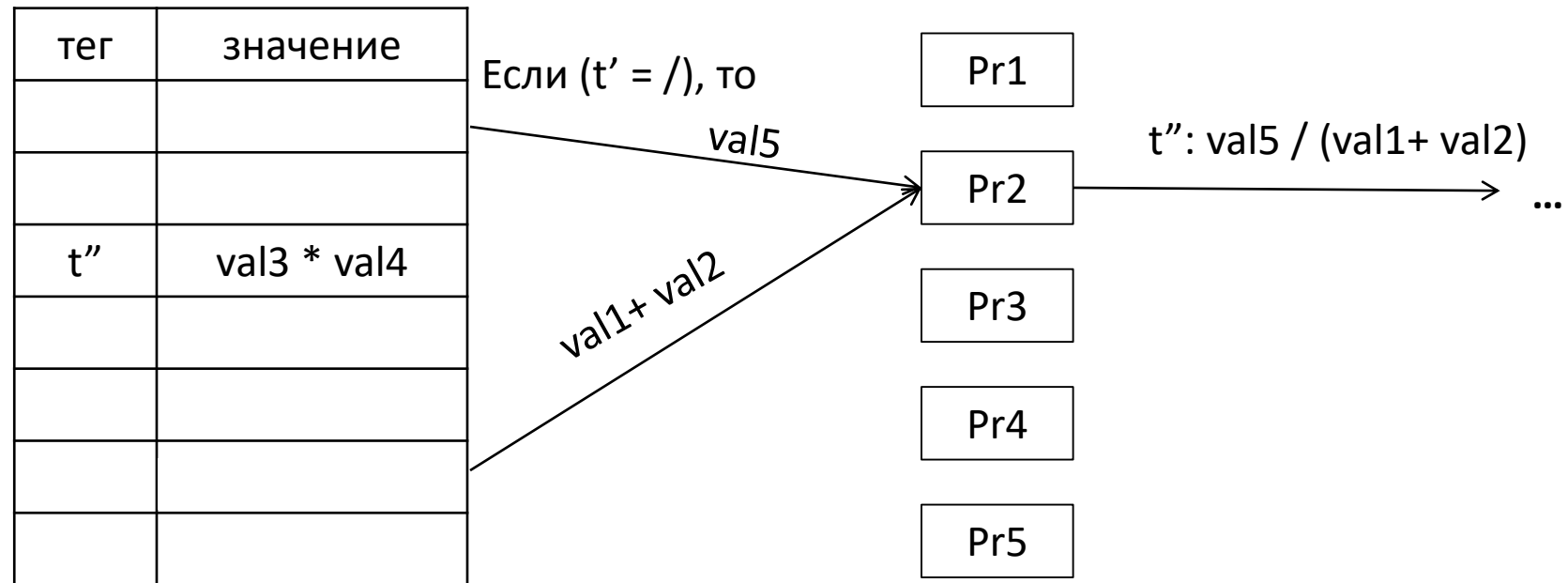
В системах управления потоком данных последовательность операций зависит от готовности операндов. Как только в памяти оказывается пара операндов для того или иного оператора, он немедленно инициируется на одном из нескольких исполнительных устройств (если есть свободные). При этом последователь-

Ассоциативные вычислительные системы (на примере процессора Бурцева)

- **Элементы системы** — активные данные: <ключ, называемый *тегом*, значение>
- **Спаривание элементов** с одинаковыми тегами \Rightarrow выполнение действия (заданного в **любой модели вычислений**)
- **Результат выполнения действия** — новые элементы
- Это одна из форм коммуникационной схемы.



Ассоциативные вычислительные системы (на примере процессора Бурцева)



Ассоциативные вычислительные системы (на примере процессора Бурцева)

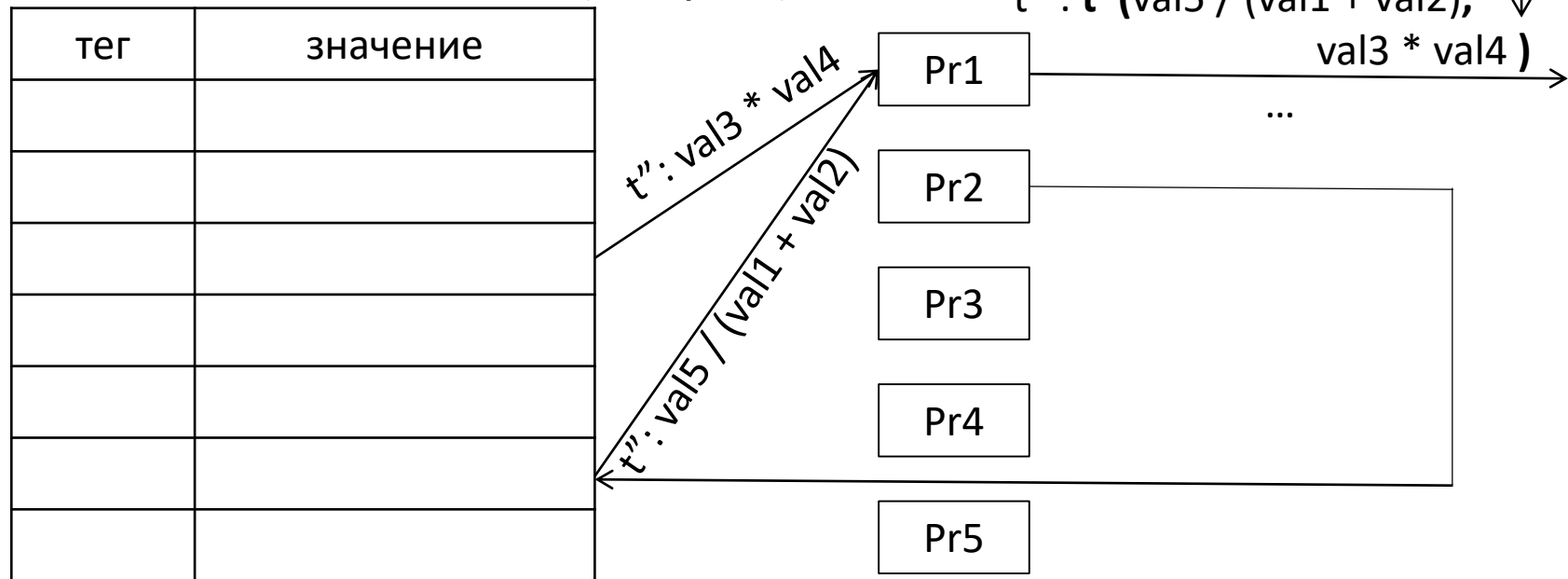
Это какие-то действия, связанные с вычислениями для тега t'' .

Они представлены в формате вызова функции $t''(arg1, arg2)$

Вопрос: Как задается порядок операндов?

Ответы (разные):

- Дополнительный тег, (спорно).
- Учет времени ожидания



Аксиоматические системы как попытка построить новую модель вычислений

- Элементы:
 - Аксиомы
 - Факты
 - Описание знаний на фиксированном языке
- Теоремы — предложения, из которых строятся программы
- Классическая логика – соотношение между данными не подходит →
 - Конструктивная логика
 - Вывод теорем (конструктивный!), т.е. новых фактов – программа (что здесь будет элементарным актом — ?)
 - Реализуемость – ?

Пример нарушения принципа обобщения без потерь:
переход от исчисления высказываний к исчислению предикатов

Вывод про попытку построить новую модель вычислений на основе аксиоматической системы терпит фиаско

Про совместное использование разных моделей

- PL1 — рождественская елка (Дейкстра): где сейчас PL1?
- История повторяется: язык Python
 - «В нем есть все, что надо, а для того, чего нет, строится обширный ряд полезных библиотек» — к этому есть вопросы...
- Пример (к язык Python имеет косвенные отношение)

```
var House : ... Color...           // описание типа, класса и др. для оперирования
                                   // со строениями, имеющих атрибут Цвет
```

...

```
print ('Этот дом — ' + if House . Color = Red
                        then 'пожарная станция'
                        else if House . Color = White
                              then 'больница'
                              else 'что-то другое'
                        fi
                        fi
fi );
```

- Текст не выдерживает никакой критики: не наглядно, громоздко, ...
- Почему?
 - Смешение стилей, относящихся к разным моделям вычислений

Разграничение моделей вычислений

1. Ассоциативная модель Бурцева:
2. Модель вычислений языка Рефал (сентенциальность + любая другая модель):

\underline{k} /детерминатив/ левая часть $\underline{_}$ → правая часть

\underline{k} и $\underline{_}$ — конкретизационные скобки

v_1, \dots, v_n — переменные, означиваемые при сопоставлении левой части с обрабатываемой строкой

Арифметические вычисления нерациональны:

$\underline{k}/\text{sum}/v_1+v_2.\underline{_}$ → $\underline{k}/\text{sum}/\underline{k}/\text{plus1}/v_1.\underline{_} +$

$\underline{k}/\text{minus1}/v_2.\underline{_}$

$\underline{k}/\text{sum}/v_1+1.\underline{_}$ → $\underline{k}/\text{plus1}/v_1.\underline{_}$

$\underline{k}/\text{sum}/v_1+0.\underline{_}$ → v_1

... ← нужны и другие правила

А как проще?

Обратиться к другой модели вычислений

тег	значение
/	val5
+	val1
*	val3
+	val2
*	val4

Ассоциативность

Pr1

Pr2

Pr3

Pr4

Pr5

Любые другие модели с эффективной реализацией арифметики

$\underline{k}/\text{sum}/v_1+v_2 \rightarrow \text{результат}.\underline{_}$

sum — имя внешней функции

v_1 и v_2 — входные параметры (для корректной работы должны быть означены как данные, соответствующие спецификациям sum)

результат — выходной параметр (приводится к строковому виду)

+, →, $\underline{_}$ и — можно рассматривать как оформление фактических параметров функции

Выводы

- Переход к новым моделям вычислений — неизбежность.
- Каждая из моделей максимально эффективна (во всех смыслах!) лишь для своих стилей программирования
- При реализации вычислений на «чужой» модели их эффективность применения снижается. Поэтому требуется решить задачу отображение используемой модели на модель реального вычислителя (традиционная для фоннеймановских вычислителей).
- Перспективные модели: функциональная и ассоциативная

Спасибо за внимание