

A diagram on the left side of the slide shows three horizontal white bars stacked vertically. Below each bar is a yellow arrow pointing upwards. The arrows increase in height from bottom to top, with the top arrow being the tallest and reaching the top bar. The middle arrow reaches the middle bar, and the bottom arrow reaches the bottom bar.

Университет города Переславля
им. А. К. Айламазяна

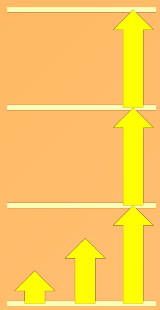
**Метавычисления
и их применение**
(слайды к циклу лекций)

Абрамов С.М.
Парменова Л.В.



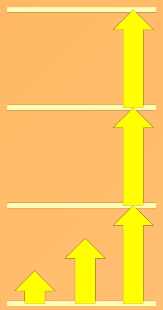
Литература и ссылки

- ★ <http://www.botik.ru/~abram/mca/>
- ★ http://www.botik.ru/~xsg/tsg_book
- ★ ftp://ftp.botik.ru/rented/xsg/www/tsg_book
- ★ Абрамов С.М. ``Метавычисления и их применение"//М., Наука-Физматлит, 1995, с. 127
- ★ Позлевич Л.В., Федосова М.А.
``Метавычисления и их применение.
Суперкомпиляция"//Выпускная работа на соискание степени бакалавра,
Филиал Российского университета дружбы народов, Переславль--Залесский, с. 68



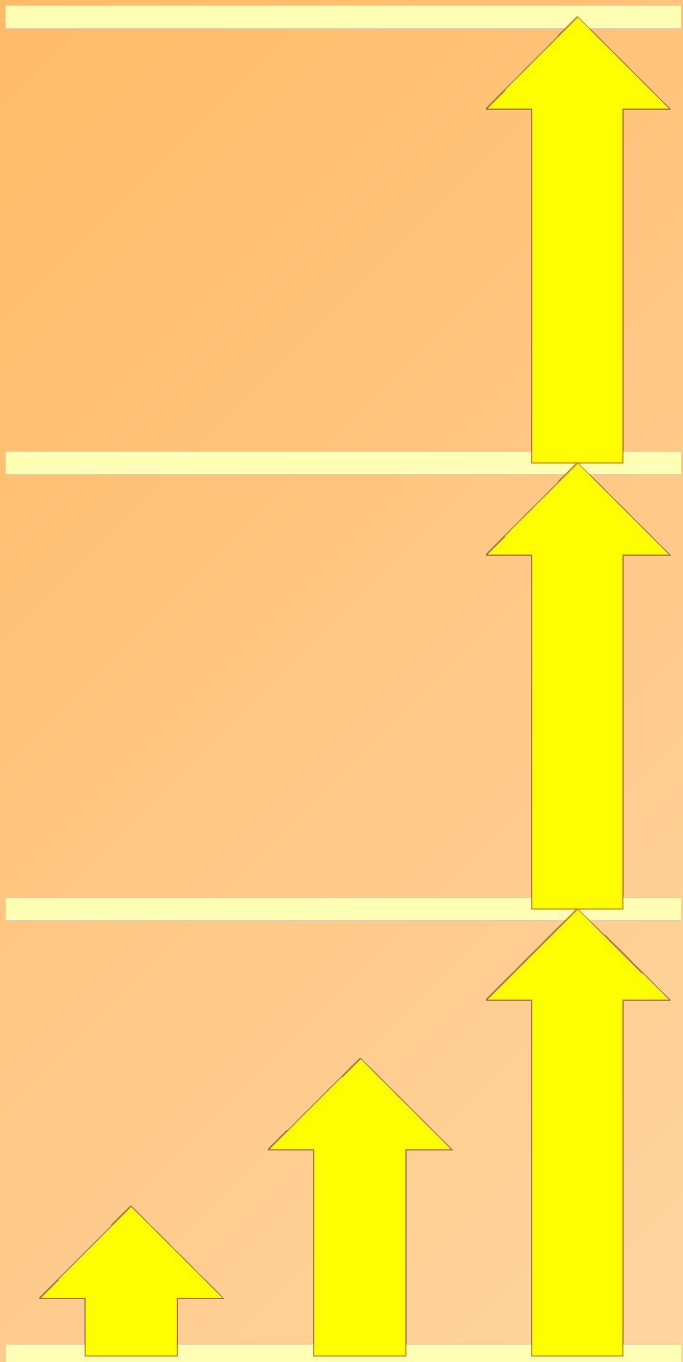
Литература и ссылки

- ★ Абрамов С.М. Метавычисления и их применение//Диссертация на соискание степени доктора физико-математических наук, ИПС РАН, декабрь 1995. с. 287
- ★ Sergei Abramov, Robert Glück, Semantics Modifiers: An Approach To Non-standard Semantics of Programming Languages'//FLOPS98 (<ftp://ftp.botik.ru/pub/local/Sergei.Abramov/FLOPS98.zip>)

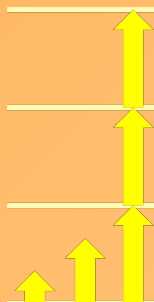


Контрольные мероприятия

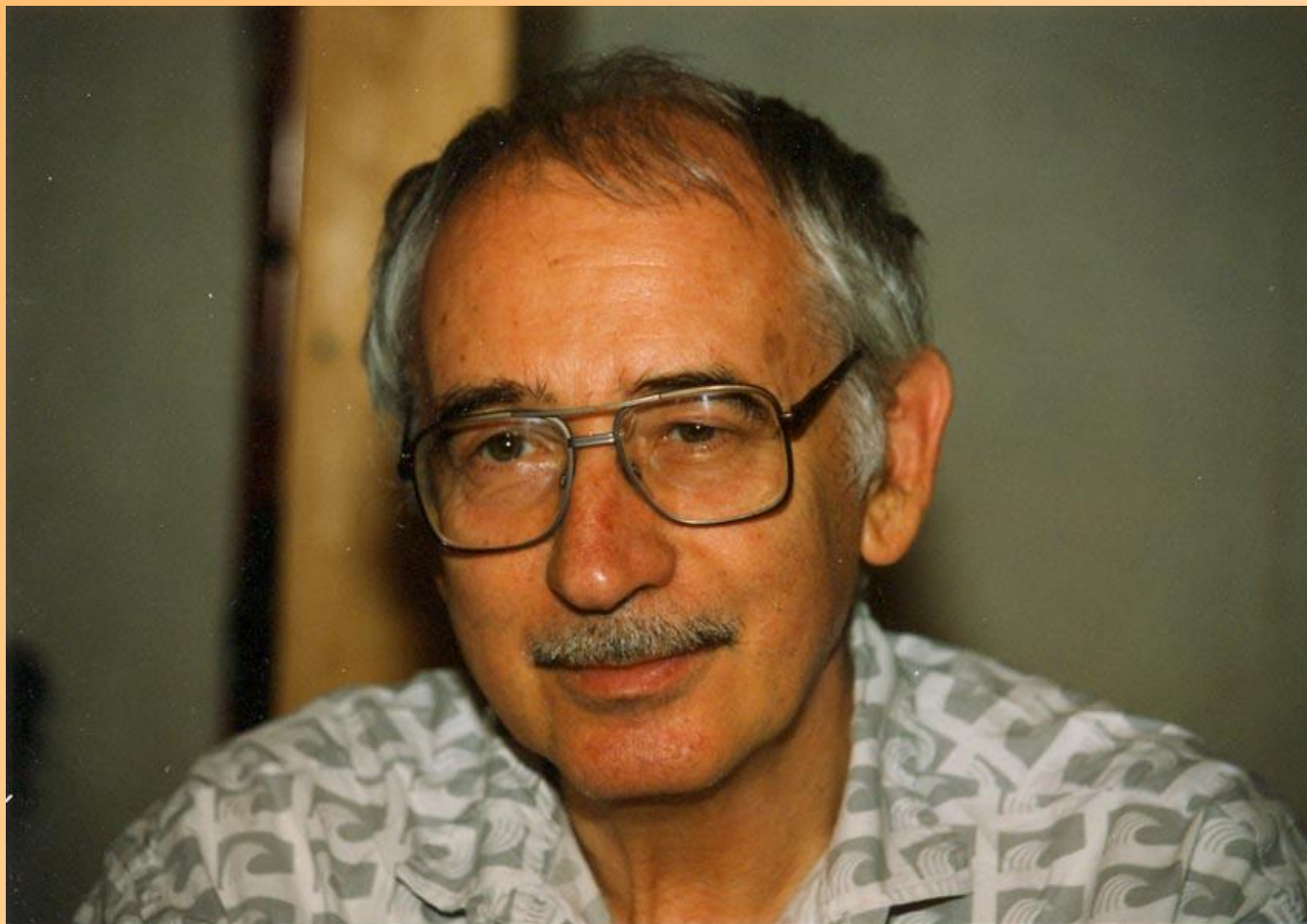
- ★ Написание тестов
 - ★ Задачи
 - ★ Рубежный контроль (1)
 - ★ Зачет (дифференцированный)
 - ★ Рубежный контроль (2)
 - ★ Экзамен
-
- ★ Ассистент(ы)
 - ★ Работа в botik.edu
 - ★ Посещаемость: старосты



Глава 1. Введение

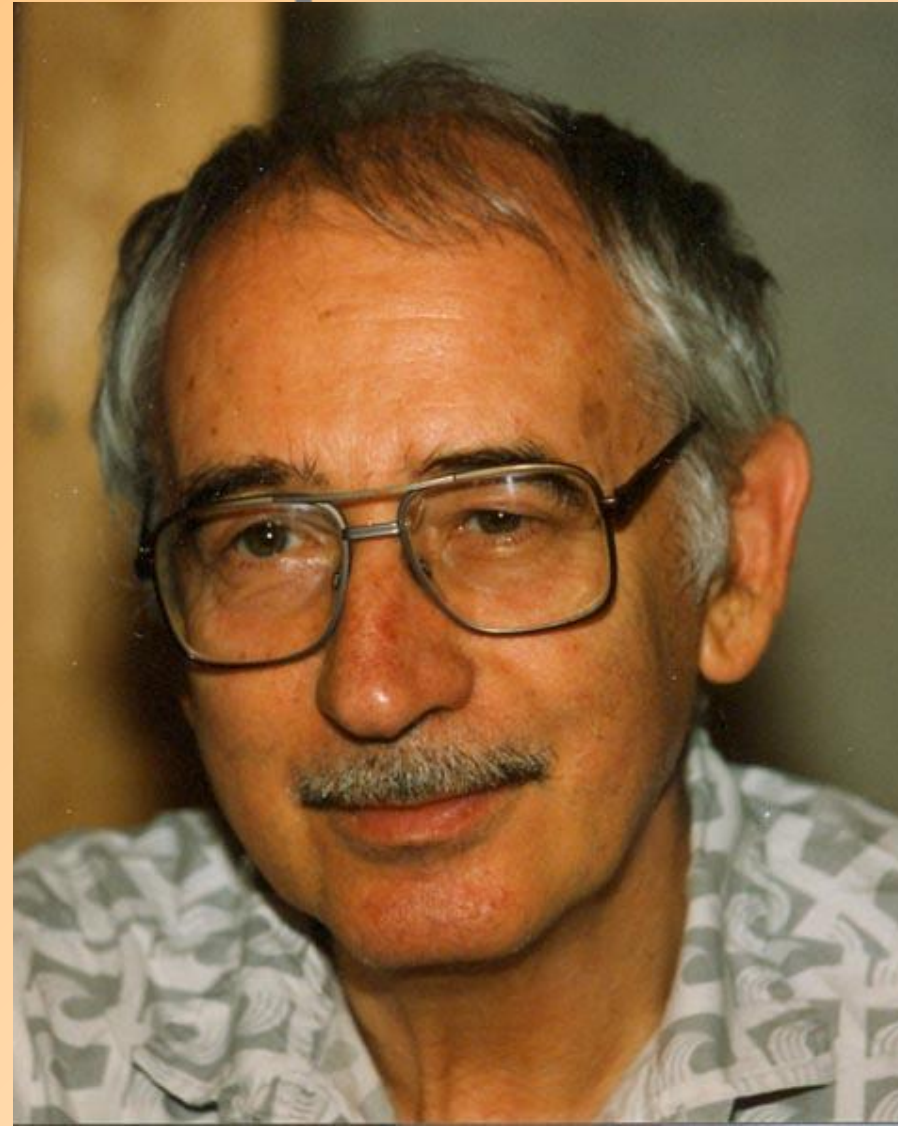


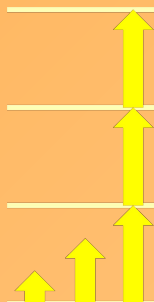
**В. Ф. Турчин — основатель
метавычислений**



В. Ф. Турчин. Краткая биографическая справка

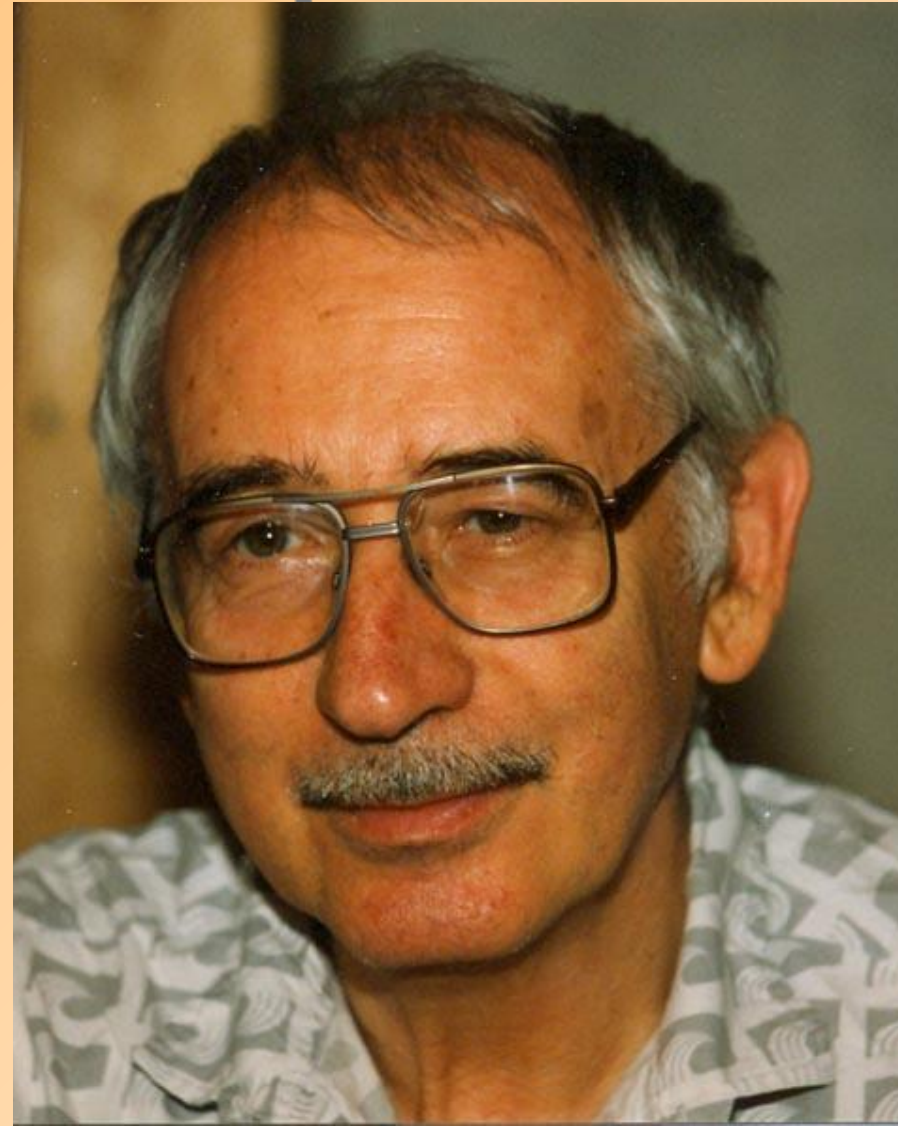
- ★ 14.02.1931–07.04.2010
- ★ физик-ядерщик
- ★ философ: «Феномен науки» — общая теория эволюции (обсуждается далее)
- ★ классик в области теоретических основ информатики, основатель метавычислений, создатель языка рефал





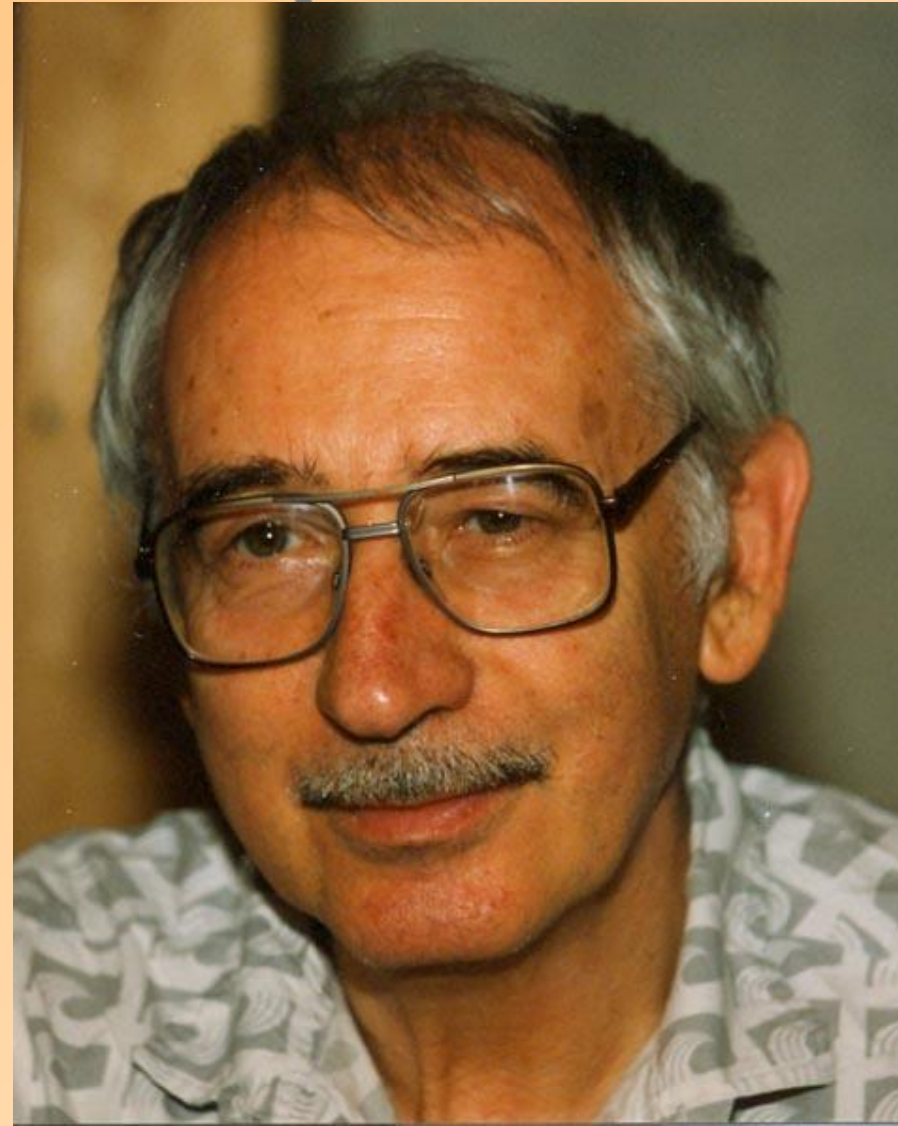
В. Ф. Турчин. Краткая биографическая справка

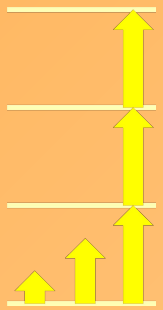
- ★ наставник молодых: Колмогоровский интернат, рефал-мальчики, рефал-семинары
- ★ популяризатор науки (журнал «Знание — сила», сборники «Физики шутят» и «Физики продолжают шутить»)



В. Ф. Турчин. Краткая биографическая справка

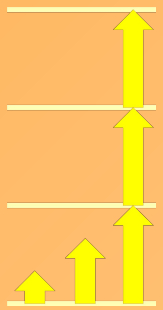
- ★ человек высокой совести, гражданин,
- ★ диссидент, правозащитник, лидер хельсинской группы в Москве
- ★ Книга В.Ф.Турчина «Инерция страха»
- ★ выдворение из СССР (1977)
- ★ персональная «железная занавесь»





Теория метасистемных переходов (MST)

- ★ «Общая теория эволюции» В.Ф.Турчина:
 - ❑ квант эволюции — метасистемный переход;
 - ❑ эволюция — цепочка метасистемных переходов
 - ❑ общие черты метасистем и метасистемных переходов
 - для метасистем объектами (анализа, преобразования, управления...) являются системы
 - часто метасистемы работают с множествами (объектов)
 - ❑ условия для метасистемного перехода
 - ❑ предпосылки неограниченной эволюции
 - ❑ системы с неограниченными возможностями развиваться
 - ❑ системы, обреченные жить без развития

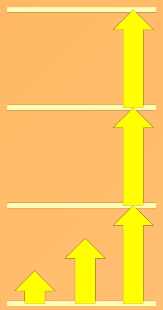


Метавычисления: применение MST к программированию

★ Основная идея:

- чтобы обеспечить эволюцию в мире программ, используем в программировании теорию эволюции В. Ф. Турчина;
- результат этого: метавычисления.

★ Метапрограммы — конструктивные метасистемы над программами



Метавычисления: применение MST к программированию

- ★ Общие черты метапрограмм заимствованы из MST — повторяют общие черты метасистем:
 - ❑ метапрограммы работают над программами: анализ, преобразование, (нестандартные) вычисления и т.п.
 - ❑ метапрограммы работают с множествами (с их представлениями)
 - ❑ можно реализовать условие «неограниченной цепочки метавычислений»: возможность совершения метасистемного перехода над метапрограммой — применение метапрограммы к метапрограмме

Простейшие примеры

метасистемных переходов

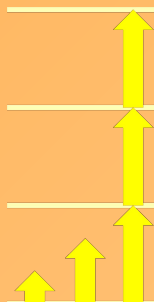
★ Программа p из языка L ($p \in P_L$) на данных $d \in D$ дает результат $r \in D$: $p d *_{L} r$

★ Что такое **(L/R)-интерпретатор** int языка L , написанный на языке R ? Мнение?

$$\text{int} \in P_R: \begin{array}{l} \square p \in P_L \quad \square d \in D \\ \text{int}(p, d) *_{R} r \quad \square p d *_{L} r \end{array}$$

★ Что такое **(L1→L2/R)-компилятор** comp с языка $L1$ на язык $L2$, написанный на языке R ? Мнение?

$$\text{comp} \in P_R: \begin{array}{l} \square p1 \in P_{L1} \quad \text{comp } p1 *_{R} p2 \in P_{L2} \\ \square d \in D \quad p1 d *_{L1} r \quad \square p2 d *_{L2} r \end{array}$$



Простейшие примеры метасистемных переходов

- ★ Даже простейшие метапрограммы позволяют достичь серьезных результатов
- ★ Назовем $(R \rightarrow R/R)$ -специализатором $s \in P_R$:
 - $\square p \in P_R \quad \square d_1 \in D \quad s(p, d_1) * \square_R p_1 \in P_R :$
 - $\square d_2 \in D \quad p(d_1, d_2) * \square_R r \quad \square p_1 d_2 * \square_R r$
- ★ Коротко (**основное свойство специализатора**):
 - $\square p \in P_R \quad \square d_1, d_2 \in D \quad p(d_1, d_2) = s(p, d_1) d_2$
- ★ Тривиальный и нетривиальный специализатор: как отличить? Touch stone будет на следующем слайде

Простейшие примеры метасистемных переходов

★ $p(d_1, d_2) = s(p, d_1) d_2$

★ $p_L d \text{ * } \square_L r_R \text{ * } \text{int}(p_L, d) =$

результат (L→R)-
компиляции p_L

(L→R/R)-
компилятор

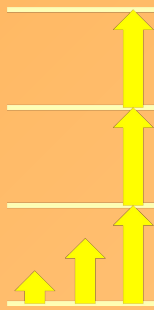
генератор $s(\text{int}, p_L) d =$

компиляторов $s(s, \text{int}) p_L d =$

$s(s, s) \text{int } p_L d$

★ *Проекция Футамуры-Турчина*

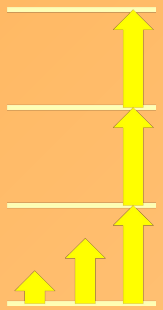
★ Дистанция в ~15 лет между «это теоретически возможно (1971)» и «это сделано на практике (1985)». Touch stone для специалистов.



История проекций Футамуры-Турчина

- ★ Футamura открыл первые две проекции
 - Y. Futamura: Partial Evaluation of Computation Process — An approach to a Compiler-Compiler, 1971
- ★ Турчин независимо от Футамуры открыл (1972...) все три проекции, опубликовать не мог
 - Первое сообщение об этом: в работе академика Ершова
- ★ Первая «неразборчивая» реализация: Neil D. Jones, DIKU, 1985
- ★ Более совершенная реализация: С.А. Романенко, 1987
- ★ Компьютерра «Турчин и другие...»
 - <http://offline.computerra.ru/2001/402/10913/>

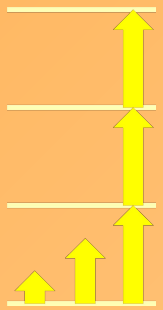




Другие примеры

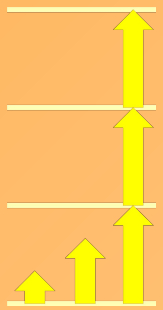
метасистемных переходов

- ★ В нашем курсе мы рассмотрим более сложные метасистемы (многоуровневые метасистемы), более сложные примеры метасистемных переходов, более сложные проекции (например, для нестандартных семантик).



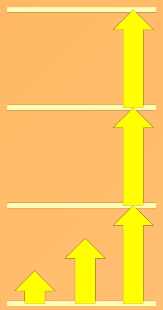
Метавычисления

- ★ **Метавычисления:** раздел программирования, посвященный разработке методов анализа и преобразования программ за счет реализации конструктивных метасистем (метапрограмм) над программами.
- ★ Программы в метавычислениях рассматриваются как **объект** анализа и/или преобразования.



Базовые идеи

- ★ **применение теории метасистем и метасистемных переходов** к программам
- ★ **процесс-ориентированный подход** к построению методов анализа и преобразования программ
 - разработка метапрограмм **M**, которые «наблюдают» за **процессами** вычисления исходной программы **p** (на одиночных данных **d**, на классах данных **C**) и управляют этими процессами.
- ★ фиксация языка реализации **R**
 - на нем должны быть написаны все программы **p**, к которым будут применяться метапрограммы
- ★ если некоторая метапрограмма **M** написана на **R**, то к ней применима она сама (самоприменимость) или другие метапрограммы (метасистемная лестница)



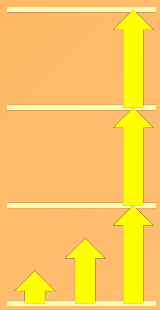
Цель исследований

- ★ Реализация средств анализа программ и средств выполнения преобразований программ
 - эквивалентных преобразований (например, оптимизаций) и
 - построения новых программ, функции которых сложным образом определяются через функции исходных программ



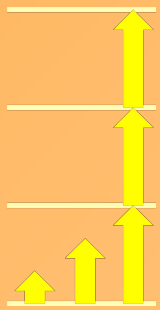
Мотивация

- ★ В.Ф.Турчин “Феномен науки”:
- ★ *«Метод современной науки по своему существу есть не что иное, как построение формальных лингвистических моделей изучаемых явлений»...*
- ★ *«Очередной шаг развития лингвистического моделирования, то есть очередной метасистемный переход в эволюции науки — использование компьютеров в процессе построения формальных лингвистических моделей»...*
- ★ *«И это может быть сделано при помощи использования методов метавычислений»*
- ★ Сегодняшние результаты убеждают, что данная роль метавычислений действительно может быть осуществлена на практике при условии дальнейшего развития методов метавычислений и методов их применения.



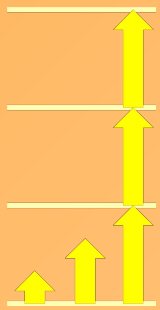
Мотивация

- ★ В области программирования речь идет о создании мощных средств автоматического анализа и преобразований программ и методов применения этих средств
 - *Автоматизация программирования*
 - *Обеспечение надежности ПО*



Мотивация

- ★ Стремление к скорейшему завершению создания теории суперкомпиляции и практической реализации суперкомпилятора привело к тому, что некоторые понятия и методы метавычислений, некоторые вопросы применения метавычислений в программировании были недостаточно полно развиты, исследованы, обоснованы и изложены.
 - *До сих пор актуальная задача — закрыть данный пробел*



Мотивация

- ★ Близкие технологии (частичные вычисления, дефорестация, расширенная дефорестация, обобщенные частичные вычисления и т. д.) упрощения суперкомпиляции и метавычислений
- *Исследователи, ранее занимающиеся данными направлениями, проявляют все больший интерес к метавычислениям*

Структура курса



[1] С.М.Абрамов «Метавычисления и их применения»

[2] Л.В. Парменова «Метавычисления и их применения. Суперкомпиляция»