

# Параллельное программирование на языке Рефал

Юрий Климов (ИПМ РАН)  
Антон Орлов (ИПС РАН)

<http://rfp.botik.ru>

# План

- Цели и задачи
- Язык Рефал
- Средства распараллеливания

# Очень трудоёмкие задачи

Например:

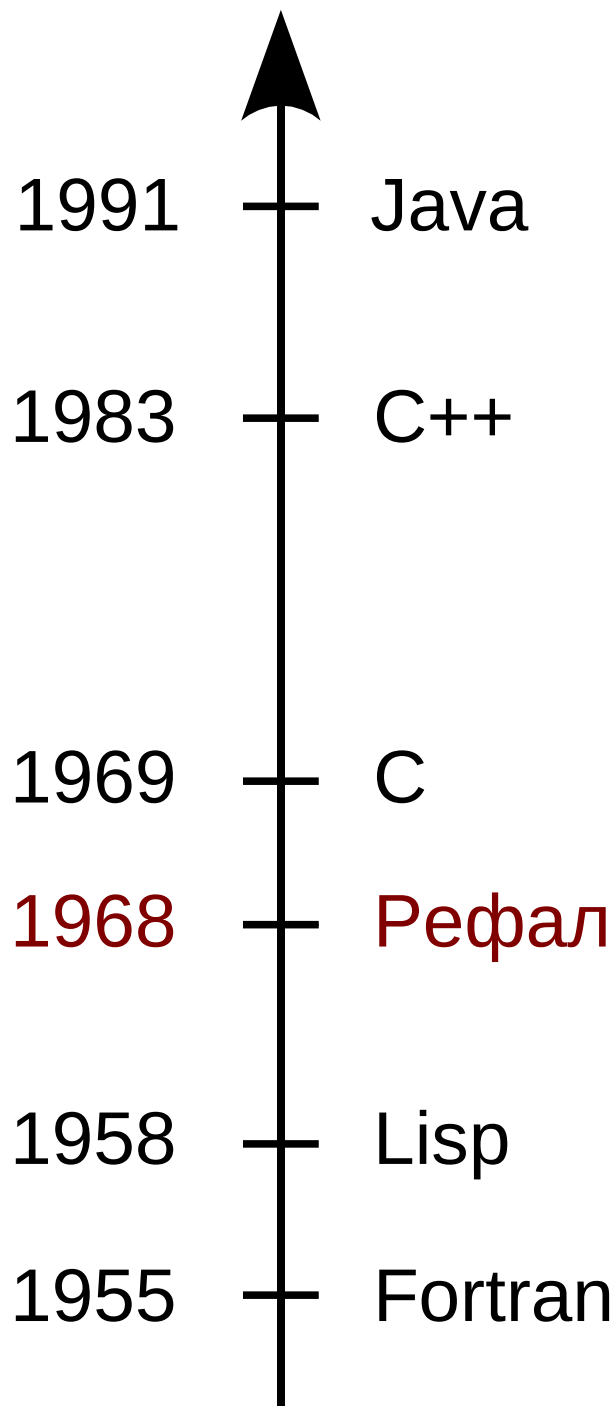
- Обработка текстов на естественных языках
- Data mining
- Семантический анализ и преобразование программ (суперкомпиляция, инверсные вычисления, метавычисления в целом)
- Символьные вычисления

# Цели

- Дать функциональным программистам возможность использовать суперкомпьютеры
- Единая программа: один раз написал - работает везде
- Универсальный инструмент важнее производительного
- Производительность программиста важнее эффективности программы

# Высокоуровневость облегчает распараллеливание

- Низкоуровневая программа заточена под конкретную платформу
- Высокоуровневая программа более абстрактна, её легче анализировать
- Функциональная программа содержит неявный параллелизм



# Рефал —

**РЕ**курсивный

**Ф**ункциональный

**АЛ**горитмический язык

Автор: В. Ф. Турчин

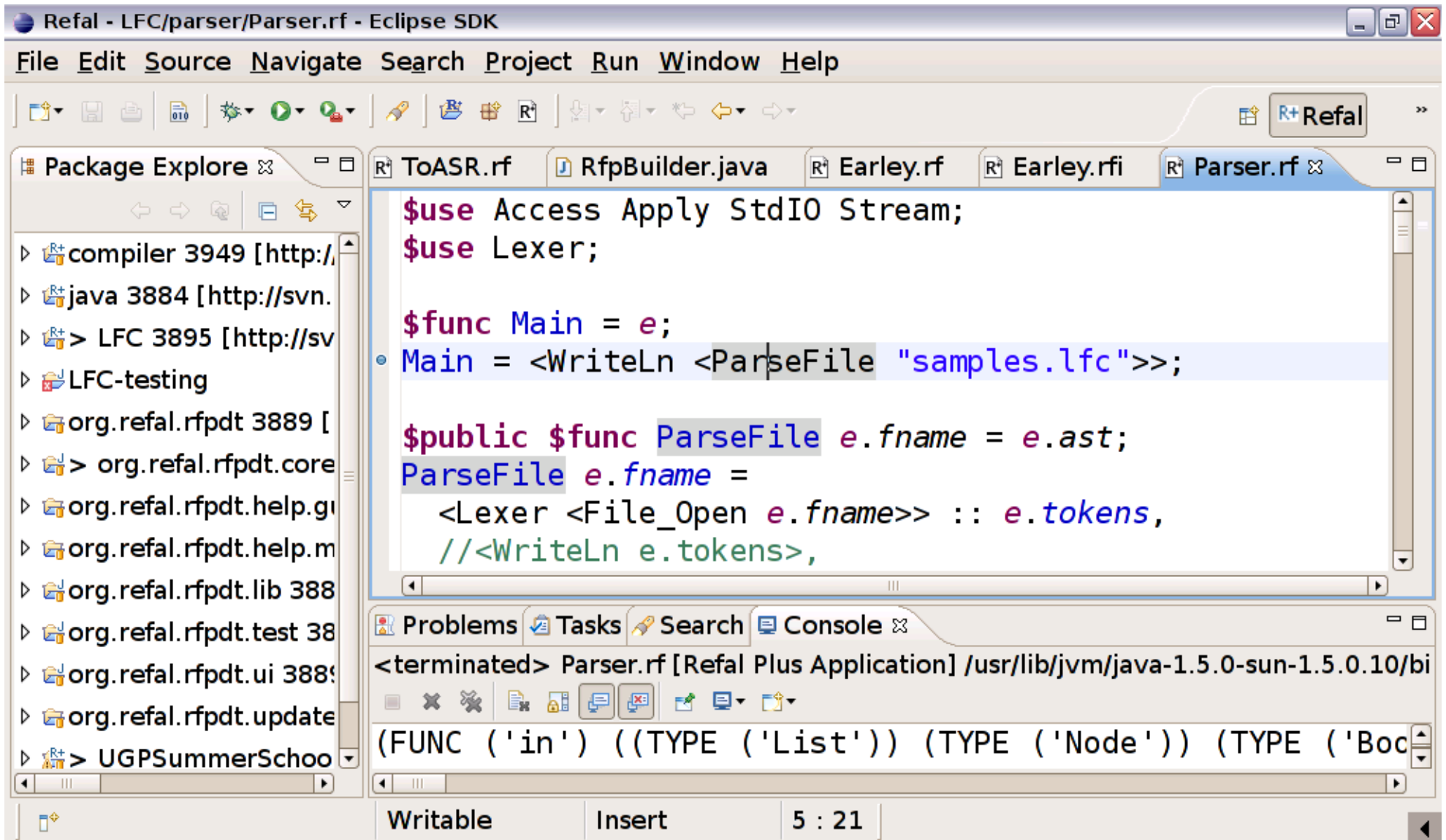
## Диалекты:

- Рефал-2 — 1986
- Рефал-5 — 1989, 2000
- Рефал-6 — 1992, 2004
- Рефал+ — 1991, 2006

# Рефал

- Функциональный язык («чистые» функции)
- Уникальная основная структура данных — рефал-выражение
- Удобен для задач обработки символьной информации
- Хорошая интеграция с языками C/C++/Java
- Современная среда разработки (IDE)

# Среда разработки





# Параллельное программирование в функциональном стиле

## **За:**

- Чистые функции можно вычислять параллельно — нет обмена информацией

## **Против:**

- Предполагается, что вызов функции дешёв, что не так на системах с распределённой памятью
- Отсюда необходимость разделения на дешёвые и дорогие вызовы

# T-Система OpenTS

- Система автоматизированного динамического распараллеливания программ
- Надстройка над C/C++ — несколько дополнительных ключевых слов:  
**tfun, tval, tout, ...**
- Гранулы параллелизма — чистые функции (*m-функции*)

# Параллельный Рефал

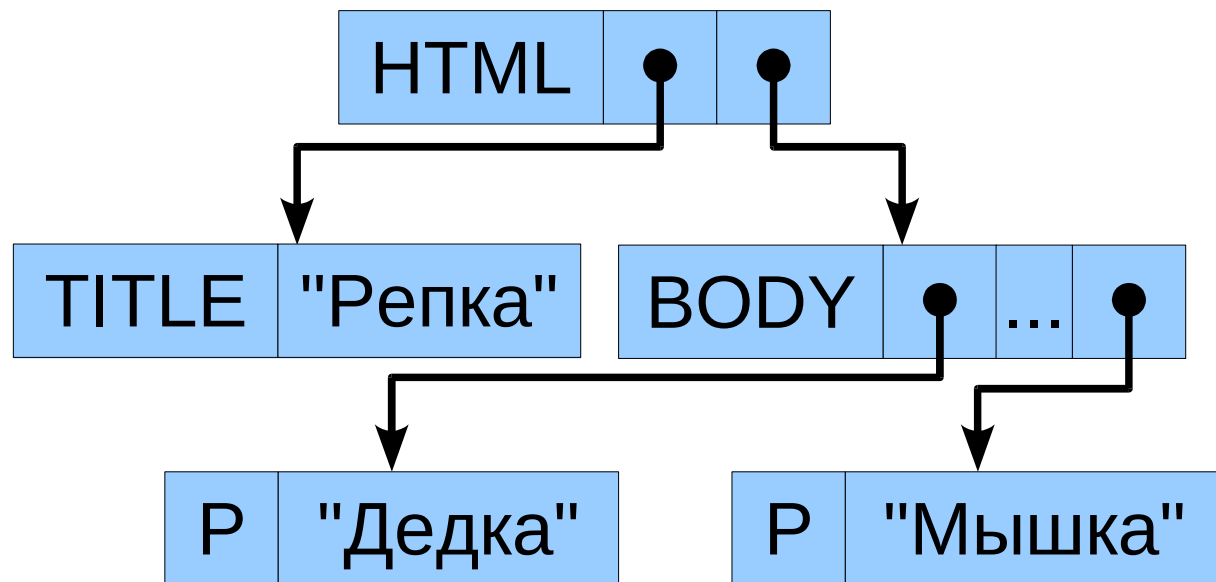
- Компилятор Рефала+ в T++ — входной язык OpenTS
- Функции переходят в функции
- Рефал-выражение — в распределенную структуру данных

# Рефал-выражение

- Последовательность символов с правильно расставленными скобками
  - Выражение — последовательность термов
  - Терм — символ или выражение в скобках

- Пример:

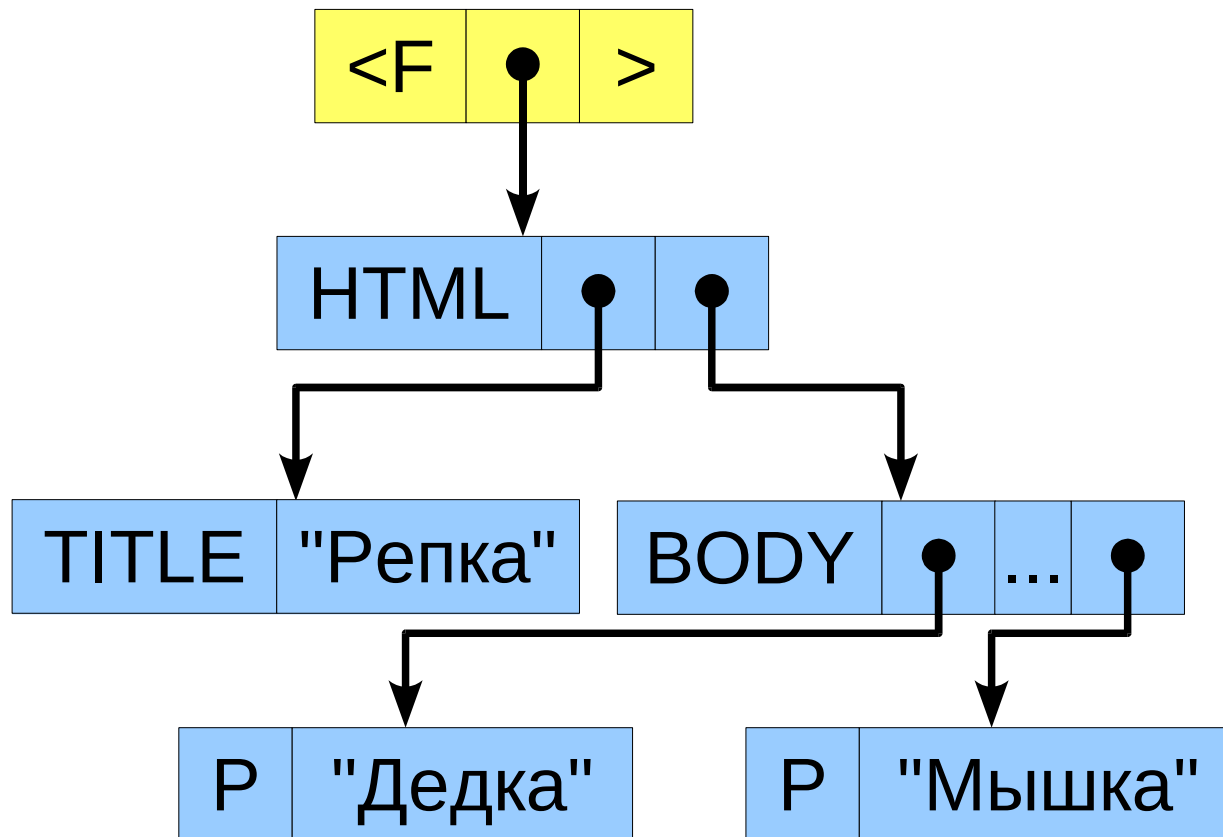
```
(HTML  
  (TITLE "Репка")  
  (BODY  
    (P "Дедка")  
    ...  
    (P "Мышка")  
  )  
)
```



# Распределенное Рефал-выражение

(HTML (TITLE "Репка")  
(BODY (P "Дедка") ... (P "Мышка"))) )

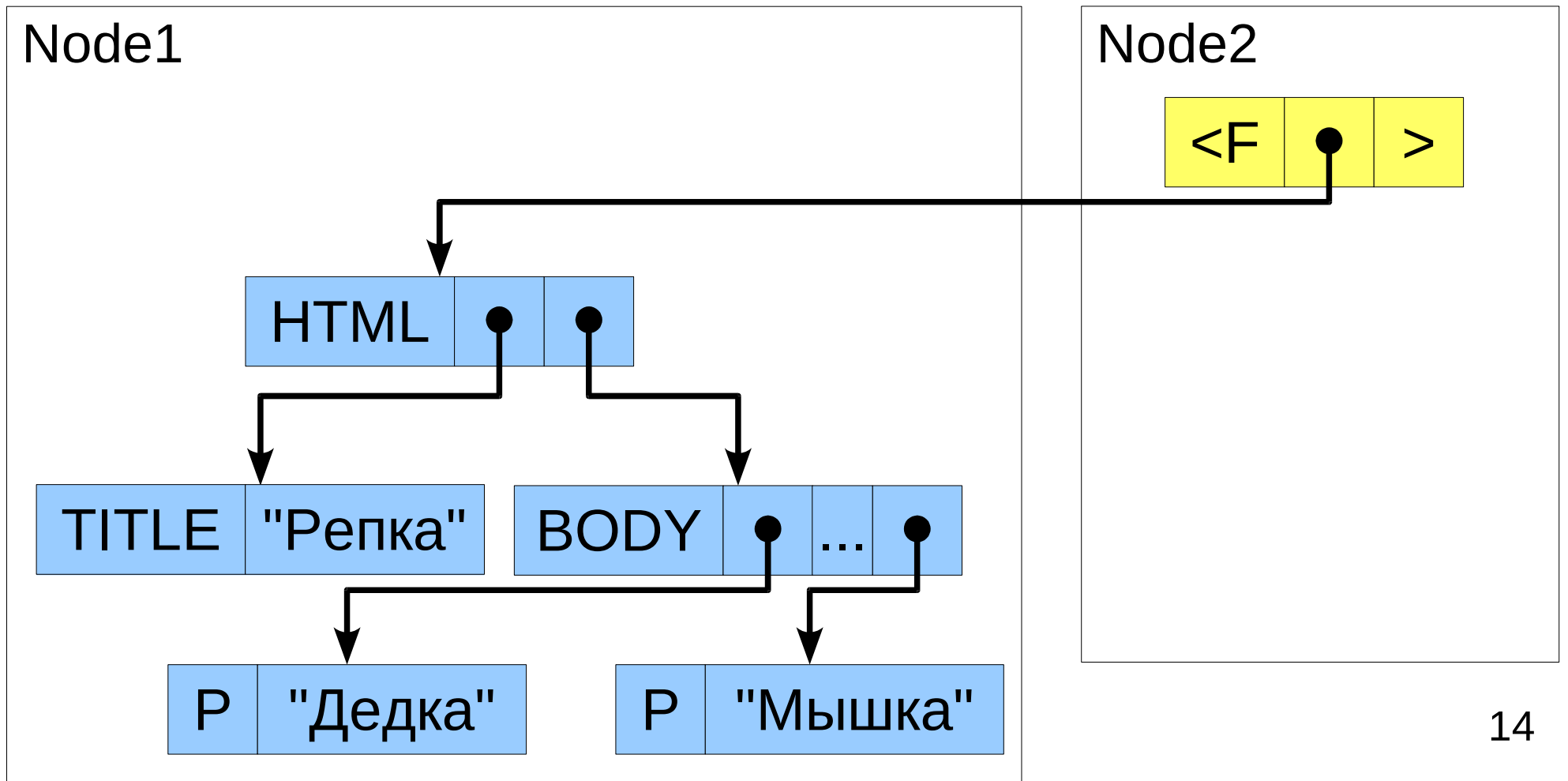
Node1



Node2

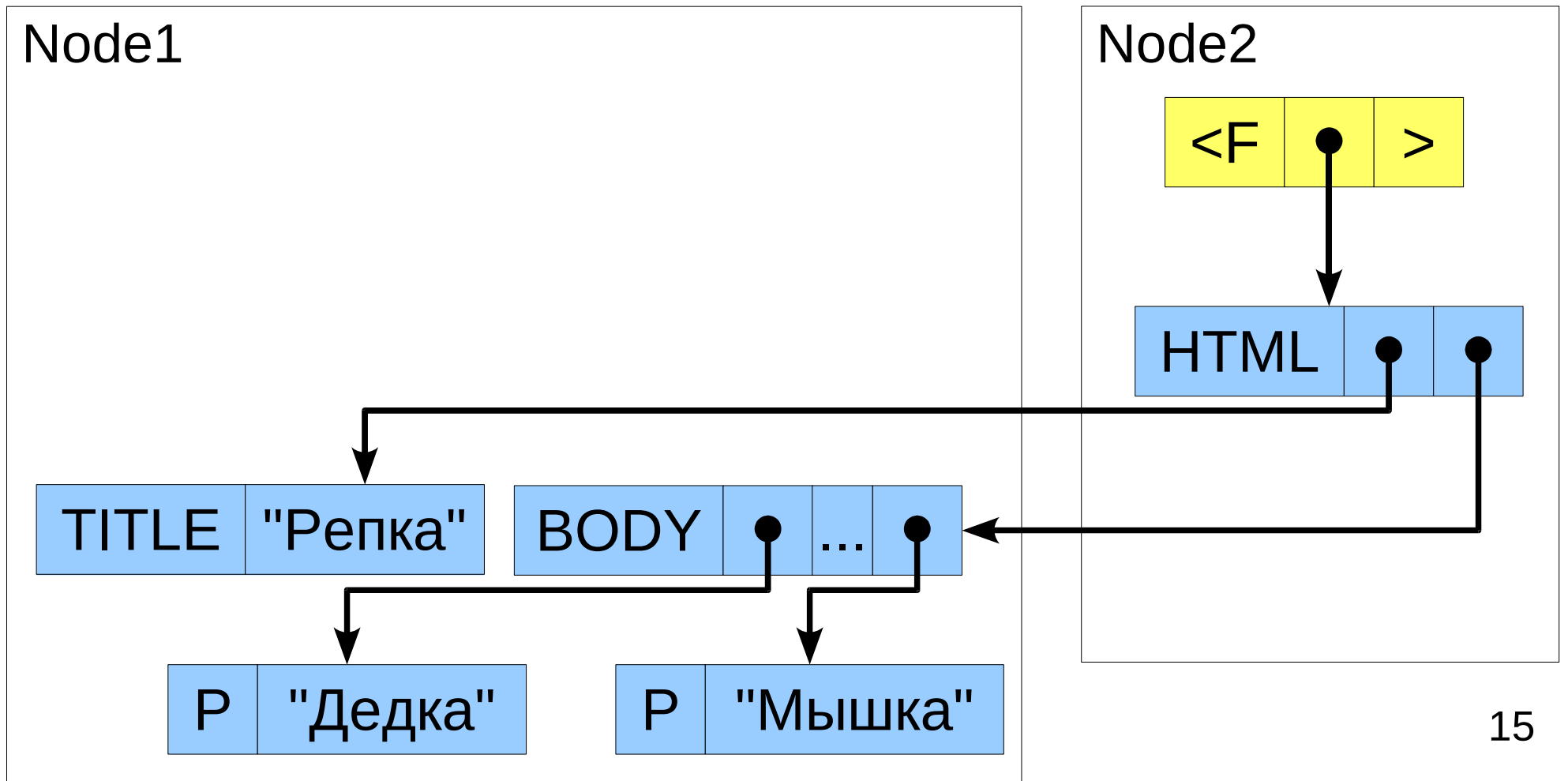
# Распределенное Рефал-выражение

(HTML (TITLE "Репка")  
(BODY (P "Дедка") ... (P "Мышка"))) )



# Распределенное Рефал-выражение

(HTML (TITLE "Репка")  
(BODY (P "Дедка") ... (P "Мышка"))) )



# Планы на будущее

- Аннотирование функции оценкой времени её работы в зависимости от структуры аргументов
- Статистическая оценка времени вычисления функции в зависимости от размера выражения



# Спасибо!

<http://rfp.botik.ru>

Вопросы?