



# Разработка средств реинжиниринга

## Студенческий проект

Руководитель: Григорьев Семён Вячеславович

Санкт-Петербургский государственный университет  
Математико-Механический факультет  
Кафедра системного программирования

13 мая 2013г.

- Название: YaccConstructor
- Сайт проекта: <http://recursive-ascent.googlecode.com>
- YaccConstructor
  - ▶ Модульный инструмент для разработки парсеров и трансляторов для платформы .NET.
  - ▶ Реализован на F#.
  - ▶ Основная область применения – реинжиниринг программного обеспечения.

- Разработка парсера (подмножества) языка T-SQL.
- Апробация YaccConstructor.
- Воспроизвести процесс разработки, характерный для реинжиниринга:
  - ▶ разработка по доступной документации
  - ▶ цель – разбор конкретного исходного кода

# Особенности реализации

- В исходной документации присутствуют неточности, которые нужно обрабатывать вручную
- Правила из документации нужно переносить по мере необходимости
- Для этого важны:
  - ▶ мощный синтаксис языка спецификации трансляций
  - ▶ алгоритм синтаксического анализа – GLR

- Рекомендации по доработке YARD/YaccConstructor:
  - ▶ Дополнительные средства отладки
  - ▶ Необходима полноценная IDE для языка YARD
  - ▶ Нужен контроль качества кода на уровне парсера
  - ▶ light-синтаксис (F#)
- Грамматика подмножества T-SQL
  - ▶ Основные управляющие конструкции (while, if then else, ...)
  - ▶ DML (Select, update)
  - ▶ Парсер тестировался на 2 млн строк кода. (Уникальных процедур – 11, 1200 строк кода).

# Задача

- Создание библиотеки привязки к исходному коду:
  - ▶ Функции сжатия и разжатия тройки координат (id, строка, колонка) и пары координат (id, смещение в файле)
  - ▶ Класс, определяющий биекцию:
    - ★ между id и именем файла
    - ★ между id и картой переносов
  - ▶ Метод, позволяющий по строке и колонке определить абсолютный отступ (линейное время)
  - ▶ Метод, позволяющий по абсолютному отступу определить строку и колонку (логарифмическое время)
  - ▶ Единицы измерения для данных
- Разработка тестов, апробация на парсере T-SQL

## Использование на примере парсера T-SQL

- Сообщение об ошибке: координаты начала и конца
- Сокращение объема памяти (за счет сжатия координат):
  - ▶ Файл: 11 Мб, 300 тыс. строк
  - ▶ Расход оперативной памяти:
    - ★ до сжатия: 665 Мб
    - ★ после сжатия: 450 Мб
  - ▶ Сокращения объема использования оперативной памяти: 32%

- Бакаева Алиса. Использование грамматик с весами для автоматической обработки диалектов.
- Иванов Андрей. Восстановление после ошибок в GLR-алгоритме.
- Алефиров Алексей. Декомпиляция MSIL в F#.



- В рамках проекта выполняются курсовые работы.
- Участники проекта успешно выступили на конференции "Технологии Microsoft в теории и практике программирования 2013"

# Демонстрация

```
D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql\bin\Debug - Far 20107 x64

D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql>ast.dot.svg

D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql>dot ast.dot -O -Tpng

D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql\bin\Debug>MsSql.exe -f D:\projects\yc\recursive-ascend\Tests\materials\ms-sql\sysprocs\test.sql

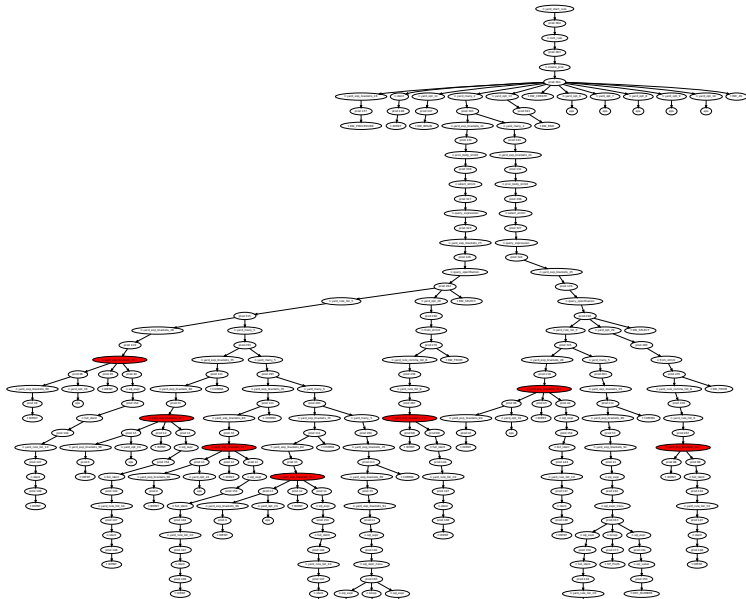
tkm# 50000 Tks/s: 12790 - p
tkm# 100000 Tks/s: 98814 - p
tkm# 150000 Tks/s: 91575 - p
tkm# 200000 Tks/s: 100401 - p
tkm# 250000 Tks/s: 99601 - p
tkm# 300000 Tks/s: 98231 - p
tkm# 350000 Tks/s: 97847 - p
tkm# 400000 Tks/s: 82372 - p
tkm# 450000 Tks/s: 100000 - p
tkm# 500000 Tks/s: 98814 - p
tkm# 550000 Tks/s: 99206 - p
tkm# 600000 Tks/s: 99601 - p
tkm# 650000 Tks/s: 98425 - p
tkm# 700000 Tks/s: 98039 - p
tkm# 750000 Tks/s: 97465 - p
tkm# 800000 Tks/s: 94161 - p
tkm# 850000 Tks/s: 81833 - p
tkm# 900000 Tks/s: 98231 - p
tkm# 950000 Tks/s: 97847 - p
tkm# 1000000 Tks/s: 96339 - p
tkm# 1050000 Tks/s: 99403 - p
tkm# 1100000 Tks/s: 99403 - p
tkm# 1150000 Tks/s: 96153 - p
tkm# 1200000 Tks/s: 97656 - p
tkm# 1250000 Tks/s: 97276 - p

Time for parse file D:\projects\yc\recursive-ascend\Tests\materials\ms-sql\sysprocs\test.sql = 00:00:17.0539754

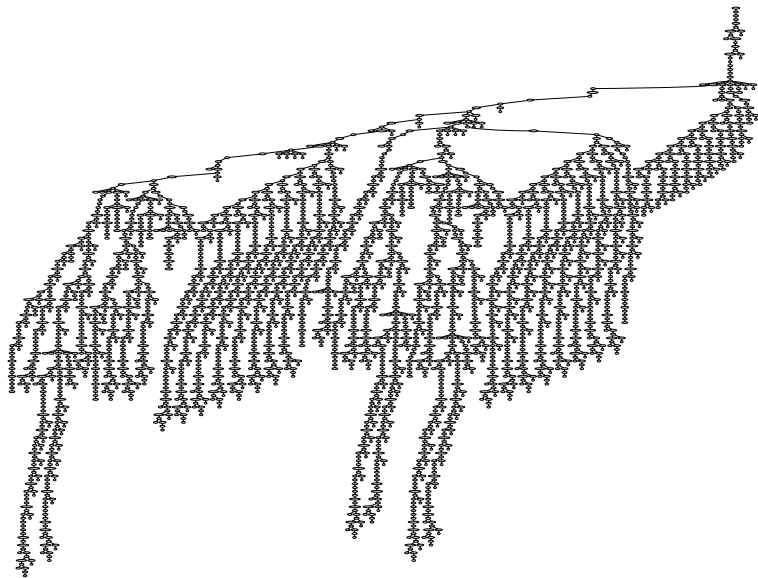
Необработанное исключение: System.Exception: Can't find family
  в Yard.Generators.RMGLR.AST.Tree`1.TraverseWithRanges(Position)(FSharpFunc`2 tokenToRange, FSharpFunc`2 dispose, FSharpFunc`2 f)
  в Yard.Generators.RMGLR.AST.Tree`1.collectMarnings(a)(FSharpFunc`2 tokenToRange)
  в MSSqlParser.Parse(String srcFilePath) в D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql\Driver.fs:crpoka 105
  в MSSqlParser.close133-2.Invoke(String srcFilePath) в D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql\Driver.fs:crpoka 133
  в <StartupCode$MSSql>.MSSqlParser.main() в D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql\Driver.fs:crpoka 133

D:\projects\yc\recursive-ascend\Examples\RMGLRMSSql\bin\Debug>
```

## Демонстрация



# Демонстрация



- Сайт проекта: <http://recursive-ascent.googlecode.com>
- Контакты: [Semen.Grigorev@lanit-tercom.com](mailto:Semen.Grigorev@lanit-tercom.com)