

Восстановление после ошибок в GLR-алгоритме

Иванов А.В., 361 группа
Руководитель: Григорьев С.В.

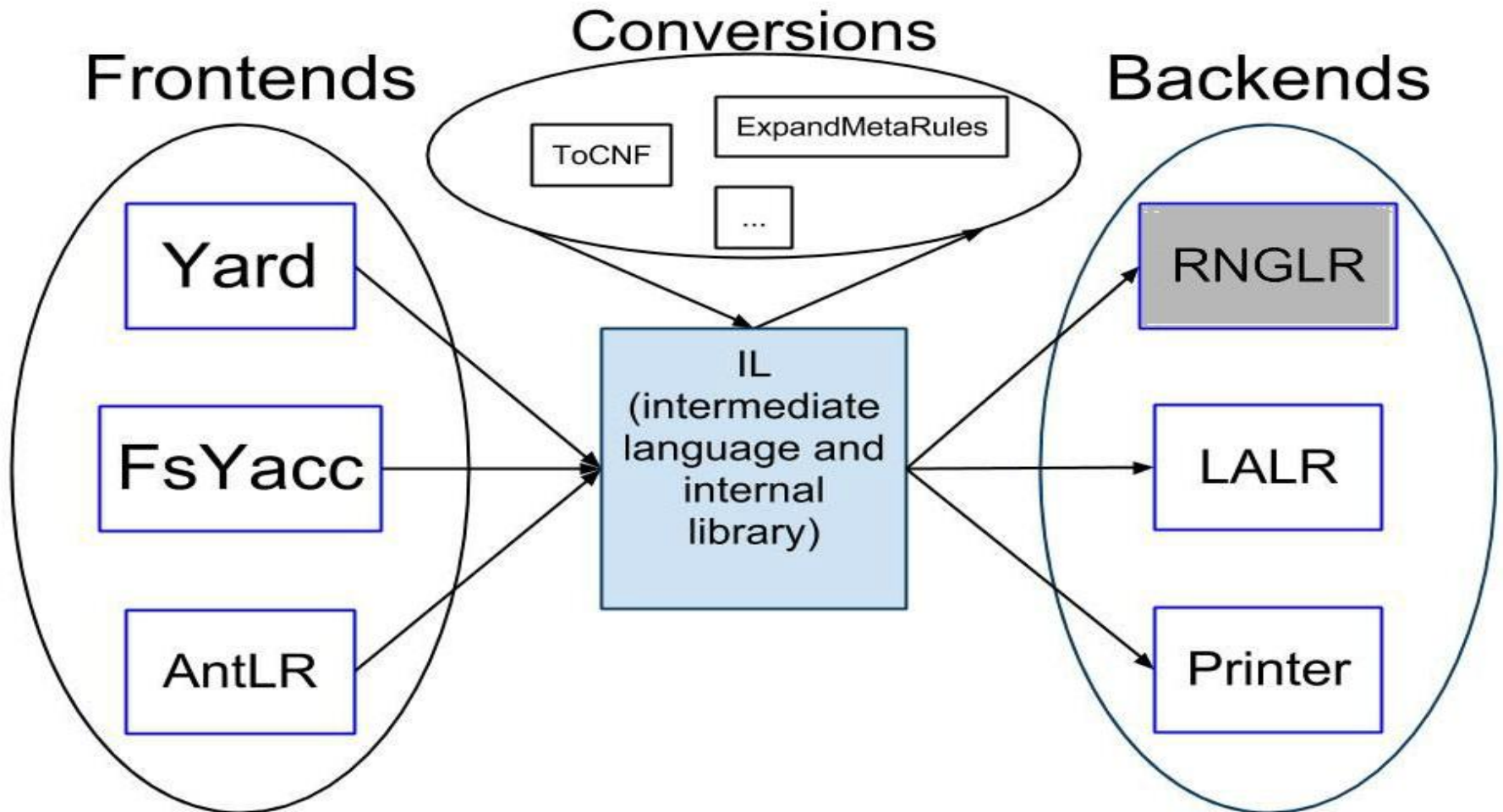
Санкт-Петербургский государственный университет
2013

Область применения

- Автоматизированный реинжиниринг ПО
- Синтаксический анализ
 - Генераторы парсеров
 - Отсутствие или неточность документации
 - Итеративная разработка
 - Неполнота грамматики

YaccConstructor

Студенческий проект кафедры системного программирования.



GLR

- Может работать с неоднозначными грамматиками.
- Охватывает большой класс языков.
- Возвращает все деревья разбора.

Постановка задачи

Восстановление после ошибок в GLR

- Изучение существующих подходов
- Оценка их применимости к GLR
- Реализация для RNLGR модуля YaccConstructor-a

Сложность восстановления в GLR

- Много ветвей разбора.
- Сложно определить, по какой нужно восстанавливаться.
- В результате можно получить много лишних результатов и лишних сообщений об ошибках.

Подходы

- Panic mode: пропуск токенов до тех пор, пока не будет найден один из синхронизирующих.
- Error token: генерация специального токена.
- Follow-set error recovery: синхронизирующее множество вычисляется из текущего состояния.
- Suffix grammar: разбор после ошибки осуществляется по суффиксной грамматике.

Error Token

Была выбрана стратегия, использующая в своей работе error token.

- Позволяет сохранить семантику.
- Позволяет влиять на восстановление изменением грамматики.

Реализована в модуле RNLGR.

Результаты

- Изучены существующие подходы к восстановлению после ошибок.
- Реализовано восстановление после ошибок в RNLGR-модуле YaccConstructor'a
 - Реализован набор тестов.

Результаты

- Принято участие в конференции "Технологии Microsoft в теории и практике программирования".
 - Тезисы опубликованы в сборнике материалов конференции.