Разработка инструмента для добавления верификации

Автор: Маллабаев Азамат Нурмухамадович, 143 группа
Научный руководитель: ст.пр. Григорьев Семён Вячеславович

Санкт-Петербургский государственный университет

1 июня 2016г.

Введение

- Тестирование — доказательство некорректности программы
- Верификация — доказательство корректности программы

Инструменты верификации

- AutoProof — верификатор языка Effel
- Coq — интерактивное средство для доказательства теорем
- F$*$ — язык с поддержкой верификации
  - ML-подобный, каким также является F#
  - Работает в DotNet
  - Транслируется в F# без учета ограничений
- Z3 — низкоуровневый инструмент верификации

Атрибуты

Атрибуты — поля, применимые к элементам программы

Атрибуты в F#

```
1 [<Obsolete("Код -- баян")>] //устаревший метод
2 [<EntryPoint>] //атрибут начала программы
3 let helloSayer () = //метод, к которому применены атрибуты
4    printfn "Hello, world!!!"
```

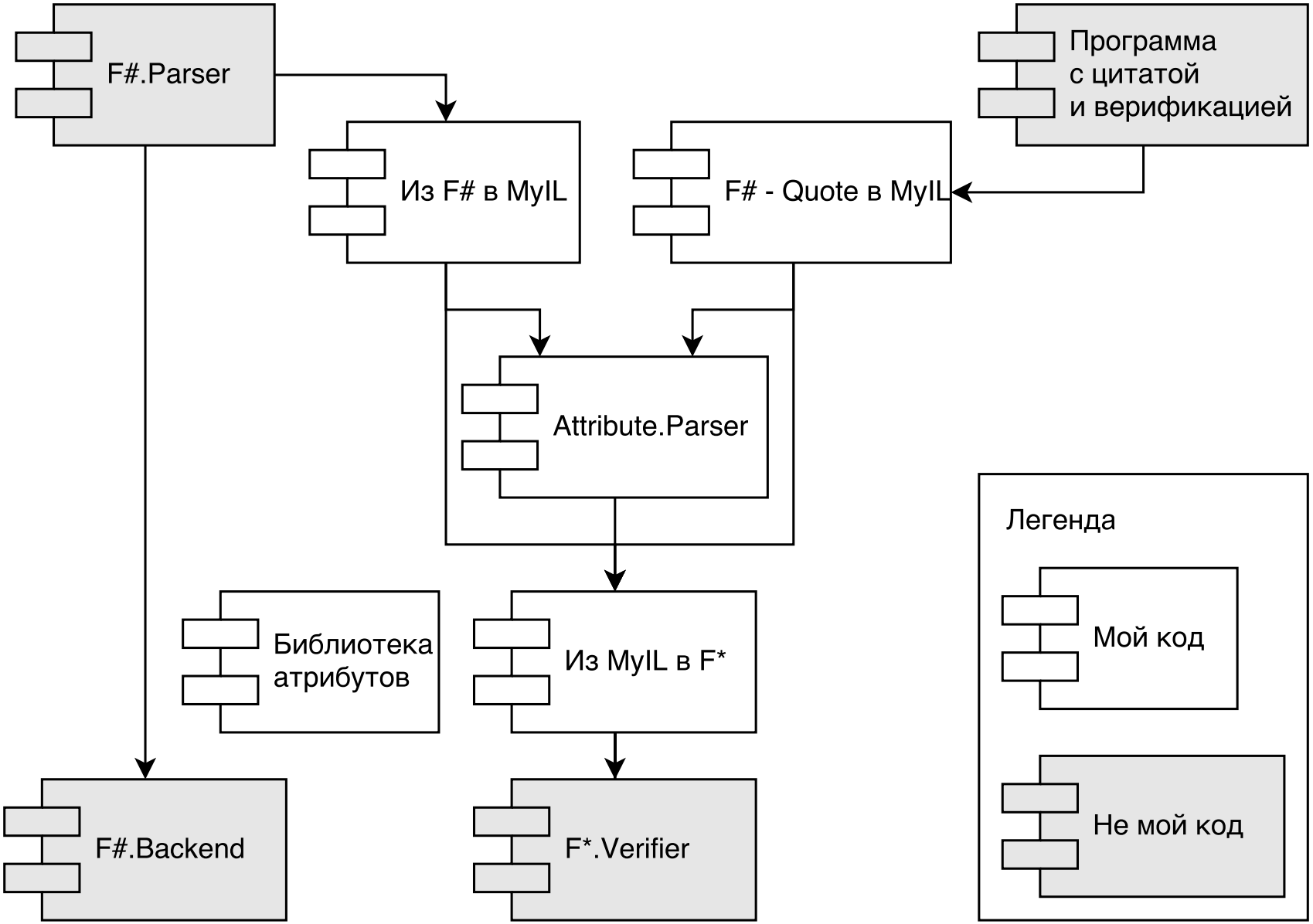
Постановка задачи

Цель: разработка инструмента для добавления верификации в код на F# с использованием верификатора F*

Задачи

- Изучить компилятор F# и верификатор F*
- Разработать архитектуру системы
- Разработать транслятор из F# в F*
- Разработать транслятор из цитат языка F# в F*

Аритектура инструмента



Поддерживаемые выражения

- Модули
- Типизированное определение функции
- Условный оператор if-then-else
- Применение функции
- Бинарные операции (+,-,*,/,>,<,==,!=,>=,<=,<==>,==>)
- Целые числа
- Выражения верификации forall и exist

Фронтенды

- Tehanu.FSharp - на базе FSharp.Compiler.Service
  - Для верификации кода на F#
- Tehanu.FSharp.Quote - на базе цитат F#
  - Для верификации F#-DSL, например, Brahma.FSharp

Пример использования атрибута верификации

```
///Верифицируемая функция
1 [<Total "forall x y . x < y ==> i x + 2 <= i y">]
2 let f x =
    if x > 1
    then x * x * x
    else x - 2

//Неверифицируемая функция
1 [<Total "forall x y . x < y ==> i x < i y">]
2 let g x = x * x
```

Пример использования цитат с верификацией

Трансляция модуля Oh с функцией haha в F*

```
createModule "Oh" ["haha",
                   "forall x . x > 0 <==> i x > 0",
                   <@@fun x -> x * x@@>]
|> sprintModule
```

Результаты

- Изучены компилятор F# и верификатор F*
- Разработана архитектура системы
- Разработан транслятор из F# в F*
- Разработан транслятор из цитат языка F# в F*