Декомпиляция MSIL в F#

Алефиров Алексей
361 группа
Руководитель: Григорьев Семён Вячеславович

2013 г.

Введение

Декомпиляция - преобразование низкоуровнего объектного кода компиляции в код языка программирования высокого уровня.

- Разработка собственного кода.
- Работа с чужим кодом.
- Изучение платформы.

Введение

- F# - мультипарадигмальный язык программирования из семейства языков .NET Framework.
- Помимо F# в семейство .NET входят такие языки, как C#, Visual Basic и другие.
- Программы, написанные на языках семейства, компилируются в единый для .NET байт-код Common Intermediate Language (CIL, MSIL)

Цель

Разработать декомпилятор из MSIL в F#.

Основная проблема

F# со всеми своими функциональными конструкциями компилируется в MSIL, представляющий объектно-ориентированную и императивную парадигму языков программирования.

Постановка задачи

- Произвести обзор программного обеспечения, занимающегося декомпиляцией .NET сборок.
- Изучить язык программирования F#.
- Изучить общие основы декомпиляции.
- Изучить, каким образом F# компилируется в MSIL.

Постановка задачи

- Если искомое ПО будет найдено, изучить его архитектуру.
- Разработать декомпилятор из MSIL в F# в виде дополнения для найденного ПО или как самостоятельное приложение.

Обзор существующих решений

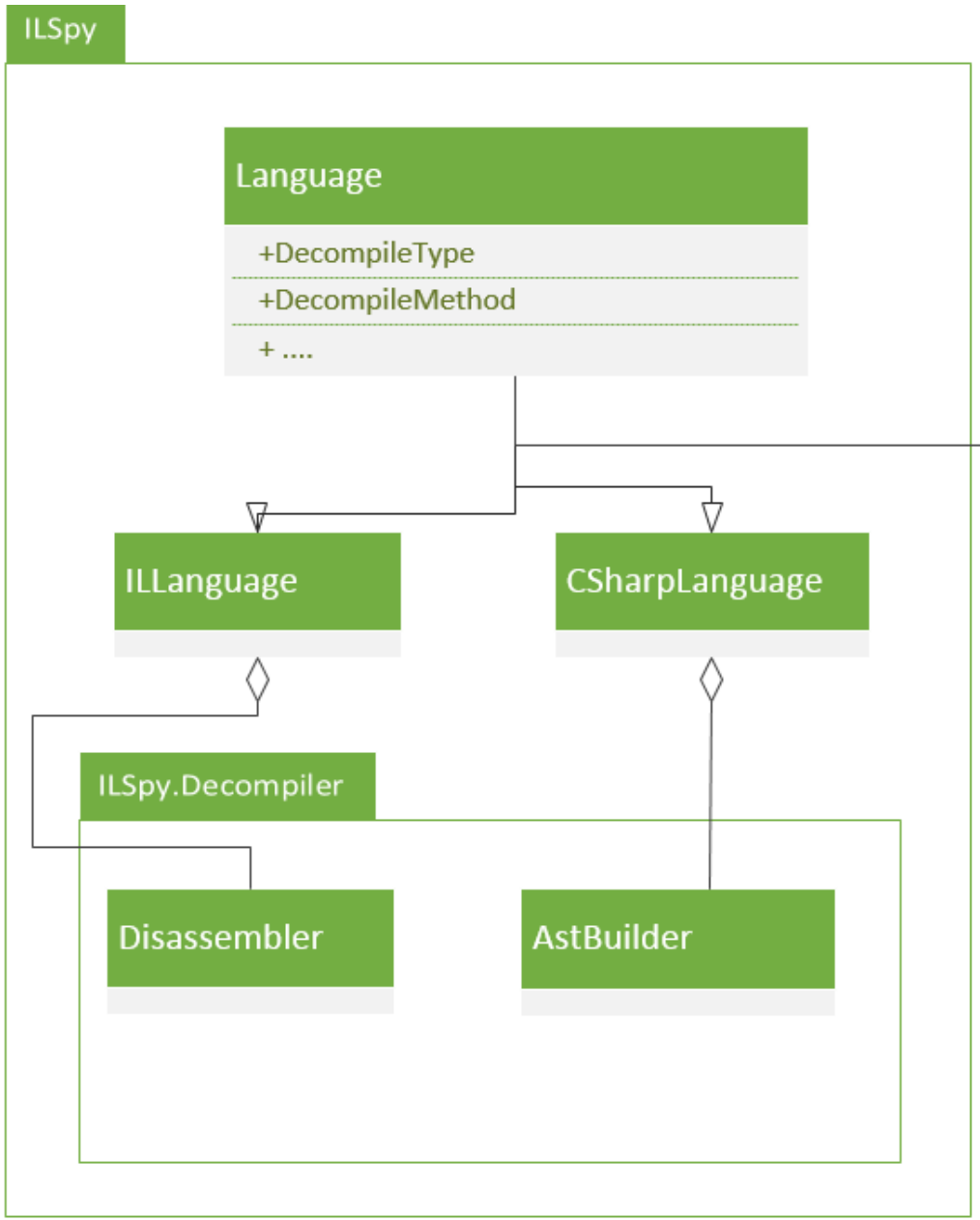
- **.NET Reflector** - в прошлом open source, ныне коммерческий проект с закрытым кодом, поддерживает декомпиляцию C# и Visual Basic.
- **ILSpy** - open source проект, запущенный после закрытия свободной версии Reflector, предоставляет удобную платформу для разработки плагинов

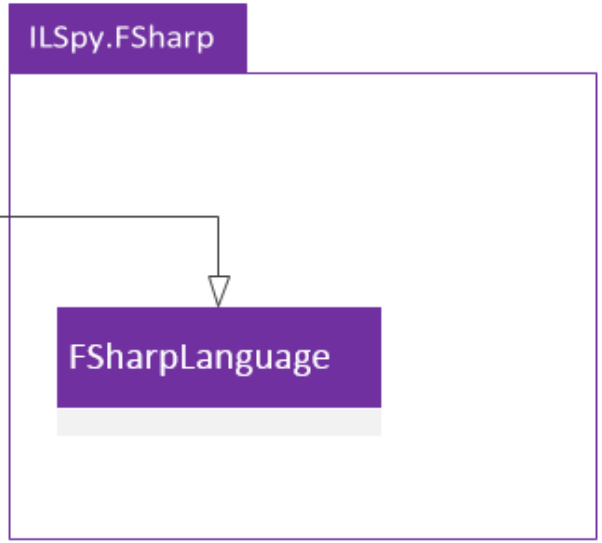
Обзор существующих решений

- **Dotnet IL Editor** - open source, однако, не поддерживает разработку плагинов.
- **MonoDevelop** - open source, однако, хоть и поддерживает разработку плагинов, однако, в отличе от ILSpy, не дает возможности “на лету” навигироваться и просматривать декомпилированный код.

ILSpy. Архитектура

- ILSpy представляет собой определенное количество библиотек и приложение ILSpy.exe.
- В проекте ILSpy определен абстрактный класс Language, который представляет интерфейс для реализации декомпиляции конкретного языка.
- При запуске декомпиляции в представлении приложения вызываются методы Language, перегруженные в соответствующем данному языку потомке.





Language - “точка прикрепления” плагина для декомпиляции нового языка

Апробация плагина на примере анонимных функций

Исходный код программы на F#:

```
module Sample3

let x y = [1;2;3;4]  |> List.map (fun x -> x*(x+y))
```

Результат декомпиляции в C#:

```
using ...
[CompilationMapping(SourceConstructFlags.Module)]
public static class Sample3
{
    [Serializable]
    internal class x@3 : FSharpFunc<int, int>
    {
        public int y;
        internal x@3(int y)
        {
            this.y = y;
        }
        public override int Invoke(int x)
        {
            return x * (x + this.y);
        }
    }
    public static FSharpList<int> x(int y)
    {
        return ListModule.Map<int, int>(new Sample3.x@3(y), FSharpList<int>.Cons(1, FSharpList<int>.Cons(2, F
    }
}
```

Апробация плагина на примере анонимных функций

Результат декомпиляции в F#:

```
type Sample3 () =
    let x@3  y  = fun x  ->  x * (x + y)
    member this.x  y =
        ListModule.Map<int, int>(fun x -> x * x + y, [1;2;3;4])
```

Результаты

- Изучен язык программирования F#.
- Изучены основы декомпиляции.
- Изучено, каким образом F# декомпилируется в MSIL.
- Изучена архитектура ILSpy.
- Разработан декомпилятор из MSIL в F# в виде плагина для ILSpy на языке F#.

Результаты

- Работа была представлена на межвузовском конкурсе-конференции студентов, аспирантов и молодых ученых “Технологии Microsoft в теории и практике программирования” в марте 2013 года, проходившем в Санкт-Петербургском Политехническом университете, тезисы работы опубликованы в сборнике материалов конференции.