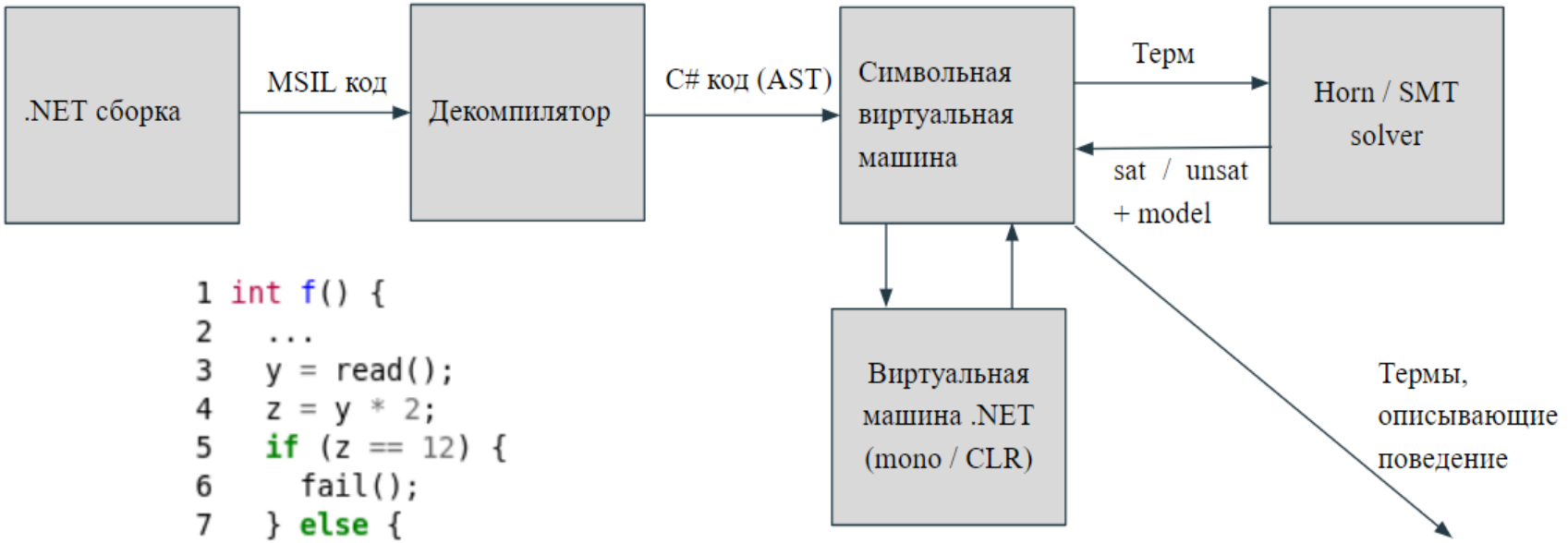
ПОДДЕРЖКА СТРОК В СИМВОЛЬНОЙ ВИРТУАЛЬНОЙ МАШИНЕ .NET

Костицын Михаил, СПбГУ
Научный руководитель: Кириленко Я.А.

БЛИЗКИЕ РАБОТЫ (АНАЛИЗАТОРЫ)

- Pex (dynamic symbolic execution, .NET)
- JPF extensions (Java)
- BLAST (model checker, static, C, unbounded)
- CHorn (LLVM)
- JayHorn (самый близкий) (Java)
- Ultimate automizer (LLVM, Boogie)
- CPAchecker (LLVM)
- Infer (static, C/C++)
- ...

СИМВОЛЬНАЯ ВИРТУАЛЬНАЯ МАШИНА .NET




```
1 int f() {
2   ...
3   y = read();
4   z = y * 2;
5   if (z == 12) {
6     fail();
7   } else {
8     printf("OK");
9   }
10 }
```

ОБЛАСТЬ ПРИМЕНЕНИЯ ПРОЕКТА

- **Верификация**
- **Символьная отладка**
- **Символьное профилирование**
- **Порождение тестов**
- **Синтез программ**
- **…**

ЗАДАЧИ

- **Базовые**
  - Придумать представление строк в символьной памяти
  - Разработать механизм инициализации строк в символьной памяти
- **Сложные**
  - Интернирование
  - Символьная хэш-функция для строк

```
public static string Obstacle(string s)
{
    int h = s.GetHashCode();
    if (h > 42)
        String.Intern(s);
    return String.IsInterned(s);
}
```

```
var s = "Strings are immutable";
int length = s.Length;
unsafe
 {
  fixed (char* c = s)
   {
    for (int i = 0; i < length / 2; i++)
     {
      var temp = c[i];
      c[i] = c[length - i - 1];
      c[length - i - 1] = temp;
     }
   }
 }
Console.WriteLine("Strings are immutable");
```

ЗАДАЧИ

- **Исследовательские проблемы**
  - Композициональность пула интернирования
  - Поиск подстроки
  - Регулярные выражения
  - Взаимодействие с решателями

```
public static string MaybeInvokedBefore(string s)
{
    Regex rgx = new Regex("\\s+");
    string res = rgx.Replace(s, " ");
    return String.Intern(res);
}

public static string Obstacle(string s)
{
    int h = s.GetHashCode();
    if (h > 42)
        String.Intern(s);
    return String.IsInterned(s);
}
```

КОНЦЕПЦИИ

- **Основная идея решения**

  Кодирование строк в линейную арифметику с использованием теории массивов и неинтерпретированных функций

- **Представление строк**

```
[NonSerialized]
private int m_stringLength;
[NonSerialized]
private char m_firstChar;
```

1. Длина строки
2. Массив символов

КОНЦЕПЦИИ

- **Интернирование**
  - Пул интернирования – символьная хэш-таблица
  - Интернирование строковых литералов на уровне декомпилятора

- **Символьная хэш-функция**
  1. Конкретная строка
  2. Частично-символьная строка
  3. Символьная строка

ИНТЕРНИРОВАНИЕ

1. Состояние виртуальной машины расширено пулом интернирования
2. Интернирование строковых литералов

```
string s = "should be interned";
```

Эмуляция уровня JIT-компиляции

3. Интернирование строки в общем случае

```
String.Intern(s);

String.IsInterned(s);
```

Интерфейс взаимодействия с пулом интернирования

ЗАКЛЮЧЕНИЕ

- ✓ **Базовые**
  - ✓ Придумать представление строк в символьной памяти
  - ✓ Разработать механизм инициализации строк в символьной памяти
- ✓ **Сложные**
  - ✓ Интернирование
  - ✓ Символьная хэш-функция для строк
- **Исследовательские проблемы**
  - ✓ Композициональность пула интернирования
  - Поиск подстроки
  - Регулярные выражения
  - Взаимодействие с решателями