FACULTY OF MATHEMATICS AND PHYSICS
Charles University

PHP Generators in .NET

Compilation of a dynamic langauge Generators into MSIL

PEACHPIE

Petr Houška | Mgr. Jakub Míšek | Department of Software engeneering | github.com/peachpiecompiler/peachpie | github.com/petrroll/bachelor-thesis

Goals

- Generators support within Peachpie
- Keeping reference PHP semantics
- Reusing already existing facilities
- Both design and implementation

Peachpie

- Bridge between PHP and .NET
- PHP to MSIL (CIL) compiler
- Reimplementation of PHP class library
- Written in C#, .NET foundation member

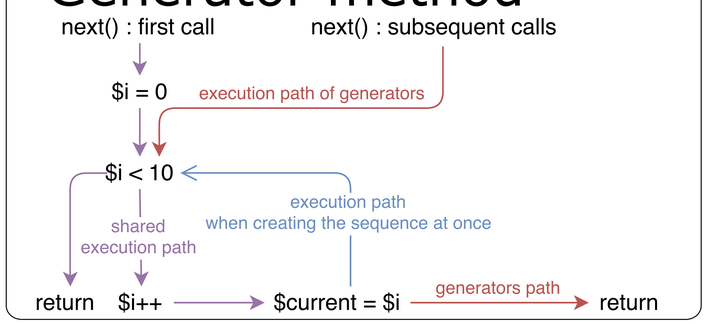
MSIL

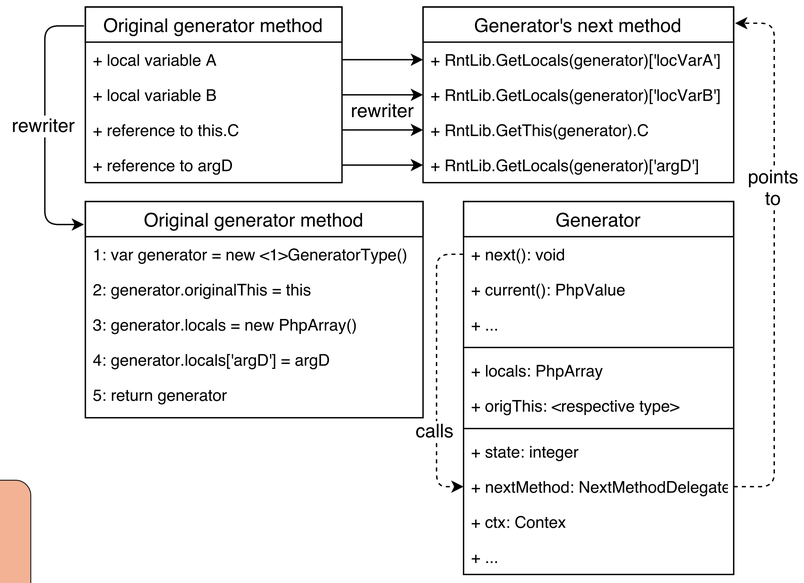
- Intermediate language for .NET platform
- High level stack based assembler
- Strongly typed, object oriented
- Native understanding of exception handling
- No concept of generators or execution state saving

Generators

- Contain a sequential alghoritm, return an Iterator
- Natively supported by the reference PHP runtime
- Lowered by compilers in .NET languages
- Execution pauses at *yield* keyword
  - Statement in most .NET languages (C#, F#)
  - Value carying expression in PHP

Generator method

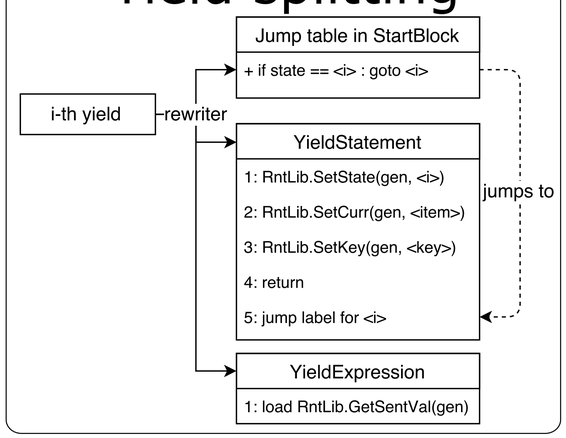




Generator method

```
function by_one_generator(){
  for($i = 0; $i < 10; $i++){
    yield $i;
  }
}
```

Yield splitting



Solution

- New Generator type representing the returned Iterator
- Changes to the original generator method's body:
  - Transformed to Generator's state machine as a new method
  - Replaced with code that initiates and returns the Generator
- State machine transformations:
  - Local variables lifted to Generator instance
  - Exlicit state saving on each yield
  - State backed jump table in the beginning
- Semantic tree transformation to lower yields:
  - Moves yields under expression trees' roots
  - Handles conditioned branches with yields
  - Maintains the original order of execution
  - Splitting yield into a statement and an expression

Semantic tree transformation