Forth Golfscript Interpreter

Golfscript

Golforth

Code Golf

- shortest possible source code that implements an algorithm
- solving problems (holes) in as few keystrokes as possible

Golforth

Code Golf

- shortest possible source code that implements an algorithm
- solving problems (holes) in as few keystrokes as possible

Golfscript

- stack oriented, variables exist
- single symbols represent high level operations
- strong typed
- heavy use of operator overloading and type coercion

Golforth

Golfscript Types

- Integer: `1 2`
- Arrays: `[1 2 3] [3]`
- Strings: `"one two three"`
- Blocks: `{1+}`

Golforth

Golfscript Types

- Integer: `1 2`
- Arrays: `[1 2 3] [3]`
- Strings: `"one two three"`
- Blocks: `{1+}`

Golfscript Operator Example

- `12 3 * -> 36`
- `[50 51 52]’ ’* -> "50 51 52"`

Forth Implementation

Golforth

Typesystem

- typisierter ausdruck ist scalar am stack
- Anonyme functions (words) vs Storage
- werden anonyme funktionen wieder geloescht wenn sie nicht referenziert werden?
- vorteil speicher: es kommen nur benoetigte elemente auf den stack (zb nur den typ fetchen)
- garbage collector vom ertl?

```
12 anon_int s" 1 anon_int golf_+" anon_block -> 2
elements on stack
```

Golforth

Arrays

- Not in syntax, operator
- code

Golforth

Blocks

- Stored as already translated strings
- code

Golforth

Preprocessor

- translates golfscript to forth based intermediate
- based on regular expression of reference implementation
- Responsible for:
  - infer initial type from syntax
  - symbol table for variables

```
"2 {1+}:x"
```

↓

```
2 anon_int s" 1 anon_int golf_+" anon_block create golf_x ,
```

Golforth

Type Coercion and Overloading

- einstellige operation fuer overloading
- zweistellige mit coerce
- hinweis das hier jede menge operationen in forth zu coden sind, evtl nicht vollstaendig

Golforth

Conditionals and Loops

- functions which take code blocks as arguments

Golforth

Cutbacks

- Error Handling differs
- Probably not all operators implemented

Golforth

Usage of Idiomatic Forth

- stack paradigma mapped to typed language
- Macros?