Arrays und Array Slicing

Mit Arrays arbeiten, leicht gemacht.

Mark Zeman, Philip Stark

FHNW - Compilerbau

18.11.2014

- `http://github.com/hellerbarde/cpib`

Übersicht

Sprachdesign

Erreichtes und Demo

Implementation

Abschliessendes

Arrays und Array-Slicing

- Bekannte Datenstruktur

```
var a:array (5) int;

a[5];
```

- Extraktion eines Teilarrays

```
a := [1,2,3,4,5];

print(a[0..3]);

--------------------
output => [1,2,3,4]
```

Lexikalische Erweiterungen

- Nur weniges nötig
- `FILL` für den syntaktischen Zucker
- `RANGE` damit wir Minus nicht überladen

Pattern	Token
`[`	LBRACKET
`]`	RBRACKET
`fill`	FILL
`array`	ARRAY
`..`	RANGE

Grammatikalische Produktionen

```
cmd      ::= SKIP
           | expr ':=' [FILL] expr
           [...]
typedIdent ::= IDENT ':'(ATOMTYPE
                 | ARRAY '(' expr {',' expr} ')' TYPE)
factor ::= LITERAL
         | arrayLiteral
         | IDENT [INIT | exprList | arrayIndex]
         [...]

arrayIndex::= '[' expr ['..' expr] ']' {arrayIndex}
arrayLiteral ::= '[' arrayContent ']'
arrayContent ::= LITERAL {',' LITERAL}
               | arrayLiteral {',' arrayLiteral}
```

Erreichte Ziele

- Generell IML
  - Variablen, global und lokal
  - Prozeduren und Funktionen
  - Kontrollstrukturen
- Arrays
  - Input und Output
  - Zugriff auf Elemente
  - Literale
  - Fixed-Length Slices

Demonstration

Bubble Sort

Abstract Syntax Tree und Code Generation

AST

- Noch zu generell
- Namen der Knoten nicht intuitiv
  - Spät realisiert

Code Generation

- In AST Knoten `generateCode`
  - Rekursive Code Generation
  - Interface ist Top of the Stack

VM Erweiterungen

- ArrayInput und -Output
  - Länge als Argument
- ArrayAccess: Nimmt 3 Werte vom Stack
  - Adresse des Arrays
  - Start-Index
  - End-Index Lädt so bestimmten Teil des Arrays auf den Stack

Rückblick

- Viel gelernt
  - Genug um einiges anders machen zu wollen
- Selbst mit Warnung vom Professor Aufwand unterschätzt
  - AST
  - Codeerzeugung
- Keine dynamischen Array Slices
  - Passt konzeptuell nicht zu IML
- Arrays auf Heap ablegen
  - Viel zu spät realisiert
  - Hat ein paar Sachen erleichtert
  - Hat vieles verkompliziert

Ausblick

- AST neu planen
  - Würde Code Generation erleichtern

Fragen

Vielen Dank!