

Switch-Case Anweisung über int32 für IML

CPIB, HS2016 – Gruppe 10: Remo Rossi und Roman Meier

Einleitung

Switch-Case mit Beispiel

IML verfügt über keine Switch-Case Anweisung. Wenn auch diese über eine verschachtelte IF-Anweisung umsetzbar wäre, ist eine Switch-Case oft übersichtlicher und einfacher verständlich. Um die Funktionsweise und den Einsatz zu verdeutlichen: ein Beispiel.

```
program SwitchCase(in n:int32, out f:int32)
global
fun switchIntFun(n:int32) returns var res:int32
do
  res init := 0;
  switch n
    case 1 : res := n + n;
             res := n - n
    case 2 : res := n * n
    case 3 : res := n * n + n
    casedefault :
  endswitch
endfun;
do
  f init := switchInt(n)
endprogram
```

Da IML momentan zwei Typen (int und bool) unterstützt, lag es auf der Hand die Switch-Case Anweisung für diese beiden Typen zu implementieren. Während der Implementierung ist uns aufgefallen, dass die implementierung für bool sehr viele Fragen aufwirft und die Grammatik unnötig komplizierter macht. So haben wir uns entschlossen diese nicht zu entwerfen, zumal sie einfach mit der IF-Anweisung umsetzbar ist.

Lexikalische Erweiterung

Lexikalisch wird die Syntax um folgende Token erweitert:

Pattern	Token
switch	SWITCHINT32
endswitch	ENDSWITCH
case	CASE
casedefault	CASEDEFAULT

Grammatikalische Erweiterung

Man sieht, es muss mindestens ein Case geben und auf jeden Fall den default Case. Die restliche Grammatik wurde 1:1 von IML übernommen. Diese Erweiterung wurde mit FixFoxi überprüft und als LL(1) Grammatik eingestuft.

```
Cmd ::= SKIP
      | expr BECOMES expr
      | IF expr THEN cpsCmd ELSE cpsCmd ENDIF
      | WHILE expr DO cpsCmd ENDWHILE
      | CALL IDENT exprList [globInits]
      | DEBUGIN expr
      | DEBUGOUT expr
      | SWITCH IDENT
          CASE LITERAL COLON cpsCmd
          [{CASE LITERAL COLON cpsCmd}]
          CASEDEFAULT COLON cpsCmd
      ENDSWITCH
```

Implementations Details

Implementationstechnisch halten wir uns möglichst genau an die Vorlage der Slides. Programmiert wird in Java. Um die Implementation in IML möglichst einfach zu halten, versuchen wir uns möglichst gut an die bisherige Syntax zu halten.