

Bilag B - Aftesting af faserne fra scan til type

Bilag B - Aftesting af faserne fra scan til type	1
Start på test af test_com1.tony.....	5
test_com1.tony.....	5
tony.exe consol output.....	5
tonyprettyTxt.txt.....	6
tonysymbol_xref.txt	6
Start på test af test_com2.tony.....	6
test_com2.tony.....	6
tony.exe consol output.....	6
Start på test af test_com3.tony.....	7
tony.exe consol output.....	7
Start på test af test_com4.tony.....	8
test_com4.tony.....	8
tony.exe consol output.....	8
tonyprettyTxt.txt.....	9
tonysymbol_xref.txt	9
Start på test af test_decl1.tony.....	9
test_decl1.tony	9
tony.exe consol output.....	9
tonyprettyTxt.txt.....	10
tonysymbol_xref.txt	11
Start på test af test_decl1b.tony	12
test_decl1b.tony	12
tony.exe consol output.....	13
tonyprettyTxt.txt.....	13
tonysymbol_xref.txt	14
Start på test af test_decl2.tony.....	15
test_decl2.tony	15
tony.exe consol output.....	16
tonyprettyTxt.txt.....	16
tonysymbol_xref.txt	17
Start på test af test_decl3.tony.....	17
test_decl3.tony	17
tony.exe consol output.....	18
tonyprettyTxt.txt.....	18
tonysymbol_xref.txt	19
Start på test af test_decl4.tony.....	19
test_decl4.tony	19
Start på test af test_stm1.tony	20
test_stm1.tony	20
tony.exe consol output.....	21
tonyprettyTxt.txt.....	21
tonysymbol_xref.txt	22
Start på test af test_stm2.tony	22
test_stm2.tony	22
tony.exe consol output.....	23
tonyprettyTxt.txt.....	23
tonysymbol_xref.txt	25
Start på test af test_stm3.tony	25
test_stm3.tony	25
tony.exe consol output.....	25
tonyprettyTxt.txt.....	26
tonysymbol_xref.txt	27
Start på test af test_precedens1.tony	27
test_precedens1.tony	27

tony.exe consol output.....	27
tonyprettyTxt.txt.....	28
tonysymbol_xref.txt	28
Start på test af test_precedens2.tony	29
test_precedens2.tony	29
tony.exe consol output.....	29
tonyprettyTxt.txt.....	30
tonysymbol_xref.txt	30
Start på test af test_precedens3.tony	31
test_precedens3.tony	31
tony.exe consol output.....	31
tonyprettyTxt.txt.....	31
tonysymbol_xref.txt	32
Start på test af test_precedens4.tony	32
test_precedens4.tony	32
tony.exe consol output.....	32
tonyprettyTxt.txt.....	33
tonysymbol_xref.txt	33
Start på test af test_term1.tony.....	34
test_term1.tony	34
tony.exe consol output.....	34
tonyprettyTxt.txt.....	35
tonyprettyXml.xml.....	35
tonysymbol_xref.txt	39
Start på test af test_weed1.tony	39
test_weed1.tony	39
tony.exe consol output.....	40
tonyprettyTxt.txt.....	40
tonysymbol_xref.txt	41
Start på test af test_weed2.tony	41
test_weed2.tony	41
tony.exe consol output.....	43
tonyprettyTxt.txt.....	44
tonysymbol_xref.txt	47
Start på test af test_weed3.tony	48
test_weed3.tony	48
tony.exe consol output.....	48
tonyprettyTxt.txt.....	49
tonysymbol_xref.txt	49
Start på test af test_weed4.tony	50
test_weed4.tony	50
tony.exe consol output.....	50
tonyprettyTxt.txt.....	51
tonysymbol_xref.txt	51
Start på test af test_typereturn1.tony.....	52
test_typereturn1.tony	52
tony.exe consol output.....	52
tonyprettyTxt.txt.....	53
tonysymbol_xref.txt	53
Start på test af test_typereturn2.tony.....	54
test_typereturn2.tony	54
tony.exe consol output	54
tonyprettyTxt.txt.....	55
tonysymbol_xref.txt	55
Start på test af test_typereturn3.tony.....	56
test_typereturn3.tony	56
tony.exe consol output.....	56
tonyprettyTxt.txt.....	57
tonysymbol_xref.txt	58

Start på test af test_typecall1.tony.....	58
test_typecall1.tony.....	58
tony.exe consol output.....	59
tonyprettyTxt.txt.....	59
tonysymbol_xref.txt.....	60
Start på test af test_typecall2.tony.....	61
test_typecall2.tony.....	61
tony.exe consol output.....	61
tonyprettyTxt.txt.....	62
tonysymbol_xref.txt.....	63
Start på test af test_typeassign1.tony.....	64
test_typeassign1.tony.....	64
tony.exe consol output.....	65
tonyprettyTxt.txt.....	66
tonysymbol_xref.txt.....	66
Start på test af test_typewrite1.tony.....	67
test_typewrite1.tony.....	67
tony.exe consol output.....	68
tonyprettyTxt.txt.....	68
tonysymbol_xref.txt.....	69
Start på test af test_typenew1.tony.....	69
test_typenew1.tony.....	69
tony.exe consol output.....	70
tonyprettyTxt.txt.....	71
tonysymbol_xref.txt.....	71
Start på test af test_typeif1.tony.....	72
test_typeif1.tony.....	72
tony.exe consol output.....	72
tonyprettyTxt.txt.....	73
tonysymbol_xref.txt.....	74
Start på test af test_typewhile1.tony.....	74
test_typewhile1.tony.....	74
tony.exe consol output.....	74
tonyprettyTxt.txt.....	75
tonysymbol_xref.txt.....	75
Start på test af test_typeexp1.tony.....	76
test_typeexp1.tony.....	76
tony.exe consol output.....	76
tonyprettyTxt.txt.....	77
tonysymbol_xref.txt.....	77
Start på test af test_typeexp2.tony.....	77
test_typeexp2.tony.....	77
tony.exe consol output.....	78
tonyprettyTxt.txt.....	78
tonysymbol_xref.txt.....	79
Start på test af test_typeexp3.tony.....	79
test_typeexp3.tony.....	79
tony.exe consol output.....	80
tonyprettyTxt.txt.....	80
tonysymbol_xref.txt.....	81
Start på test af test_typeexp4.tony.....	81
test_typeexp4.tony.....	81
tony.exe consol output.....	81
tonyprettyTxt.txt.....	82
tonysymbol_xref.txt.....	82
Start på test af test_typestruct1.tony.....	83
test_typestruct1.tony.....	83
tony.exe consol output.....	83
tonyprettyTxt.txt.....	84

tonysymbol_xref.txt	85
Start på test af test_typearray1.tony	86
test_typearray1.tony	86
tony.exe consol output	87
tonyprettyTxt.txt	87
tonysymbol_xref.txt	88
Start på test af test_typearray2.tony	90
test_typearray2.tony	90
tony.exe consol output	90
tonyprettyTxt.txt	91
tonysymbol_xref.txt	92
Start på test af test_typescope1.tony	93
test_typescope1.tony	93
tony.exe consol output	94
tonyprettyTxt.txt	94
tonysymbol_xref.txt	96
Start på test af test_typeminus.tony	97
test_typeminus.tony	97
tony.exe consol output	98
tonyprettyTxt.txt	98
tonysymbol_xref.txt	99
Start på test af test_typefor1.tony	99
test_typefor1.tony	99
tony.exe consol output	99
tonyprettyTxt.txt	100
tonysymbol_xref.txt	101

:

:

Start på test af test_com1.tony

test_com1.tony

:

001: (*Test af TONY kommentar OK*)

002: (*

003: var

004: tal1: int;

005: tal1 = 2;

006: *)

007: var

008: tal1: int;

009: tal1 = 2;

:

tony.exe consol output

:

Options:

-msg sat til test_com1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
var
    vd#tal1: t#int;

v_int#tal1 = c#2;

```

```

-----
:
tonyprettyXml.xml
:
-----
:

```

tonysymbol_xref.txt

```

:
-----

```

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

tal1 <8> variabel af typen: int ref: 1 <9>

```

-----
Slut på test af test_com1.tony
:
:
:
:
-----
:

```

Start på test af test_com2.tony**test_com2.tony**

```

:
001: (*Test af TONY kommentar - mangler slut*)
002: (*
003: var
004: tal1: int;
005: tal1 = 2;
006:
-----
:

```

tony.exe consol output

```

:
Options:

-msg sat til test_com2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

```

7: syntaksfejl ved <Fejl comment ikke sluttet>

:
tonyprettyTxt.txt

:

:
tonyprettyXml.xml

:

:
tonysymbol_xref.txt

:

Slut på test af test_com2.tony

:
:.....

:
:.....

:
:.....

:
:.....

:
:.....

:
:.....

:
:.....

:
:.....

Start på test af test_com3.tony

test_com3.tony

:
001: (*Test af TONY slut uden start *)
002: var
003: tal1: int;
004: tal1 = 2;
005: *)

:

:

tony.exe consol output

:
Options:

-msg sat til test_com3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

5: syntaksfejl ved *) <comment slut uden start>

:
tonyprettyTxt.txt

:

:
tonyprettyXml.xml

:

:
tonysymbol_xref.txt

Slut på test af test_com3.tony

:
:.....
:
:

:

Start på test af test_com4.tony

test_com4.tony

:
001: (*Test af TONY nestede kommentarer *)
002: var (* her (* er (* nestede *) kommentar *) indlagt *)
003: tal1: int;
004: tal1 = 2;

tony.exe consol output

:
Options:

-msg sat til test_com4.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:
var
 vd#tal1: t#int;

v_int#tal1 = c#2;

:
tonyprettyXml.xml
:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

tal1 <3> variabel af typen: int ref: 1 <4>

Slut på test af test_com4.tony
:
:.....
:
:

:

Start på test af test_decl1.tony

test_decl1.tony
:
001: (*Test af TONY TYPE-declaration *)
002: type heltal = int;
003: type rec1 = record of {tal1: int, tal2: int, tal3: heltal};
004: type rec2 = record of {r1: rec1, r2: rec2};
005: type rec3 = record of {tab: array of int};
006: type rec4 = rec4;
007: type taltab = array of int;
008: type rectab = array of rec2;
009: type liste = record of {val: int, next: liste}; (* selv reference *)
010: type arliste = array of liste; (* reference tilbage *)
011:
012: type ar1 = array of ar2; (* kan ikke bruges til så meget*)
013: type ar2 = array of ar3; (* i praksis, men viser ekstem *)
014: type ar3 = array of ar1;
015: (* grammatik kræver mindst 1 statement *)
016: return 0;

:

tony.exe consol output

:

Options:

-msg sat til test_decl1.console.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

6: Fejl navn<rec4> og type er identisk

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

12: Fejl: typen <ar2> er ikke tilgængelig - erklæret senere<13>

13: Fejl: typen <ar3> er ikke tilgængelig - erklæret senere<14>

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 2

Antal fejl: 1

Antal advarsler: 0

:-----

tonyprettyTxt.txt

:

type td#heltal :t#int;

type td#rec1 :t#record of {

vd#tal1: t#int,

vd#tal2: t#int,

vd#tal3: t#heltal

};

type td#rec2 :t#record of {

vd#r1: t#rec1,

vd#r2: t#rec2

};

type td#rec3 :t#record of {

vd#tab: t#array of t#int

};

inakt#type td#rec4 :t#rec4;

```

type td#taltab : t#array of t#int;
type td#rectab : t#array of t#rec2;
type td#liste : t#record of {
    vd#val: t#int,
    vd#next: t#liste
};
type td#arliste : t#array of t#liste;
type td#ar1 : t#array of t#ar2;
type td#ar2 : t#array of t#ar3;
type td#ar3 : t#array of t#ar1;
return c#0;

```

```

-----
:
tonyprettyXml.xml
:
-----
:

```

tonysymbol_xref.txt

```

:

```

```

-----
SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

```

```

ar1 <12> type = array of ar2 => ikke defineret ref: 1 <14>
ar2 <13> type = array of ar3 => ikke defineret ref: 1 <12>
ar3 <14> type = array of ar1<12> => array of ar2 => ikke defineret ref: 1<13>
rectab <8> type = array of rec2<4> => record<4>
liste <9> type = record<9> ref: 2 <9, 10>
arliste <10> type = array of liste<9> => record<9>
taltab <7> type = array of int
heltal<2> type = int ref: 1 <3>
rec1 <3> type = record<3> ref: 1 <4>
rec2 <4> type = record<4> ref: 2 <4, 8>
rec3 <5> type = record<5>

```

```

-----
SymbolTable for rec1: <3> tabelnr:2 scope-level:0 - start

```

```

tal1 <3> variabel af typen: int

```

tal2 <3> variabel af typen: int

tal3 <3> variabel af typen: heltal<0> => int

SymbolTable for rec2: <4> tabelNr: 3 scope-level: 0 - start

r1 <4> variabel af typen: rec1<3> => record<3>

r2 <4> variabel af typen: rec2<4> => record<4>

SymbolTable for rec3: <5> tabelNr: 4 scope-level: 0 - start

tab <5> variabel af typen: array of int

SymbolTable for liste: <9> tabelNr: 5 scope-level: 0 - start

next <9> variabel af typen: liste<9> => record<9>

val <9> variabel af typen: int

Slut på test af test_decl1.tony

:

.....

:

:

:

Start på test af test_decl1b.tony

test_decl1b.tony

:

001: (*Test af TONY TYPE-declaration *)

002: type heltal = int;

003: type rec1 = record of {tal1: int, tal2: int, tal3: heltal};

004: type rec2 = record of {r1: rec1, r2: rec2};

005: type rec3 = record of {tab: array of int};

006: type rec4 = rec4;

007: type taltab = array of int;

008: type rectab = array of rec2;

009: type liste = record of {val: int, next: liste}; (* selv reference *)

010: type arliste = array of liste; (* reference tilbage *)

011:

012: type ar1 = array of ar2; (* kan ikke bruges til så meget*)

013: type ar2 = array of ar3; (* i praksis, men viser ekstem *)

014: type ar3 = array of ar1;

015: (* grammatik kræver mindst 1 statement *)

016: return 0;

:

tony.exe consol output

:

Options:

-msg sat til test_decl1b.consol.txt

-S er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

6: Fejl navn<rec4> og type er identisk

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

12: Advarsel: reference til typen <ar2> som foerst erklaeres senere<13>

13: Advarsel: reference til typen <ar3> som foerst erklaeres senere<14>

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 1

Antal advarsler: 2

:

tonyprettyTxt.txt

:

type td#heltal :t#int;

```
type td#rec1 :t#record of {
  vd#tal1: t#int,
  vd#tal2: t#int,
  vd#tal3: t#heltal
};
```

```
type td#rec2 :t#record of {
  vd#r1: t#rec1,
  vd#r2: t#rec2
};
```

```
type td#rec3 :t#record of {
  vd#tab: t#array of t#int
};
```

```

inakt#type td#rec4 :t#rec4;
type td#taltab :t#array of t#int;
type td#rectab :t#array of t#rec2;
type td#liste :t#record of {
    vd#val: t#int,
    vd#next: t#liste
};
type td#arliste :t#array of t#liste;
type td#ar1 :t#array of t#ar2;
type td#ar2 :t#array of t#ar3;
type td#ar3 :t#array of t#ar1;

return c#0;

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

```

ar1  <12>  type = array of ar2<13> => array of ar3<14> => array of ar1<12> => ref:
1  <14>

ar2  <13>  type = array of ar3<14> => array of ar1<12> => array of ar2<13> => ref:
1  <12>

ar3  <14>  type = array of ar1<12> => array of ar2<13> => array of ar3<14> => ref:
1  <13>

rectab  <8>  type = array of rec2<4> => record<4>

liste  <9>  type = record<9>  ref: 2  <9, 10>

arliste  <10>  type = array of liste<9> => record<9>

taltab  <7>  type = array of int

heltal<2>  type = int  ref: 1  <3>

rec1  <3>  type = record<3>  ref: 1  <4>

rec2  <4>  type = record<4>  ref: 2  <4, 8>

rec3  <5>  type = record<5>

```

 SymbolTable for rec1: <3> tabelNr:2 scope-level:0 - start

tal1 <3> variabel af typen: int

tal2 <3> variabel af typen: int

tal3 <3> variabel af typen: heltal<0> => int

 SymbolTable for rec2: <4> tabelNr:3 scope-level:0 - start

r1 <4> variabel af typen: rec1<3> => record<3>

r2 <4> variabel af typen: rec2<4> => record<4>

 SymbolTable for rec3: <5> tabelNr:4 scope-level:0 - start

tab <5> variabel af typen: array of int

 SymbolTable for liste: <9> tabelNr:5 scope-level:0 - start

next <9> variabel af typen: liste<9> => record<9>

val <9> variabel af typen: int

 Slut på test af test_decl1b.tony

:
 :::
 :
 :
 :

 :

Start på test af test_decl2.tony

test_decl2.tony

```
:
001: (*Test af TONY Var-declaration *)
002: type heltal = int;
003: var
004: tal1: int,
005: tal2: heltal,
006: ok: bool
007: ;
008: var
009: tal3: int,
010: tal4: int,
011: ok2: bool,
012: tab: array of int,
```

```
013: rec: record of {t1: int}
014: ;
015: (* grammatik kræver mindst 1 statement *)
016: return 0;
```

```
-----
:
```

tony.exe consol output

```
:
```

Options:

-msg sat til test_decl2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

```
-----
:
```

tonyprettyTxt.txt

```
:
```

```
type td#heltal : t#int;
```

```
var
```

```
  vd#tal1: t#int,
  vd#tal2: t#heltal,
  vd#ok: t#bool;
```

```
var
```

```
  vd#tal3: t#int,
  vd#tal4: t#int,
  vd#ok2: t#bool,
  vd#tab: t#array of t#int,
```



```

    vd#rec: t#record of {
      vd#t1: t#int
    };

```

```
return c#0;
```

```

-----
:
tonyprettyXml.xml
:

```

```
:
```

tonysymbol_xref.txt

```
:
```

```
-----
SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start
```

```
ok <6> variabel af typen: bool
```

```
ok2 <11> variabel af typen: bool
```

```
tab <12> variabel af typen: array of int
```

```
rec <13> variabel af typen: record<13>
```

```
heltal<2> type = int ref: 1 <5>
```

```
tal1 <4> variabel af typen: int
```

```
tal2 <5> variabel af typen: heltal<0> => int
```

```
tal3 <9> variabel af typen: int
```

```
tal4 <10> variabel af typen: int
```

```
-----
```

```
SymbolTable for rec: <13> tabelnr:2 scope-level:0 - start
```

```
t1 <13> variabel af typen: int
```

```
-----
```

```
Slut på test af test_decl2.tony
```

```
:
```

```
.....
```

```
:
```

```
:
```

```
-----
```

```
:
```

Start på test af test_decl3.tony

```
-----
```

test_decl3.tony

```
:
```

```
001: (*Test af TONY func-declaration *)
```

```
002: func f1 (tal1: int, tal2: int): int (* med exp_list *)
```

```
003: func f11 (): int          (* uden exp_list *)
004:   return 1;
005: end f11;
006: var                      (* flere indre declarationer *)
007:   tal3: int;
008: return tal1 + tal2 + f11();
009: end f1;
010: (* grammatik kræver mindst 1 statement *)
011: return f1(1,2);
```

:

tony.exe consol output

:

Options:

-msg sat til test_decl3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

```
func fd#f1 (  
    vd#tal1: t#int,  
    vd#tal2: t#int    ):t#int
```

```
    func fd#f11 ( ):t#int
```

```
        return c#1;
```

```

end fd#f11;

var
  vd#tal3: t#int;

return t_int#( t_int#( v_int#tal1 + v_int#tal2 ) + f_int#f11 ( ) );

end fd#f1;

return f_int#f1 (c#1, c#2);

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

```

:
-----
SymbolTable for MAIN: <0>  tabelNr:1 scope-level:0 - start
f1 <2>    function af typen: int  ref: 1  <11>

```

```

-----
SymbolTable for f1: <2>  tabelNr:2 scope-level:1 - start
f11 <3>    function af typen: int  ref: 1  <8>
tal1 <2>    variabel af typen: int  ref: 1  <8>
tal2 <2>    variabel af typen: int  ref: 1  <8>
tal3 <7>    variabel af typen: int

```

```

-----
SymbolTable for f11: <3>  tabelNr:3 scope-level:2 - start

```

```

-----
Slut på test af test_decl3.tony
:
:
:
:
:
-----
:

```

Start på test af test_decl4.tony

test_decl4.tony

```

:
001: (*Test af TONY Var-declaration - fejl *)
002: type heltal = int;

```

```

003: var
004: tal1: int,
005: tal2: heltal,
006: ok:      bool
007: ;
008: (* grammatik kræver mindst 1 statement *)
009: tal1 = 1;
010: var
011: tal3: int
012: ;

```

```

-----
:
tony.exe consol output
:
Options:

```

-msg sat til test_decl4.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

10: syntaksfejl ved var <syntax error>

```

-----
:
tonyprettyTxt.txt
:

```

```

-----
:
tonyprettyXml.xml
:

```

```

-----
:
tonysymbol_xref.txt
:

```

Slut på test af test_decl4.tony

```

:
:
:
:
:
-----
:

```

Start på test af test_stm1.tony

test_stm1.tony

```

:
001: (*Test af TONY assign med expression statements *)
002: var
003: tal1: int,
004: tal2: int
005: ;

```

```

006: tal1 = 1;
007: tal1 = tal1 + 1;
008: tal1 = tal1 + 1 + 3 * tal1 + 40 / tal1;
009: tal1 = ( tal1 + 1 + 3 ) * (tal1 + 40) / tal1;
-----

```

:

tony.exe consol output

:

Options:

-msg sat til test_stm1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

tonyprettyTxt.txt

:

var

vd#tal1: t#int,

vd#tal2: t#int;

v_int#tal1 = c#1;

v_int#tal1 = t_int#(v_int#tal1 + c#1);

v_int#tal1 = t_int#(t_int#(t_int#(v_int#tal1 + c#1) + t_int#(c#3 * v_int#tal1)) + t_int#(c#40 / v_int#tal1));

v_int#tal1 = t_int#(t_int#(t_int#(t_int#(v_int#tal1 + c#1) + c#3)) * (t_int#(v_int#tal1 + c#40))) / v_int#tal1);

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

tal1 <3> variabel af typen: int ref: 11 <6, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9>

tal2 <4> variabel af typen: int

Slut på test af test_stm1.tony

:

:::

:

:

:

Start på test af test_stm2.tony

test_stm2.tony

:

001: (*Test af TONY if statements *)

002: var

003: tal1: int,

004: tal2: int,

005: ok: bool

006: ;

007: if tal1 > 1 then (* if uden else *)

008: tal2 = 1;

009: if tal1 > 2 then (* if med else *)

010: tal2 = 2;

011: else

012: tal2 = 3;

013: if tal2 > 3 then (* if med compound *)

014: {

015: tal2 = 3;

016: ok = false;

017: }

018: if tal2 > 4 then (* if med compound + else *)

019: {

020: tal2 = 4;

021: ok = false;

022: }

023: else

024: ok = false;

025: if tal2 > 5 then (* if med if else *)

026: if tal1 < 8 then

027: tal2 = 4;

028: else

029: tal2 = 3;

```
030: if tal1 > 51 then (* if if med else else *)
031: if tal1 > 52 then
032:   tal2 = 2;
033: else
034:   tal2 = 3;
035: else
036: tal2 = 4;
037: if tal1 > 56 then (* if med else if else if *)
038:   tal2 = 1;
039: else if tal1 > 62 then
040:   tal2 = 2;
041: else if tal1 > 63 then
042:   tal2 = 3;
```

:

tony.exe consol output

:

Options:

-msg sat til test_stm2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

vd#tal1: t#int,

vd#tal2: t#int,

```

vd#ok: t#bool;

if t_bool#( v_int#tal1 > c#1 ) then {
  v_int#tal2 = c#1;
}
if t_bool#( v_int#tal1 > c#2 ) then {
  v_int#tal2 = c#2;
}
else {
  v_int#tal2 = c#3;
}
if t_bool#( v_int#tal2 > c#3 ) then {
  #{
    v_int#tal2 = c#3;
    v_bool#ok = c#false;
  #}
}
if t_bool#( v_int#tal2 > c#4 ) then {
  #{
    v_int#tal2 = c#4;
    v_bool#ok = c#false;
  #}
}
else {
  v_bool#ok = c#false;
}
if t_bool#( v_int#tal2 > c#5 ) then {
  if t_bool#( v_int#tal1 < c#8 ) then {
    v_int#tal2 = c#4;
  }
  else {
    v_int#tal2 = c#3;
  }
}
if t_bool#( v_int#tal1 > c#51 ) then {
  if t_bool#( v_int#tal1 > c#52 ) then {
    v_int#tal2 = c#2;
  }
  else {
    v_int#tal2 = c#3;
  }
}
else {
  v_int#tal2 = c#4;
}
if t_bool#( v_int#tal1 > c#56 ) then {
  v_int#tal2 = c#1;
}
else {
  if t_bool#( v_int#tal1 > c#62 ) then {
    v_int#tal2 = c#2;
  }
  else {
    if t_bool#( v_int#tal1 > c#63 ) then {
      v_int#tal2 = c#3;
    }
  }
}
}

```

:
tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <5> variabel af typen: bool ref: 3 <16, 21, 24>

tal1 <3> variabel af typen: int ref: 8 <7, 9, 26, 30, 31, 37, 39, 41>

tal2 <4> variabel af typen: int ref: 16 <8, 10, 12, 13, 15, 18, 20, 25, 27, 29, 32, 34, 36, 38, 40, 42>

Slut på test af test_stm2.tony

:

::

:

:

:

Start på test af test_stm3.tony

test_stm3.tony

:

001: (* Test af TONY statement typer excl. if *)

002: var

003: tal1: int,

004: tab: array of int,

005: rec: record of {t1: int},

006: ok: bool

007: ;

008: ok = true; (* assign *)

009: tal1 = 1; (* assign *)

010: while (tal1 < 5) do (* while *)

011: tal1 = tal1 + 1;

012: new tab of length 10; (* new med length *)

013: new rec; (* new uden length *)

014: write tal1; (* write *)

015: { (* compound *)

016: tal1 = 1;

017: tal1 = tal1 + 1;

018: }

019: return 0;

:

tony.exe consol output

:

Options:

-msg sat til test_stm3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

```

    vd#tal1: t#int,
    vd#tab: t#array of t#int,
    vd#rec: t#record of {
        vd#t1: t#int
    },
    vd#ok: t#bool;

```

```

v_bool#ok = c#true;

```

```

v_int#tal1 = c#1;

```

```

while (t_bool#(v_int#tal1 < c#5)) {
    v_int#tal1 = t_int#(v_int#tal1 + c#1);
}

```

```

new v_array#tab of length c#10;

```

```

new v_record#rec;

```

```

write v_int#tal1;

```

```

#{

```

```

    v_int#tal1 = c#1;

```

```

    v_int#tal1 = t_int#(v_int#tal1 + c#1);

```

```

#}

```

```

return c#0;

```

:
tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <6> variabel af typen: bool ref: 1 <8>

tab <4> variabel af typen: array of int ref: 1 <12>

rec <5> variabel af typen: record<5> ref: 1 <13>

tal1 <3> variabel af typen: int ref: 8 <9, 10, 11, 11, 14, 16, 17, 17>

SymbolTable for rec: <5> tabelnr:2 scope-level:0 - start

t1 <5> variabel af typen: int

Slut på test af test_stm3.tony

:

::

:

:

:

:

Start på test af test_precedens1.tony

test_precedens1.tony

:

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24

002:

003: Test af TONY operatorer og præcendes regler

004: *)

005: var

006: tal: int,

007: ok: bool

008: ;

009: tal = 1 + 2 * 3;

010: tal = 1 + 2 / 3;

011: tal = 1 + 2 * 3 / 4 * 5;

012: tal = 1 - 2 + 3 - 4;

:

tony.exe consol output

:

Options:

-msg sat til test_precedens1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

vd#tal: t#int,
vd#ok: t#bool;

v_int#tal = t_int#(c#1 + t_int#(c#2 * c#3));

v_int#tal = t_int#(c#1 + t_int#(c#2 / c#3));

v_int#tal = t_int#(c#1 + t_int#(t_int#(t_int#(c#2 * c#3) / c#4) * c#5));

v_int#tal = t_int#(t_int#(t_int#(c#1 - c#2) + c#3) - c#4);

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <7> variabel af typen: bool

tal <6> variabel af typen: int ref: 4 <9, 10, 11, 12>

Slut på test af test_precedens1.tony

:
:.....
:
:
:-----
:

Start på test af test_precedens2.tony

test_precedens2.tony

:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Test af TONY operatorer og præcendes regler
004: *)
005: var
006: ok: bool, b1: bool, b2: bool, b3: bool, b4: bool;
007: b1 = true;
008: b2 = false;
009: b3 = b1;
010: b4 = b2;
011: ok = b1 || b2 && b3 || b4;

:

tony.exe consol output

:
Options:

-msg sat til test_precedens2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
var
  vd#ok: t#bool,
  vd#b1: t#bool,
  vd#b2: t#bool,
  vd#b3: t#bool,
  vd#b4: t#bool;

v_bool#b1 = c#true;
v_bool#b2 = c#false;
v_bool#b3 = v_bool#b1;
v_bool#b4 = v_bool#b2;
v_bool#ok = t_bool#( t_bool#( v_bool#b1 || t_bool#( v_bool#b2 && v_bool#b3 ) ) ||
v_bool#b4 );

```

:

tonyprettyXml.xml

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

```

ok <6>   variabel af typen: bool   ref: 1 <11>
b1 <6>   variabel af typen: bool   ref: 3 <7, 9, 11>
b2 <6>   variabel af typen: bool   ref: 3 <8, 10, 11>
b3 <6>   variabel af typen: bool   ref: 2 <9, 11>
b4 <6>   variabel af typen: bool   ref: 2 <10, 11>

```

Slut på test af test_precedens2.tony

:

:

:

Start på test af test_precedens3.tony

test_precedens3.tony

```
:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Test af TONY operatorer og præcendes regler
004: *)
005: var
006: tal: int,
007: ok: bool
008: ;
009: tal = 1;
010: ok = tal == 1+2 || tal != 3+4 && tal < 5+6;
```

:

tony.exe consol output

:

Options:

-msg sat til test_precedens3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

```
vd#tal: t#int,
vd#ok: t#bool;
```

```
v_int#tal = c#1;
v_bool#ok = t_bool#( t_bool#( v_int#tal == t_int#( c#1 + c#2 ) ) || t_bool#( t_bool#(
v_int#tal != t_int#( c#3 + c#4 ) ) && t_bool#( v_int#tal < t_int#( c#5 + c#6 ) ) ) );
```

```
-----
:
tonyprettyXml.xml
:
-----
:
```

tonysymbol_xref.txt

```
:
-----
```

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <7> variabel af typen: bool ref: 1 <10>

tal <6> variabel af typen: int ref: 4 <9, 10, 10, 10>

```
-----
Slut på test af test_precedens3.tony
:
:
:
:
:
-----
:
```

Start på test af test_precedens4.tony

```
-----
```

test_precedens4.tony

```
:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Test af TONY operatorer og præcendes regler
004: *)
005: var
006: tal: int,
007: ok: bool
008: ;
009: tal = 1;
010: ok = ! ok && ! ok || ! (tal < 20);
```

```
-----
:
```

tony.exe consol output

```
:
Options:

-msg sat til test_precedens4.consol.txt

-S er sat
```


-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

vd#tal: t#int,
vd#ok: t#bool;

v_int#tal = c#1;

v_bool#ok = t_bool#(t_bool#(!(v_bool#ok) && !(v_bool#ok)) || !(t_bool#(v_int#tal < c#20))));

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <7> variabel af typen: bool ref: 3 <10, 10, 10>

tal <6> variabel af typen: int ref: 2 <9, 10>

Slut på test af test_precedens4.tony

:

```

:
:
:
-----
:

```

Start på test af test_term1.tony

test_term1.tony

```

:
001: (*Test af TONY term's *)
002: func f1 (tal1: int, tal2: int): int
003:   return tal1 + tal2;
004: end f1;
005: var
006:   tal3: int,
007:   tal4: int,
008:   ok:   bool,
009:   tab: array of int,
010:   rec: record of {t1: int}
011:   ;
012:
013: tal3 = 1;      (* num *)
014: tal4 = tal3;   (* variable *)
015: tal4 = (tal3+1); (* (expression) *)
016:
017: ok = true;     (* true *)
018: ok = false;    (* false *)
019: ok = !ok;      (* ! *)
020: tab = null;    (* null *)
021: tal3 = |tab|;  (* | expression | *)
022:
023: tal3 = f1(1,2); (* call *)
-----

```

```

:
```

tony.exe consol output

```

:
```

Options:

-msg sat til test_term1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

-prettyXml er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i XML-format til <tonyprettyXml.xml>

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
func fd#f1 (
    vd#tal1: t#int,
    vd#tal2: t#int    ):t#int

    return t_int#( v_int#tal1 + v_int#tal2 );

end fd#f1;

var
    vd#tal3: t#int,
    vd#tal4: t#int,
    vd#ok: t#bool,
    vd#tab: t#array of t#int,
    vd#rec: t#record of {
        vd#t1: t#int
    };

v_int#tal3 = c#1;
v_int#tal4 = v_int#tal3;
v_int#tal4 = (t_int#( v_int#tal3 + c#1 )) ;
v_bool#ok = c#true;
v_bool#ok = c#false;
v_bool#ok = !(v_bool#ok) ;
v_array#tab = c#null;
v_int#tal3 = |v_array#tab| ;
v_int#tal3 = f_int#f1 (c#1, c#2);

```

:

tonyprettyXml.xml

```

:
<program linienr='1 23'>
<boddy linienr='1 23'>
<decl_List>
<declaration linienr='1 4'>
<function linienr='1 4'>
<head linienr='1 2'>
<id>f1</id>

```

```
<Par_decl_list linienr='1 2'>
<var_type linienr='1 2'>
<id>tal1</id>
<type linienr='1 0'>
<type>int</type>
</type>
</var_type>
<var_type linienr='1 2'>
<id>tal2</id>
<type linienr='1 0'>
<type>int</type>
</type>
</var_type>
</Par_decl_list>
<type linienr='1 0'>
<type>int</type>
</type>
</head>
<boddy linienr='1 4'>
<decl_List>
</decl_List>
<statement_list>
<statement linienr='1 3'>
<return linienr='1 3'>
<expression operation='add' linienr='1 3' type='int'>
<expression linienr='1 3' type='int'>
<variabel_id type='int'>tal1</variabel_id>
</expression>
<expression linienr='1 3' type='int'>
<variabel_id type='int'>tal2</variabel_id>
</expression>
</expression>
</return>
</statement>
</statement_list>
</boddy>
<tail linienr='1 4'>
<id>f1</id>
</tail>
</function>
</declaration>
<declaration linienr='1 11'>
<var_type linienr='1 6'>
<id>tal3</id>
<type linienr='1 0'>
<type>int</type>
</type>
</var_type>
<var_type linienr='1 7'>
<id>tal4</id>
<type linienr='1 0'>
<type>int</type>
</type>
</var_type>
<var_type linienr='1 8'>
<id>ok</id>
<type linienr='1 0'>
<type>bool</type>
</type>
```

```
</var_type>
<var_type linienr='l 9'>
<id>tab</id>
<type linienr='l 9'>
<array_of>
<type linienr='l 0'>
<type>int</type>
</type>
</array_of>
</type>
</var_type>
<var_type linienr='l 10'>
<id>rec</id>
<type linienr='l 10'>
<record_of>
<var_type linienr='l 10'>
<id>t1</id>
<type linienr='l 0'>
<type>int</type>
</type>
</var_type>
</record_of>
</type>
</var_type>
</declaration>
</decl_List>
<statement_list>
<statement linienr='l 13'>
<assign linienr='l 13'>
<variabel_id type='int'>tal3</variabel_id>
<expression linienr='l 13' type='int'>
<const>1</const>
</expression>
</assign>
</statement>
<statement linienr='l 14'>
<assign linienr='l 14'>
<variabel_id type='int'>tal4</variabel_id>
<expression linienr='l 14' type='int'>
<variabel_id type='int'>tal3</variabel_id>
</expression>
</assign>
</statement>
<statement linienr='l 15'>
<assign linienr='l 15'>
<variabel_id type='int'>tal4</variabel_id>
<expression linienr='l 15' type='int'>
<parantes linienr='l 15' type='int'>
<expression operation='add' linienr='l 15' type='int'>
<expression linienr='l 15' type='int'>
<variabel_id type='int'>tal3</variabel_id>
</expression>
<expression linienr='l 15' type='int'>
<const>1</const>
</expression>
</expression>
</parantes>
</expression>
</assign>
```

```
</statement>
<statement linienr='l 17'>
<assign linienr='l 17'>
<variabel_id type='bool'>ok</variabel_id>
<expression linienr='l 17' type='bool'>
<const>true</const>
</expression>
</assign>
</statement>
<statement linienr='l 18'>
<assign linienr='l 18'>
<variabel_id type='bool'>ok</variabel_id>
<expression linienr='l 18' type='bool'>
<const>>false</const>
</expression>
</assign>
</statement>
<statement linienr='l 19'>
<assign linienr='l 19'>
<variabel_id type='bool'>ok</variabel_id>
<expression linienr='l 19' type='bool'>
<not linienr='l 19' type='bool'>
<variabel_id type='bool'>ok</variabel_id>
</not>
</expression>
</assign>
</statement>
<statement linienr='l 20'>
<assign linienr='l 20'>
<variabel_id type='array'>tab</variabel_id>
<expression linienr='l 20' type='null'>
<const>>null</const>
</expression>
</assign>
</statement>
<statement linienr='l 21'>
<assign linienr='l 21'>
<variabel_id type='int'>tal3</variabel_id>
<expression linienr='l 21' type='int'>
<numeric linienr='l 21' type='int'>
<expression linienr='l 21' type='array'>
<variabel_id type='array'>tab</variabel_id>
</expression>
</numeric>
</expression>
</assign>
</statement>
<statement linienr='l 23'>
<assign linienr='l 23'>
<variabel_id type='int'>tal3</variabel_id>
<expression linienr='l 23' type='int'>
<call linienr='l 23' type='int'>
<id>f1</id>
<act_list linienr='l 23'>
<expression linienr='l 23' type='int'>
<const>1</const>
</expression>
<expression linienr='l 23' type='int'>
<const>2</const>
```

```

</expression>
</act_list>
</call>
</expression>
</assign>
</statement>
</statement_list>
</boddy>
</program>

```

:

tonysymbol_xref.txt

:

```

-----
SymbolTable for MAIN: <0>  tabelNr:1 scope-level:0 - start
ok <8>   variabel af typen: bool  ref: 4 <17, 18, 19, 19>
tab  <9>   variabel af typen: array of int  ref: 2  <20, 21>
rec  <10>  variabel af typen: record<10>
f1 <2>   function af typen: int  ref: 1  <23>
tal3 <6>   variabel af typen: int  ref: 5  <13, 14, 15, 21, 23>
tal4 <7>   variabel af typen: int  ref: 2  <14, 15>

```

SymbolTable for f1: <2> tabelNr:2 scope-level:1 - start

```

tal1 <2>   variabel af typen: int  ref: 1  <3>
tal2 <2>   variabel af typen: int  ref: 1  <3>

```

SymbolTable for rec: <10> tabelNr:3 scope-level:0 - start

```

t1 <10>   variabel af typen: int

```

Slut på test af test_term1.tony

:

```

:.....:
:
:

```

:

Start på test af test_weed1.tony

test_weed1.tony

:

```
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Aftester om kontrol af om navn i head og tail matcher
004: *)
005:
006: func f0 (): int    (* navn i head og tail forskellige *)
007: return 1;
008: end f0x;
009:
010: var t: int;
011: (* krav om mindst eet statement *)
012: t = 1;
013:
014: (* ingen krav om return i "main" rutine *)
-----
```

```
:
```

tony.exe consol output

```
:
```

Options:

-msg sat til test_weed1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

8: Advarsel funktion afslutter ikke med samme navn <f0x> som den er defineret med <f0> i linie: <6>

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 1

Antal advarsler: 0

```
-----
```

```
:
```

tonyprettyTxt.txt

```
:
```



```
func fd#f0 ( ):t#int
```

```
    return c#1;
```

```
end fd#f0x;
```

```
var
```

```
    vd#t: t#int;
```

```
v_int#t = c#1;
```

```
-----
```

```
:
```

```
tonyprettyXml.xml
```

```
:
```

```
-----
```

```
:
```

tonysymbol_xref.txt

```
:
```

```
-----
```

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

t <10> variabel af typen: int ref: 1 <12>

f0 <6> function af typen: int

```
-----
```

SymbolTable for f0: <6> tabelNr:2 scope-level:1 - start

```
-----
```

Slut på test af test_weed1.tony

```
:
```

```
.....
```

```
:
```

```
:
```

```
-----
```

```
:
```

Start på test af test_weed2.tony

```
-----
```

test_weed2.tony

```
:
```

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24

002:

003: Aftester kontrol af garanteret return i funktioner

004: *)

005:

006: func f1 (): int (* normal med return *)

007: return 1;

008: end f1;

009:

010: func f2 (): int (* uden return *)

011: var tal: int;

012: tal = 2;

013: end f2;

```
014:
015: func f3 (): int    (* normal med nested return *)
016: var tal: int;
017: tal = 1;
018: if (tal >= 2) then
019:   if (tal >= 3) then
020:     return 4;
021:   else
022:     return 5;
023: else
024:   if (tal >= 6) then
025:     return 7;
026:   else
027:     return 8;
028: end f3;
029:
030: func f4 (): int    (* med manglende nested return *)
031: var tal: int;
032: tal = 1;
033: if (tal >= 2) then
034:   if (tal >= 3) then
035:     return 4;
036:   else
037:     return 5;
038: else
039:   if (tal >= 6) then
040:     return 7;
041: end f4;
042:
043: func f5 (): int    (* med manglende return uden for while *)
044: var tal: int;
045: tal = 1;
046: while (tal < 10) do
047: {
048:   return 1;
049: }
050: end f5;
051:
052:
053:
054: func f6 (): int    (* normal med nested return i compound *)
055: var tal: int;
056: tal = 1;
057: if (tal >= 2) then
058: {
059:   if (tal >= 3) then
060:   {
061:     return 4;
062:   }
063:   else
064:   {
065:     return 5;
066:   }
067: }
068: else
069: {
070:   if (tal >= 6) then
071:   {
072:     return 7;
```

```
073:   }
074:   else
075:   {
076:       return 8;
077:   }
078: }
079: end f6;
080:
081: func f7 (): int    (* med manglende nested return og compound *)
082: var tal: int;
083: tal = 1;
084: if (tal >= 2) then
085: {
086:     if (tal >= 3) then
087:     {
088:         return 4;
089:     }
090:     else
091:     {
092:         return 5;
093:     }
094: }
095: else
096: {
097:     if (tal >= 6) then
098:         return 7;
099: }
100: end f7;
101:
102:
103:
104:
105:
106: var t: int;
107: (* krav om mindst eet statement *)
108: t = 1;
109:
110: (* ingen krav om return i "main" rutine *)
```

:

tony.exe consol output

:

Options:

-msg sat til test_weed2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

alvorligFejl: funktion <f2> linie: <10> mangler return statement

alvorligFejl: funktion <f4> linie: <30> mangler return statement
 alvorligFejl: funktion <f5> linie: <43> mangler return statement
 alvorligFejl: funktion <f7> linie: <81> mangler return statement
 TONY weeding fase slut
 TONY opbygning af symboltabel start
 TONY opbygning af symboltabel slut
 TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
 TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 4

Antal fejl: 0

Antal advarsler: 0

 :

tonyprettyTxt.txt

```

:
func fd#f1 ( ):t#int
    return c#1;

end fd#f1;

func fd#f2 ( ):t#int
    var
        vd#tal: t#int;

        v_int#tal = c#2;

end fd#f2;

func fd#f3 ( ):t#int
    var
        vd#tal: t#int;

        v_int#tal = c#1;
    if (t_bool#( v_int#tal >= c#2 )) then {
        if (t_bool#( v_int#tal >= c#3 )) then {
            return c#4;
        }
        else {
            return c#5;
        }
    }
    else {
        if (t_bool#( v_int#tal >= c#6 )) then {
            return c#7;
        }
        else {

```

```
        return c#8;
    }
}
```

```
end fd#f3;
```

```
func fd#f4 ( ):t#int
```

```
    var
        vd#tal: t#int;

    v_int#tal = c#1;
    if (t_bool#( v_int#tal >= c#2 )) then {
        if (t_bool#( v_int#tal >= c#3 )) then {
            return c#4;
        }
        else {
            return c#5;
        }
    }
    else {
        if (t_bool#( v_int#tal >= c#6 )) then {
            return c#7;
        }
    }
}
```

```
end fd#f4;
```

```
func fd#f5 ( ):t#int
```

```
    var
        vd#tal: t#int;

    v_int#tal = c#1;
    while (t_bool#( v_int#tal < c#10 )) {
        #{
            return c#1;
        #}
    }
}
```

```
end fd#f5;
```

```
func fd#f6 ( ):t#int
```

```
    var
        vd#tal: t#int;

    v_int#tal = c#1;
    if (t_bool#( v_int#tal >= c#2 )) then {
        #{
            if (t_bool#( v_int#tal >= c#3 )) then {
                #{
                    return c#4;
                #}
            }
            else {
                #{
                    return c#5;
                #}
            }
        #}
    }
}
```

```

    }
  #}
}
else {
  #{
    if (t_bool#( v_int#tal >= c#6 ))      then {
      #{
        return c#7;
      #}
    }
    else {
      #{
        return c#8;
      #}
    }
  #}
}
}

```

end fd#f6;

func fd#f7 ():t#int

```

var
  vd#tal: t#int;

v_int#tal = c#1;
if (t_bool#( v_int#tal >= c#2 ))  then {
  #{
    if (t_bool#( v_int#tal >= c#3 ))  then {
      #{
        return c#4;
      #}
    }
    else {
      #{
        return c#5;
      #}
    }
  #}
}
else {
  #{
    if (t_bool#( v_int#tal >= c#6 ))  then {
      return c#7;
    }
  #}
}
}

```

end fd#f7;

```

var
  vd#t: t#int;

```

```

v_int#t = c#1;

```

```

-----
:
tonyprettyXml.xml
:
-----

```

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelNr: 1 scope-level: 0 - start

t <106> variabel af typen: int ref: 1 <108>

f1 <6> function af typen: int

f2 <10> function af typen: int

f3 <15> function af typen: int

f4 <30> function af typen: int

f5 <43> function af typen: int

f6 <54> function af typen: int

f7 <81> function af typen: int

SymbolTable for f1: <6> tabelNr: 2 scope-level: 1 - start

SymbolTable for f2: <10> tabelNr: 3 scope-level: 1 - start

tal <11> variabel af typen: int ref: 1 <12>

SymbolTable for f3: <15> tabelNr: 4 scope-level: 1 - start

tal <16> variabel af typen: int ref: 4 <17, 18, 19, 24>

SymbolTable for f4: <30> tabelNr: 5 scope-level: 1 - start

tal <31> variabel af typen: int ref: 4 <32, 33, 34, 39>

SymbolTable for f5: <43> tabelNr: 6 scope-level: 1 - start

tal <44> variabel af typen: int ref: 2 <45, 46>

SymbolTable for f6: <54> tabelNr: 7 scope-level: 1 - start

tal <55> variabel af typen: int ref: 4 <56, 57, 59, 70>

SymbolTable for f7: <81> tabelNr:8 scope-level:1 - start

tal <82> variabel af typen: int ref: 4 <83, 84, 86, 97>

 Slut på test af test_weed2.tony

:
 :::
 :
 :
 :

 :

Start på test af test_weed3.tony

test_weed3.tony

:
 001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
 002:
 003: Aftester om statements efter garanteres return fanges
 004: *)
 005: func f6 (): int (* normal med statement efter return *)
 006: var tal: int;
 007: tal = 1;
 008: return 1;
 009: tal = 2;
 010: tal = 3;
 011: end f6;
 012:
 013: var t: int;
 014: (* krav om mindst eet statement *)
 015: t = 1;
 016:
 017: (* ingen krav om return i "main" rutine *)

:

tony.exe consol output

:

Options:

-msg sat til test_weed3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

9: Advarsel statement vil aldrig blive udfoert p.g.a. tidligere return

10: Advarsel statement vil aldrig blive udfoert p.g.a. tidligere return

TONY weeding fase slut
 TONY opbygning af symboltabel start
 TONY opbygning af symboltabel slut
 TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
 TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 2

 :

tonyprettyTxt.txt

```

:
func fd#f6 ( ):t#int
    var
        vd#tal: t#int;

    v_int#tal = c#1;
    return c#1;
    inakt# v_int#tal = c#2;
    inakt# v_int#tal = c#3;

```

end fd#f6;

```

var
    vd#t: t#int;

```

```

v_int#t = c#1;

```

 :
 tonyprettyXml.xml
 :

 :

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

t <13> variabel af typen: int ref: 1 <15>

f6 <5> function af typen: int

SymbolTable for f6: <5> tabelNr:2 scope-level:1 - start

tal <6> variabel af typen: int ref: 3 <7, 9, 10>

Slut på test af test_weed3.tony

:
:.....
:
:
:-----
:

Start på test af test_weed4.tony

test_weed4.tony

:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Aftester om selvrefererende type declarationer fanges
004: *)
005:
006: type heltal = int; (* i "main" *)
007: func f7 (): int (* normal *)
008: type heltal = heltal; (* skal fanges *)
009: type nytHeltal = heltal; (* ok *)
010: return 7;
011: end f7;
012:
013: var t: int;
014: (* krav om mindst eet statement *)
015: t = 1;
016:
017: (* ingen krav om return i "main" rutine *)

:

tony.exe consol output

:
Options:

-msg sat til test_weed4.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start
8: Fejl navn<heltal> og type er identisk
TONY weeding fase slut
TONY opbygning af symboltabel start
TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 1

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
type td#heltal :t#int;

func fd#f7 ( ):t#int
    inakt#type td#heltal :t#heltal;
    type td#nytHeltal :t#heltal;
    return c#7;

end fd#f7;

var
    vd#t: t#int;

v_int#t = c#1;

```

:
tonyprettyXml.xml
:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

t <13> variabel af typen: int ref: 1 <15>

heltal<6> type = int ref: 1 <9>

f7 <7> function af typen: int

SymbolTable for f7: <7> tabelNr:2 scope-level:1 - start

nytHeltal <9> type = heltal<0> => int

 Slut på test af test_weed4.tony

:
 :::
 :
 :
 :

 :

Start på test af test_typereturn1.tony

test_typereturn1.tony

:
 001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
 002:
 003: Aftester om type-check af for return fra funktioner
 004: *)
 005: type tab = array of int;
 006: func f1 (t: tab, i: int): int (* test af retur typer *)
 007: if (i == 1) then
 008: return 1; (* retur type ok *)
 009: else
 010: return t; (* retur type fejl *)
 011: end f1;
 012:
 013: var t: int;
 014: (* krav om mindst eet statement *)
 015: t = 1;

tony.exe consol output

:
 Options:

-msg sat til test_typereturn1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

10: Fejl: returtype <tab -> array> svarer ikke til funktionshoved<int -> int>

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 1

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:
type td#tab : t#array of t#int;

```
func fd#f1 (
    vd#t: t#tab,
    vd#i: t#int    ):t#int

    if (t_bool#( v_int#i == c#1 ))    then {
        return c#1;
    }
    else {
        return v_tab#t;
    }

end fd#f1;
```

```
var
    vd#t: t#int;
```

```
v_int#t = c#1;
```

:
tonyprettyXml.xml
:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelNr: 1 scope-level: 0 - start

t <13> variabel af typen: int ref: 1 <15>

tab <5> type = array of int ref: 1 <6>

f1 <6> function af typen: int

SymbolTable for f1: <6> tabelNr: 2 scope-level: 1 - start

i <6> variabel af typen: int ref: 1 <7>

t <6> variabel af typen: tab<5> => array of int ref: 1 <10>

Slut på test af test_typereturn1.tony

```
:
```

.....

```
:
```

```
:
```

Start på test af test_typereturn2.tony

test_typereturn2.tony

```
:
```

```
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
```

```
002:
```

```
003: Aftester om type-check af for return fra funktioner
```

```
004: I min udgave tillades returnering af forskellig type, der
```

```
005: fører til samme grundtype
```

```
006: *)
```

```
007: type heltal = int;
```

```
008: type h2 = int;
```

```
009: func f2 (h: heltal, i: int): h2 (* test af retur typer *)
```

```
010:   if (i == 1) then
```

```
011:     return 1; (* retur type ok - grundtype int*)
```

```
012:   else
```

```
013:     return h; (* retur type ok - grundtype int *)
```

```
014: end f2;
```

```
015:
```

```
016: var t: int;
```

```
017: (* krav om mindst eet statement *)
```

```
018: t = 1;
```

```
:
```

tony.exe consol output

```
:
```

```
Options:
```

-msg sat til test_typereturn2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start
TONY weeding fase slut

TONY opbygning af symboltabel start
 TONY opbygning af symboltabel slut
 TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
 TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

 :

tonyprettyTxt.txt

```

:
type td#heltal : t#int;

type td#h2 : t#int;

func fd#f2 (
    vd#h: t#heltal,
    vd#i: t#int      ):t#h2

    if (t_bool#( v_int#i == c#1 )) then {
        return c#1;
    }
    else {
        return v_heltal#h;
    }

end fd#f2;

var
    vd#t: t#int;

v_int#t = c#1;

```

 :
 tonyprettyXml.xml
 :

 :

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

t <16> variabel af typen: int ref: 1 <18>

heltal<7> type = int ref: 1 <9>

-msg sat til test_typereturn3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

18: Fejl: returtype <tab3 -> array> svarer ikke til funktionshoved<tab -> array>

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 1

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

type td#tab : t#array of t#int;

type td#tab2 : t#tab;

type td#tab3 : t#array of t#int;

```
func fd#f3 (
    vd#t1: t#tab,
    vd#t2: t#tab2,
    vd#t3: t#tab3,
    vd#i: t#int    ):t#tab

    if (t_bool#( v_int#i == c#1 ))    then {
        return v_tab#t1;
    }
    else {
        if (t_bool#( v_int#i == c#2 ))    then {
            return v_tab2#t2;
        }
        else {
            return v_tab3#t3;
        }
    }
}
```

```
end fd#f3;

var
  vd#t: t#int;
```

```
v_int#t = c#1;
```

```
-----
:
tonyprettyXml.xml
:
-----
:
```

tonysymbol_xref.txt

```
:
-----
```

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

t <21> variabel af typen: int ref: 1 <23>

tab <9> type = array of int ref: 3 <10, 12, 12>

f3 <12> function af typen: tab<9> => array of int

tab2 <10> type = tab<9> => array of int ref: 1 <12>

tab3 <11> type = array of int ref: 1 <12>

```
-----
```

SymbolTable for f3: <12> tabelNr:2 scope-level:1 - start

i <12> variabel af typen: int ref: 2 <13, 15>

t1 <12> variabel af typen: tab<9> => array of int ref: 1 <14>

t2 <12> variabel af typen: tab2<10> => array of int ref: 1 <16>

t3 <12> variabel af typen: tab3<11> => array of int ref: 1 <18>

```
-----
```

Slut på test af test_typereturn3.tony

```
:
:
:
:
:
-----
:
```

Start på test af test_typecall1.tony

```
-----
```

test_typecall1.tony

```
:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
```

```

003: Aftester om type-check af parametre og ved funktionskald
004: *)
005: type tab = array of int;
006: type rec = record of { t: tab, i: int };
007: func f1 (t: tab, r: rec, i: int): int
008: return 1;
009: end f1;
010:
011: var mt: tab, mr: rec, mi: int;
012:
013: mi = f1 (mt, mr, mi); (* ok *)
014: mi = f1 (mt, mi); (* fejl i antal *)
015: mi = f1 (mr, mr, mi); (* fejl i typer *)
-----

```

```

:
```

tony.exe consol output

```

:
```

Options:

-msg sat til test_typecall1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

14: Fejl: i antal argumenter(2) i kald af funktion f1<7> - forventet(3)

15: Fejl: argument type <rec> passer ikke med parameter type <tab>

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 2

Antal fejl: 0

Antal advarsler: 0

```

-----
:
```

tonyprettyTxt.txt

```

:
```

type td#tab : t#array of t#int;

```

type td#rec : t#record of {
  vd#t: t#tab,
  vd#i: t#int
};

func fd#f1 (
  vd#t: t#tab,
  vd#r: t#rec,
  vd#i: t#int    ):t#int

  return c#1;

end fd#f1;

var
  vd#mt: t#tab,
  vd#mr: t#rec,
  vd#mi: t#int;

v_int#mi = f_int#f1 (v_tab#mt, v_rec#mr, v_int#mi);
v_int#mi = f_int#f1 (v_???#mt, v_???#mi);
v_int#mi = f_int#f1 (v_rec#mr, v_rec#mr, v_int#mi);

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

```

:
-----
SymbolTable for MAIN: <0>  tabelNr:1 scope-level:0 - start

mi <11>  variabel af typen: int  ref: 5  <13, 13, 14, 15, 15>

mr <11>  variabel af typen: rec<6> => record<6>  ref: 3  <13, 15, 15>

mt <11>  variabel af typen: tab<5> => array of int  ref: 1  <13>

tab  <5>  type = array of int  ref: 3  <6, 7, 11>

rec  <6>  type = record<6>  ref: 2  <7, 11>

f1 <7>  function af typen: int  ref: 3  <13, 14, 15>

```

```

:
-----
SymbolTable for rec: <6>  tabelNr:2 scope-level:0 - start

```

```

i  <6>  variabel af typen: int

t  <6>  variabel af typen: tab<5> => array of int

```

```

:
-----
SymbolTable for f1: <7>  tabelNr:3 scope-level:1 - start

```

i <7> variabel af typen: int
 r <7> variabel af typen: rec<6> => record<6>
 t <7> variabel af typen: tab<5> => array of int

 Slut på test af test_typecall1.tony

:
 :::
 :
 :
 :

 :

Start på test af test_typecall2.tony

test_typecall2.tony

```
:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Aftester om type-check for funktioners returnværdier
004: Checkkes med brug af assign
005: *)
006: type tab = array of int;
007: type rec = record of {i:int};
008: type rectab = array of rec;
009: func f1 (i:int): int
010: return i;
011: end f1;
012: func f2 (t:tab): tab
013: return t;
014: end f2;
015: func f3 (r:rec): rec
016: return r;
017: end f3;
018: func f4 (rt:rectab): rectab
019: return rt;
020: end f4;
021:
022: var mi: int, mt: tab, mr: rec, mrt: rectab;
023: mi = f1(mi); (* ok *)
024: mt = f2(mt); (* ok *)
025: mr = f3(mr); (* ok *)
026: mrt = f4(mrt); (* ok *)
027: mi = f2(mt); (* fejl *)
028: mi = f3(mr); (* fejl *)
029: mi = f4(mrt); (* fejl *)
```

 :

tony.exe consol output

:
 Options:

-msg sat til test_typecall2.console.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

27: Fejl: type i assignment <int> = <tab> : der tilades kun ens typer

28: Fejl: type i assignment <int> = <rec> : der tilades kun ens typer

29: Fejl: type i assignment <int> = <rectab> : der tilades kun ens typer

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 3

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

type td#tab : t#array of t#int;

type td#rec : t#record of {

 vd#i: t#int

};

type td#rectab : t#array of t#rec;

func fd#f1 (
 vd#i: t#int):t#int

 return v_int#i;

end fd#f1;

func fd#f2 (
 vd#t: t#tab):t#tab

 return v_tab#t;

end fd#f2;

```

func fd#f3 (
    vd#r: t#rec    ):t#rec

    return v_rec#r;

end fd#f3;

func fd#f4 (
    vd#rt: t#rectab    ):t#rectab

    return v_rectab#rt;

end fd#f4;

var
    vd#mi: t#int,
    vd#mt: t#tab,
    vd#mr: t#rec,
    vd#mrt: t#rectab;

v_int#mi = f_int#f1 (v_int#mi);
v_tab#mt = f_tab#f2 (v_tab#mt);
v_rec#mr = f_rec#f3 (v_rec#mr);
v_rectab#mrt = f_rectab#f4 (v_rectab#mrt);
v_int#mi = f_tab#f2 (v_tab#mt);
v_int#mi = f_rec#f3 (v_rec#mr);
v_int#mi = f_rectab#f4 (v_rectab#mrt);

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

```

SymbolTable for MAIN: <0>  tabelNr:1 scope-level:0 - start

```

```

mi <22>  variabel af typen: int  ref: 5  <23, 23, 27, 28, 29>

```

```

mr <22>  variabel af typen: rec<7> => record<7>  ref: 3  <25, 25, 28>

```

```

mt <22>  variabel af typen: tab<6> => array of int  ref: 3  <24, 24, 27>

```

```

tab  <6>  type = array of int  ref: 3  <12, 12, 22>

```

```

rec  <7>  type = record<7>  ref: 4  <8, 15, 15, 22>

```

```

mrt  <22>  variabel af typen: rectab<8> => array of rec<7> => record<7>  ref: 3<26,
26, 29>

```

```

rectab  <8>  type = array of rec<7> => record<7>  ref: 3  <18, 18, 22>

```

```

f1  <9>  function af typen: int  ref: 1  <23>

```

```

f2  <12>  function af typen: tab<6> => array of int  ref: 2  <24, 27>

```

f3 <15> function af typen: rec<7> => record<7> ref: 2 <25, 28>

f4 <18> function af typen: rectab<8> => array of rec<7> => record<7> ref: 2 <26, 29>

SymbolTable for rec: <7> tabelNr:2 scope-level:0 - start

i <7> variabel af typen: int

SymbolTable for f1: <9> tabelNr:3 scope-level:1 - start

i <9> variabel af typen: int ref: 1 <10>

SymbolTable for f2: <12> tabelNr:4 scope-level:1 - start

t <12> variabel af typen: tab<6> => array of int ref: 1 <13>

SymbolTable for f3: <15> tabelNr:5 scope-level:1 - start

r <15> variabel af typen: rec<7> => record<7> ref: 1 <16>

SymbolTable for f4: <18> tabelNr:6 scope-level:1 - start

rt <18> variabel af typen: rectab<8> => array of rec<7> => record<7> ref: 1 <19>

Slut på test af test_typecall2.tony

:

.....

:

:

:

Start på test af test_typeassign1.tony

test_typeassign1.tony

:

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24

002:

003: Aftester om type-check ved assign

004: Kun assign af samme type tillades

005: null tillades for array og record

006: *)

007: type tab = array of int;

008: type rec = record of {i:int};


```

009: type rec2 = rec;
010: type tal = int;
011:
012: var i: int, b: bool, t: tab, r: rec, r2: rec2, i2: tal;
013:
014: i = i;    (* ok *)
015: b = b;    (* ok *)
016: t = t;    (* ok *)
017: r = r;    (* ok *)
018: r2 = r2; (* ok *)
019: i = t[1]; (* ok *)
020: i = r.i;  (* ok *)
021: r2.i = r.i; (* ok *)
022: i2 = i2; (* ok *)
023: b = true; (* ok *)
024: b = false; (* ok *)
025: t = null; (* ok *)
026: r = null; (* ok *)
027: r2 = r;   (* fejl *)
028: i = r;    (* fejl *)
029: i = i2;   (* fejl *)
030: i2 = i;   (* fejl *)

```

:

tony.exe consol output

:

Options:

-msg sat til test_typeassign1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

27: Fejl: type i assignment <rec2> = <rec> : der tilades kun ens typer

28: Fejl: type i assignment <int> = <rec> : der tilades kun ens typer

29: Fejl: type i assignment <int> = <tal> : der tilades kun ens typer

30: Fejl: type i assignment <tal> = <int> : der tilades kun ens typer

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 4

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
type td#tab : t#array of t#int;

type td#rec : t#record of {
    vd#i: t#int
};

type td#rec2 : t#rec;

type td#tal : t#int;

var
    vd#i: t#int,
    vd#b: t#bool,
    vd#t: t#tab,
    vd#r: t#rec,
    vd#r2: t#rec2,
    vd#i2: t#tal;

v_int#i = v_int#i;
v_bool#b = v_bool#b;
v_tab#t = v_tab#t;
v_rec#r = v_rec#r;
v_rec2#r2 = v_rec2#r2;
v_int#i = v_tab#t[c#1];
v_int#i = v_rec#r.v_int#i;
v_rec2#r2.v_int#i = v_rec#r.v_int#i;
v_tal#i2 = v_tal#i2;
v_bool#b = c#true;
v_bool#b = c#false;
v_tab#t = c#null;
v_rec#r = c#null;
v_rec2#r2 = v_rec#r;
v_int#i = v_rec#r;
v_int#i = v_tal#i2;
v_tal#i2 = v_int#i;

```

:
tonyprettyXml.xml

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

b <12> variabel af typen: bool ref: 4 <15, 15, 23, 24>

i <12> variabel af typen: int ref: 7 <14, 14, 19, 20, 28, 29, 30>
 r <12> variabel af typen: rec<8> => record<8> ref: 7 <17, 17, 20, 21, 26, 27, 28>
 t <12> variabel af typen: tab<7> => array of int ref: 4 <16, 16, 19, 25>
 tab <7> type = array of int ref: 1 <12>
 rec <8> type = record<8> ref: 2 <9, 12>
 tal<10> type = int ref: 1 <12>
 i2 <12> variabel af typen: tal<0> => int ref: 4 <22, 22, 29, 30>
 r2 <12> variabel af typen: rec2<9> => record<8> ref: 4 <18, 18, 21, 27>
 rec2 <9> type = rec<8> => record<8> ref: 1 <12>

 SymbolTable for rec: <8> tabelNr:2 scope-level:0 - start

i <8> variabel af typen: int ref: 3 <20, 21, 21>

 Slut på test af test_typeassign1.tony

:
 :::::::::::::::::::::::::::::::::::::::
 :
 :
 :

 :

Start på test af test_typewrite1.tony

test_typewrite1.tony

:
 001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
 002:
 003: Aftester om type-check ved write
 004: Min udgave tillader alle typer baseret på grundtypen int
 005: *)
 006: type tab = array of int;
 007: type rec = record of {i:int};
 008: type tal = int;
 009:
 010: var i: int, b: bool, t: tab, r: rec, i2: tal;
 011:
 012: write i; (* ok *)
 013: write t[1]; (* ok *)
 014: write r.i; (* ok *)
 015: write i2; (* ok *)
 016: write b; (* fejl *)
 017: write t; (* fejl *)
 018: write r; (* fejl *)

:

tony.exe consol output

:

Options:

-msg sat til test_typewrite1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

16: Fejl: expression i write er af typen<bool> men kun <int> er accepteret

17: Fejl: expression i write er af typen<array> men kun <int> er accepteret

18: Fejl: expression i write er af typen<record> men kun <int> er accepteret

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 3

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

type td#tab : t#array of t#int;

type td#rec : t#record of {
 vd#i: t#int
};

type td#tal : t#int;

var

 vd#i: t#int,
 vd#b: t#bool,
 vd#t: t#tab,
 vd#r: t#rec,
 vd#i2: t#tal;

```

write v_int#i;
write v_tab#t[c#1];
write v_rec#r.v_int#i;
write v_tal#i2;
write v_bool#b;
write v_tab#t;
write v_rec#r;

```

```

-----
:
tonyprettyXml.xml
:

```

```

-----
:

```

tonysymbol_xref.txt

```

:

```

```

-----
SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

```

```

b <10>  variabel af typen: bool  ref: 1 <16>
i <10>  variabel af typen: int   ref: 1  <12>
r <10>  variabel af typen: rec<7> => record<7>  ref: 2  <14, 18>
t <10>  variabel af typen: tab<6> => array of int  ref: 2  <13, 17>
tab <6>   type = array of int  ref: 1  <10>
rec <7>   type = record<7>    ref: 1  <10>
tal<8>   type = int           ref: 1  <10>
i2 <10>  variabel af typen: tal<0> => int  ref: 1  <15>

```

```

-----
SymbolTable for rec: <7> tabelNr:2 scope-level:0 - start

```

```

i <7>   variabel af typen: int  ref: 1  <14>

```

```

-----
Slut på test af test_typewrite1.tony

```

```

:

```

```

:

```

```

:

```

```

:

```

```

-----
:

```

Start på test af test_typenew1.tony

```

-----

```

test_typenew1.tony

```

:

```

```

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24

```

```

002:
003: Aftester om type-check ved new og new .. length of
004: *)
005: type tab = array of int;
006: type rec = record of {i:int};
007: type rec2 = rec;
008:
009: var i: int, b: bool, t: tab, r: rec, r2: rec2;
010:
011: new t of length 10;    (* ok *)
012: new r;                (* ok *)
013: new r2;               (* ok *)
014: new i of length 10; (* fejl *)
015: new b of length 10;  (* fejl *)
016: new r of length 10;  (* fejl *)
017: new r2 of length 10; (* fejl *)
018: new i;                (* fejl *)
019: new b;                (* fejl *)
020: new t;                (* fejl *)
021: new t of length b;   (* fejl *)
-----
:

```

tony.exe consol output

```

:
Options:

-msg sat til test_typenew1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

```

TONY Parser afsluttet normal

```

TONY weeding fase start
TONY weeding fase slut
TONY opbygning af symboltabel start
TONY opbygning af symboltabel slut
TONY type-check start

```

```

14: Fejl: variabel i new i length of .. har af typen <int> men kun <array> er accepteret
15: Fejl: variabel i new b length of .. har af typen <bool> men kun <array> er accepteret
16: Fejl: variabel i new r length of .. har af typen <record> men kun <array> er accepteret
17: Fejl: variabel i new r2 length of .. har af typen <record> men kun <array> er accepteret
18: Fejl: variabel i new i af typen <int> men kun <record> er accepteret
19: Fejl: variabel i new b af typen <bool> men kun <record> er accepteret
20: Fejl: variabel i new t af typen <array> men kun <record> er accepteret
21: Fejl: expression i new t length of .. er af typen <bool> men kun <int> er accepteret
TONY type-check slut

```

```

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

```

Antal alvorligFejl: 8

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

type td#tab : t#array of t#int;

type td#rec : t#record of {
 vd#i: t#int
 };

type td#rec2 : t#rec;

var

 vd#i: t#int,
 vd#b: t#bool,
 vd#t: t#tab,
 vd#r: t#rec,
 vd#r2: t#rec2;

new v_tab#t of length c#10;

new v_rec#r;

new v_rec2#r2;

new v_int#i of length c#10;

new v_bool#b of length c#10;

new v_rec#r of length c#10;

new v_rec2#r2 of length c#10;

new v_int#i;

new v_bool#b;

new v_tab#t;

new v_tab#t of length v_bool#b;

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

b <9> variabel af typen: bool ref: 3 <15, 19, 21>

i <9> variabel af typen: int ref: 2 <14, 18>

r <9> variabel af typen: rec<6> => record<6> ref: 2 <12, 16>

t <9> variabel af typen: tab<5> => array of int ref: 3 <11, 20, 21>

```

tab <5>    type = array of int  ref: 1  <9>
rec  <6>    type = record<6>    ref: 2  <7, 9>
r2 <9>     variabel af typen: rec2<7> => record<6>  ref: 2 <13, 17>
rec2 <7>    type = rec<6> => record<6>  ref: 1  <9>

```

SymbolTable for rec: <6> tabelNr:2 scope-level:0 - start

i <6> variabel af typen: int

Slut på test af test_typenew1.tony

```

:
:
:
:
:
:
:
:
:
:

```

Start på test af test_typeif1.tony

test_typeif1.tony

```

:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Aftester om type-check ved if og if-else
004: *)
005:
006: var i: int, b: bool;
007:
008: if (b) then          (* ok *)
009: b = b;
010: if (b) then          (* ok *)
011: b = b;
012: else
013: b = b;
014:
015: if (i) then          (* fejl *)
016: b = b;
017: if (i) then          (* fejl *)
018: b = b;
019: else
020: b = b;

```

tony.exe consol output

```

:
Options:

-msg sat til test_typeif1.consol.txt

```


-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

15: Fejl: expression i if er af typen <int> men kun <bool> er accepteret

17: Fejl: expression i if-else er af typen <int> men kun <bool> er accepteret

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 2

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

```
    vd#i: t#int,
    vd#b: t#bool;
```

```
if (v_bool#b) then {
    v_bool#b = v_bool#b;
}
```

```
if (v_bool#b) then {
    v_bool#b = v_bool#b;
}
```

```
else {
    v_bool#b = v_bool#b;
}
```

```
if (v_int#i) then {
    v_bool#b = v_bool#b;
}
```

```
if (v_int#i) then {
    v_bool#b = v_bool#b;
}
```

```
else {
    v_bool#b = v_bool#b;
}
```

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

b <6> variabel af typen: bool ref: 14 <8, 9, 9, 10, 11, 11, 13, 13, 16, 16, 18, 18, 20, 20>

i <6> variabel af typen: int ref: 2 <15, 17>

Slut på test af test_typeif1.tony

:

::

:

:

:

Start på test af test_typewhile1.tony

test_typewhile1.tony

:

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24

002:

003: Aftester om type-check ved while

004: *)

005:

006: var i: int, b: bool;

007:

008: while (b) do (* ok *)

009: b = b;

010: while (i) do (* fejl *)

011: b = b;

:

tony.exe consol output

:

Options:

-msg sat til test_typewhile1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start
 TONY weeding fase slut
 TONY opbygning af symboltabel start
 TONY opbygning af symboltabel slut
 TONY type-check start

11: Fejl: expression i while er af typen <int> men kun <bool> er accepteret
 TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
 TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 1

Antal fejl: 0

Antal advarsler: 0

 :

tonyprettyTxt.txt

```

:
var
  vd#i: t#int,
  vd#b: t#bool;

while (v_bool#b) {
  v_bool#b = v_bool#b;
}
while (v_int#i) {
  v_bool#b = v_bool#b;
}

```

 :
 tonyprettyXml.xml
 :

 :

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

b <6> variabel af typen: bool ref: 5 <8, 9, 9, 11, 11>

i <6> variabel af typen: int ref: 1 <10>

 Slut på test af test_typewhile1.tony

```

:
:
:
:

```

```

:
-----
:

```

Start på test af test_typeexp1.tony

test_typeexp1.tony

```

:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Test af TONY operatorer, præcendens regler og typer
004: *)
005: var
006: tal:  int,
007: ok:    bool
008: ;
009: tal = 1 + 2 * 3;      (* ok *)
010: tal = 1 + 2 / 3;    (* ok *)
011: tal = 1 + 2 * 3 / 4 * 5; (* ok *)
012: tal = 1 - 2 + 3 - 4; (* ok *)
013: tal = tal + ok + ok * ok; (* fejl *)
014: tal = |tal|;        (* ok *)
015: tal = |ok|;         (* fejl *)
-----

```

tony.exe consol output

```

:
Options:

-msg sat til test_typeexp1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start
TONY weeding fase slut
TONY opbygning af symboltabel start
TONY opbygning af symboltabel slut
TONY type-check start

13: Fejl: type i expression <int> + <bool> der tilades kun type int
13: Fejl: type i expression <bool> * <bool> der tilades kun type int
15: Fejl: numeric operator ikke mulig med typen <bool> men kun med typen <array og int>
TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

```

Antal alvorligFejl: 3

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
var
  vd#tal: t#int,
  vd#ok: t#bool;

v_int#tal = t_int#( c#1 + t_int#( c#2 * c#3 ) );
v_int#tal = t_int#( c#1 + t_int#( c#2 / c#3 ) );
v_int#tal = t_int#( c#1 + t_int#( t_int#( t_int#( c#2 * c#3 ) / c#4 ) * c#5 ) );
v_int#tal = t_int#( t_int#( t_int#( c#1 - c#2 ) + c#3 ) - c#4 );
v_int#tal = t_int#( t_int#( v_int#tal + v_bool#ok ) + t_int#( v_bool#ok * v_bool#ok ) );
v_int#tal = |v_int#tal| ;
v_int#tal = |v_bool#ok| ;

```

:
tonyprettyXml.xml

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <7> variabel af typen: bool ref: 4 <13, 13, 13, 15>

tal <6> variabel af typen: int ref: 9 <9, 10, 11, 12, 13, 13, 14, 14, 15>

Slut på test af test_typeexp1.tony

:

:

Start på test af test_typeexp2.tony

:

test_typeexp2.tony

```

:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Test af TONY operatorer, præcendes regler og typer
004: *)
005: var

```

```

006: ok:  bool, b1: bool, b2: bool, b3: bool, b4: bool, t: int;
007: b1 = true;           (* ok *)
008: b2 = false;         (* ok *)
009: b3 = b1;            (* ok *)
010: b4 = b2;            (* ok *)
011: ok = b1 || b2 && b3 || b4; (* ok *)
012:
013: ok = t || t && t || t;   (* fejl *)
-----

```

:

tony.exe consol output

:

Options:

-msg sat til test_typeexp2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

13: Fejl: type i expression <int> && <int> der tilades kun type bool

13: Fejl: type i expression <int> || <bool> der tilades kun type bool

13: Fejl: type i expression <bool> || <int> der tilades kun type bool

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 3

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

vd#ok: t#bool,

vd#b1: t#bool,

vd#b2: t#bool,

vd#b3: t#bool,

```
vd#b4: t#bool,
vd#t: t#int;
```

```
v_bool#b1 = c#true;
v_bool#b2 = c#false;
v_bool#b3 = v_bool#b1;
v_bool#b4 = v_bool#b2;
v_bool#ok = t_bool#( t_bool#( v_bool#b1 || t_bool#( v_bool#b2 && v_bool#b3 ) ) ||
v_bool#b4 );
v_bool#ok = t_bool#( t_bool#( v_int#t || t_bool#( v_int#t && v_int#t ) ) || v_int#t );
```

```
-----
:
tonyprettyXml.xml
```

```
:
```

```
-----
:
```

tonysymbol_xref.txt

```
:
```

```
-----
SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start
```

```
ok <6> variabel af typen: bool ref: 2 <11, 13>
```

```
t <6> variabel af typen: int ref: 4 <13, 13, 13, 13>
```

```
b1 <6> variabel af typen: bool ref: 3 <7, 9, 11>
```

```
b2 <6> variabel af typen: bool ref: 3 <8, 10, 11>
```

```
b3 <6> variabel af typen: bool ref: 2 <9, 11>
```

```
b4 <6> variabel af typen: bool ref: 2 <10, 11>
```

```
-----
Slut på test af test_typeexp2.tony
```

```
:
```

```
.....
```

```
:
```

```
-----
:
```

Start på test af test_typeexp3.tony

```
-----
```

test_typeexp3.tony

```
:
```

```
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
```

```
002:
```

```
003: Test af TONY operatorer, præcendes regler og typer
```

```
004: *)
```

```
005: var
```

```
006: tal: int,
```

```
007: ok: bool
```

```
008: ;
```

```
009: tal = 1;
```

010: ok = tal == 1+2 || tal != 3+4 && tal < 5+6; (* ok *)
 011: ok = ok == true || ok != false && false != ok; (* ok *)
 012: ok = tal == ok || tal != ok && tal < ok; (* fejl *)

 :

tony.exe consol output

:

Options:

-msg sat til test_typeexp3.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

12: Fejl: type i expression <int> == <bool> der tilades kun ens typer

12: Fejl: type i expression <int> != <bool> der tilades kun ens typer

12: Fejl: type i expression <int> < <bool> der tilades kun type int

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 3

Antal fejl: 0

Antal advarsler: 0

 :

tonyprettyTxt.txt

:

var

vd#tal: t#int,

vd#ok: t#bool;

v_int#tal = c#1;

v_bool#ok = t_bool#(t_bool#(v_int#tal == t_int#(c#1 + c#2)) || t_bool#(t_bool#(v_int#tal != t_int#(c#3 + c#4)) && t_bool#(v_int#tal < t_int#(c#5 + c#6))));

v_bool#ok = t_bool#(t_bool#(v_bool#ok == c#true) || t_bool#(t_bool#(v_bool#ok != c#false) && t_bool#(c#false != v_bool#ok)));


```
v_bool#ok = t_bool#( t_bool#( v_int#tal == v_bool#ok ) || t_bool#( t_bool#( v_int#tal !=
v_bool#ok ) && t_bool#( v_int#tal < v_bool#ok ) ) );
```

```
-----
```

```
:
tonyprettyXml.xml
```

```
-----
```

```
:
```

tonysymbol_xref.txt

```
:
-----
```

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <7> variabel af typen: bool ref: 9 <10, 11, 11, 11, 11, 12, 12, 12, 12>

tal <6> variabel af typen: int ref: 7 <9, 10, 10, 10, 12, 12, 12>

```
-----
```

Slut på test af test_typeexp3.tony

```
:
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
:
```

```
:
```

```
-----
```

```
:
```

Start på test af test_typeexp4.tony

```
-----
```

test_typeexp4.tony

```
:
```

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24

002:

003: Test af TONY operatorer, præcendes regler og typer

004: *)

005: var

006: tal: int,

007: ok: bool

008: ;

009: tal = 1;

010: ok = ! ok && ! ok || ! (tal < 20); (* ok *)

011: ok = ! tal && ! tal || ! (ok < 20); (* fejl *)

```
-----
```

```
:
```

tony.exe consol output

```
:
```

Options:

-msg sat til test_typeexp4.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

11: Fejl: negation ikke mulig med typen <int> men kun med typen <bool>

11: Fejl: negation ikke mulig med typen <int> men kun med typen <bool>

11: Fejl: type i expression <bool> < <int> der tilades kun type int

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 3

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

```
    vd#tal: t#int,
    vd#ok: t#bool;
```

```
v_int#tal = c#1;
```

```
v_bool#ok = t_bool#( t_bool#( !(v_bool#ok) && !(v_bool#ok) ) || !( t_bool#( v_int#tal
< c#20 )) ) );
```

```
v_bool#ok = t_bool#( t_bool#( !(v_int#tal) && !(v_int#tal) ) || !( t_bool#( v_bool#ok <
c#20 )) ) );
```

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

ok <7> variabel af typen: bool ref: 5 <10, 10, 10, 11, 11>

tal <6> variabel af typen: int ref: 4 <9, 10, 11, 11>

 Slut på test af test_typeexp4.tony

:
 :::
 :
 :
 :

 :

Start på test af test_typestruct1.tony

test_typestruct1.tony

```

:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24
002:
003: Test af TONY record strukturer
004: *)
005: type recdef = record of {niv2: record of {niv3: record of {t: int, b:bool}}};
006: var
007: tal: int,
008: ok: bool,
009: rec: recdef,
010: rec2: recdef,
011: rec3: record of {niv2: record of {niv3: record of {t: int, b:bool}}}
012: ;
013: tal = rec.niv2.niv3.t; (* ok *)
014: rec.niv2.niv3.t = tal; (* ok *)
015: ok = rec.niv2.niv3.b; (* ok *)
016: rec.niv2.niv3.b = ok; (* ok *)
017: rec2 = rec; (* ok *)
018: rec = rec2; (* ok *)
019: rec2.niv2 = rec.niv2; (* ok *)
020: rec.niv2 = rec2.niv2; (* ok *)
021: rec.niv2.niv3 = rec2.niv2.niv3; (* ok *)
022: rec.niv2.niv3.t = rec2.niv2.niv3.t; (* ok *)
023: tal = rec.niv2; (* fejl *)
024: tal = rec.niv2.t; (* fejl *)
025: ok = rec.niv2.niv3.t; (* fejl *)
026: rec3 = rec2; (* fejl *)

```

 :

tony.exe consol output

```

:
Options:

-msg sat til test_typestruct1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

```

TONY Parser afsluttet normal

TONY weeding fase start
 TONY weeding fase slut
 TONY opbygning af symboltabel start
 TONY opbygning af symboltabel slut
 TONY type-check start

23: Fejl: type i assignment <int> = <record> : der tilades kun ens typer
 24: Fejl: variabel <t> er ikke defineret
 24: Fejl: assign ikke mulig da expression type ikke def
 25: Fejl: type i assignment <bool> = <int> : der tilades kun ens typer
 26: Fejl: type i assignment <record> = <recdef> : der tilades kun ens typer
 TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt
 TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 5

Antal fejl: 0

Antal advarsler: 0

 :

tonyprettyTxt.txt

```

:
type td#recdef : t#record of {
  vd#niv2: t#record of {
    vd#niv3: t#record of {
      vd#t: t#int,
      vd#b: t#bool
    }
  }
};

var
  vd#tal: t#int,
  vd#ok: t#bool,
  vd#rec: t#recdef,
  vd#rec2: t#recdef,
  vd#rec3: t#record of {
    vd#niv2: t#record of {
      vd#niv3: t#record of {
        vd#t: t#int,
        vd#b: t#bool
      }
    }
  }
};

v_int#tal = v_recdef#rec.v_record#niv2.v_record#niv3.v_int#t;
v_recdef#rec.v_record#niv2.v_record#niv3.v_int#t = v_int#tal;
v_bool#ok = v_recdef#rec.v_record#niv2.v_record#niv3.v_bool#b;
v_recdef#rec.v_record#niv2.v_record#niv3.v_bool#b = v_bool#ok;
v_recdef#rec2 = v_recdef#rec;
v_recdef#rec = v_recdef#rec2;

```

```

v_recdef#rec2.v_record#niv2 = v_recdef#rec.v_record#niv2;
v_recdef#rec.v_record#niv2 = v_recdef#rec2.v_record#niv2;
v_recdef#rec.v_record#niv2.v_record#niv3 = v_recdef#rec2.v_record#niv2.v_record#niv3;
v_recdef#rec.v_record#niv2.v_record#niv3.v_int#t =
v_recdef#rec2.v_record#niv2.v_record#niv3.v_int#t;
v_int#tal = v_recdef#rec.v_record#niv2;
v_int#tal = v_recdef#rec.v_record#niv2.v_???#t;
v_bool#ok = v_recdef#rec.v_record#niv2.v_record#niv3.v_int#t;
v_record#rec3 = v_recdef#rec2;

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

```

:

```

```

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

```

```

ok <8> variabel af typen: bool ref: 3 <15, 16, 25>

```

```

recdef <5> type = record<5> ref: 2 <9, 10>

```

```

rec <9> variabel af typen: recdef<5> => record<5> ref: 13 <13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25>

```

```

tal<7> variabel af typen: int ref: 4 <13, 14, 23, 24>

```

```

rec2 <10> variabel af typen: recdef<5> => record<5> ref: 7 <17, 18, 19, 20, 21, 22,
26>

```

```

rec3 <11> variabel af typen: record<11> ref: 1 <26>

```

```

SymbolTable for recdef:<5> tabelNr:2 scope-level:0 - start

```

```

niv2 <5> variabel af typen: record<5> ref: 15 <13, 14, 15, 16, 19, 19, 20, 20, 21, 21,
22, 22, 23, 24, 25>

```

```

SymbolTable for niv2:<5> tabelNr:3 scope-level:0 - start

```

```

niv3 <5> variabel af typen: record<5> ref: 9 <13, 14, 15, 16, 21, 21, 22, 22, 25>

```

```

SymbolTable for niv3:<5> tabelNr:4 scope-level:0 - start

```

```

b <5> variabel af typen: bool ref: 2 <15, 16>

```

```

t <5> variabel af typen: int ref: 5 <13, 14, 22, 22, 25>

```

SymbolTable for rec3: <11> tabelnr:5 scope-level:0 - start

niv2 <11> variabel af typen: record<11>

SymbolTable for niv2: <11> tabelnr:6 scope-level:0 - start

niv3 <11> variabel af typen: record<11>

SymbolTable for niv3: <11> tabelnr:7 scope-level:0 - start

b <11> variabel af typen: bool

t <11> variabel af typen: int

Slut på test af test_typestruct1.tony

:
:.....
:
:

:

Start på test af test_typearray1.tony

test_typearray1.tony

:

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.26

002:

003: Test af TONY array og strukturer

004: *)

005: type rec = record of {x: int};

006: var

007: tal: int,

008: ar1: array of int,

009: ar2: array of array of int,

010: ar3: array of array of array of int,

011: arrec: array of rec,

012: arrec1: array of record of {x: int},

013: arrec2: array of record of {ar: array of int},

014: arrec3: array of record of {ar: array of record of {x: int, y: int}}

015: ;

016:

017: ar1[1] = 10; (* ok *)

018: ar2[1][2]=10; (* ok *)

019: ar3[1][2][3]=10; (* ok *)

020: arrec1[1].x = 10; (* ok *)

021: ar1[1] = 10; (* ok *)

022: ar2[1][1] = 10; (* ok *)

023: arrec[1].x = 10; (* ok *)

024: arrec1[1].x = 10; (* ok *)

025: arrec2[1].ar[1] = 10; (* ok *)

026: arrec3[1].ar[1].x = 10; (* ok *)

```

027: ar1[1][2] = 10;      (* fejl for mange niveauer *)
028:
029: tal = ar1[1];      (* ok *)
030: tal = ar2[1][2];   (* ok *)
031: tal = ar3[1][2][3]; (* ok *)
032: tal = arrec1[1].x; (* ok *)
033: tal = ar1[1];      (* ok *)
034: tal = ar2[1][1];   (* ok *)
035: tal = arrec[1].x;  (* ok *)
036: tal = arrec1[1].x; (* ok *)
037: tal = arrec2[1].ar[1]; (* ok *)
038: tal = arrec3[1].ar[1].x; (* ok *)
039: tal = ar1[1][2];   (* fejl for mange niveauer *)
-----

```

```

:
```

tony.exe consol output

```

:
```

Options:

-msg sat til test_typearray1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

27: Fejl: indexering ikke mulig paa typen <int>

27: Fejl: assign ikke mulig da variabelens type ikke er def

39: Fejl: indexering ikke mulig paa typen <int>

39: Fejl: assign ikke mulig da expression type ikke def

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 4

Antal fejl: 0

Antal advarsler: 0

```

-----
:
```

tonyprettyTxt.txt

```

:
type td#rec : t#record of {
  vd#x: t#int
};

var
  vd#tal: t#int,
  vd#ar1: t#array of t#int,
  vd#ar2: t#array of t#array of t#int,
  vd#ar3: t#array of t#array of t#array of t#int,
  vd#arrec: t#array of t#rec,
  vd#arrec1: t#array of t#record of {
    vd#x: t#int
  },
  vd#arrec2: t#array of t#record of {
    vd#ar: t#array of t#int
  },
  vd#arrec3: t#array of t#record of {
    vd#ar: t#array of t#record of {
      vd#x: t#int,
      vd#y: t#int
    }
  }
};

v_array#ar1[c#1] = c#10;
v_array#ar2[c#1][c#2] = c#10;
v_array#ar3[c#1][c#2][c#3] = c#10;
v_array#arrec1[c#1].v_int#x = c#10;
v_array#ar1[c#1] = c#10;
v_array#ar2[c#1][c#1] = c#10;
v_array#arrec[c#1].v_int#x = c#10;
v_array#arrec1[c#1].v_int#x = c#10;
v_array#arrec2[c#1].v_array#ar[c#1] = c#10;
v_array#arrec3[c#1].v_array#ar[c#1].v_int#x = c#10;
v_array#ar1[c#1][c#2] = c#10;
v_int#tal = v_array#ar1[c#1];
v_int#tal = v_array#ar2[c#1][c#2];
v_int#tal = v_array#ar3[c#1][c#2][c#3];
v_int#tal = v_array#arrec1[c#1].v_int#x;
v_int#tal = v_array#ar1[c#1];
v_int#tal = v_array#ar2[c#1][c#1];
v_int#tal = v_array#arrec[c#1].v_int#x;
v_int#tal = v_array#arrec1[c#1].v_int#x;
v_int#tal = v_array#arrec2[c#1].v_array#ar[c#1];
v_int#tal = v_array#arrec3[c#1].v_array#ar[c#1].v_int#x;
v_int#tal = v_array#ar1[c#1][c#2];

```

```

:
tonyprettyXml.xml
:

```

```

:
-----
tonysymbol_xref.txt
:

```

```

SymbolTable for MAIN: <0> tabelnr: 1 scope-level: 0 - start

```


ar1 <8> variabel af typen: array of int ref: 6 <17, 21, 27, 29, 33, 39>
 ar2 <9> variabel af typen: array of array of int ref: 4 <18, 22, 30, 34>
 ar3 <10> variabel af typen: array of array of array of int ref: 2 <19, 31>
 arrec <11> variabel af typen: array of rec<5> => record<5> ref: 2 <23, 35>
 rec <5> type = record<5> ref: 1 <11>
 tal<7> variabel af typen: int ref: 11 <29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39>
 arrec1 <12> variabel af typen: array of record<12> ref: 4 <20, 24, 32, 36>
 arrec2 <13> variabel af typen: array of record<13> ref: 2 <25, 37>
 arrec3 <14> variabel af typen: array of record<14> ref: 2 <26, 38>

SymbolTable for rec: <5> tabelnr:2 scope-level:0 - start

x <5> variabel af typen: int ref: 2 <23, 35>

SymbolTable for arrec1: <12> tabelnr:3 scope-level:0 - start

x <12> variabel af typen: int ref: 4 <20, 24, 32, 36>

SymbolTable for arrec2: <13> tabelnr:4 scope-level:0 - start

ar <13> variabel af typen: array of int ref: 2 <25, 37>

SymbolTable for arrec3: <14> tabelnr:5 scope-level:0 - start

ar <14> variabel af typen: array of record<14> ref: 2 <26, 38>

SymbolTable for ar: <14> tabelnr:6 scope-level:0 - start

x <14> variabel af typen: int ref: 2 <26, 38>

y <14> variabel af typen: int

Slut på test af test_typearray1.tony

:

.....

:

:

:

Start på test af test_typearray2.tony

test_typearray2.tony

```
:
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.26
002:
003: Test af TONY array og strukturer
004: *)
005: type rec = record of {x: int};
006: var
007: tal: int,
008: ar1: array of int,
009: ar2: array of array of int,
010: ar3: array of array of array of int,
011: arrec: array of rec,
012: arrec1: array of record of {x: int},
013: arrec2: array of record of {ar: array of int},
014: arrec3: array of record of {ar: array of record of {x: int, y: int}}
015: ;
016:
017: tal = |ar1|; (* ok *)
018: tal = |ar2|; (* ok *)
019: tal = |ar2[1]|; (* ok *)
020: tal = |ar3|; (* ok *)
021: tal = |ar3[1]|; (* ok *)
022: tal = |ar3[1][1]|; (* ok *)
023: tal = |arrec|; (* ok *)
024: tal = |arrec1|; (* ok *)
025: tal = |arrec2|; (* ok *)
026: tal = |arrec2[1].ar|; (* ok *)
027: tal = |arrec3[1].ar|; (* ok *)
028: tal = |arrec[1]|; (* fejl ikke array / int *)
```

:

tony.exe consol output

:

Options:

-msg sat til test_typearray2.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

28: Fejl: numeric operator ikke mulig med typen <record> men kun med typen <array og int>

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 1

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

```

:
type td#rec : t#record of {
  vd#x: t#int
};

var
  vd#tal: t#int,
  vd#ar1: t#array of t#int,
  vd#ar2: t#array of t#array of t#int,
  vd#ar3: t#array of t#array of t#array of t#int,
  vd#arrec: t#array of t#rec,
  vd#arrec1: t#array of t#record of {
    vd#x: t#int
  },
  vd#arrec2: t#array of t#record of {
    vd#ar: t#array of t#int
  },
  vd#arrec3: t#array of t#record of {
    vd#ar: t#array of t#record of {
      vd#x: t#int,
      vd#y: t#int
    }
  }
};

```

```

v_int#tal = |v_array#ar1| ;
v_int#tal = |v_array#ar2| ;
v_int#tal = |v_array#ar2[c#1]| ;
v_int#tal = |v_array#ar3| ;
v_int#tal = |v_array#ar3[c#1]| ;
v_int#tal = |v_array#ar3[c#1][c#1]| ;
v_int#tal = |v_array#arrec| ;
v_int#tal = |v_array#arrec1| ;
v_int#tal = |v_array#arrec2| ;
v_int#tal = |v_array#arrec2[c#1].v_array#ar| ;
v_int#tal = |v_array#arrec3[c#1].v_array#ar| ;
v_int#tal = |v_array#arrec[c#1]| ;

```

:
tonyprettyXml.xml
:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

ar1 <8> variabel af typen: array of int ref: 1 <17>

ar2 <9> variabel af typen: array of array of int ref: 2 <18, 19>

ar3 <10> variabel af typen: array of array of array of int ref: 3 <20, 21, 22>

arrec <11> variabel af typen: array of rec<5> => record<5> ref: 2 <23, 28>

rec <5> type = record<5> ref: 1 <11>

tal<7> variabel af typen: int ref: 12 <17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28>

arrec1 <12> variabel af typen: array of record<12> ref: 1 <24>

arrec2 <13> variabel af typen: array of record<13> ref: 2 <25, 26>

arrec3 <14> variabel af typen: array of record<14> ref: 1 <27>

SymbolTable for rec: <5> tabelNr:2 scope-level:0 - start

x <5> variabel af typen: int

SymbolTable for arrec1: <12> tabelNr:3 scope-level:0 - start

x <12> variabel af typen: int

SymbolTable for arrec2: <13> tabelNr:4 scope-level:0 - start

ar <13> variabel af typen: array of int ref: 1 <26>

SymbolTable for arrec3: <14> tabelNr:5 scope-level:0 - start

ar <14> variabel af typen: array of record<14> ref: 1 <27>

SymbolTable for ar: <14> tabelNr:6 scope-level:0 - start

x <14> variabel af typen: int

y <14> variabel af typen: int

Slut på test af test_typeofarray2.tony

```
:  
:.....  
:  
:  
-----  
:
```

Start på test af test_typescope1.tony

test_typescope1.tony

```
:  
001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.03.24  
002:  
003:   Aftester scoperegler  
004:   Herunder tilgængelighed ved declaration forud og efter  
005: *)  
006: type def0 = array of int;  
007: var v0: def0, tal: int;  
008:  
009: func f1 (p1_1: def0): int  
010:  
011: var fejl: def0a;           (* fejl def0a def. efter *)  
012: var  v1: int, tal: int, tal1: int;  
013: type def1 = array of int;  
014:  
015: func f2 (p2_1: def0, p2_2: def1): int  
016:   var  tal1: int;  
017:  
018:   tal1 = p2_1[1]; (* ok inderste scope *)  
019:   v1 = p2_1[1];   (* ok f2-scope *)  
020:   tal = 1;        (* ok f2-scope *)  
021:   v0 = p2_1;      (* ok yderste scope *)  
022:   v1a = p2_1[1]; (* fejl først def. efter *)  
023:   tal1 = f3(p2_1); (* ok samme level *)  
024:   tal1 = f1(p2_1); (* ok et level ud level *)  
025:   return 1;  
026: end f2;  
027:  
028: func f3 (p3_1: def0): int  
029:   var  talf3: int;  
030:   func f1 (): int  
031:     return 1;  
032:   end f1;  
033:   talf3 = tal1; (* fejl def. i f2's scope *)  
034:   return f1(); (* ok funktion fra indre scope vælges *)  
035: end f3;  
036:  
037: type def1a = array of int;  
038: var  v1a: int;  
039: v1 = p1_1[1]; (* ok *)  
040: v1a = p1_1[1]; (* ok *)  
041: v0 = p1_1; (* ok ydre scope *)  
042: tal = 1; (* ok indre scope *)  
043: v0a = 1; (* fejl først def. efter *)  
044: return 1;  
045:
```

```

046: end f1;
047:
048: type def0a = array of int;
049: var v0a: int;
050: v0a = f1(v0);    (* ok *)
051: v0a = f3(v0);    (* fejl indre scope i f1 skjult *)
-----

```

```

:
```

tony.exe consol output

```

:
```

Options:

-msg sat til test_typescope1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

11: Fejl: typen <def0a> er ikke tilgaengelig - erklaret senere<48>

22: Fejl: variabelen <v1a> er ikke tilgaengelig - erklaret senere<38>

22: Fejl: assign ikke mulig da variabelens type ikke er def

43: Fejl: variabelen <v0a> er ikke tilgaengelig - erklaret senere<49>

43: Fejl: assign ikke mulig da variabelens type ikke er def

51: Fejl: funktion f3 er ikke defineret/tilgaengelig

51: Fejl: assign ikke mulig da expression type ikke def

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 7

Antal fejl: 0

Antal advarsler: 0

```

-----
:
```

tonyprettyTxt.txt

```

:
```

```

type td#def0 :t#array of t#int;
```

```

var
```

```

vd#v0: t#def0,
vd#tal: t#int;

func fd#f1 (
    vd#p1_1: t#def0    ):t#int

var
    vd#fejl: t#def0a;

var
    vd#v1: t#int,
    vd#tal: t#int,
    vd#tal1: t#int;

type td#def1 :t#array of t#int;

func fd#f2 (
    vd#p2_1: t#def0,
    vd#p2_2: t#def1    ):t#int

var
    vd#tal1: t#int;

v_int#tal1 = v_def0#p2_1[c#1];
v_int#v1 = v_def0#p2_1[c#1];
v_int#tal = c#1;
v_def0#v0 = v_def0#p2_1;
v_???#v1a = v_def0#p2_1[c#1];
v_int#tal1 = f_int#f3 (v_def0#p2_1);
v_int#tal1 = f_int#f1 (v_def0#p2_1);
return c#1;

end fd#f2;

func fd#f3 (
    vd#p3_1: t#def0    ):t#int

var
    vd#talf3: t#int;

func fd#f1 () :t#int

    return c#1;

end fd#f1;

v_int#talf3 = v_int#tal1;
return f_int#f1 ();

end fd#f3;

type td#def1a :t#array of t#int;

var
    vd#v1a: t#int;

v_int#v1 = v_def0#p1_1[c#1];
v_int#v1a = v_def0#p1_1[c#1];
v_def0#v0 = v_def0#p1_1;

```

```

    v_int#tal = c#1;
    v_???#v0a = c#1;
    return c#1;

end fd#f1;

type td#def0a : t#array of t#int;

var
    vd#v0a: t#int;

v_int#v0a = f_int#f1 (v_def0#v0);
v_int#v0a = f_???#f3 (v_???#v0);

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

```

SymbolTable for MAIN: <0> tabelNr:1 scope-level:0 - start

```

```

v0a <49> variabel af typen: int ref: 3 <43, 50, 51>

```

```

tal <7> variabel af typen: int

```

```

def0a <48> type = array of int ref: 1 <11>

```

```

def0 <6> type = array of int ref: 4 <7, 9, 15, 28>

```

```

f1 <9> function af typen: int ref: 2 <24, 50>

```

```

v0 <7> variabel af typen: def0<6> => array of int ref: 3 <21, 41, 50>

```

```

SymbolTable for f1: <9> tabelNr:2 scope-level:1 - start

```

```

v1a <38> variabel af typen: int ref: 2 <22, 40>

```

```

p1_1 <9> variabel af typen: def0<6> => array of int ref: 3 <39, 40, 41>

```

```

tal <12> variabel af typen: int ref: 2 <20, 42>

```

```

def1a <37> type = array of int

```

```

def1 <13> type = array of int ref: 1 <15>

```

```

f2 <15> function af typen: int

```

```

f3 <28> function af typen: int ref: 1 <23>

```

```

fejl <11> variabel af typen: def0a => ikke defineret

```

```

v1 <12> variabel af typen: int ref: 2 <19, 39>

```


tal1 <12> variabel af typen: int ref: 1 <33>

SymbolTable for f2: <15> tabelnr:3 scope-level:2 - start

p2_1 <15> variabel af typen: def0<6> => array of int ref: 6 <18, 19, 21, 22, 23, 24>

p2_2 <15> variabel af typen: def1<13> => array of int

tal1 <16> variabel af typen: int ref: 3 <18, 23, 24>

SymbolTable for f3: <28> tabelnr:4 scope-level:2 - start

p3_1 <28> variabel af typen: def0<6> => array of int

tal3 <29> variabel af typen: int ref: 1 <33>

f1 <30> function af typen: int ref: 1 <34>

SymbolTable for f1: <30> tabelnr:5 scope-level:3 - start

Slut på test af test_typescope1.tony

:
 :::
 :
 :
 :

 :

Start på test af test_typeminus.tony

test_typeminus.tony

:
 001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.05.11
 002:
 003: Test af TONY minus (-) foran konstanter og variable
 004: *)
 005: var
 006: tal: int;
 007: tal = -1; (* ok *)
 008: tal = -1 + -2 * -3; (* ok *)
 009: tal = -1 + -2 / -3; (* ok *)
 010: tal = -1 + -2 * -3 / -4 * -5; (* ok *)
 011: tal = -1 - -2 + -3 - -4; (* ok *)
 012: tal = |-10|; (* ok *)
 013: tal = |-tal|; (* ok *)
 014: tal = -tal; (* ok *)
 015: tal = 5 + -tal; (* ok *)

 :

tony.exe consol output

:

Options:

-msg sat til test_typeminus.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 0

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

vd#tal: t#int;

v_int#tal = c#-1;

v_int#tal = t_int#(c#-1 + t_int#(c#-2 * c#-3));

v_int#tal = t_int#(c#-1 + t_int#(c#-2 / c#-3));

v_int#tal = t_int#(c#-1 + t_int#(t_int#(t_int#(c#-2 * c#-3) / c#-4) * c#-5));

v_int#tal = t_int#(t_int#(t_int#(c#-1 - c#-2) + c#-3) - c#-4);

v_int#tal = |c#-10| ;

v_int#tal = | (t_int#(c#0 - v_int#tal)) | ;

v_int#tal = (t_int#(c#0 - v_int#tal)) ;

v_int#tal = t_int#(c#5 + (t_int#(c#0 - v_int#tal)));

:

tonyprettyXml.xml

:

:

tonysymbol_xref.txt

:

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

tal <6> variabel af typen: int ref: 12 <7, 8, 9, 10, 11, 12, 13, 13, 14, 14, 15, 15>

 Slut på test af test_typeminus.tony

:

.....

:

:

:

Start på test af test_typefor1.tony

test_typefor1.tony

:

001: (* programmør: Bjørk Busch - bjbu@tietgen.dk - 2005.05.11

002:

003: Aftester om type-check ved for-to og for-downto

004: *)

005:

006: var i: int, b: bool;

007:

008: for i=1 to 10 do (* ok *)

009: i = i;

010: for i=1+1 to i+10 do (* ok *)

011: i = i;

012:

013: for b=false to 10 do (* fejl *)

014: b = b;

015: for i=1 to false do (* fejl *)

016: b = b;

017:

018: for i=1 downto 10 do (* ok *)

019: i = i;

020: for i=1+1 downto i+10 do (* ok *)

021: i = i;

022:

023: for b=false downto 10 do (* fejl *)

024: b = b;

025: for i=1 to downfalse do (* fejl *)

026: b = b;

:

tony.exe consol output

:

Options:

-msg sat til test_typefor1.consol.txt

-S er sat

-restriktiv er sat

-prettyTxt er sat

-xref er sat

TONY Parser afsluttet normal

TONY weeding fase start

TONY weeding fase slut

TONY opbygning af symboltabel start

TONY opbygning af symboltabel slut

TONY type-check start

14: Fejl: variabel i for-to er af typen <bool> men kun <int> er accepteret

14: Fejl: assign-expression i for-to er af typen <bool> men kun <int> er accepteret

16: Fejl: to expression i for-to er af typen <bool> men kun <int> er accepteret

24: Fejl: variabel i for-to er af typen <bool> men kun <int> er accepteret

24: Fejl: assign-expression i for-to er af typen <bool> men kun <int> er accepteret

25: Fejl: variabel <downfalse> er ikke defineret

26: Fejl: to expression i for-to er ikke en identificerbar type

TONY type-check slut

TONY SYMBOL XREF udskrives til tonysymbol_xref.txt

TONY abstract syntax tree udskrives i TXT-format til <tonyprettyTxt.txt>

Antal alvorligFejl: 7

Antal fejl: 0

Antal advarsler: 0

:

tonyprettyTxt.txt

:

var

```
    vd#i: t#int,
    vd#b: t#bool;
```

```
for v_int#i = c#1 to c#10 {
```

```
    v_int#i = v_int#i;
```

```
}
```

```
for v_int#i = t_int#( c#1 + c#1 ) to t_int#( v_int#i + c#10 ) {
```

```
    v_int#i = v_int#i;
```

```
}
```

```
for v_bool#b = c#false to c#10 {
```

```
    v_bool#b = v_bool#b;
```

```
}
```

```
for v_int#i = c#1 to c#false {
```

```
    v_bool#b = v_bool#b;
```

```
}
```

```
for v_int#i = c#1 downto c#10 {
```

```
    v_int#i = v_int#i;
```

```
}
```

```

for v_int#i = t_int#( c#1 + c#1 ) downto t_int#( v_int#i + c#10 ) {
  v_int#i = v_int#i;
}
for v_bool#b = c#false downto c#10 {
  v_bool#b = v_bool#b;
}
for v_int#i = c#1 to v_???#downfalse {
  v_bool#b = v_bool#b;
}

```

```

:
tonyprettyXml.xml
:

```

tonysymbol_xref.txt

```

:
-----

```

SymbolTable for MAIN: <0> tabelnr:1 scope-level:0 - start

b <6> variabel af typen: bool ref: 10 <13, 14, 14, 16, 16, 23, 24, 24, 26, 26>

i <6> variabel af typen: int ref: 16 <8, 9, 9, 10, 10, 11, 11, 15, 18, 19, 19, 20, 20, 21, 21, 25>

```

-----
Slut på test af test_typefor1.tony
:
:
:
:

```