Statische und dynamische Semantik

- **Statische Semantik**
  Typumgebungen: $T \in TE = Id \xrightarrow{fin} Ty$
  Statische Semantik: $SS \subseteq TE \times Exp \times Ty$
  Notation: $T \vdash e : t$

- **Dynamische Semantik**
  Werteumgebungen: $V \in VE = Id \xrightarrow{fin} Val$
  Dynamische Semantik: $DS \subseteq VE \times Exp \times Val$
  Notation: $V \vdash e \triangleright \nu$

Elaborierung

```
type 'a env = id -> 'a
exception Unbound
fun empty x = raise Unbound of id
fun update env x a y = if y=x then a else env y
exception Error of string

fun elabCon True = Bool
  | elabCon False = Bool
  | elabCon (IC _) = Int

fun elab f (Con c) = elabCon c
  | elab f (Id x) = f x
  | ...
```

- **Strue** $\dfrac{}{T \vdash \mathit{true} : \mathit{bool}}$
- **Snum** $\dfrac{z \in \mathbb{Z}}{T \vdash z : \mathit{int}}$
- **Sid** $\dfrac{Tx = t}{T \vdash x : t}$

Elaborierung

```
fun elabOpr Add Int Int = Int
  | elabOpr Sub Int Int = Int
  | elabOpr Mul Int Int = Int
  | elabOpr Leq Int Int = Bool
  | elabOpr _ _ _ = raise Error "T Opr"

fun elab f (Con c) = elabCon c
  | elab f (Id x) = f x
  | elab f (Opr(opr,e1,e2))
        = elabOpr opr (elab f e1) (elab f e2)
  | ...
```

- **Soai** $\dfrac{o \in \{+, -, *\} \quad T \vdash e_1 : int \quad T \vdash e_2 : int}{T \vdash e_1 \ o \ e_2 : int}$
- **Soab** $\dfrac{T \vdash e_1 : int \quad T \vdash e_2 : int}{T \vdash e_1 \leq e_2 : bool}$

Elaborierung

```
| elab f (If(e1,e2,e3)) =
   (case (elab f e1, elab f e2, elab f e3) of
     (Bool, t2, t3) => if t2=t3 then t2
       else raise Error "T If1"
     | _ => raise Error "T If2")
| elab f (Abs(x,t,e)) = Arrow(t, elab(update f x t) e)
| elab f (App(e1,e2)) =
   (case elab f e1 of
     Arrow(t,t') => if t=elab f e2 then t'
       else raise Error "T App1"
| _ => raise Error "T App2")
```

- **Sif** $\dfrac{T \vdash e_1 : bool \quad T \vdash e_2 : t \quad T \vdash e_3 : t}{T \vdash if\ e_1\ then\ e_2\ else\ e_3\ :\ t}$
- **Sabs** $\dfrac{T[x := t] \vdash e_1 : t'}{T \vdash fn\ x : t \Rightarrow e : t \rightarrow t'}$
- **Sapp** $\dfrac{T \vdash e_1 : t \rightarrow t' \quad T \vdash e_2 : t}{T \vdash e_1 e_2 :\ t'}$

Statische Semantik

- **Sfalse** $\dfrac{}{T \vdash \mathit{false} : \mathit{bool}}$
- **Snum** $\dfrac{z \in \mathbb{Z}}{T \vdash z : \mathit{int}}$
- **Strue** $\dfrac{}{T \vdash \mathit{true} : \mathit{bool}}$
- **Sid** $\dfrac{Tx = t}{T \vdash x : t}$
- **Soai** $\dfrac{o \in \{+, -, *\} \quad T \vdash e_1 : \mathit{int} \quad T \vdash e_2 : \mathit{int}}{T \vdash e_1 \ o \ e_2 : \mathit{int}}$
- **Soab** $\dfrac{T \vdash e_1 : \mathit{int} \quad T \vdash e_2 : \mathit{int}}{T \vdash e_1 \leq e_2 : \mathit{bool}}$
- **Sif** $\dfrac{T \vdash e_1 : \mathit{bool} \quad T \vdash e_2 : t \quad T \vdash e_3 : t}{T \vdash \mathit{if} \ e_1 \ \mathit{then} \ e_2 \ \mathit{else} \ e_3 \ : \ t}$
- **Sabs** $\dfrac{T[x := t] \vdash e_1 : t'}{T \vdash \mathit{fn} \ x : t \Rightarrow e : t \rightarrow t'}$
- **Sapp** $\dfrac{T \vdash e_1 : t \rightarrow t' \quad T \vdash e_2 : t}{T \vdash e_1 e_2 : \ t'}$

$$f[x := y] := f + \{(x, y)\}$$

Dynamische Semantik

- **Dfalse** $\dfrac{}{V \vdash \mathit{false} \triangleright 0}$
- **Dnum** $\dfrac{z \in \mathbb{Z}}{V \vdash z \triangleright z}$
- **Dtrue** $\dfrac{}{V \vdash \mathit{true} \triangleright 1}$
- **Did** $\dfrac{Vx = \nu}{V \vdash x \triangleright \nu}$
- **D+** $\dfrac{V \vdash e_1 \triangleright \nu_1 \quad V \vdash e_2 \triangleright \nu_2 \quad \nu = \nu_1 + \nu_2}{V \vdash e_1 + e_2 \triangleright \nu}$
- **D-**, **D***, **D**$\leq$ analog
- **Diftrue** $\dfrac{V \vdash e_1 \triangleright 1 \quad V \vdash e_2 \triangleright \nu}{V \vdash \mathit{if}\ e_1\ \mathit{then}\ e_2\ \mathit{else}\ e_3\ \triangleright \nu}$
- **Diffalse** analog
- **Dabs** $\dfrac{}{V \vdash \mathit{fn}\ x : t \Rightarrow e \triangleright \langle x, e, V \rangle}$
- **Dapp** $\dfrac{V \vdash e_1 \triangleright \langle x, e, V' \rangle \quad V \vdash e_2 : \nu_2 \quad V'[x := \nu_2] \vdash e \triangleright \nu}{V \vdash e_1\, e_2 \triangleright \nu}$

Eigenschaften

- **Determinismus:**
  Statische Semantik: Sei $T \vdash e : t$ und $T \vdash e : t'$. Dann $t = t'$.
  Dynamische Semantik: Sei $V \vdash e \triangleright \nu$ und $V \vdash e \triangleright \nu'$. Dann $\nu = \nu'$.

- **Auswertbarkeit:** Sei $\emptyset \vdash e : t$.
  Dann existiert genau ein Wert ν mit $\emptyset \vdash e \triangleright \nu$.

- **Typkorrektheit:**
  1. Sei $\emptyset \vdash e : \mathit{int}$ und $\emptyset \vdash e \triangleright \nu$. Dann $\nu \in \mathbb{Z}$.
  2. Sei $\emptyset \vdash e : \mathit{bool}$ und $\emptyset \vdash e \triangleright \nu$. Dann $\nu \in \{0, 1\}$.

Evaluierung

```
datatype value =
    IV of int
  | Proc of id * exp * value env

eval: value env -> exp -> value

fun evalCon True = IV 1
  | evalCon False = IV 0
  | evalCon (IC x) = IV x

fun evalOpr Add (IV x1) (IV x2) = IV(x1+x2)
  | evalOpr Sub (IV x1) (IV x2) = IV(x1-x2)
  | evalOpr Mul (IV x1) (IV x2) = IV(x1*x2)
  | evalOpr Leq (IV x1) (IV x2) = IV(if x1<=x2 then 1 else 0)
  | evalOpr _ _ _ = raise Error "R Opr"

fun eval f (Con c) = evalCon c
  | eval f (Id x) = f x
  | eval f (Opr(opr,e1,e2)) = evalOpr opr (eval f e1) (eval f e2)
```

Evaluierung

```
| eval f (If(e1,e2,e3)) =
      (case eval f e1 of
          IV 1 => eval f e2
        | IV 0 => eval f e3
        | _ => raise Error "R If")
| eval f (Abs(x,t,e)) = Proc(x,e,f)
| eval f (App(e1,e2)) =
      (case (eval f e1, eval f e2) of
          (Proc(x,e,f'),v) => eval (update f' x v) e
        | _ => raise Error "R App")
```