# Reduktsiooni järjekorrad

## Reduktsioonistrateegiad

Aplikatiivne järjekord — väärtustatakse seest välja

$$\begin{aligned} \mathit{double}\ \underline{(2+3)} &\Longrightarrow \underline{\mathit{double}\ 5} \\ &\Longrightarrow \underline{2 \times 5} \\ &\Longrightarrow 10 \end{aligned}$$

Normaaljärjekord — väärtustatakse väljast sisse

$$\begin{aligned} \underline{\mathit{double}\ (2+3)} &\Longrightarrow 2 \times \underline{(2+3)} \\ &\Longrightarrow \underline{2 \times 5} \\ &\Longrightarrow 10 \end{aligned}$$

## Church-Rosser'i teoreem

Avaldise normaalkuju ei sõltu reduktsioonijadast.

# Reduktsiooni järjekorrad

## NB!

Normaalkuju leidumine sõltub reduktsioonijärjekorrast!

$$
\begin{aligned}
&\mathit{const}\ x\ y = x \\
&\mathit{loop} \qquad\quad = \mathit{loop} \\
&\mathit{const}\ 5\ \underline{\mathit{loop}} \Longrightarrow \mathit{const}\ 5\ \underline{\mathit{loop}} \\
&\qquad\qquad\quad \Longrightarrow \mathit{const}\ 5\ \underline{\mathit{loop}} \\
&\qquad\qquad\quad \Longrightarrow \ldots \\
&\underline{\mathit{const}\ 5\ \mathit{loop}} \Longrightarrow 5
\end{aligned}
$$

## Normaliseerimisteoreem

Kui avaldisel leidub normaalkuju, siis normaaljärjekorras reduktsioonijada on lõplik.

# Reduktsiooni järjekorrad

**NB!**

Normaaljärjekord võib olla ebaefektiivne!

$$
\begin{aligned}
\underline{square\ (2+3)} &\Longrightarrow \underline{square\ 5} \\
&\Longrightarrow \underline{5 \times 5} \\
&\Longrightarrow 25
\end{aligned}
$$

$$
\begin{aligned}
\underline{square\ (2+3)} &\Longrightarrow (\underline{2+3}) \times (2+3) \\
&\Longrightarrow 5 \times (\underline{2+3}) \\
&\Longrightarrow \underline{5 \times 5} \\
&\Longrightarrow 25
\end{aligned}
$$

# Reduktsiooni järjekorrad

## Laisk väärtustamine

Laisk väärtustamine = normaaljärjekord + graafireduktsioon

## Näide

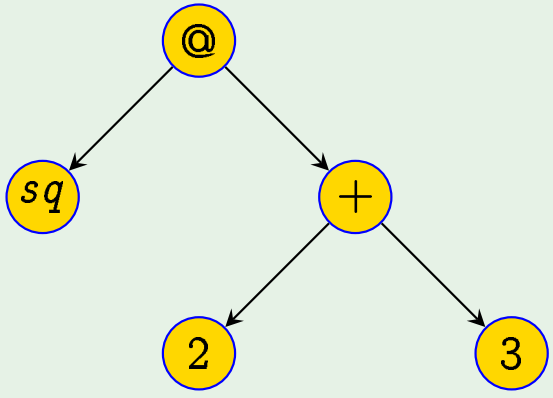
# Reduktsiooni järjekorrad

## Laisk väärtustamine

Laisk väärtustamine = normaaljärjekord + graafireduktsioon

## Näide

$\underline{square\ (2+3)}$



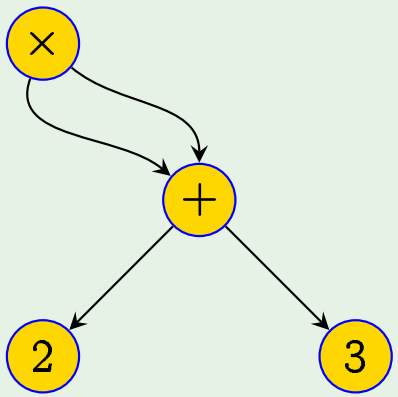
# Reduktsiooni järjekorrad

## Laisk väärtustamine

Laisk väärtustamine = normaaljärjekord + graafireduktsioon

## Näide

$$\underline{\mathit{square}\ (2+3)} \quad \Longrightarrow \quad x \times x \quad \textbf{where}\ x = \underline{2+3}$$



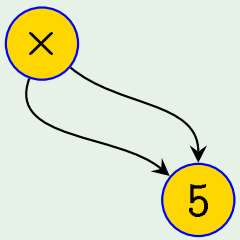
# Reduktsiooni järjekorrad

## Laisk väärtustamine

Laisk väärtustamine = normaaljärjekord + graafireduktsioon

## Näide

$$\underline{square\ (2+3)} \quad \Longrightarrow \quad x \times x \quad \textbf{where } x = \underline{2+3}$$
$$\Longrightarrow \quad \underline{x \times x} \quad \textbf{where } x = 5$$



# Reduktsiooni järjekorrad

## Laisk väärtustamine

Laisk väärtustamine = normaaljärjekord + graafireduktsioon

## Näide

$$\begin{aligned} \underline{\mathit{square}\ (2+3)} &\Longrightarrow x \times x \quad \textbf{where } x = \underline{2+3} \\ &\Longrightarrow \underline{x \times x} \quad \textbf{where } x = 5 \\ &\Longrightarrow 25 \end{aligned}$$

# Reduktsiooni järjekorrad

## Samaselt tõesuse kontroll *foldr* abil

$$
\begin{aligned}
& \mathit{foldr}\ (\&\&)\ \mathit{True}\ [\mathit{False}, \mathit{True}, \mathit{False}] \\
\Longrightarrow\ & \mathit{False}\ \&\&\ (\mathit{foldr}\ (\&\&)\ \mathit{True}\ [\mathit{True}, \mathit{False}]) \\
\Longrightarrow\ & \mathit{False}
\end{aligned}
$$

## Samaselt tõesuse kontroll *foldl* abil

$$
\begin{aligned}
& \mathit{foldl}\ (\&\&)\ \mathit{True}\ [\mathit{False}, \mathit{True}, \mathit{False}] \\
\Longrightarrow\ & \mathit{foldl}\ (\&\&)\ (\mathit{True}\ \&\&\ \mathit{False})\ [\mathit{True}, \mathit{False}] \\
\Longrightarrow\ & \mathit{foldl}\ (\&\&)\ ((\mathit{True}\ \&\&\ \mathit{False})\ \&\&\ \mathit{True})\ [\mathit{False}] \\
\Longrightarrow\ & \mathit{foldl}\ (\&\&)\ (((\mathit{True}\ \&\&\ \mathit{False})\ \&\&\ \mathit{True})\ \&\&\ \mathit{False})\ [\,] \\
\Longrightarrow\ & ((\mathit{True}\ \&\&\ \mathit{False})\ \&\&\ \mathit{True})\ \&\&\ \mathit{False} \\
\Longrightarrow\ & \mathit{False}
\end{aligned}
$$

# Reduktsiooni järjekorrad

## Listi elementide summa *foldr* abil

$$
\begin{aligned}
\mathit{foldr}\ (+)\ 0\ [1,2,3] &\Longrightarrow 1 + \mathit{foldr}\ (+)\ 0\ [2,3] \\
&\Longrightarrow 1 + (2 + \mathit{foldr}\ (+)\ 0\ [3]) \\
&\Longrightarrow 1 + (2 + (3 + \mathit{foldr}\ (+)\ 0\ [\,])) \\
&\Longrightarrow 1 + (2 + (3 + 0)) \\
&\Longrightarrow 6
\end{aligned}
$$

## Listi elementide summa *foldl* abil

$$
\begin{aligned}
\mathit{foldl}\ (+)\ 0\ [1,2,3] &\Longrightarrow \mathit{foldl}\ (+)\ (0 + 1)\ [2,3] \\
&\Longrightarrow \mathit{foldl}\ (+)\ ((0 + 1) + 2)\ [3] \\
&\Longrightarrow \mathit{foldl}\ (+)\ (((0 + 1) + 2) + 3)\ [\,] \\
&\Longrightarrow ((0 + 1) + 2) + 3 \\
&\Longrightarrow 6
\end{aligned}
$$

# Reduktsiooni järjekorrad

## Agarad funktsioonid

- Funktsioon on agar, kui funktsiooni tulemuse leidmiseks tuleb argument alati väärtustada
- Pole vahet millist reduktsiooni järjekorda kasutada!
- Efektiivsem on aplikatiivne (või mingi sega-)järjekord!?!

## Listi elementide summa *foldl* abil (segajärjekorras)

$$
\begin{aligned}
\mathit{foldl}\ (+)\ 0\ [1,2,3] &\Longrightarrow \mathit{foldl}\ (+)\ (0+1)\ [2,3] \\
&\Longrightarrow \mathit{foldl}\ (+)\ 1\ [2,3] \\
&\Longrightarrow \mathit{foldl}\ (+)\ (1+2)\ [3] \\
&\Longrightarrow \mathit{foldl}\ (+)\ 3\ [3] \\
&\Longrightarrow \mathit{foldl}\ (+)\ (3+3)\ [\,] \\
&\Longrightarrow \mathit{foldl}\ (+)\ 6\ [\,] \\
&\Longrightarrow 6
\end{aligned}
$$

# Reduktsiooni järjekorrad

## Agar aplikatsioon

- Operaator $! tähistab agarat funktsiooni aplikatsiooni
- Avaldise $f$ \$! $x$ korral väärtustatakse argument $x$ enne funktsiooni rakendamist
- Võimaldab juhtida redutseerimise järjekorda

## Agar *foldl*

$$
\begin{aligned}
&foldl' :: (a \to b \to a) \to a \to [b] \to a \\
&foldl'\ f\ a\ [\,] \qquad\quad = a \\
&foldl'\ f\ a\ (x : xs) = (foldl'\ f\ \$!\ f\ a\ x)\ xs
\end{aligned}
$$

## NB!

$foldl'$ on defineeritud Hugs'i prelüüdis, kuid pole eksporditud

# Laisk väärtustamine

## Lõpmatud listid

- Laisk väärtustamine võimaldab kasutada lõpmatuid andmestruktuure
- Kui lõpmatust andmestrutuurist kasutatakse ainult lõplikku osa, siis on ka väärtustamine lõplik

  $$take\ 5\ [1\,..] \Longrightarrow [1,2,3,4,5]$$

- Lõpmatud struktuurid võivad sisaldada omakorda lõpmatuid strutuure!

  $$multiples :: [[Int]]$$
  $$multiples = [[m * n \mid m \leftarrow [1\,..]] \mid n \leftarrow [1\,..]]$$

# Laisk väärtustamine

## Funktsioon *iterate* (eeldefineeritud)

$$iterate\ f\ x = [x, f\ x, f^2\ x, f^3\ x, \ldots]$$

## Definitsioon

$$iterate :: (a \to a) \to a \to [a]$$
$$iterate\ f\ x = x : iterate\ f\ (f\ x)$$

## Näited

$$powertables :: [[Int]]$$
$$powertables = [iterate\ (*n)\ 1 \mid n \leftarrow [2..]]$$

$$digits :: Int \to [Int]$$
$$digits = reverse\ .\ map\ (`mod`10)\ .\ takeWhile\ (/= 0)$$
$$.\ iterate\ (`div`10)$$

# Laisk väärtustamine

## Ruutjuur Newton–Raphson'i meetodil

$$a_{i+1} = \frac{a_i + \frac{n}{a_i}}{2}$$

```
next n x = (x + n / x) / 2.0
within eps (x1 : x2 : xs)
    | abs (x1 - x2) <= eps = x2
    | otherwise            = within eps (x2 : xs)
sqrt1 n = within eps (iterate (next n) a0)
  where eps = 0.00001
        a0  = 1.0
```

# Laisk väärtustamine

## Erasthostenese sõel

1. Kirjuta üles kõik positiivsed täisarvud alates arvust 2;
2. Tähista jada esimene element $p$ kui algarv;
3. Kustuta jadast kõik arvu $p$ kordsed;
4. Mine tagasi sammule 2.

$$
\begin{aligned}
primes &= map\ head\ (iterate\ sieve\ [2..]) \\
sieve\ (p:xs) &= [x \mid x \leftarrow xs, x\ `mod`\ p\ /= 0]
\end{aligned}
$$

# Laisk väärtustamine

## Kaheksa lippu

$$
\begin{aligned}
&\mathit{queens}\ 0 \;\; = [[\,]] \\
&\mathit{queens}\ m = [p +\!\!+ [n] \mid p \leftarrow \mathit{queens}\ (m-1) \\
&\qquad\qquad\qquad\qquad\quad ,\ n \leftarrow [1\,..\,8] \\
&\qquad\qquad\qquad\qquad\quad ,\ \mathit{safe}\ p\ n] \\
&\mathit{safe}\ p\ n \;\; = \mathit{and}\ [\mathit{not}\ (\mathit{check}\ (i,j)\ (m,n)) \\
&\qquad\qquad\qquad\qquad\qquad\quad \mid (i,j) \leftarrow \mathit{zip}\ [1\,..]\ p] \\
&\qquad\qquad \mathbf{where}\ m = 1 + \mathit{length}\ p \\
&\mathit{check}\ (i,j)\ (m,n) = j == n \ \mid\mid\ (i+j == m+n) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \mid\mid\ (i-j == m-n)
\end{aligned}
$$

# Laisk väärtustamine

## Liita listi kõigile elementidele listi minimaalne element

$$
\begin{aligned}
&incmin\ xs = map\ (minv+)\ xs \\
&\qquad\qquad \textbf{where}\ minv = minimum\ xs
\end{aligned}
$$

## Listi ühekordse läbimisega versioon

$$
\begin{aligned}
&incmin\ [\,] = [\,] \\
&incmin\ xs = newlist \\
&\quad \textbf{where}\ (minv, newlist) = onepass\ xs \\
&\qquad\qquad onepass\ [a] \quad\ = (a, [a + minv]) \\
&\qquad\qquad onepass\ (a : as) = (a\ `min`\ b, (a + minv) : bs) \\
&\qquad\qquad\qquad\qquad\qquad \textbf{where}\ (b, bs) = onepass\ as
\end{aligned}
$$