Aufgabe 1

```
--

concat l = foldr append Nil l;

Kons x xs = Cons{4,2} x xs;

append xs ys = foldr Kons ys xs;

foldr f z l =
   case l of
      <3>      -> z;
      <4> x xs -> f x (foldr f z xs);



length l =
   case l of
      <3>      -> 0;
      <4> x xs -> 1 + (length xs);

sum l = foldl plus Top l;

plus x y = x + y;

foldl f z l =
   case l of
      <3>      -> z;
      <4> x xs -> foldl f (f z x) xs;


--
-- Analyse
--
Nil = Cons{3,0};

nils = Kons Nil nils;

inf =
```

```
   case Top of
      <1> -> Kons Top inf;
      <2> -> Bot;

botelem =
   case Top of
      <5> -> Kons Bot Top;
      <6> -> Kons Top botelem;
      <7> -> Bot;


main1 x = concat nils;
main2 x = length inf;
main3 x = sum botelem;
main4a x = length (append inf Top);
main4b x = length (append Top inf);

lit x ys = case ys of
       <3> -> Nil;
       <4> y ys ->(lit2 x y ys);
lit2 x y ys = case (gl x y) of
       <2> -> (Kons ys Nil);
       <1> -> Nil;

gl x y = Top x y;

alt p q x = append (p x) (q x);

seq p q xs = case (p xs) of
       <3> -> Nil;
       <4> x xs -> seq2 q x xs;
seq2 q x xs = append (q x) (seq3 q xs);
seq3 q xs = case xs of
       <3> -> Nil;
       <4> x xs -> seq2 q x xs;

prek xs = alt (seq prek (lit Top)) (lit Top) xs;

loopsprek x = prek Top;
```

```
-------Ergebnisse

--concat [strict]
--Kons [?, ?]
--append [strict, ?]
--foldr [?, ?, strict]
--length [strict]
--sum [strict]
--plus [strict, strict]
--foldl [?, ?, strict]
--main1 [strict]
--main2 [strict]
--main3 [?]
--main4a [strict]
--main4b [strict]
--lit [?, strict]
--lit2 [?, ?, ?]
--gl [?, ?]
--alt [strict, ?, ?]
--seq [strict, ?, ?]
--seq2 [strict, ?, strict]

--seq3 [?, strict]
--prek [strict]
--loopsprek [strict]

--
```

Aufgabe 2

Wir wollen ein Beispiel für die Analyse einer detailierten Striktheitsbeschreibung einer Funktion auf binären markierten Bäumen geben. Hierzu deklarieren wir in Haskell-Syntax den entsprechenden Typ:

```
data BinBaum
  =  BNil
    |BCons BinBaum Int BinBaum
```

Der entsprechende algebraische Typ sei in der Kernsprache gegeben und damit die **case**-Konstante vom Typ:
$\mathit{BinBaum} \to \tau \to (\mathit{BinBaum} \to \mathit{Int} \to \mathit{BinBaum} \to \tau) \to \tau$

Für den algebraischen Datentypen müssen wir jetzt einen abstrakten *Domain* kreieren. Wir gehen dabei erst einmal von der gegebenen, natürlich-sprachlichen Beschreibung aus:

$\perp$: ein Ausdruck hat keine Kopfnormalform.

$\bowtie$: ein Ausdruck hat höchstens eine Kopfnormalform.

$R\perp$: es kann höchstens der linkeste Pfad eines Baumes in der Länge bis zum Ende ausgewertet werden.

$L\perp$: es kann höchstens der rechteste Pfad eines Baumes in der Länge bis zum Ende ausgewertet werden.

$R\perp\!\!\in$: es kann höchstens der linkeste Pfad eines Baumes komplett mit allen auf diesem Pfad enthaltenen Elementen ausgewertet werden.

$L\perp\!\!\in$: es kann höchstens der rechteste Pfad eines Baumes komplett mit allen auf diesem Pfad enthaltenen Elementen ausgewertet werden.

$\perp\!\!\in$: die komplette Baum*struktur* läßt sich unter Umständen auswerten aber auch nicht mehr, d.h. mindestens einmal ist `Bot` die Markierung eines Knotens oder der Baum enthält mindestens einen unendlichen Pfad. Wir können dies auch folgendermaßen charakterisieren: Mehr als ein Pfad eines Baumes kann komplett mit allen auf diesem Pfad enthaltenen Elementen ausgewertet werden, aber auch für mindestens einen Pfad trifft dies nicht zu. Somit leuchtet ein, daß $\perp\!\!\in$ sowohl $L\perp\!\!\in$ als auch $R\perp\!\!\in$ umfaßt.

$\top$: der Ausdruck kann eine Normalform haben.

Sei $D_{BinBaum}$ nun die Menge $\{\perp, \bowtie, L\perp, R\perp, L\perp\!\!\in, R\perp\!\!\in, \perp\!\!\in, \top\}$ dann können wir nach der obigen Beschreibung eine Ordnung angeben, die in Abbildung 1 zu sehen ist.

Schließlich gilt es, exakte abstrakte Interpretationen der eingebauten Konstanten und Funktionen zu geben. Wir werden uns dies anhand der obigen Beschreibung intuitiv überlegen, um uns vorerst nicht mit dem Formalismus belasten zu müssen. Ganz wichtig ist, zu sehen, daß `BNil` nicht nur eine Kopfnormalform, sondern sogar eine *Normalform* ist. Daher kann für dessen abstrakte Interpretation einzig gelten:
$\texttt{BNil}_{abs} = \top$ oder $[\![\texttt{BNil}]\!]^a = \top$ oder $\alpha(\texttt{BNil}) = \top$, um alle Schreibweisen einmal bedient zu haben.

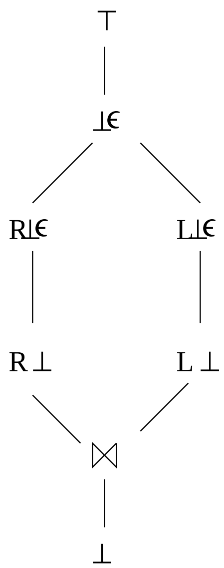

Abbildung 1: Der abstrakte *Domain* für *BinBaum*.

Für $\texttt{BCons}_{abs}\ x \perp y$ versuchen wir es auf eine ähnliche Art und Weise: Indem wir uns z.B. überlegen, was dabei herauskommt, wenn die Wurzel eines Baumes mit `Bot` markiert ist, der linke Teilbaum einer ist, dessen linkester Pfad in der Länge bis zum Ende ausgewertet werden kann, und der rechte Teilbaum nur eine Kopfnormalform hat. Man sieht, daß man wieder einen Baum erhält, dessen linkester Pfad in der Länge bis zum Ende ausgewertet werden kann, sprich $\texttt{BCons}_{abs}\ R\bot \perp \bowtie\ \in R\bot$. Mit Hilfe dieser Überlegungen erhalten wir eine Wertetabelle zunächst für $\texttt{BCons}_{abs}\ x \perp y$:

$\bot$	$\bot$	$\bowtie$	$L\bot$	$R\bot$	$L\bot\!\in$	$R\bot\!\in$	$\bot\!\in$	$\top$
$\bot$	$\bowtie$	$L\bot$	$L\bot$	$\bot\!\in$	$L\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$\bowtie$	$R\bot$	$\bot\!\in$	$L\bot$	$\bot\!\in$	$L\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$L\bot$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$L\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$R\bot$	$R\bot$	$R\bot$	$\bot\!\in$	$\bot\!\in$	$L\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$L\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$R\bot\!\in$	$R\bot\!\in$	$R\bot\!\in$	$R\bot\!\in$	$R\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$
$\top$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$	$\bot\!\in$

$\texttt{BCons}_{abs}\ x \perp y$

Ein paar grundsätzliche Gedanken dazu sind hilfreich:

- $\texttt{BCons}_{abs}\ x\ \bot\ y$ bzw. im symmetrischen Fall $\texttt{BCons}_{abs}\ y\ \bot\ x$ für $x \in \{\bot\in, \top\}$ ist immer $\bot\in$ (für alle y). Man könnte auch sagen: Da wir die Markierung der Wurzel als $\bot$ annehmen, kann der Baum ja gar nie eine Normalform haben.

- $\texttt{BCons}_{abs}\ R\bot\in\ \bot\ x$ ist $R\bot\in$ für $x \in \{\bot, \bowtie, R\bot, L\bot\}$ und $\bot\in$ sonst. Symmetrisch dazu: $\texttt{BCons}_{abs}\ x\ \bot\ L\bot\in$ ist $L\bot\in$ für $x \in \{\bot, \bowtie, L\bot, R\bot\}$ und $\bot\in$ sonst.

- Ein analoges Ergebnis erhält man für $\texttt{BCons}_{abs}\ R\bot\ \bot\ x$ mit $x \in \{\bot, \bowtie\}$ bzw. $\texttt{BCons}_{abs}\ x\ \bot\ L\bot$.

- Dagegen ist klar, daß $\texttt{BCons}_{abs}\ x\ \bot\ R\bot\in$ bzw. $\texttt{BCons}_{abs}\ L\bot\in\ \bot\ x$ immer $\bot\in$ ist.

Einen Haken hat die Sache leider: Die Funktion, die wir definiert haben, ist nicht monoton. Auf den ersten Blick sieht es zwar schlimmer aus als es ist, da $L\bot$ und $R\bot$ sowie $L\bot\in$ und $R\bot\in$ ohnehin nicht vergleichbar sind, aber Schwierigkeiten macht uns

1. daß wir $\texttt{BCons}_{abs}\ \bowtie\ \bot\ \bowtie$ mit $\bot\in$ approximieren müssen, obwohl offenbar $\texttt{BCons}_{abs}\ \bowtie\ \bot\ \bowtie\ \in \bigcap\{L\bot, R\bot, L\bot\in, R\bot\in, \bot\in\}$ ist.

2. $\texttt{BCons}_{abs}\ R\bot\ \bot\ L\bot = \bot\in$ wegen $\texttt{BCons}_{abs}\ R\bot\ \bot\ L\bot\in = L\bot\in$, obwohl ersteres intuitiv in $L\bot\in$ *und* $R\bot\in$ enthalten ist. Das gleiche gilt für den gleichsam symmetrischen Fall $\texttt{BCons}_{abs}\ R\bot\in\ \bot\ L\bot = \bot\in$.

3. $\texttt{BCons}_{abs}\ L\bot\ \bot\ x$ für $x \in \{\bot, \bowtie, L\bot\}$ und $\texttt{BCons}_{abs}\ y\ \bot\ R\bot$ für $y \in \{\bot, \bowtie, R\bot\}$.

Wir können diese Verletzungen der Monotonie am einfachsten dadurch beseitigen, daß wir $\texttt{BCons}_{abs}\ x\ \bot\ y = \bot\in$ setzen für alle $x, y \geq \bowtie$. Wir erhielten dann folgende Wertetabelle für$\texttt{BCons}_{abs}\ x\ \bot\ y$:

$\bot$	$\bot$	$\bowtie$	$L\bot$	$R\bot$	$L\bot\in$	$R\bot\in$	$\bot\in$	$\top$
$\bot$	$\bowtie$	$L\bot$	$L\bot$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$\bowtie$	$R\bot$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$L\bot$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$R\bot$	$R\bot$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$L\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$R\bot\in$	$R\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$
$\top$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$	$\bot\in$

$\texttt{BCons}_{abs}\ x\ \bot\ y$

Und ebenso für $\texttt{BCons}_{abs}\ x\ \top\ y$:

$\top$	$\bot$	$\bowtie$	$L\bot$	$R\bot$	$L\bot\!\!\in$	$R\bot\!\!\in$	$\bot\!\!\in$	$\top$
$\bot$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$\bowtie$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$L\bot$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$R\bot$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$L\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$R\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$
$\top$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$	$\bot\!\!\in$

$\texttt{BCons}_{abs}\ x\ \top\ y$

Dadurch, daß wir schon $\texttt{BCons}_{abs}\ \bot\ \top\ \bot$ in $\bot\!\!\in$ einordnen müssen und eine monotone Funktion definieren wollen, fällt die abstrakte Interpretation hier sehr einfach aus [1].

Wir könnten auch noch weitere Werte in den abstrakten Domain aufnehmen, so zum Beispiel:

∞: ein Ausdruck ist lazy-terminierend, d.h. alle Äste des Baumes sind unendlich.

$\bigcirc$: es kann höchstens ein Pfad eines Baumes in der Länge bis zum Ende ausgewertet werden.

$\odot$: es kann höchstens ein Pfad eines Baumes komplett mit allen auf diesem Pfad enthaltenen Elementen ausgewertet werden.

So könnte man dann z.B. $\texttt{BCons}_{abs}\ \bowtie\ \bot\ \bowtie\ = \infty$ setzen und die Probleme mit der Monotonie auf eine andere Weise lösen. Wir werden hier aber nur ∞ für die Definition zu Hilfe nehmen:

- $\bot :=< \texttt{Bot} >$.
- $\bowtie\ :=< \texttt{Bot}, \texttt{BCons Bot Bot Bot} >$
- $\infty :=< \texttt{Bot}, \texttt{BCons}\ \infty\ \top\ \infty >$
- $L\bot\ :=< \infty, \texttt{BCons}\ \infty\ \bot\ L\bot' >$ mit $L\bot' :=< \texttt{BNil}, L\bot >$ also $L\bot :=< \texttt{Bot}, \texttt{BCons}\ \infty\ \top\ \infty, \texttt{BCons}\ \infty\ \bot\ \texttt{BNil}, \texttt{BCons}\ \infty\ \bot\ L\bot >$.

[1]Eine andere Möglichkeit wäre, dort wo ein Wert sowohl in $L\bot$ ($L\bot\!\!\in$) als auch in $R\bot$ ($R\bot\!\!\in$) ist, einfach eine der beiden Mengen auszuwählen.

- $R\bot := < \infty, \texttt{BCons}\ R\bot'\ \bot\ \infty >$ mit $R\bot' := < \texttt{BNil}, R\bot >$ also $R\bot := < \texttt{Bot}, \texttt{BCons}\ \infty\ \top\ \infty, \texttt{BCons}\ \texttt{BNil}\ \bot\ \infty, \texttt{BCons}\ R\bot\ \bot\ \infty >$.
- $L\bot\!\in\ := < \infty, \texttt{BCons}\ \infty\ \top\ L\bot\!\in' >$ mit $L\bot\!\in' := < \texttt{BNil}, L\bot\!\in >$.
- $R\bot\!\in\ := < \infty, \texttt{BCons}\ R\bot\!\in'\ \top\ \infty >$ mit $R\bot\!\in' := < \texttt{BNil}, R\bot\!\in >$.
- $\bot\!\in\ := < \texttt{Bot}, \texttt{BCons}\ \top\ \bot\ \top, \texttt{BCons}\ \bot\!\in'\ \top\ \top, \texttt{BCons}\ \top\ \top\ \bot\!\in' >$ mit $\bot\!\in' := < \texttt{BNil}, \bot\!\in >$, das bedeutet $\bot\!\in\ := < \texttt{Bot}, \texttt{BCons}\ \top\ \bot\ \top$, $\texttt{BCons}\ \bot\!\in\ \top\ \top, \texttt{BCons}\ \texttt{BNil}\ \top\ \top, \texttt{BCons}\ \top\ \top\ \bot\!\in, \texttt{BCons}\ \top\ \top\ \texttt{BNil} >$.
- $\top := < \texttt{Top} >$.

Schließlich fehlt uns noch eine abstrakte Interpretation für `case`. Wir geben sie im folgenden an:

$$
\begin{array}{rcl}
\texttt{case}_{abs}\ \bot\ e_1\ e_2 & = & \bot \\
\texttt{case}_{abs}\ \bowtie\ e_1\ e_2 & = & e_2\ \bot\ \bot \\
\texttt{case}_{abs}\ L\bot\ e_1\ e_2 & = & \sqcup\{e_2\ \infty\ \top\ \infty, e_2\ \infty\ \bot\ \texttt{BNil}, e_2\ \infty\ \bot\ L\bot\} \\
\texttt{case}_{abs}\ R\bot\ e_1\ e_2 & = & \sqcup\{e_2\ \infty\ \top\ \infty, e_2\ \texttt{BNil}\ \bot\ \infty, e_2\ R\bot\ \bot\ \infty\} \\
\texttt{case}_{abs}\ L\bot\!\in\ e_1\ e_2 & = & \sqcup\{e_2\ \infty\ \top\ \infty, e_2\ \infty\ \top\ \texttt{BNil}, e_2\ \infty\ \top\ L\bot\!\in\} \\
\texttt{case}_{abs}\ R\bot\!\in\ e_1\ e_2 & = & \sqcup\{e_2\ \infty\ \top\ \infty, e_2\ \texttt{BNil}\ \top\ \infty, e_2\ R\bot\!\in\ \top\ \infty\} \\
\texttt{case}_{abs}\ \bot\!\in\ e_1\ e_2 & = & \sqcup\{e_2\ \top\ \bot\ \top, e_2\ \bot\!\in\ \top\ \top, e_2\ \texttt{BNil}\ \top\ \top, \\
 & & \quad e_2\ \top\ \top\ \bot\!\in, e_2\ \top\ \top\ \texttt{BNil}\} \\
\texttt{case}_{abs}\ \top\ e_1\ e_2 & = & \sqcup\{e_1, e_2\ \top\ \top\ \top\}
\end{array}
$$

Ein anderer Domain

Wir wollen für den alternativen Domain die Ordnung aus Abbildung 2 angeben und die noch fehlenden Werte definieren:

- $\bigcirc := < \infty, \texttt{BCons}\ \bigcirc'\ \bot\ \infty, \texttt{BCons}\ \infty\ \bot\ \bigcirc' >$ mit $\bigcirc' := < \texttt{BNil}, \bigcirc >$.
- $\bigodot := < \infty, \texttt{BCons}\ \bigodot'\ \top\ \infty, \texttt{BCons}\ \infty\ \top\ \bigodot' >$ mit $\bigodot' := < \texttt{BNil}, \bigodot >$.

Dabei fällt auf, daß noch eine sprachliche Charakterisierung von $\bot\!\in$ möglich ist, die den Unterschied zu $\bigodot$ besonders hervorhebt: Beliebig viele Pfade eines Baumes können komplett mit allen auf diesen Pfaden enthaltenen Elementen ausgewertet werden, aber auch für mindestens einen Pfad trifft dies nicht zu.

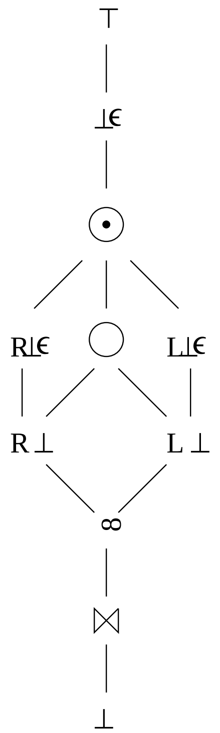

Abbildung 2: Der alternative abstrakte *Domain* für *BinBaum*.