A unified approach to sequences, bags, and trees

Citation for published version (APA):

Document status and date:
Published: 01/01/1988

Publisher Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
A unified approach to sequences, bags, and trees

by

A. Bijlsma

August, 1988
This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science Eindhoven University of Technology. Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review. Copies of these notes are available from the author or the editor.
A unified approach to sequences, bags, and trees

Introduction

This paper contains a formal definition of constructs, a new concept of which sequences, bags, and trees may be regarded as special cases. This unification of data structures commonly regarded as unrelated leads to a simpler treatment of several operations that are usually defined by structural induction for each case separately.

Constructs

DEFINITION. An ordered set $A$ is a pair $(V_A, \preceq_A)$, where $V_A$ is a set and $\preceq_A$ is a relation on $V_A$ satisfying the following laws: for arbitrary $a, b, c$ in $V_A$,

(i) $a \preceq_A a$ (reflexivity),
(ii) $a \preceq_A b \land b \preceq_A c = a \preceq_A c$ (transitivity),
(iii) $a \preceq_A b \land b \preceq_A a = b$ (antisymmetry).

For simplicity's sake, we usually write $a \in A$ instead of $a \in V_A$. We shall interpret $a \preceq_A b$ as $b \preceq_A a$, and $a \prec_A b$ as $a \preceq_A b \land a \neq b$.

For an ordered set $A$ and an arbitrary set $X$, the set $A \rightarrow X$ consists of the pairs $(\preceq_A, f)$ with $f \in V_A \rightarrow X$. Note that this definition ensures that $A \neq B$ implies $(A \rightarrow X) \cap (B \rightarrow X) = \emptyset$. Once again for simplicity, we will write $f \in A \rightarrow X$ instead of $(\preceq_A, f) \in A \rightarrow X$.

DEFINITION. Let $A$ and $B$ be ordered sets. An isomorphism of $A$ onto $B$ is a bijection $\sigma$ of $V_A$ onto $V_B$ that satisfies

$$a_0 \preceq_A a_1 = \sigma(a_0) \preceq_B \sigma(a_1)$$

for all $a_0, a_1$ in $A$. 

DEFINITION. Let $A$, and $B$ be ordered sets, $X$ and $Y$ arbitrary sets. Mappings $f \in A \rightarrow X$, and $g \in B \rightarrow Y$ are called equivalent (notation: $f \equiv g$) if there exists an isomorphism $\sigma$ of $A$ onto $B$ such that $f = g \circ \sigma$. (Note that this implies $X = Y$.)

THEOREM 0. Let $F$ denote the class of all mappings that belong to a set of the form $A \rightarrow X$, where $A$ is an ordered set and $X$ any set. Then $\equiv$ is reflexive, symmetric and transitive on $F$.

PROOF. It is reflexive, since for any $f$ in $F$,

$$f = f \circ \text{id}$$

where id denotes the identity mapping on the domain of $f$. It is symmetric, since

$$f = g \circ \sigma \Rightarrow g = f \circ \sigma^{-1}$$

Finally, it is transitive, since

$$f = g \circ \sigma \land g = h \circ \tau \Rightarrow f = h \circ (\tau \circ \sigma)$$

THEOREM 1. It is possible to associate with every $f$ in $F$ a set $C(f)$ in such a way that

$$C(f) = C(g) \iff f \equiv g$$

PROOF. If $F$ had been a set, one could have taken $C(f)$ to be the equivalence class of $f$ with respect to the equivalence relation $\equiv$. However, $F$ is not a set. We shall circumvent this objection by constructing nonempty sets $C(f)$ that consist of some, but not all, mappings equivalent to $f$.

By transfinite induction, we define for every ordinal $\alpha$ a set $R_\alpha$ in the following way: $R_0 = \emptyset$, $R_{\alpha+1} = P(R_\alpha)$, and for every limit ordinal $\alpha$, $R_\alpha$
A unified approach to sequences, bags, and trees

is the union of all sets $\mathbb{R}_\beta$ with $\beta < \alpha$. For every set $X$, the rank of $X$ is defined as the least $\alpha$ with $X \subseteq \mathbb{R}_\alpha$; in Zermelo-Fraenkel set theory, every set has rank $[3, 5]$.

The elements of $F$ are themselves sets: if $f \in A - X$, then

$$f \in \mathcal{P}(\mathbb{R}_\alpha) \times \mathcal{P}(\mathbb{R}_\alpha)$$

Now define $C(f)$ as the class of all $g$ in $F$ with $g = f$ such that $g$ is of minimal rank. If $\alpha$ is the rank of the elements of $C(f)$, we find that every element of $C(f)$ belongs to $\mathcal{P}(\mathcal{P}(\mathbb{R}_\alpha))$, so $C(f)$ is indeed a set. 

We shall call the sets $C(f)$ constructs; as we shall see below, they are sufficiently general for such a vague term to be appropriate.

In order to escape the obligation to give every mapping a name, we introduce an alternative notation for constructs. If $A$ is an ordered set and $E$ an expression in the single unbound identifier $i$, we define

$$(\forall i: i \in A : E) = C(f)$$

where $f$ is the mapping that satisfies $f(a) = E(i:= a)$ for every $a$ in $A$ and is surjective (in other words, has codomain $\{E(i:=a) \mid a \in A\}$).

This new notation also provides us with an easy way to restrict the domain of a construct, which is sometimes called filtering. Let $p$ be a boolean function defined on $A$. We define

$$(\forall i: i \in A \land p(i) : E) = (\forall i: i \in B : E)$$

where $B$ is the subset of those $a$ in $A$ that satisfy $p(a)$, and $B$ is the restriction of $\mathbb{R}_\alpha$ to $B \times B$. In case $A$ consists of the natural numbers with their usual ordering, the conjunct $i \in A$ is usually omitted.

Sequences

EXAMPLE 0. Let $A$ denote the set $\{0, 1, 2\}$ with the usual ordering.
A unified approach to sequences, bags, and trees

Define \( f : A \to \{2, 3, 6\} \) by \( f(0) = 2, f(1) = 3, f(2) = 6 \). Now a mapping \( g : B \to Y \) satisfies \( g = f \) if and only if \( Y = \{2, 3, 6\} \) and \( V_B \) consists of three elements, say \( V_B = \{b_0, b_1, b_2\} \) with \( b_0 \leq_{_A} b_1 \leq_{_A} b_2 \), such that \( g(b_0) = 2, g(b_1) = 3, g(b_2) = 6 \). What these different \( g \) have in common is that they take the values \( 2, 3 \) and \( 6 \) in that order. Therefore we wish to identify \( C(f) \) with the finite sequence \( \langle 2, 3, 6 \rangle \). Below we shall give a definition of finite sequences that achieves precisely this effect.

**Definition.** A finite sequence is a construct, say \( C(f) \), with \( f : A \to X \), such that \( f \) is surjective, \( V_A \) is finite and \( \leq_{_A} \) is a linear ordering (i.e., for every \( a_0 \) and \( a_1 \) in \( V_A \), it is true that \( a_0 \leq_{_A} a_1 \lor a_1 \leq_{_A} a_0 \)).

**Remark.** The condition that \( f \) be surjective has the effect that sequences are determined by the terms and their order alone, not by the type of these terms. For instance, there is only one sequence consisting of the numbers \( 0 \) and \( 1 \) in that order, regardless of whether these are considered as elements of the set of integers or of the set \( \{0, 1, 25, 1988\} \). There is also only one empty sequence. If some application should require the introduction of typed sequences, this is achieved by the removal of the surjectivity condition. Note, however, that Theorem 2 below will then lose its validity.

In particular, if \( p \) is a boolean function on the natural numbers satisfying

\[
(N i:: p(i)) < =
\]

the construct

\[
(C i: p(i): x_i)
\]

is a finite sequence, one that the present author has been in the habit of denoting by \( \langle \text{SEQ} i: p(i): x_i \rangle \). In the literature, many other notations can be found, for instance...
from [4]. Note, however, that not all authors make a clear distinction between \( f \) and \( C(f) \); this makes it uncertain whether dummy transformations are permitted. In case \( p(i) = 1 \leq i \leq n \), it is common practice to denote the above sequence by \( \langle x_1, x_2, \ldots, x_n \rangle \).

**THEOREM 2.** Every finite sequence can be written as \( \langle x_1, x_2, \ldots, x_n \rangle \) in precisely one way.

**PROOF.** Let \( C(f) \) be a finite sequence, where \( f \in A \rightarrow X \). Let \( n \) be the number of elements of \( VA \). Define \( \sigma \in VA \rightarrow \{1, 2, \ldots, n\} \) by

\[
\sigma(a) = (\{i: i \in A: i \leq_A a\})
\]

We shall now prove that \( \sigma \) is an isomorphism from \( A \) onto \( \{1, 2, \ldots, n\} \) with the usual ordering.

Take \( a_0 \) and \( a_1 \) in \( VA \) with \( a_0 \leq_A a_1 \) and \( a_0 \neq a_1 \) (remember that this is abbreviated as \( a_0 <_A a_1 \)). By the antisymmetry of \( \leq_A \), this implies \( \neg a_1 \leq a_0 \). Now

\[
\sigma(a_0) < \sigma(a_1)
\]

= [definition of \( \sigma \)]

\[
(\{i: i \in A: i \leq_A a_0\}) < (\{i: i \in A: i \leq_A a_1\})
\]

= [domain split]

\[
(\{i: i \in A: i \leq_A a_0\}) < (\{i: i \in A \land i \leq_A a_0: i \leq_A a_1\}) + (\{i: i \in A \land \neg i \leq_A a_0: i \leq_A a_1\})
\]

= [\( a_0 \leq a_1 \), transitivity of \( \leq_A \)]

\[
0 < (\{i: i \in A \land \neg i \leq a_0: i \leq a_1\})
\]

= [\( \neg a_1 \leq a_0 \), reflexivity of \( \leq_A \)]

true

This proves

\[
a_0 <_A a_1 \Rightarrow \sigma(a_0) < \sigma(a_1)
\]
By symmetry, the same formula holds with \(a_0\) and \(a_1\) interchanged. Now we can (finally!) use the linearity of \(\preceq_a\) to obtain

\[ a_0 \preceq_a a_1 = \sigma(a_0) \preceq \sigma(a_1) \]

and also

\[ a_0 \neq a_1 \Rightarrow \sigma(a_0) \neq \sigma(a_1) \]

in other words, \(\sigma\) is injective. Since domain and codomain of \(\sigma\) have the same finite number of elements, \(\sigma\) is in fact a bijection.

We have now proved that \(\sigma\) is an isomorphism from \(A\) onto \([1, 2, \ldots, n]\) with the usual ordering. Since \(\sigma^{-1}\) must then also be an isomorphism, the definition of equivalence gives \(f \approx f \circ \sigma^{-1}\), hence,

\[
C(f) = \{ f \approx f \circ \sigma^{-1} \text{, definition of } C \}
C(f \circ \sigma^{-1})
= \{ \text{surjectivity of } f \text{, definition of } C \}
(C i : 1 \leq i \leq n : f(\sigma^{-1}(i))
= \{ \text{definition of } \langle \ldots \rangle \}
\langle f(\sigma^{-1}(1)), f(\sigma^{-1}(2)), \ldots, f(\sigma^{-1}(n)) \rangle
\]

It remains to show uniqueness. Suppose that also \(C(f) = \langle x_1, x_2, \ldots, x_n \rangle\). Then there exists an isomorphism \(\tau\) from \([1, 2, \ldots, n]\) with the usual ordering onto itself, such that for every \(i\),

\[ x_i = f(\sigma^{-1}(\tau(i))) \]

However, such a \(\tau\) can only be the identity mapping, as we shall now prove. Consider a minimal \(i\) with \(\tau(i) \neq i\). Then

\[ \tau(i) \neq i \]
A unified approach to sequences, bags, and trees

= [bijectivity of \( \tau \)]
\[ \tau(\tau(i)) \neq \tau(i) \land \tau^{-1}(\tau(i)) \neq \tau^{-1}(i) \]

= [minimality of \( i \)]
\[ \tau(i) \geq i \land \tau^{-1}(i) \geq i \]

= [\( \tau \) is an isomorphism]
\[ \tau(i) \geq i \land i \geq \tau(i) \]

= [reflexivity and antisymmetry of \( \leq \)]
\[ \tau(i) = i \]

so no such \( i \) exists. Hence \( \tau \) must be the identity mapping. \( \Box \)

**EXAMPLE 1.** By means of dummy transformations and filtering, the same finite sequence can be written in a variety of ways. For instance,

\[
\begin{align*}
(C \ i: & \ 0 \leq i \leq 2: \ i^2 + 2) \\
(C \ i: & \ 3 \leq i \leq 5: \ i^2 - 6i + 11) \\
(C \ i: & \ 2 \leq i \leq 6 \land i \neq 4 \land i \neq 5: \ i) \\
(C \ i: & \ 2 \leq i \leq 3 \lor i = 6: \ i) \\
(C \ i: & \ i > 1 \land 6 \mod i = 0: \ i)
\end{align*}
\]

all denote the sequence \( \langle 2, 3, 6 \rangle \). \( \Box \)

**DEFINITION.** An infinite sequence is a construct, say \( C(f) \) with \( f \in A \to X \), such that \( f \) is surjective and \( A \) is isomorphic to the natural numbers with their usual ordering.

**EXAMPLE 2.** The constructs

\[(C \ i:: 2i + 1)\]

and
A unified approach to sequences, bags, and trees

\[(C \ i: \ i \mod 2 \neq 0: \ i)\]

both denote the same infinite sequence, the one whose terms are the odd natural numbers in their usual order. It is not uncommon to denote this sequence by

\[\langle 1, 3, 5, 7, \ldots \rangle\]

However, the usefulness of such a notation heavily depends on how strongly the first few terms suggest the general law. From

\[\langle 2, 3, 5, 7, 11, \ldots \rangle\]

we can probably guess that

\[(C \ i: \ i > 1 \land (A \ j, k: j > 1 \land k > 1: j \cdot k \neq i): i)\]

is meant. But what about

\[\langle 0, 1, 25, 1988, \ldots \rangle\]

REMARK. It is easy to define the reverse of a construct, say \(C(f)\) with \(f \in A \to X\): simply replace the ordering \(\leq_A\) by its inverse relation \(\geq_A\). This corresponds with the well-known reversal operation on finite sequences. If we wish to have the set of all sequences closed under reversal, we must also admit left-infinite sequences.

Bags

EXAMPLE 3. Let \(A\) denote the set \(\{0, 1, 2\}\), ordered discretely: \(a_0 \leq_A a_1 = a_0 = a_1\). Define \(f \in A \to \{7, 10\}\) by \(f(0) = 7\), \(f(1) = 10\), \(f(2) = 7\). A mapping \(g \in B \to Y\) satisfies \(g \cong f\) if and only if \(Y = \{7, 10\}\) and \(V_m\) consists of three elements, ordered discretely, such that \(g\) maps two of these onto 7 and one onto 10. What these different \(g\) have in common is that
they take the value 7 twice and the value 10 once. Therefore we wish to identify \( C(f) \) with the bag \([7, 7, 10]\). Below we shall give a definition of bags that achieves precisely this effect.

**DEFINITION.** A **bag** is a construct, say \( C(f) \) with \( f \in A \rightarrow X \), such that \( f \) is surjective and \( \preceq_A \) is discrete (i.e., \( a_0 \preceq_A a_1 \Rightarrow a_0 = a_1 \)).

The \( B \)-quantifier that is sometimes used for bags relates to our \( C \)-quantifier as follows: if \( p \) is a boolean function on the set \( N \) of natural numbers,

\[
(B \ i: p(i): x_i) = (C \ i: i \in (N, =) \land p(i): x_i)
\]

In case \( p(i) = 1 \leq i \leq n \), this bag may also be denoted by \([x_1, x_2, ..., x_n]\).

A bag \( C(f) \), where \( f \in A \rightarrow X \), is called **finite** if \( V_A \) is finite.

**THEOREM 3.** Every finite bag can be written as \([x_1, x_2, ..., x_n]\). The elements \( x_1, x_2, ..., x_n \) are uniquely determined but for permutations.

**PROOF.** Let \( C(f) \) be a finite bag, where \( f \in A \rightarrow X \). Let \( n \) denote the number of elements of \( V_A \). This means that there exists a bijection \( \sigma \) from \( \{1, 2, ..., n\} \) onto \( V_A \); such a \( \sigma \) is also an isomorphism from \( \{1, 2, ..., n\} \) with the discrete order onto \( A \). By the definition of equivalence, \( f = f \circ \sigma \), hence

\[
C(f)
= \{f = f \circ \sigma, \text{ definition of } C\}
= \{\text{surjectivity of } f, \text{ definition of } C\}
= \{\text{definition of } C\}
= \{f(\sigma(1)), f(\sigma(2)), ..., f(\sigma(n))\}
\]

It remains to show uniqueness but for permutations. Suppose that also \( C(f) \)
A unified approach to sequences, bags, and trees

= \{x_1, x_2, \ldots, x_n\} \). Then there exists an isomorphism \( \tau \) from \( \{1, 2, \ldots, n\} \) with the discrete ordering onto itself, i.e., a permutation of these numbers, such that

\[
x_1 = f(\sigma(\tau(i)))
\]

We shall now prove that our definition of bags is equivalent to the usual one.

DEFINITION. For a bag \( B \), say \( B = C(f) \) with \( f \in A \to X \), and for \( x \in X \), the number \( \#(B, x) \) is defined as

\[
\#(B, x) = (\{ i : i \in A : f(i) = x \})
\]

THEOREM 4. For any two bags \( B_0, B_1 \), say \( B_0 = C(f) \) and \( B_1 = C(g) \) with \( f \in A \to X \) and \( g \in B \to Y \),

\[
(0) \quad B_0 = B_1 \iff X = Y \land (\forall x : x \in X : \#(B_0, x) = \#(B_1, x))
\]

PROOF. According to the definition of \( C \), the bags \( C(f) \) and \( C(g) \) are equal if and only if there exists an isomorphism \( \sigma \) from \( A \) onto \( B \) such that \( f = g \circ \sigma \). As \( A \) and \( B \) are discretely ordered, such an isomorphism is nothing but a bijection from \( V_A \) onto \( V_B \). From the existence of such a bijection, the first conjunct of (0) clearly follows; we now prove that the second one does too. For any \( x \) in \( X \),

\[
\#(B_0, x)
\]

= \{definition of \#\}

\[
(\{ i : i \in A : f(i) = x \})
\]

= \{f = g \circ \sigma\}

\[
(\{ i : i \in A : g(\sigma(i)) = x \})
\]

= \{dummy transformation, \( j := \sigma(i) \}\}

\[
(\{ j : j \in B : g(j) = x \})
\]
A unified approach to sequences, bags, and trees

On the other hand, assume that \( \#(B_0, x) = \#(B_1, x) \) for every \( x \) in \( X \). Then there exists, for every \( x \) in \( X \), a bijection \( \sigma_x \) from \( f^{-1}[x] \) onto \( g^{-1}[x] \). Define \( \sigma \in V_A \to V_b \) by

\[
\sigma(a) = \sigma_{f(a)}(a)
\]

We prove the injectivity of \( \sigma \). For all \( a_0, a_1 \) in \( V_A \),

\[
\sigma(a_0) = \sigma(a_1)
\]

= (definition of \( \sigma \))

\[
\sigma_{f(a_0)}(a_0) = \sigma_{f(a_1)}(a_1)
\]

= (application of \( g \))

\[
g(\sigma_{f(a_0)}(a_0)) = g(\sigma_{f(a_1)}(a_1)) \land \sigma_{f(a_0)}(a_0) = \sigma_{f(a_1)}(a_1)
\]

= (definition of \( \sigma_x \) implies \( g(\sigma_x(y)) = x \))

\[
f(a_0) = f(a_1) \land \sigma_{f(a_0)}(a_0) = \sigma_{f(a_1)}(a_1)
\]

= (injectivity of \( \sigma_{f(a_0)} \))

\[
a_0 = a_1
\]

Moreover, \( \sigma \) is surjective: for any \( b \) in \( V_b \) we have \( b \in g^{-1}[g(b)] \), so by the bijectivity of the \( \sigma_x \), there exists an \( a \) in \( f^{-1}[g(b)] \) with \( \sigma_{g(b)}(a) = b \). Since \( a \in f^{-1}[g(b)] \), we have \( f(a) = g(b) \), so \( \sigma(a) = \sigma_{f(a)}(a) = \sigma_{g(b)}(a) = b \).

We have now established that \( \sigma \) is a bijection from \( V_A \) onto \( V_b \). For any \( a \) in \( V_A \), \( g(\sigma(a)) = g(\sigma_{f(a)}(a)) = f(a) \). Together with \( X = Y \) this yields \( f = g \circ \sigma \).

Theorem 4 shows that a bag \( B \) may equally well be identified with the function \( \#(B,m) \in X \to N \); that is the usual definition.
**Concatenation and summation**

**DEFINITION.** Let $A$ and $B$ be ordered sets. The *concatenation* of $A$ and $B$, denoted $A \ ++ \ B$, is the ordered set $(V_{A++B}, S_{A++B})$, where

$$V_{A++B} = \{(0, a) \mid a \in V_A\} \cup \{(1, b) \mid b \in V_B\}$$

and

$$(i_0, c_0) \leq_{A++B} (i_1, c_1) \iff (i_0 = 0 \land i_1 = 1) \lor (i_0 = i_1 = 0 \land c_0 \leq_{A} c_1) \lor (i_0 = i_1 = 1 \land c_0 \leq_{B} c_1).$$

**DEFINITION.** Consider constructs $C(f)$ and $C(g)$, where $f \in A \to X$ and $g \in B \to Y$. The *concatenation* of $C(f)$ and $C(g)$, denoted $C(f) \ ++ \ C(g)$, is the construct $C(h)$, where $h \in (A ++ B) \to X \cup Y$ is defined by

$$(1) \quad h(0, a) = f(a) \land h(1, b) = g(b)$$

for $a \in A$, $b \in B$.

**THEOREM 5** ("domain split law"). For $f$ defined on $A ++ B$,

$$(C \ i: i \in A ++ B: f(i)) = (C \ i: i \in A: f(0, i)) ++ (C \ i: i \in B: f(1, i))$$

**EXAMPLE 4.**

(a) $<1, 2, 5> ++ <5, 3, 0> = <1, 2, 5, 5, 3, 0>$. 
(b) $(C \ i: 0 \leq i < m: f(i)) ++ (C \ i: 0 \leq i < n: g(i))$

$$= (C \ i: 0 \leq i < m+n: \text{if } i < m \to f(i) \lor i \geq m \to g(i-m) \ f(i)).$$

(c) $(C \ i: i \in (N, \geq): f(i)) ++ (C \ i: i \geq 1: f(i))$

$$= (C \ i: i \in (Z, \leq): f(|i|)).$$

(d) $(C \ i: f(i)) ++ <2>

$$= (C \ i: i \in (\omega, 1, \leq): \text{if } i < \omega \to f(i) \cup i = \omega \to 2 \ f(i)).$$
Concatenation of constructs reduces to the normal concatenation operator when the constructs are sequences. There is also an operation on constructs that generalizes the normal operator on bags:

**DEFINITION.** Consider constructs $C(f)$ and $C(g)$, where $f : A \to X$ and $g : B \to Y$. The sum of $C(f)$ and $C(g)$, denoted $C(f) + C(g)$, is the construct $C(h)$, where $h : (A + B) \to X \cup Y$ is defined by (1).

**THEOREM 6 ("domain split law").** For $f$ defined on $A + B$,

$$(\sum_{i : i \in A + B} f(i)) = (\sum_{i : i \in A} f(0, i)) + (\sum_{i : i \in B} f(1, i))$$

In the final part of this section, we explore the relationship between $++$ and $+$. 

**DEFINITION.** Let $A$ be an ordered set. Consider $(\leq_A, f) \in A \to X$. Define

$$\text{Bag}(\leq_A, f) = C(=, f)$$

i.e., the construct that corresponds to $f$ considered as a mapping defined on
A unified approach to sequences, bags, and trees

a discretely ordered set.

THEOREM 7. For \( f, g \) in \( F \),

\[ C(f) = C(g) \Rightarrow \text{Bag}(f) = \text{Bag}(g) \]

PROOF. Let \( \sigma \) be an isomorphism from \( A \) onto \( B \). Then \( \sigma \) is injective, so for \( a_0, a_1 \) in \( A \),

\[ a_0 = a_1 = \sigma(a_0) = \sigma(a_1) \]

Therefore \( \sigma \) is an isomorphism from \((VA, =)\) onto \((VB, =)\).

COROLLARY. We can define \( \text{BAG}(CO) \) for every construct \( CO \) in the following way: choose an \( f \) such that \( CO = C(f) \) and let \( \text{BAG}(CO) = \text{Bag}(f) \). The theorem ensures that the result is independent of the choice of \( f \). Every bag \( B \) can be obtained in this way, since \( \text{BAG}(B) = B \).

THEOREM 8. For constructs \( CO, CI \),

\[ \text{BAG}(CO ++ CI) = \text{BAG}(CO) + \text{BAG}(CI) \]

PROOF. Write \( CO = C(\leq_A, f) \), where \( f \in VA \rightarrow X \), and \( CI = C(\leq_B, g) \), where \( g \in B \rightarrow Y \). According to the definition of ++, we have \( CO ++ CI = C(\leq_{A+B}, h) \), where \( h \) is defined by (1). Therefore

\[
\begin{align*}
\text{BAG}(CO ++ CI) &= \{\text{definition of BAG}\} \\
&= \{\text{definition of Bag}\} \\
&= \{\text{definition of } ++\} \\
&= \{\text{definition of Bag}\} \\
&= \{\text{definition of Bag}\} \\
&= \text{Bag}(\leq_A, f) + \text{Bag}(\leq_B, g)
\end{align*}
\]
A unified approach to sequences, bags, and trees

= \{\text{definition of BAG}\}
BAG(C0) + BAG(C1)

Canonical decomposition

We shall call a construct \( C(f) \), with \( f \in A \times X \), empty if \( A \) is empty. It follows that there is only one empty construct, namely \( \{\emptyset\} \).

We shall call a construct irreducible if it cannot be written as the concatenation or sum of nonempty constructs. Clearly, every finite construct may be built from irreducible constructs with the operators ++ and +. The question naturally occurs whether this decomposition is in some sense unique. Obviously, it cannot be completely unique, since both ++ and + are associative and + is commutative (as operators between constructs; they are so up to isomorphism as operators between ordered sets). Therefore, we must allow rearrangement of terms and introduction of parentheses that exploit these properties; for instance, \([2] + ([1] + [0]) = [0] + [1] + [2] \). However, apart from this the decomposition turns out to be unique. The next three theorems have the purpose of establishing this.

THEOREM 9. There are no nonempty constructs \( C_0, C_1, C_2, C_3 \) with

\[ C_0 ++ C_1 = C_2 + C_3 \]

PROOF. For elements \( a \) and \( b \) of an ordered set \( A \) we define

\[ a \rightarrow_A b = (\exists x : x \in A : (x \leq_A a \lor x \geq_A a) \land (x \leq_A b \lor x \geq_A b)) \]

Now ++ has the property that for nonempty ordered sets \( A \) and \( B \),

\[ (A, a, b : a \in A ++ B \land b \in A ++ B : a \rightarrow_A b) \]

This can be seen in the following way. If \( a \) and \( b \) are both in the part of \( A ++ B \) that corresponds to \( A \), every element \( x \) of the part that corresponds to \( B \) is comparable to both \( a \) and \( b \). Similarly if they are both
in the other part. If they are in different parts, they are themselves comparable and we may take the \( x \) in the definition of \( \leftrightarrow_A \) to be \( a \) or \( b \).

On the other hand, \( + \) has the property that for nonempty ordered sets \( A \) and \( B \),

\[
(E \ a, \ b: \ a \in A + B \land b \in A + B: \neg a \leftrightarrow_B b)
\]

To see this, it suffices to take \( a \) in the part of \( A + B \) that corresponds to \( A \) and \( b \) in the part that corresponds to \( B \).

Therefore no ordered set can be written nontrivially as both a concatenation and a sum; it follows that no construct can either.

THEOREM 10. If \( C_0, C_1, C_2, C_3 \) are constructs such that

\[
C_0 ++ C_1 = C_2 ++ C_3
\]

there exists a construct \( C_4 \) such that

\[
C_0 = C_2 ++ C_4 \lor C_2 = C_0 ++ C_4
\]

PROOF. Let \( C_0 ++ C_1 \) be \( C(f) \) with \( f \in A \rightarrow X \). Let \( A_0, A_1, A_2, A_3 \) denote the parts of \( A \) corresponding to \( C_0, C_1, C_2, C_3 \) respectively. For \( a \in A_0 \setminus A_2 \) and \( b \in A_2 \setminus A_0 \) we have \( a \prec_A b \) since \( C(f) = C_0 ++ C_1 \), and \( a \succ_A b \) since \( C(f) = C_2 ++ C_3 \). These are mutually contradictory, so

\[
A_0 \setminus A_2 = \emptyset \lor A_2 \setminus A_0 = \emptyset
\]

We only consider the case \( A_0 \setminus A_2 = \emptyset \); the other one is symmetric. Since in this case \( A_0 = A_0 \cap A_2 \), it follows that \( C_2 = C_0 ++ C(f \mid A_2 \setminus A_0) \).

THEOREM 11. Suppose that \( C_0 \) is a nonempty construct that is not the sum of two nonempty constructs. If \( C_1, C_2, C_3 \) are constructs such that
C₀ + C₁ = C₂ + C₃ ,

there exists a construct C₄ such that

C₂ = C₀ + C₄ ∨ C₃ = C₀ + C₄ .

PROOF. Let A₀ through A₃ be defined as in the preceding proof. Obviously,

A₀ ∩ A₂ ≠ ∅ ∨ A₀ ∩ A₃ ≠ ∅ .

We assume the first disjunct; the other one is treated by exchanging the roles of C₂ and C₃ .

If A₀ \ A₂ ≠ ∅ , we have C₀ = C(f | A₀ ∩ A₂) + C(f | A₀ \ A₂) , where both terms are nonempty. Since it is given that C₀ is not such a sum, it follows that A₀ \ A₂ = ∅ . Therefore A₀ = A₀ ∩ A₂ , so C₂ = C₀ + C(f | A₂ \ A₀).

COROLLARY. Decomposition of a finite construct into irreducible constructs by means of the operators ++ and + is unique except for the trivial variations that correspond to the associativity of both operators and the commutativity of + .

Trees

EXAMPLE 5. Let A denote the set of words of length at most 2 over the alphabet {0, 1, 2} . For words t and u we let t A u mean that t is a prefix of u . Define f on A as the sum of the numerical values of the symbols in a word. We claim that C(f) is a proper abstraction for the ternary tree

```
0
/\  \\
0 1 2
/\ /\  \\
0 1 2 1 2 3 2 3 4
```
Note that the left-right ordering of the successors of any given node is purely arbitrary: interchanging, for instance, the leftmost 0 and 1 on the bottom row would not change $C(f)$. 

It is our intention to give the name "tree" to a construct of the kind considered in Example 5. We shall now proceed to formalize the definition and explore the relationship between those constructs and the concept of a tree as it is known from graph theory.

**DEFINITION.** Let $A$ be an ordered set. The relation $(\langle A \rangle)$ is defined on $A$ by

$$a \langle A \rangle b \equiv a \langle A \rangle b \land \exists x : x \in A : a \langle A \rangle x \langle A \rangle b$$

If $a$ and $b$ satisfy $a \langle A \rangle b$, we say that $a$ is a predecessor of $b$, and that $b$ is a successor of $a$.

**DEFINITION.** An ordered set $A$ is called tree-like if $A$ is finite and there exists an $r$ in $A$ such that for all $a$ in $A$,

$$(\forall x : x \in A : x \langle A \rangle r = \{ a = r \land a \neq r \lor 1 \}$$

If such an $r$ exists, it is obviously unique; we call it the root of $A$.

**THEOREM 12.** For every tree-like ordered set $A$,

$$(\forall x, y : x \in A \land y \in A : x \langle A \rangle y = |V_A| - 1$$

**PROOF.** Let $A$ be tree-like with root $r$. Then

$$(\forall x, y : x \in A \land y \in A : x \langle A \rangle y)$$

= \{domain split\}

$$(\exists y : y \in A : (\forall x : x \in A : x \langle A \rangle y)$$
A unified approach to sequences, bags, and trees

\[ A \text{ is tree-like with root } r \]
\[ \{ y \in A : \text{if } y = r \rightarrow 0 \land y \neq r \rightarrow 1 \} \]
\[ \{ \text{domain split} \} \]
\[ \{ y \in A \land y \neq r : 1 \} \]
\[ \{ \text{definition of } N \} \]
\[ \{ r \in A \} \]
\[ |V_A| - 1 \]

THEOREM 13. In a tree-like ordered set, the root is the least element.

PROOF. Let \( A \) be tree-like with root \( r \). Define \( d \in A \rightarrow N \) by

\[ d(a) = (N x : x \in A : x <_A a) \]

The claim that every \( a \) in \( A \) satisfies \( r \leq_A a \) will be proved by induction on \( d(a) \). If \( d(a) = 0 \), we deduce from \( x (<_A a) \Rightarrow x <_A a \) that \( a = r \) and therefore, by reflexivity, \( r \leq_A a \). If \( d(a) > 0 \), there is an element \( b \) in \( A \) with \( b <_A a \). Now

\[ d(a) \]
\[ = \{ \text{definition of } d \} \]
\[ (N x : x \in A : x <_A a) \]
\[ = \{ \text{domain split} \} \]
\[ (N x : x \in A \land x <_A b : x <_A a) + (N x : x \in A \land \neg x <_A b : x <_A a) \]
\[ = \{ b <_A a \Rightarrow x <_A b = x <_A a \} \]
\[ (N x : x \in A : x <_A b) + (N x : x \in A \land \neg x <_A b : x <_A a) \]
\[ > \{ \neg b <_A b \text{ and } b <_A a \} \]
\[ (N x : x \in A : x <_A b) \]
\[ = \{ \text{definition of } d \} \]
\[ d(b) \]

Applying the induction hypothesis to \( b \), we obtain \( r \leq_A b \). Hence, by transitivity, \( r \leq_A a \).
A unified approach to sequences, bags, and trees

DEFINITION. For any tree-like ordered set $A$ with root $r$, the predecessor function $p_a$ is determined by $p_a \in A \setminus \{r\} \rightarrow A$ and $p_a(a) \in \{>) a$ for all $a$ in $A \setminus \{r\}$. 

THEOREM 14. For all $a$ and $b$ in a tree-like ordered set $A$,

$$a \leq_A b = (\exists i : i \in \mathbb{N} : a = p_A^i(b))$$

PROOF. Define $d : A \times A \rightarrow \mathbb{N}$ by

$$d(a, b) = (\exists x : x \in A : a <_A x <_A b)$$

We shall prove the implication from left to right by induction on $d(a, b)$. The theorem then follows by observing that the other implication is a direct consequence of the transitivity of $\leq_A$.

Base: assume $a \leq_A b$. Then

$$d(a, b) = 0$$

$$= \{|\text{definition of} ~ d|\}$$

$$= (\exists x : x \in A : a <_A x <_A b)$$

$$= \{|a \leq_A b = a = b \lor a <_A b|\}$$

$$= \{|\text{definition of} ~ p_A|\}$$

$$= \{|\text{instantiation at} ~ i = 0 ~ \text{and} ~ i = 1 \text{~respectively}|\}$$

$$(\exists i : i \in \mathbb{N} : a = p_A^i(b))$$

Step: suppose $d(a, b) > 0$. Then there exists an element $c$ of $A$ with $a <_A c <_A b$. As $d(a, c) < d(a, b)$, the induction hypothesis gives an $i$ with $a = p_A^i(c)$. As also $d(c, b) < d(a, b)$, the induction hypothesis gives an $j$ with $c = p_A^j(b)$. Therefore $a = p_A^{i+j}(b)$.

Given an ordered set $A$, we may construct an undirected graph $G$ (to be referred to as "the graph of $A"$") as follows. Vertices of $G$ are the
A unified approach to sequences, bags, and trees

elements of $V_A$; there is an edge between $a$ and $b$ iff $a \prec_A b \lor b \prec_A a$. If $A$ is tree-like, Theorems 13 and 14 together imply that $G$ is connected (since any two points are connected by way of the root). Moreover, by Theorem 12, the number of edges of $G$ is $1$ less than the number of vertices. This establishes that $G$ is a tree in the sense of graph theory. The next theorem shows that every graph-theoretical tree can be obtained in this way.

THEOREM 15. Let $G$ be a connected undirected graph of which the number of edges is $1$ less than the number of vertices. Then there exists a tree-like ordered set $A$ such that $G$ is the graph of $A$.

PROOF. For $V_A$, we take the collection of vertices of $G$. A root $r$ is chosen arbitrarily in $V_A$. It is known from graph theory (e.g. [1], Theorem 4.1) that $G$ is free of cycles; therefore, for every $a$ in $V_A$ there exists precisely one path from $a$ to $r$. Let $a \preceq_A b$ mean that $a$ occurs on the path from $b$ to $r$. Then $A = (V_A, \preceq_A)$ is an ordered set.

Moreover, $a \prec_A b$ now has the meaning that $a$ is the first node after $b$ on the path from $b$ to $r$; therefore $a \prec_A b \lor b \prec_A a$ implies the existence of an edge between $a$ and $b$.

Conversely, let $a$ and $b$ be connected by an edge of $G$. If $\lnot a \preceq_A b$, the node $a$ does not occur on the path from $b$ to $r$. Then the edge from $a$ to $b$, followed by the path from $b$ to $r$, constitutes a path from $a$ to $r$ on which $b$ occurs as the first node after $a$. Hence $b \prec_A a$. We have now proved that

$$a \preceq_A b \lor b \prec_A a$$

By symmetry, the same formula holds with $a$ and $b$ interchanged. Distributing $\land$ over $\lor$ yields

$$a \prec_A b \lor b \prec_A a$$

The last two paragraphs together show that two nodes are connected by an edge iff one precedes the other in the ordering of $A$. Therefore $G$ is the
A unified approach to sequences, bags, and trees

graph of $A$.

REMARK. The difference in style between this proof and the preceding ones is one of the reasons why we have chosen to define constructs in the language of ordered sets rather than that of graphs.

The idea of a tree as we shall now define it is slightly different from the concept in graph theory. By a tree, we shall mean an arrangement of (not necessarily distinct) values; isomorphic trees that have the same values in corresponding places will be identified. The following definition is intended to reflect this point of view.

DEFINITION. A tree is a construct $C(f)$, say with $f \in A \to X$, such that $A$ is tree-like and $f$ is surjective.

EXAMPLE 6. In Example 5, a tree was introduced as the construct $C(f)$ of an explicitly given function $f$. However, it could also have been described with the aid of the operators $++$ and $\cdot$, since the tree involved is exactly equal to

$$[0] ++ ([0] ++ ([0] + [1] + [2]))$$
$$+ ([1] ++ ([1] + [2] + [3]))$$
$$+ ([2] ++ ([2] + [3] + [4]))$$

This is no accident: in the rest of this section we shall show that every tree can be decomposed in this way.

THEOREM 16. Every tree may be built from singleton bags with the operators $++$ and $\cdot$.

PROOF. Consider a tree $C(f)$, where $f \in A \to X$. Let $r$ be the root of $A$. We prove the statement by induction on the number of elements of $A$. If $|A| = 1$, we have $C(f) = [f(r)]$. If $|A| > 1$, consider

$$B = \{b \in A \mid r (\langle A \rangle b)\}$$
We shall prove that

(2) \[ C(f) = [f(r)] ++ (\forall b: b \in B: (\exists a: a \in T(b): f(a))) \]

where \( \forall \) denotes the quantifier corresponding to the operator \( ++ \) and where

\[ T(b) = ([a \in A | b \preceq_A a], \preceq_A) \]

Note that

\[ T(b) \text{ is tree-like with root } b \]
\[ = \{ \text{definition of tree-like} \}
\[ (\forall a: a \in T(b): (\exists x: x \in T(b): x (<_A) a) = \begin{cases} \text{if } a = b \to 0 & \text{if } a \neq b \to 1 \end{cases} \}
\]
\[ = \{ \text{definition of } p_A \}
\[ (\forall a: a \in T(b): (\exists x: x \in T(b): x = p_A(a)) = \begin{cases} \text{if } a = b \to 0 & \text{if } a \neq b \to 1 \end{cases} \}
\]
\[ = \{ \text{one-point rule for } N \}
\[ (\forall a: a \in T(b): p_A(a) \in T(b) = a \neq b) \]
\[ = \{ \text{definition of } T \}
\[ (\forall a: b \preceq_A a: b \preceq_A p_A(a) = a \neq b) \]
\[ = \{ \text{domain split} \}
\[ (\forall a: b <_A a: b \preceq_A p_A(a)) \land \neg b \preceq_A p_A(b) \]
\[ = \{ p_A(b) <_A b , antisymmetry \}
\[ (\forall a: b <_A a: b \preceq_A p_A(a)) \]
\[ = \{ \text{Theorem 14} \}
\[ (\exists i: i > 0: b = p_A^i(a)): (\exists i: i \geq 0: b = p_A^i(p_A(a))) \]
\[ = [] \]
\[ \text{true} \]

As \( |T(b)| < |A| \) for all \( b \) in \( B \), the theorem then follows by induction.
It remains to prove (2). Due to Theorem 13, it is sufficient to prove that for \( a_0, a_1 \) in \( A \setminus \{ r \} \) and \( b_0, b_1 \) in \( B \),

\[
(3) \quad a_0 \preceq a_1 \Rightarrow (\exists b : b \in B : a_0 \in T(b) \land a_1 \in T(b))
\]

\[
(4) \quad b_0 \neq b_1 \Rightarrow T(b_0) \cap T(b_1) = \emptyset
\]

Proof of (3):

\[
a_0 \preceq a_1
\]

\[
= \{ \text{Theorem 13, } a_0 \neq r \}
\]

\[
a_0 \preceq a_1 \land r \prec a_0
\]

\[
= \{ \text{Theorem 14} \}
\]

\[
a_0 \preceq a_1 \land (\exists i : i > 0 : r = p_{a_1}^i(a_0))
\]

\[
= \{ \text{dummy transformation, } b := p_{a_1}^{-1}(a_0) \}
\]

\[
a_0 \preceq a_1 \land (\exists b : b \in A : r (\preceq a) b \land b \preceq a_0)
\]

\[
= \{ \text{transitivity} \}
\]

\[
(\exists b : b \in A : r (\preceq a) b \land b \preceq a_0 \land b \preceq a_1)
\]

\[
= \{ \text{definitions of } B \text{ and } T \}
\]

\[
(\exists b : b \in B : a_0 \in T(b) \land a_1 \in T(b))
\]

Proof of (4):

\[
a \in T(b_0) \land a \in T(b_1)
\]

\[
= \{ \text{definition of } T \}
\]

\[
b_0 \preceq a \land b_1 \preceq a
\]

\[
= \{ \text{Theorem 14} \}
\]

\[
(\exists i, j : i \in \mathbb{N} \land j \in \mathbb{N} : b_0 = p_{a_1}^i(a) \land b_1 = p_{a_1}^j(a))
\]

\[
= \{ k := |j-i| \}
\]

\[
(\exists k : k \in \mathbb{N} : b_0 = p_{a_k}^k(b_1) \lor b_1 = p_{a_k}^k(b_0))
\]

\[
= \{ p_{a_k}(b_0) = r, p_{a_k}(b_1) = r, r \text{ not in the domain of } p_{a_k} \}
\]

\[
b_0 = b_1 \lor b_0 = r \lor b_1 = r
\]

\[
= \{ b_0 \in B, \text{ so } b_0 \neq r ; b_1 \in B, \text{ so } b_1 \neq r \}
\]

\[
b_0 = b_1
\]
Theorem 16 establishes that every tree may be decomposed into singleton bags. Since these are obviously irreducible, the corollary to Theorem 15 shows that this decomposition is essentially unique.

**Remark.** It is not true that every finite construct can be decomposed into singleton bags. As a counterexample, consider the ordered set A consisting of \{2, 3, 6, 9\} with the divisibility ordering. If f is the identity function on A, then C(f) is the construct 

![Diagram](image)

This is not a concatenation, since we have, in the notation of the proof of Theorem 9, \( \triangleleft 2 \rightarrow 9 \). Neither is it a sum, since its graph is connected.

**Structural induction**

Let X be a set and let Co(X) denote the set of all constructs that can be built from singleton bags over X by means of ++ and +. Then all finite nonempty sequences, bags and trees with values from X belong to Co(X). Due to the essential uniqueness of the canonical decomposition, we may define functions on Co(X) by describing their values on singleton bags and their behaviour under ++ and +.

**Definition.** Let $\$ denote a commutative, associative operator on X. Then $\$/ is the X-valued function on Co(X) that satisfies

\[
\begin{align*}
\$/[x] &= x \text{ for all } x \text{ in } X, \\
\$/([C_0] + [C_1]) &= ([\$/C_0] \$/[C_1]) \text{ for all } C_0, C_1 \text{ in } Co(X), \\
\$/([C_0] + [C_1]) &= ([\$/C_0] \$/[C_1]) \text{ for all } C_0, C_1 \text{ in } Co(X).
\end{align*}
\]

(In [0] and [2] the symbol / was introduced for sequences and bags separate-
A unified approach to sequences, bags, and trees

ly. The pronunciation of * is variously reported as "reduce" [0] or "inserted in" [2].)

EXAMPLE 7. If * denotes multiplication in \( \mathbb{Z} \), then

(a) \( */\langle 2, 3, 5 \rangle = */(\langle 2 \rangle ++ \langle 3 \rangle ++ \langle 5 \rangle) = 2 \cdot 3 \cdot 5 = 30 \);

(b) \( */\langle 2, 3, 5 \rangle = */(\langle 2 \rangle + \langle 3 \rangle + \langle 5 \rangle) = 2 \cdot 3 \cdot 5 = 30 \);

(c) \( */(\langle 2, 3 \rangle ++ \langle 2, 3, 5 \rangle) = */(\langle 2 \rangle + \langle 3 \rangle + \langle 2, 3, 5 \rangle) = 2 \cdot 3 \cdot 2 \cdot 3 \cdot 5 = 180 \).

EXAMPLE 8. Many familiar quantifiers can be defined as the application of a function of the form \( \$ / \) on \( \mathsf{Co}(X) \). For instance,

\[
\begin{align*}
+/ (\mathcal{C} \ i: i \in A: x_t) &= (\mathcal{S} \ i: i \in A: x_t), \\
+/ (\mathcal{C} \ i: i \in A: \{x_t\}) &= (\mathcal{B} \ i: i \in A: x_t), \\
*/ (\mathcal{C} \ i: i \in A: x_t) &= (\mathcal{P} \ i: i \in A: x_t), \\
\min/ (\mathcal{C} \ i: i \in A: x_t) &= (\mathcal{M} \ i: i \in A: x_t), \\
\max/ (\mathcal{C} \ i: i \in A: x_t) &= (\mathcal{M} \ i: i \in A: x_t), \\
\land/ (\mathcal{C} \ i: i \in A: b_t) &= (\mathcal{A} \ i: i \in A: b_t), \\
\lor/ (\mathcal{C} \ i: i \in A: b_t) &= (\mathcal{A} \ i: i \in A: b_t).
\end{align*}
\]

DEFINITION. For \( f \in X \rightarrow Y \) the mapping \( f^* \in \mathsf{Co}(X) \rightarrow \mathsf{Co}(Y) \) is defined by

\[
\begin{align*}
f^*[x] &= [f(x)] \text{ for all } x \text{ in } X, \\
f^*(\mathcal{C} \ C_0 ++ C_1) &= (f^*C_0) ++ (f^*C_1) \text{ for all } C_0, C_1 \text{ in } \mathsf{Co}(X), \\
f^*(\mathcal{C} \ C_0 + C_1) &= (f^*C_0) + (f^*C_1) \text{ for all } C_0, C_1 \text{ in } \mathsf{Co}(X).
\end{align*}
\]

(See again [0] and [2] for sequences and bags separately. The pronunciation is "map" [0] or "applied to all" [2].)

EXAMPLE 9. Let \( f \in \mathbb{Z} \rightarrow \mathbb{Z} \) be defined by \( f(x) = x^2 \). Then

(a) \( f^*\langle 2, 3, 5 \rangle = \langle 4, 9, 25 \rangle \);

(b) \( f^*[2, 3, 5] = \langle 4, 9, 25 \rangle \);

(c) \( f^*(\langle 2, 3 \rangle ++ \langle 2, 3, 5 \rangle) = \langle 4, 9 \rangle ++ \langle 4, 9, 25 \rangle \).
EXAMPLE 10. Let \([\mathcal{M}] \in X \to \mathcal{C}(X)\) be defined by \([\mathcal{M}](x) = [x]\). Then

\[
\text{BAG } | \text{Co}(X) = +/ \circ [\mathcal{M}]^*.
\]

Heaps and search trees

We would like to use the word "heap" to refer to trees in which the values occurring on any root path are ascending. It is very easy to phrase this in the language of constructs:

DEFINITION. Let \(A\) be a tree-like ordered set, \(B\) a linearly ordered set, \(f \in V_A \to V_B\). The tree \(C(\leq_A, f)\) is called a heap with respect to \(\leq_B\) if for every \(a_0\) and \(a_1\) in \(A\),

\[
a_0 \leq_A a_1 \Leftrightarrow f(a_0) \leq_B f(a_1).
\]

The modelling of search trees presents much more difficulties, mainly because the definition of tree is such that the successors of a node cannot be distinguished.

DEFINITION. Let \(A\) be a tree-like ordered set, \(B\) a linearly ordered set, \(f \in V_A \to V_B\). An element \(a\) of \(A\) is called left with respect to \((\leq_A, \leq_B, f)\) if \(a\) is not the root of \(A\) and

\[
(\exists x: a \leq_A x: f(x) \leq_B f(p_A(a))).
\]

An element \(a\) is called right with respect to \((\leq_A, \leq_B, f)\) if \(a\) is left with respect to \((\leq_A, \geq_B, f)\).

THEOREM 17. Let \(A\) be a tree-like ordered set, \(B\) a linearly ordered set, \(f \in V_A \to V_B\). Assume that

(5) every element of \(A\) except the root is either left or right.
Then the following conditions are equivalent:

(6) every element of A has at most one left successor and at most one right successor;

(7) for all $a, b, c$ in A such that $b$ and $c$ are different successors of $a$,

$$f(b) \leq f(c) \Rightarrow f(b) \leq f(a) \leq f(c).$$

(Here "left" and "right" are with respect to $(\leq_A, \leq_B, f)$.)

PROOF. Assume (6). Take $a, b, c$ in A such that $b$ and $c$ are different successors of $a$ with $f(b) \leq f(c)$. Suppose $f(a) \leq f(b)$. Then $b$ is not left and by transitivity of $\leq_B$ we have $f(a) \leq f(c)$, so $c$ is not left either. From (5) we conclude that $b$ and $c$ are both right. This contradicts (6), so $f(a) \leq f(b)$. From the linearity of $\leq_B$ we see that $f(b) \leq f(a)$. Similarly we get $f(a) \leq f(c)$.

Now assume (7). Let $b$ and $c$ be different successors of $a$. As $\leq_B$ is linear, $f(b)$ and $f(c)$ are comparable, say $f(b) \leq f(c)$. From (7) we see that $f(b) \leq f(a)$, so $b$ is not right, and that $f(a) \leq f(c)$, so $c$ is not left.

If $A$, $B$ and $f$ have properties (5) through (7), while $(\leq_A, f) \neq (\leq_{A1}, f_1)$, then (5) through (7) also hold with $A$ and $f$ replaced by $A_1$ and $f_1$. Therefore the following definition makes sense.

DEFINITION. Let $A$ be a tree-like ordered set, $B$ a linearly ordered set, $f \in V_A \cap V_B$. The tree $C(\leq_A, f)$ is called a search tree with respect to $\leq_B$ if (5) and (6) (or, equivalently, (5) and (7)) hold.
A unified approach to sequences, bags, and trees

References


March 27, 1988

A. Bijlsma
In this series appeared:

<table>
<thead>
<tr>
<th>No.</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>85/01</td>
<td>R.H. Mak</td>
<td>The formal specification and derivation of CMOS-circuits</td>
</tr>
<tr>
<td>85/02</td>
<td>W.M.C.J. van Overveld</td>
<td>On arithmetic operations with M-out-of-N-codes</td>
</tr>
<tr>
<td>85/03</td>
<td>W.J.M. Lemmens</td>
<td>Use of a computer for evaluation of flow films</td>
</tr>
<tr>
<td>85/04</td>
<td>T. Verhoeff, H.M.J.L. Schols</td>
<td>Delay insensitive directed trace structures satisfy the foam rubber wrapper postulate</td>
</tr>
<tr>
<td>86/01</td>
<td>R. Koymans</td>
<td>Specifying message passing and real-time systems</td>
</tr>
<tr>
<td>86/02</td>
<td>G.A. Bussing, K.M. van Hee, M. Voorhoeve</td>
<td>ELISA, A language for formal specifications of information systems</td>
</tr>
<tr>
<td>86/03</td>
<td>Rob Hoogerwoord</td>
<td>Some reflections on the implementation of trace structures</td>
</tr>
<tr>
<td>86/04</td>
<td>G.J. Houben, J. Paredaens, K.M. van Hee</td>
<td>The partition of an information system in several parallel systems</td>
</tr>
<tr>
<td>86/05</td>
<td>Jan L.G. Dietz, Kees M. van Hee</td>
<td>A framework for the conceptual modeling of discrete dynamic systems</td>
</tr>
<tr>
<td>86/06</td>
<td>Tom Verhoeff</td>
<td>Nondeterminism and divergence created by concealment in CSP</td>
</tr>
<tr>
<td>86/07</td>
<td>R. Gerth, L. Shira</td>
<td>On proving communication closedness of distributed layers</td>
</tr>
<tr>
<td>86/09</td>
<td>C. Huizing, R. Gerth, W.P. de Roever</td>
<td>Full abstraction of a real-time denotational semantics for an OCCAM-like language</td>
</tr>
<tr>
<td>86/10</td>
<td>J. Hooman</td>
<td>A compositional proof theory for real-time distributed message passing</td>
</tr>
<tr>
<td>86/11</td>
<td>W.P. de Roever</td>
<td>Questions to Robin Milner - A responder's commentary (IFIP86)</td>
</tr>
<tr>
<td>86/12</td>
<td>A. Boucher, R. Gerth</td>
<td>A timed failures model for extended communicating processes</td>
</tr>
<tr>
<td>Year</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>86/14</td>
<td>R. Koymans</td>
<td>Specifying passing systems requires extending temporal logic</td>
</tr>
<tr>
<td>87/01</td>
<td>R. Gerth</td>
<td>On the existence of sound and complete axiomatizations of the monitor concept</td>
</tr>
<tr>
<td>87/02</td>
<td>Simon J. Klaver, Chris F.M. Verberne</td>
<td>Federative Databases</td>
</tr>
<tr>
<td>87/03</td>
<td>G.J. Houben, J. Paredaens</td>
<td>A formal approach to distributed information systems</td>
</tr>
<tr>
<td>87/04</td>
<td>T. Verhoeff</td>
<td>Delay-insensitive codes - An overview</td>
</tr>
<tr>
<td>87/05</td>
<td>R. Kuiper</td>
<td>Enforcing non-determinism via linear time temporal logic specification.</td>
</tr>
<tr>
<td>87/06</td>
<td>R. Koymans</td>
<td>Temporele logica specificatie van message passing en real-time systemen (in Dutch).</td>
</tr>
<tr>
<td>87/07</td>
<td>R. Koymans</td>
<td>Specifying message passing and real-time systems with real-time temporal logic.</td>
</tr>
<tr>
<td>87/08</td>
<td>H.M.J.L. Schols</td>
<td>The maximum number of states after projection.</td>
</tr>
<tr>
<td>87/10</td>
<td>T. Verhoeff</td>
<td>Three families of maximally nondeterministic automata.</td>
</tr>
<tr>
<td>87/11</td>
<td>P. Lemmens</td>
<td>Eldorado ins and outs. Specifications of a data base management toolkit according to the functional model.</td>
</tr>
<tr>
<td>87/12</td>
<td>K.M. van Hee and A. Lapinski</td>
<td>OR and AI approaches to decision support systems.</td>
</tr>
<tr>
<td>87/13</td>
<td>J.C.S.P. van der Woude</td>
<td>Playing with patterns, searching for strings.</td>
</tr>
<tr>
<td>87/14</td>
<td>J. Hooman</td>
<td>A compositional proof system for an occam-like real-time language</td>
</tr>
<tr>
<td>Year</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>87/15</td>
<td>C. Huizing, R. Gerth, W.P. de Roever</td>
<td>A compositional semantics for statecharts</td>
</tr>
<tr>
<td>87/16</td>
<td>H.M.M. ten Eikelder, J.C.F. Wilmont</td>
<td>Normal forms for a class of formulas</td>
</tr>
<tr>
<td>87/17</td>
<td>K.M. van Hee, G.J.Houben, J.L.G. Dietz</td>
<td>Modelling of discrete dynamic systems framework and examples</td>
</tr>
<tr>
<td>87/18</td>
<td>C.W.A.M. van Overveld</td>
<td>An integer algorithm for rendering curved surfaces</td>
</tr>
<tr>
<td>87/19</td>
<td>A.J. Seebregts</td>
<td>Optimalisering van file allocatie in gedistribueerde database systemen</td>
</tr>
<tr>
<td>87/20</td>
<td>G.J. Houben, J. Paredaens</td>
<td>The $R^2$-Algebra: An extension of an algebra for nested relations</td>
</tr>
<tr>
<td>87/21</td>
<td>R. Gerth, M. Codish, Y. Lichtenstein, E. Shapiro</td>
<td>Fully abstract denotational semantics for concurrent PROLOG</td>
</tr>
<tr>
<td>88/01</td>
<td>T. Verhoeff</td>
<td>A Parallel Program That Generates the Möbius Sequence</td>
</tr>
<tr>
<td>88/02</td>
<td>K.M. van Hee, G.J. Houben, L.J. Somers, M. Voorhoeve</td>
<td>Executable Specification for Information Systems</td>
</tr>
<tr>
<td>88/03</td>
<td>T. Verhoeff</td>
<td>Settling a Question about Pythagorean Triples</td>
</tr>
<tr>
<td>88/04</td>
<td>G.J. Houben, J. Paredaens, D. Tahon</td>
<td>The Nested Relational Algebra: A Tool to handle Structured Information</td>
</tr>
<tr>
<td>88/05</td>
<td>K.M. van Hee, G.J. Houben, L.J. Somers, M. Voorhoeve</td>
<td>Executable Specifications for Information Systems</td>
</tr>
<tr>
<td>88/06</td>
<td>H.M.J.L. Schols</td>
<td>Notes on Delay-Insensitive Communication</td>
</tr>
<tr>
<td>88/07</td>
<td>C. Huizing, R. Gerth, W.P. de Roever</td>
<td>Modelling Statecharts behaviour in a fully abstract way</td>
</tr>
<tr>
<td>88/08</td>
<td>K.M. van Hee, G.J. Houben, L.J. Somers, M. Voorhoeve</td>
<td>A Formal model for System Specification</td>
</tr>
<tr>
<td>88/09</td>
<td>A.T.M. Aerts, K.M. van Hee</td>
<td>A Tutorial for Data Modelling</td>
</tr>
<tr>
<td>Year</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>88/10</td>
<td>J.C. Ebergen</td>
<td>A Formal Approach to Designing Delay Insensitive Circuits</td>
</tr>
<tr>
<td>88/11</td>
<td>G.J. Houben</td>
<td>A graphical interface formalism: specifying nested relational databases</td>
</tr>
<tr>
<td></td>
<td>J.Paredaens</td>
<td></td>
</tr>
<tr>
<td>88/12</td>
<td>A.E. Eiben</td>
<td>Abstract theory of planning</td>
</tr>
<tr>
<td>88/13</td>
<td>A. Bijlsma</td>
<td>A unified approach to sequences, bags, and trees</td>
</tr>
</tbody>
</table>