LECTURE 14: BAGS (MULTISETS)

Software Engineering

Mike Wooldridge

1 Bags

- We have seen that sets are unordered collections of items, which do not contain duplicates.
- A sequence is an ordered collection of items, that *may* contain duplicates.
- A *bag* is an *unordered* collection of items that *may* contain duplicates.
- Bags are sometimes called *multisets*.

	Ordered?	Duplicates?
Set	N	N
Sequence	Y	Y
Bag	N	Y

- **Definition:** The set of all bags over type T is given by the expression

  $\text{bag}\, T.$

- Like sets and sequences, bags may be enumerated, by listing theeir contents between *Strachey brackets*: $[\![\ ]\!]$.
- EXAMPLE. Suppose

  $B : \text{bag}\, \mathbb{N}$

  then

  $B == [\![1, 1, 2, 3]\!]$

  assigns to B the bag containing the value 1 twice, the value 2 once, and the value 3 once.

  Note that this is *not* the same as the set

  $\{1, 2, 3\}.$

  However,

  $[\![1, 1, 2, 3]\!] = [\![1, 2, 3, 1]\!].$

2 Bag Membership and Sub-bags

- The 'equivalent' of the set membership predicate $\in$ is 'in'.

  (This is sometime written '   '.)

- **Definition:** If

  $B : \text{bag}\, T$
  $x : T$

  then the predicate

  $x \text{ in } B$

  is true iff x appears in B at least once.

- The 'equivalent' of the subset predicate $\subset$ is $\sqsubseteq$.
- **Definition:** If

  $B_1, B_2 : \text{bag}\, T$

  then the predicate

  $B_1 \sqsubseteq B_2$

  is true iff each element that occurs in B_1 occurs in B_1 no more often than it occurs in B_2.

- Summary:

a in b	a is a member of bag b
$b \sqsubseteq c$	b is a sub-bag of c

- EXAMPLES.

$$jan \text{ in } [\![mar, mar, feb]\!]$$
$$\neg(apr \text{ in } [\![mar, mar, feb]\!])$$
$$[\![jan, feb]\!] \sqsubseteq [\![jan, mar, feb, apr]\!]$$
$$[\![jan, feb]\!] \sqsubseteq [\![jan, feb]\!]$$

- Some theorems about bag membership and sub-bags.

$$[\![\,]\!] \sqsubseteq B$$
$$B \sqsubseteq C \land C \sqsubseteq B \Leftrightarrow B = C$$
$$B \sqsubseteq C \land C \sqsubseteq D \Rightarrow B \sqsubseteq D$$

Lecture 14 Software Engineering

3 Counting Bags

- Suppose we want to know how many times a value x occurs in bag B. We use #:

$$\_ \# \_ : \text{bag } T \times T \to \mathbb{N}$$

- EXAMPLE. If

$$storms == [\![jan, jan, feb, dec]\!]$$

then

$$storms \# jan = 2$$
$$storms \# dec = 1$$
$$storms \# apr = 0$$

- **Definition:** If

$B : \text{bag } T$
$x : T$

then the number of times x occurs in B (a natural number) is given by the expression

$$B \# x.$$

Lecture 14 Software Engineering

4 Scaling Bags

- Another common operation we want to do is *scale* bags; that is, we want to 'multiply' their contents. We do this using the bag scaling operator: $\otimes$.
- EXAMPLE. Let

$$storms == [\![jan, jan, feb]\!]$$

then

$$2 \otimes storms = [\![jan, jan, jan, jan, feb, feb]\!].$$

- **Definition:** If

$B : \text{bag } T$
$n : \mathbb{N}$

then

$$n \otimes B$$

is a bag which contains the same elements as B, except that every element that occurs m times in B occurs $n * m$ times in $n \otimes B$.

Lecture 14 Software Engineering

- Some theorems about scaling...

$$n \otimes [\![\,]\!] = [\![\,]\!]$$
$$0 \otimes B = [\![\,]\!]$$
$$1 \otimes B = B$$
$$(n * m) \otimes B = n \otimes (m \otimes B)$$

5 Bag Union

- Just as there is a set union operator, so there is a bag union operator.
- EXAMPLE. Let

  $$storms == [\![jan, jan, feb]\!]$$

  then

  $$\begin{aligned} storms \uplus [\![mar]\!] &= [\![jan, jan, feb, mar]\!] \\ storms \uplus [\![jan]\!] &= [\![jan, jan, jan, feb]\!] \end{aligned}$$

- **Definition:** If

  $$B_1, B_2 : \mathrm{bag}\, T$$

  then

  $$B_1 \uplus B_2$$

  is bag that contains just those values that occur in either B_1 or B_2, except that the number of times a value x occurs in $B_1 \uplus B_2$ is equal to $(B_1 \# x) + (B_2 \# x)$.
- There is a *bag difference* operator, $\uminus$ ...

6 Making Bags out of Sequences

- One last thing we often want to do is to make a bag out of a sequence, by counting up all number of times in a sequence. We do this using *items*.

  EXAMPLE.

  $$\begin{aligned} items\langle a, b, a, b, c\rangle &= [\![a, a, b, b, c]\!] \\ items\langle a, c, d, a, a\rangle &= [\![a, a, a, c, d]\!] \end{aligned}$$

- **Definition:** If

  $$\sigma : \mathrm{seq}\, T$$

  then $items(\sigma)$ is a bag over T such that a value x occurs in $items(\sigma)$ exactly as many times as it appears in σ.

7 A Model for Bags

- In the previous lecture, we saw that sequences are defined in terms of functions.

  $$\langle a, b, c\rangle = \{1 \mapsto a, 2 \mapsto b, 3 \mapsto c\}.$$

  Bags are defined in a similar way:

  $$\mathrm{bag}\, T == T \nrightarrow \mathbb{N}_1$$

- So the bag

  $$[\![jan, feb, jan]\!]$$

  is really the function

  $$\{jan \mapsto 2, feb \mapsto 1\}.$$

- So we can use all the function manipuating operations to manipulate bags.
- In particular:

  $$\mathrm{dom}[\![a_1, \ldots, a_n]\!] = \{a_1, \ldots, a_n\}$$

  and so

  $$\mathrm{dom}[\![jan, jan, feb]\!] = \{jan, feb\}.$$

- Taking the range of a bag is not generally as useful.

- QUESTION: If bags are defined in this way, then how do we define all the operations on them?
- The difficult one is #; given this, the others are all more or less easy...
- First, $\otimes$:

  $$a \mapsto (n * m) \in (B \otimes m) \Leftrightarrow a \mapsto n \in B$$

- Now, 'in':

  $$x \text{ in } B \Leftrightarrow (B \# x) > 0$$

- The $\sqsubseteq$ predicate is a bit more complicated.

  $$\begin{aligned} &\forall B_1, B_2 : \mathrm{bag}\, T \bullet \\ &\quad B_1 \sqsubseteq B_2 \Leftrightarrow \\ &\qquad \forall x : T \bullet B_1 \# x \leq B_2 \# x \end{aligned}$$