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.

<table>
<thead>
<tr>
<th></th>
<th>Ordered?</th>
<th>Duplicates?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Sequence</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Bag</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>
• **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 their 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 $\subseteq$.

- **Definition:** If
  
  $B_1, B_2 : \text{bag } T$
  
  then the predicate
  
  $B_1 \subseteq B_2$
  
  is true iff each element that occurs in $B_1$ occurs in $B_1$ no more often than it occurs in $B_2$. 
• Summary:

\[
a \text{ in } b \quad \text{a is a member of bag } b
\]

\[
b \subseteq c \quad \text{b is a sub-bag of } c
\]

• EXAMPLES.

\[\text{jan in } [\text{mar, mar, feb}]\]

\[\neg (\text{apr in } [\text{mar, mar, feb}])\]

\[[\text{jan, feb}] \subseteq [\text{jan, mar, feb, apr}]\]

\[[\text{jan, feb}] \subseteq [\text{jan, feb}]\]

• Some theorems about bag membership and sub-bags.

\[
[\ ] \subseteq B
\]

\[
B \subseteq C \land C \subseteq B \iff B = C
\]

\[
B \subseteq C \land C \subseteq D \Rightarrow B \subseteq D
\]
3 Counting Bags

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

\[
\_ \# \_ : \text{bag} T \times T \rightarrow \mathbb{N}
\]

• EXAMPLE. If

\[
\text{storms} == [\text{jan, jan, feb, dec}]
\]

then

\[
\text{storms} \# \text{jan} = 2
\]

\[
\text{storms} \# \text{dec} = 1
\]

\[
\text{storms} \# \text{apr} = 0
\]

• 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

\[
B \# x.
\]
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

\[
\text{storms} == [\text{jan}, \text{jan}, \text{feb}]
\]

then

\[
2 \otimes \text{storms} = [\text{jan}, \text{jan}, \text{jan}, \text{feb}, \text{feb}].
\]

• 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 $n \times m$ times in $n \otimes B$. 
Some theorems about scaling...

\[ n \otimes [ ] = [ ] \]
\[ 0 \otimes B = [ ] \]
\[ 1 \otimes B = B \]
\[ (n \ast m) \otimes B = n \otimes (m \otimes B) \]
5 Bag Union

• Just as there is a set union operator, so there is a bag union operator.

• EXAMPLE. Let

\[ \text{storms} == [\text{jan, jan, feb}] \]

then

\[ \text{storms} \cup [\text{mar}] = [\text{jan, jan, feb, mar}] \]
\[ \text{storms} \cup [\text{jan}] = [\text{jan, jan, jan, feb}] \]

• Definition: If

\[ B_1, B_2 : \text{bag } T \]

then

\[ B_1 \cup B_2 \]

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 \( B_1 \cup B_2 \) is equal to \((B_1 \#x) + (B_2 \#x)\).

• There is a bag difference operator, \( \cup \cdots \)
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 \textit{items}.

\textbf{EXAMPLE.}

\begin{align*}
\text{items}\langle a, b, a, b, c \rangle &= [a, a, b, b, c] \\
\text{items}\langle a, c, d, a, a \rangle &= [a, a, a, c, d]
\end{align*}

• \textbf{Definition: If}

\[
\sigma : \text{seq } T
\]

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

\[ \text{bag } T =\mapsto \mathbb{N}_1 \]

- So the bag

\[ [\text{jan, feb, jan}] \]

is really the function

\[ \{\text{jan} \mapsto 2, \text{feb} \mapsto 1\}. \]

- So we can use all the function manipulating operations to manipulate bags.

- In particular:

\[ \text{dom}[a_1, \ldots, a_n] = \{a_1, \ldots, a_n\} \]

and so

\[ \text{dom}[\text{jan, jan, feb}] = \{\text{jan, feb}\}. \]

- 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, \(\otimes\):

\[ a \mapsto (n \ast m) \in (B \otimes m) \iff a \mapsto n \in B \]

• Now, ‘in’:

\[ x \text{ in } B \iff (B\#x) > 0 \]

• The \(\subseteq\) predicate is a bit more complicated.

\[
\forall B_1, B_2 : \text{bag } T \bullet \\
B_1 \subseteq B_2 \iff \\
\forall x : T \bullet B_1 \#x \leq B_2 \#x
\]