Monoids

Ross Tate

September 5, 2014

Exercise 1. Given monoids \( A \) and \( B \), give a monoidal structure \( A \& B \) to the set \( A \times B \) such that the projection functions \( \pi_A \) and \( \pi_B \) are monoid homomorphisms from \( A \& B \) to \( A \) and \( B \) respectively.

Proof. Define \( \langle a_1, b_1 \rangle \ast \langle a_2, b_2 \rangle \) to be \( \langle a_1 \ast a_2, b_1 \ast b_2 \rangle \). This is associative because \( \ast \) is associative for both \( A \) and \( B \). Define \( e_{A \& B} \) to be \( (e_A, e_B) \). This is an identity because \( e_A \) and \( e_B \) are identities for \( A \) and \( B \) respectively. \( \pi_A \) is a monoid homomorphism since \( \pi_A(\langle a_1 \ast a_2, b_1 \ast b_2 \rangle) = a_1 \ast a_2 \), preserving multiplication, and \( \pi_A((e_A, e_B)) = e_A \), preserving identity. Similarly for \( \pi_B \).

Exercise 2. Determine the monoid “\( \top \)" with the property that for every monoid \( A \) there is exactly one monoid homomorphism from \( A \) to \( \top \).

Proof. The underlying set is \( \top \), and multiplication and identity are the only functions with their respective signatures. Given two monoid homomorphisms from some monoid \( A \) to \( \top \), they must both map everything to the unique inhabitant of \( \top \), making them equal.

Exercise 3. Determine the monoid “\( 0 \)" with the property that for every monoid \( A \) there is exactly one monoid homomorphism from \( 0 \) to \( A \).

Proof. The underlying set is \( \top \), and multiplication and identity are the only functions with their respective signatures. Given two monoid homomorphisms from \( 0 \) to some monoid \( A \), their only input is the identity of \( \top \) and so being monoid homomorphisms they must both map this only input to \( e_A \), making them equal.

Definition. Given monoids \( A \) and \( B \), define the equivalence relation \( \approx \) on \( L(A \times B) \) to be the least equivalence relation such that:

1. \( \forall m_1, m_1', m_2, m_2' : L(A \times B). m_1 \approx m_1' \land m_2 \approx m_2' \implies m_1 + m_2 \approx m_1' + m_2' \)
2. \( \forall b : B. [(e_A, b)] \approx [ ] \)
3. \( \forall a_1, a_2 : A, b : B. [\langle a_1, b \rangle, \langle a_2, b \rangle] \approx [\langle a_1 \ast a_2, b \rangle] \)
4. \( \forall a : A. [(a, e_B)] \approx [ ] \)
5. \( \forall a : A, b_1, b_2 : B. [\langle a, b_1 \rangle, \langle a, b_2 \rangle] \approx [\langle a, b_1 \ast b_2 \rangle] \)

We use requirement 1 to impose a monoidal structure \( A \otimes B \) on the quotient set \( \frac{L(A \times B)}{\approx} \):

<table>
<thead>
<tr>
<th>Operator</th>
<th>( \mathbf{++} )</th>
<th>( = \lambda q_1, q_2. \text{select } \tilde{m}_1 \text{ from } q_1 \text{ in } (\text{select } \tilde{m}_2 \text{ from } q_2 \text{ in } \tilde{m}_1 \mathbf{++} \tilde{m}_2 \text{ using } \cdot) \text{ using } . )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associativity</td>
<td>Follows from associativity of ( \mathbf{++} ) and the fact that quotienting only makes things more equal</td>
<td></td>
</tr>
<tr>
<td>Identity Element</td>
<td>( = \frac{[]}{} )</td>
<td></td>
</tr>
<tr>
<td>Identity</td>
<td>Follows from identity of [ ] and the fact that quotienting only makes things more equal</td>
<td></td>
</tr>
</tbody>
</table>

Exercise 4. Show that, for any monoid \( C \), there is a bijection between the set of multilinear homomorphisms from \( A \) and \( B \) to \( C \) and the set of monoid homomorphisms from \( A \otimes B \) to \( C \).

Proof. Given a function \( f : A \times B \to C \) that is a multilinear homomorphism from \( A \) and \( B \) to \( C \), define \( \hat{f} : L(A \times B) \to C \) to be \( \lambda \tilde{m}. \text{map}_f \tilde{m} \) where \( \text{map}_f \) is the function that takes a list and produces a new list by applying \( f \) to each element. \( \hat{f} \) is a monoid homomorphism:
\[
\hat{f}(\vec{m}_1 \leftrightarrow \vec{m}_2) = \Pi \text{map}_f(\vec{m}_1 \leftrightarrow \vec{m}_2) = \Pi(\text{map}_f \vec{m}_1 \leftrightarrow \text{map}_f \vec{m}_2) = (\Pi \text{map}_f \vec{m}_1) \ast (\Pi \text{map}_f \vec{m}_2) = \hat{f}(\vec{m}_1) \ast \hat{f}(\vec{m}_2)
\]

\[
\hat{f}(\vec{m}) = \Pi \text{map}_f[\vec{m}] = \Pi[\vec{m}] = e_C
\]

\(\hat{f}\) has the property that it maps related lists to equal elements (skipping the additional rules for equivalence relations below):

1. Given \(\vec{m}_1, \vec{m}_1', \vec{m}_2, \vec{m}_2' : L(A \times B)\) such that \(\vec{m}_1 \approx \vec{m}_1'\) and \(\vec{m}_2 \approx \vec{m}_2'\) hold, by induction on the proof of \(\approx\) we can assume \(\hat{f}(\vec{m}_1) = \hat{f}(\vec{m}_1')\) and \(\hat{f}(\vec{m}_2) = \hat{f}(\vec{m}_2')\). Then \(\hat{f}(\vec{m}_1 \leftrightarrow \vec{m}_2) = \hat{f}(\vec{m}_1') \ast \hat{f}(\vec{m}_2') = \hat{f}(\vec{m}_1' \leftrightarrow \vec{m}_2')\)

2. Given \(b : B\), \(\hat{f}([e_A, b]) = \Pi \text{map}_f([e_A, b]) = \Pi f(e_A, b) = f(e_A, b) = e_C\) = \(\Pi[\vec{m}] = \Pi \text{map}_f[\vec{m}] = \hat{f}([\vec{m}])\)

3. Given \(a_1, a_2 : A\) and \(b : B\), \(\hat{f}([a_1, b], [a_2, b]) = \Pi \text{map}_f([a_1, b], [a_2, b]) = \Pi f(a_1, b) \ast f(a_2, b) = f(a_1 \ast a_2, b) = \Pi f(a_1 \ast a_2, b) = \Pi \text{map}_f([a_1 \ast a_2, b]) = ([a_1 \ast a_2, b])\)

4. Given \(a : A\), \(\hat{f}([a, e_B]) = \Pi \text{map}_f([a, e_B]) = \Pi f(a, e_B) = f(a, e_B) = e_C\) = \(\Pi[\vec{m}] = \Pi \text{map}_f[\vec{m}] = \hat{f}([\vec{m}])\)

5. Given \(a : A\) and \(b_1, b_2 : B\), \(\hat{f}([a, b_1], [a, b_2]) = \Pi \text{map}_f([a, b_1], [a, b_2]) = \Pi f(a, b_1) \ast f(a, b_2) = f(a, b_1 \ast b_2) = \Pi f(a, b_1 \ast b_2) = \Pi \text{map}_f([a, b_1 \ast b_2]) = ([a, b_1 \ast b_2])\)

Consequently, we can define \(\tilde{f} : \frac{L(A \times B)}{\approx} \rightarrow C\) to be \(\lambda q. \text{select } \vec{m} \text{ from } q \text{ in } \Pi \text{map}_f \vec{m}\) using (proof above). This is a monoid homomorphism because \(\tilde{f}\) is a monoid homomorphism.

In the other direction, given a function \(g : \frac{L(A \times B)}{\approx} \rightarrow C\) that is a monoid homomorphism from \(A \otimes B\) to \(C\), define \(\tilde{g} : A \times B \rightarrow C\) to be \(\lambda (a, b). g([\langle a, b \rangle])\). \(\tilde{g}\) is a multilinear monoid homomorphism from \(A\) and \(B\) to \(C\) since related lists are in equal equivalence classes and \(g\) is a monoid homomorphism:

- Given \(b : B\), \(\tilde{g}(e_A, b) = g([\langle e_A, b \rangle]) = g([\frac{1}{e_A}]) = e_C\)
- Given \(a_1, a_2 : A\) and \(b : B\), \(\tilde{g}(a_1 \ast a_2, b) = g([\langle a_1 \ast a_2, b \rangle]) = g([\langle a_1, b \rangle \approx \langle a_2, b \rangle]) = g([\langle a_1, b \rangle]) \ast g([\langle a_2, b \rangle]) = \tilde{g}(a_1, b) \ast \tilde{g}(a_2, b)\)
- Given \(a : A\), \(\tilde{g}(a, e_B) = g([\langle a, e_B \rangle]) = g([\frac{1}{a}]) = e_C\)
- Given \(a : A\) and \(b_1, b_2 : B\), \(\tilde{g}(a, b_1 \ast b_2) = g([\langle a, b_1 \ast b_2 \rangle]) = g([\langle a, b_1 \rangle \approx \langle a, b_2 \rangle]) = g([\langle a, b_1 \rangle]) \ast g([\langle a, b_2 \rangle]) = \tilde{g}(a, b_1) \ast \tilde{g}(a, b_2)\)

Given a function \(f : A \times B \rightarrow C\) that is a multilinear homomorphism from \(A\) and \(B\) to \(C\), we have the following equality for all \(a : A\) and \(b : B\):

\[
\tilde{f}(a, b) = \hat{f}([\langle a, b \rangle]) = \text{select } \vec{m} \text{ from } [\langle a, b \rangle] \approx \text{ in } \Pi \text{map}_f \vec{m} \text{ using } = \Pi \text{map}_f([a, b]) = \Pi f(a, b) = f(a, b)
\]

In the other direction, given a function \(g : \frac{L(A \times B)}{\approx} \rightarrow C\) that is a monoid homomorphism from \(A \otimes B\) to \(C\), we have the following equality for all \(q : \frac{L(A \times B)}{\approx}\):

\[
g(q) = \text{select } \vec{m} \text{ from } q \text{ in } g([\frac{\vec{m}}{\approx}]\) using .
\[
= \text{select } \Sigma_\lambda([a_1, b_1]) \text{ from } q \text{ in } g([\frac{\Sigma_\lambda([a_1, b_1])}{\approx}]\) using .
\[
= \text{select } \Sigma_\lambda([a_1, b_1]) \text{ from } q \text{ in } \Pi \Sigma_\lambda g([\frac{[a_1, b_1]}{\approx}]\) using .
\[
= \text{select } \Sigma_\lambda([a_1, b_1]) \text{ from } q \text{ in } \Pi \text{map}_\lambda([a_1, b_1]) \Sigma_\lambda([a_1, b_1]) \) using .
\[
= \text{select } \vec{m} \text{ from } q \text{ in } \Pi \text{map}_\lambda([a_1, b_1]) \vec{m} \) using .
\[
= \text{select } \vec{m} \text{ from } q \text{ in } \Pi \text{map}_\lambda([a_1, b_1]) \vec{m} \) using .
\[
= g(q)
\]