MAT 305: Lab #8

April 15, 2016

Background

Definition. If a set S and an operation $\otimes$ satisfy the **closure**, **associative**, **identity**, and **inverse** properties, then we call S a **group** under $\otimes$. These properties are defined in the following way:

- *closure:* $x \otimes y \in S$ for all $x, y \in S$;
- *associative:* $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ for all $x, y, z \in S$;
- *identity:* we can find $\iota \in S$ such that $x \otimes \iota = x$ and $\iota \otimes x = x$ for any $x \in S$;
- *inverse:* for any $x \in S$, we can find $y \in S$ such that $x \otimes y = y \otimes x = \iota$.

Example. The integers $\mathbb{Z}$ form a group under addition, because

- adding any two integers gives you an integer ($x + y \in \mathbb{Z}$ for all $x, y \in \mathbb{Z}$);
- addition of integers is associative;
- there is an additive identity ($x + 0 = x$ and $0 + x = x$ for all $x \in \mathbb{Z}$); and
- every integer x has an additive inverse *that is also an integer* ($x + (-x) = (-x) + x = 0$).

Example. The integers $\mathbb{Z}$ *do not* form a group under multiplication, for two reasons:

- 0 has no multiplicative inverse 0^{-1}; and
- the other integers a have multiplicative inverses $1/a$, but *most are not integers*. A group only satisfies the inverse property if it contains the inverses of each element.

In this lab you will use pseudocode to write code to test whether a finite set is a group under *multiplication*. You will then test it on three sets, two of which succeed, and one of which does not. A complication in this project is that the function has to depend on the operation, so you can't just write a function for one operation, only.

Pseudocode

Closure

We must check every pair $x, y \in S$. We can test whether this is true for "every" element of a finite set using definite loops.

algorithm *is_closed*
inputs
  S, a finite set
outputs
  `true` if S is closed under multiplication; `false` otherwise
do
  for $s \in S$
    for $t \in S$
      if $st \notin S$
        print "fails closure for", s, t
        return `false`
  return `true`

Associative

We must check every triplet $x, y, z \in S$, requiring definite loops. The pseudocode is an exercise.

Identity

We can test whether "we can find" an identity using a special variable called a **flag** with Boolean value (sometimes called a **signal**). We adjust the flag's value depending on whether a candidate continues to satisfy a known property. When the loop ends, the flag indicates whether we're done (i.e., whether we've found an identity). The quantifiers' structure ("we can find… for any…") requires the pseudocode to presume an identity exists until proved otherwise.

algorithm *find_identity*
inputs
  S, a finite set
outputs
  an identity, if it can find it; otherwise, $\emptyset$
do
  for $s \in S$
    let *maybe_identity* = `true`
    for $t \in S$
      if $st \neq t$ **or** $ts \neq t$
        let *maybe_identity* = `false`
    if *maybe_identity* = `true`
      return s
  print "no identity"
  return $\emptyset$

Inverse

We are looking for an inverse for each element. Here, again, we use a flag a **flag**, as the logic requires us to find an inverse. Unlike the previous pseudocode, we presume an inverse *does not* exist until proved otherwise; this is because the order of the quantifiers is switched ("for any... we can find..." instead of "we can find... for any..."). This pseudocode also requires that we identify the set's identity in the input.

algorithm *has_inverses*
inputs
S, a finite set
ι, an identity of S under multiplication
outputs
`true` if every element of S has a multiplicative inverse; `false` otherwise
do
for $s \in S$
let *found_inverse* = `false`
for $t \in S$
if $st = \iota$ **and** $ts = \iota$
let *found_inverse* = `true`
if *found_inverse* = `false`
print "no inverse for", s
return `false`
return `true`

Putting them together

This pseudocode tests whether a set is a group under an operation by invoking all four algorithms defined above.

algorithm *is_a_group*
inputs
S, a finite set
outputs
`true` if S is a group under multiplication; `false` otherwise
do
if *is_closed*(S) **and** *is_associative*(S)
let ι = *find_identity*(S)
if $\iota \neq \emptyset$ **and** *has_inverses*(S, ι)
return `true`
return `false`

Your tasks

Use $\LaTeX$ in your Sage worksheets wherever appropriate. **Two of the sets in 3–5 are groups; one is not.**

1. Study the pseducode for closure, and write pseudocode for an algorithm named *is_associative* that tests whether a set S is associative under multiplication. You essentially modify the pseudocode for *is_closed* with a third loop, and change the condition for the **if** appropriately.

2. Write Sage code for each of the five algorithms defined above in pseudocode. You will test them on the following sets.

3. Define a ring R to be $\mathbb{Z}_{101}$, the finite ring of 101 elements. (You will want to revisit Lab #2 if you forgot how to do this.) Let $S = \{1, 2, \ldots, 100\} \subsetneq R$; that is, S should include every element of R *except* 0. Be sure to define S using elements of R, and not plain integers. (Again, you will want to revisit Lab #2 if you forgot how to do this.) Test your Sage code on S; is S a group under multiplication? If not, which property fails?

4. Redefine the ring R to be $\mathbb{Z}_{102}$, the finite ring of 102 elements. Let $S = \{1, 2, \ldots, 101\} \subsetneq R$; that is, S should include every element of R *except* 0. Be sure to define S using elements of R, and not plain integers. Test your Sage code on S; is S a group under multiplication? If not, which property fails?

5. Define the matrices
$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{i} = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$$
$$\mathbf{j} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \qquad \mathbf{k} = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$$
and the set
$$Q = \{I_2, -I_2, \mathbf{i}, -\mathbf{i}, \mathbf{j}, -\mathbf{j}, \mathbf{k}, -\mathbf{k}\}.$$
Test your Sage code on Q; is Q a group under multiplication? If not, which property fails?

   Remark. This set is sometimes called the **set of quaternions**.

6. Using the matrices of problem #4, define the set
$$S = \{I_2, -I_2, \mathbf{j}, -\mathbf{j}\}.$$
   (a) You've probably noticed that $S \subseteq Q$. Is S also a group? If so, we call S a **subgroup** of Q. If not, which property fails?

   (b) The set S actually consists of matrices of the form A from Lab #6, Problem #1. Indicate in a text cell the correct value of a for each matrix.