Answers to Selected Chapter 2 Exercises

Note: Student answers, and your answers, to any of these questions may vary from the answers here. That's okay, because one looks at these questions with a particular environment in mind (in other words, we make assumptions). The key issue is whether the answer the student, or you, give can be justified. The acceptability of the answer depends upon the quality of the justification.

1.

a.

	alicerc	*bobrc*	*cyndyrc*
Alice	ox	r	
Bob	r	ox	
Cyndy	r	rw	orwx

a.

	alicerc	*bobrc*	*cyndyrc*
Alice	ox	r	r
Bob		ox	
Cyndy	r	rw	orwx

2. One reason might be that Miller and Baldwin's work was on databases, and their focus was on the conditions intrinsic to the data and environment that would constrain access. Their environment did not require access to data constrained upon subjects; they used groups and roles. The rules in the example, for instance, all grant access based on *subject.group* and *subject.role*. Were the group and role facilities not present, or the numbers of groups and roles small, the proposed change would have simplified the controls.
3. not answered here
4. In these answers, *r, w, x, a, l, m,* and *o* represent the read, write, execute, append, list, modify, and own rights, respectively.
   a. The key observation is that anyone can delete the rights, not *p*. So:

```
command delete_all_rights(p, q, s)
      delete r in A[q, s];
      delete w in A[q, s];
      delete x in A[q, s];
      delete a in A[q, s];
      delete l in A[q, s];
      delete m in A[q, s];
      delete o in A[q, s];
```

```
end;
```

b. Here, we must condition the command on the presence of rights that *p* has over *s*:

```
command delete_all_rights(p, q, s)
     if m ∈ A[p, s] then
          delete r in A[q, s];
          delete w in A[q, s];
          delete x in A[q, s];
          delete a in A[q, s];
          delete l in A[q, s];
          delete m in A[q, s];
          delete o in A[q, s];
end;
```

c. This one is trickier. We cannot test for the *absense* of rights directly, so we build a surrogate object *z*. The idea is that *A*[*q*, *z*] will contain the right *o* if *q* does not have *o* rights over *s*, and will contain the right *m* if *q* has *o* rights over *s*. So, we need some auxiliary commands:

```
command make_aux_object(q, z)
     create object z;
     enter o in A[q, z];
end;
command fixup_aux_object(q, s, z)
     if o ∈ A[q, s] then
          delete o in A[q, z];
          enter m in A[q, z];
end;
```

Now we write the command to delete the rights if *p* has *m* rights over *s* and *q* does *not* have *o* rights over *s*. The last econdition is logically equivalent to *q* having *o* rights over *z*:

```
command prelim_delete_all_rights(p, q, s, z)
     if m ∈ A[p, s] and o ∈ A[q, z] then
          delete r in A[q, s];
          delete w in A[q, s];
          delete x in A[q, s];
          delete a in A[q, s];
          delete l in A[q, s];
          delete m in A[q, s];
          delete o in A[q, s];
end;
```

Finally, we create the actual delete command:

```
command delete_all_rights(p, q, s, z)
     make_aux_object(q, z);
     fixup_aux_object(q, s, z);
     prelim_delete_all_rights(p, q, s, z);
     destroy object z;
end;
```

5. The copy flag defines an additional set of rights. We treat *r* and *rc* as read rights, but the first may not be copied whereas the second may be.

a. We build this command from a set of smaller commands. First, we define *copy_r*(*p*, *q*, *s*) as:

```
command copy_right(r, p, q, s)
if r ∈ A[p, s] then
     enter r in A[q, s]
end
```

Then the required command is:

```
command copy_all_rights(p, q, s)
     copy_right(r, p, q, s)
     copy_right(rc, p, q, s)
     copy_right(w, p, q, s)
     copy_right(wc, p, q, s)
     copy_right(x, p, q, s)
     copy_right(xc, p, q, s)
     copy_right(a, p, q, s)
     copy_right(ac, p, q, s)
     copy_right(l, p, q, s)
     copy_right(lc, p, q, s)
     copy_right(m, p, q, s)
     copy_right(mc, p, q, s)
     copy_right(o, p, q, s)
     copy_right(oc, p, q, s)
end
```

b. We can use the same *copy_right* as in part a, and simply copy those rights with the copy flag set. *copy_all_rights* becomes:

```
command copy_all_rights(p, q, s)
     copy_right(rc, p, q, s)
     copy_right(wc, p, q, s)
     copy_right(xc, p, q, s)
     copy_right(ac, p, q, s)
     copy_right(lc, p, q, s)
     copy_right(mc, p, q, s)
     copy_right(oc, p, q, s)
end
```

c. Copying the copy flag would allow the recipient of the right to propagate the rights to others. The recipient would have the same abilities that the originator has.

6.

a. If any subject can give any rights to any other subject, then the maximal set of rights subjects within the system can acquire is all rights over all other entities in the system. If some rights can be given under specific conditions, the maximal set of rights depends upon the nature of the conditions themselves.

b. Applying attenuation of privileges to access rights only would not ameliorate the above situation, because the control rights only control the spread of rights throughout the system. If the owner can grant rights it does not have, it can grant the requisite access rights as desired, thus enabling information to flow about the system.

c. To answer this question, we need a lemma:

Lemma**:** Let q_i have the right $a_{q^{i+1}}$ for $1 \leq i < n$. Then q_1 has as its set of rights the union of the set of rights for all q_i, $1 \leq i \leq n$.

Proof: by induction on n.

Basis: $n = 1$. Immediate.

Induction hypothesis: Let $n = k–1$ and let q_i have the right $a_{q^{i+1}}$ for $1 \leq i < n$. Then q_1 has as its set of rights the union of the set of rights for all q_i, $1 \leq i \leq n$.

Induction step: Let $n = k$. By the induction hyporthesis, q_2 has as its set of rights the union of the set of rights for all q_i, $2 \leq i \leq n$. Now, as q_1 has a_{q^1} rights, it has all rights that q_2 has, plus all that it has itself. Hence it has as its set of rights the union of the set of rights for all q_i, $1 \leq i \leq n$. This completes the induction step and proves the lemma.

Hence from the lemma, we see that q has, or can acquire, any rights that any of its ancestors, or *their* ancestors, have.