Example

Suppose we want to change the value for a key in a hashtable without removing it from the table
– Assume lock guards the whole table

```
synchronized(lock) {
  v1 = table.lookup(k);
  v2 = expensive(v1);
  table.remove(k);
  table.insert(k,v2);
}
```

_Papa Bear’s critical section was too long_

_(table locked during expensive call)_
Example

Suppose we want to change the value for a key in a hashtable without removing it from the table

- Assume `lock` guards the whole table

```
Mama Bear’s critical section was too short
(if another thread updated the entry, we will lose an update)

synchronized (lock) {
    v1 = table.lookup(k);
}
    v2 = expensive(v1);
synchronized (lock) {
    table.remove(k);
    table.insert(k, v2);
}
```
Example

Suppose we want to change the value for a key in a hashtable without removing it from the table
- Assume lock guards the whole table

```
done = false;
while(!done) {
    synchronized(lock) {
        v1 = table.lookup(k);
    }
    v2 = expensive(v1);
    synchronized(lock) {
        if(table.lookup(k)==v1) {
            done = true;
            table.remove(k);
            table.insert(k,v2);
        }
    }
}
```

Baby Bear’s critical section was just right
(if another update occurred, try our update again)