```
public class BackoffLock implements Lock {
  private AtomicBoolean state = new AtomicBoolean(false);
  private static final int MIN_DELAY = ...;
  private static final int MAX_DELAY = ...;
  public void lock() {
    Backoff backoff = new Backoff(MIN_DELAY, MAX_DELAY);
    while (true) {
      while (state.get()) {};
      if (!state.getAndSet(true)) {
        return;
      } else {
        backoff.backoff();
      }
    }
  }
  public void unlock() {
    state.set(false);
  }
  ...
}
```

**FIGURE 7.6** The exponential back-off lock. Whenever the thread fails to acquire a lock that became free, it backs off before retrying.