Table 15.1 The sixteen microstates for a one-dimensional system of $N = 4$ noninteracting spins. The total energy E of each microstate is also shown. If the total energy of the system is $E = -2\mu B$, then there are four accessible microstates (see the fourth column). Hence, in this case the ensemble consists of four systems, each in a different microstate with equal probability.

$4\mu B$	$2\mu B$	0	$-2\mu B$	$-4\mu B$
↓↓↓↓	↓↓↓↑	↓↓↑↑	↓↑↑↑	↑↑↑↑
	↓↓↑↓	↓↑↓↑	↑↓↑↑	
	↓↑↓↓	↓↑↑↓	↑↑↓↑	
	↑↓↓↓	↑↓↓↑	↑↑↑↓	
		↑↓↑↓		
		↑↑↓↓		

15.3 ■ THE DEMON ALGORITHM

We found in Chapter 8 that we can do a time average of a system of many particles with E, V, and N fixed by integrating Newton's equations of motion for each particle and computing the time-averaged value of the physical quantities of interest. How can we do an ensemble average at fixed E, V, and N? And what can we do if there is no equation of motion available? One way would be to enumerate all the accessible microstates and calculate the ensemble average of the desired physical quantities as we did in Table 15.1. This approach is usually not practical because the number of microstates for even a small system is much too many to enumerate. In the spirit of Monte Carlo, we wish to develop a practical method of obtaining a representative sample of the total number of microstates. One possible procedure is to fix N, choose each spin to be up or down at random, and retain the configuration if it has the desired total energy. However, this procedure is very inefficient because most configurations would not have the desired total energy and would have to be discarded.

An efficient Monte Carlo procedure for simulating systems at a given energy was developed by Creutz in the context of lattice gauge theory. Suppose that we add an extra degree of freedom to the original macroscopic system of interest. For historical reasons, this extra degree of freedom is called a *demon*. The demon transfers energy as it attempts to change the dynamical variables of the system. If the desired change lowers the energy of the system, the excess energy is given to the demon. If the desired change raises the energy of the system, the demon gives the required energy to the system if the demon has sufficient energy. The only constraint is that the demon cannot have negative energy.

We first apply the demon algorithm to a one-dimensional classical system of N noninteracting particles of mass m (an ideal gas). The total energy of the system is $E = \sum_i mv_i^2/2$, where v_i is the velocity of particle i. In general, the demon algorithm is summarized by the following steps:

1. Choose a particle at random and make a trial change in its coordinates.
2. Compute ΔE, the change in the energy of the system due to the change.
3. If $\Delta E \leq 0$, the system gives the amount $|\Delta E|$ to the demon, that is, $E_d = E_d - \Delta E$, and the trial configuration is accepted.

4. If $\Delta E > 0$ and the demon has sufficient energy $E_d \geq \Delta E$, then the demon gives the necessary energy to the system, that is, $E_d = E_d - \Delta E$, and the trial configuration is accepted. Otherwise, the trial configuration is rejected and the configuration is not changed.

The above steps are repeated until a representative sample of states is obtained. After a sufficient number of steps, the demon and the system will agree on an average energy for each. The total energy of the system plus the demon remains constant, and because the demon is only one degree of freedom in comparison to the many degrees of freedom of the system, the energy fluctuations of the system will be of order $1/N$, which is very small for $N \gg 1$.

The ideal gas has a trivial dynamics. That is, because the particles do not interact, their velocities do not change. (The positions of the particles change, but the positions are irrelevant because the energy depends only on the velocity of the particles.) So the use of the demon algorithm is equivalent to a fictitious dynamics that lets us sample the microstates of the system. Of course, we do not need to apply the demon algorithm to an ideal gas because all its properties can be calculated analytically. However, it is a good idea to consider a simple example first.

How do we know that the Monte Carlo simulation of the microcanonical ensemble will yield results equivalent to the time-averaged results of molecular dynamics? The assumption that these two types of averages yield equivalent results is called the *quasi-ergodic* hypothesis. Although these two averages have not been proven to be identical in general, they have been found to yield equivalent results in all cases of interest.

`IdealDemon` and `IdealDemonApp` implement the microcanonical Monte Carlo simulation of the ideal classical gas in one dimension. To change a configuration, we choose a particle at random and change its velocity by a random amount. The parameter `mcs`, the number of Monte Carlo steps per particle, plays an important role in Monte Carlo simulations. On the average, the demon attempts to change the velocity of each particle once per Monte Carlo step per particle. We frequently will refer to the number of Monte Carlo steps per particle as the "time," even though this time has no obvious direct relation to a physical time.

Listing 15.1 The demon algorithm for the one-dimensional ideal gas.

```
package org.opensourcephysics.sip.ch15;
public class IdealDemon {
   public double v[];
   public int N;
   public double systemEnergy;
   public double demonEnergy;
   public int mcs = 0; // number of MC moves per particle
   public double systemEnergyAccumulator = 0;
   public double demonEnergyAccumulator = 0;
   public int acceptedMoves = 0;
   public double delta;

   public void initialize() {
      v = new double[N]; // array to hold particle velocities
      double v0 = Math.sqrt(2.0*systemEnergy/N);
      for(int i = 0;i<N;++i) {
         v[i] = v0; // give all particles the same initial velocity
      }
```