Algorithm 12.1 Forward Sampling in a Bayesian network

```
Procedure Forward-Sample (
    B    // Bayesian network over X
)
    Let X_1, ..., X_n be a topological ordering of X
    for i = 1, ..., n
        u_i ← x⟨Pa_{X_i}⟩    // Assignment to Pa_{X_i} in x_1, ..., x_{i-1}
        Sample x_i from P(X_i | u_i)
    return (x_1, ..., x_n)
```

heads, so that we pick the value d^1 for D. Similarly, we sample I from its distribution; say that the result is i^0. Given those, we know the right distribution from which to sample G: $P(G \mid i^0, d^1)$, as defined by G's CPD; we therefore pick G to be g^1 with probability 0.05, g^2 with probability 0.25, and g^3 with probability 0.7. The process continues similarly for S and L. ■

As shown in algorithm 12.1, we sample the nodes in some order consistent with the partial order of the BN, so that by the time we sample a node we have values for all of its parents. We can then sample from the distribution defined by the CPD and by the chosen values for the node's parents. Note that the algorithm requires that we have the ability to sample from the distributions underlying our CPD. Such sampling is straightforward in the discrete case (see box 12.A), but subtler when dealing with continuous measures (see section 14.5.1).

Box 12.A — Skill: Sampling from a Discrete Distribution. *How do we generate a sample from a distribution? For a uniform distribution, we can use any pseudo-random number generator on our machine. Other distributions require more thought, and much work has been devoted in statistics to the problem of sampling from a variety of parametric distributions. Most obviously, consider a multinomial distribution $P(X)$ for $Val(X) = \{x^1, \ldots, x^k\}$, which is defined by parameters $\theta_1, \ldots, \theta_k$. This process can be done quite simply as follows: We generate a sample s uniformly from the interval $[0, 1]$. We then partition the interval into k subintervals: $[0, \theta_1), [\theta_1, \theta_1 + \theta_2), \ldots$; that is, the ith interval is $[\sum_{j=1}^{i-1} \theta_j, \sum_{j=1}^{i} \theta_j)$. If s is in the ith interval, then the sampled value is x^i. We can determine the interval for s using binary search in time $O(\log k)$.*

This approach gives us a general-purpose solution for generating samples from the CPD of any discrete-valued variable: given a parent assignment $\boldsymbol{u}$, we can always generate the full conditional distribution $P(X \mid \boldsymbol{u})$ and sample from it. (Of course, more efficient methods may exist if X has a large value space or a CPD that requires an expensive computation.) As we discuss in section 14.5.1, the problem of sampling from continuous CPDs is considerably more complex.

convergence bound

Using basic *convergence bounds* (see appendix A.2), we know that from a set of particles $\mathcal{D} = \{\xi[1], \ldots, \xi[M]\}$ generated via this sampling process, we can estimate the expectation of