

# Announcement

---

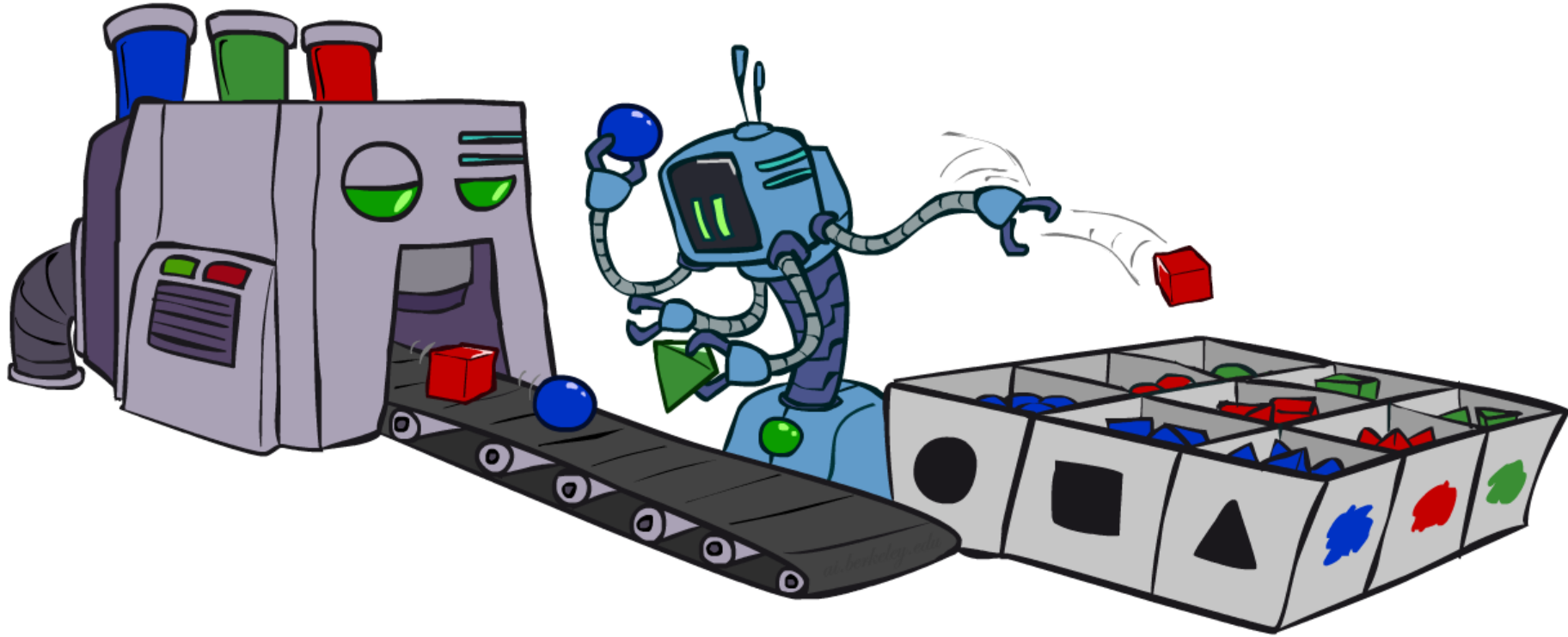
- Homework 3
  - Due: April 7, 11:59pm
- Programming Assignment 3
  - Due: April 12, 11:59pm

# Announcement

---

- Midterm Review @April 12
- Midterm @April 14 (in class)
  - Location: TBA
  - “Introduction” – “Approximate inference on BN”
  - Format
    - Closed-book. You can bring an A4-size cheat sheet and nothing else.
    - Around 5 problems
  - Grade
    - 25% of the total grade

# Bayes Nets: Approximate Inference



AIMA Chapter 14.5, PRML Chapter 11

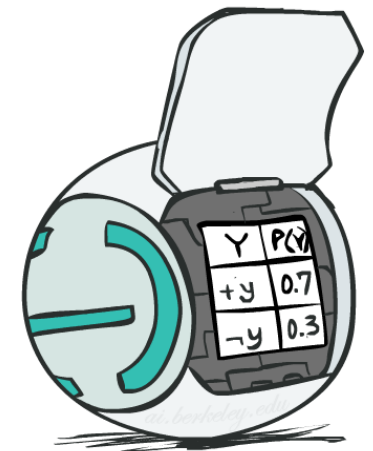
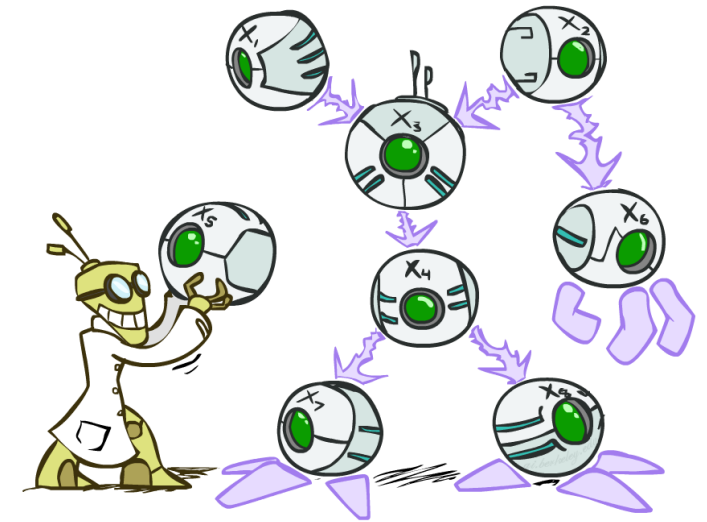
# Bayes' Net Representation

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over  $X$ , one for each combination of parents' values

$$P(X|a_1 \dots a_n)$$

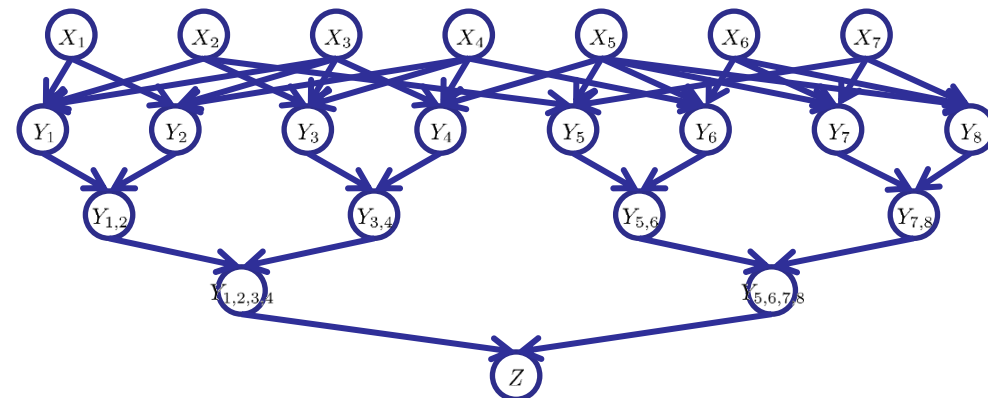
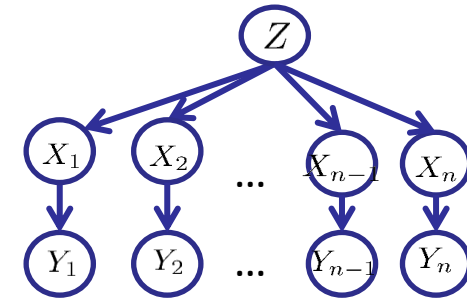
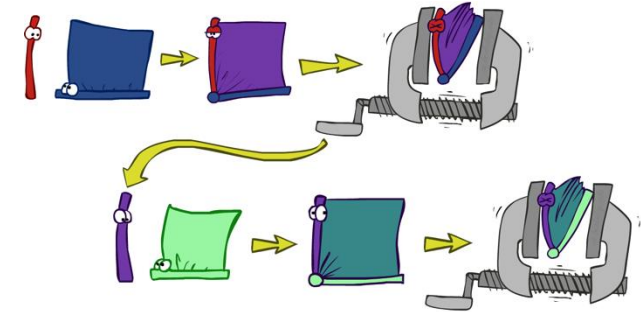
- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$



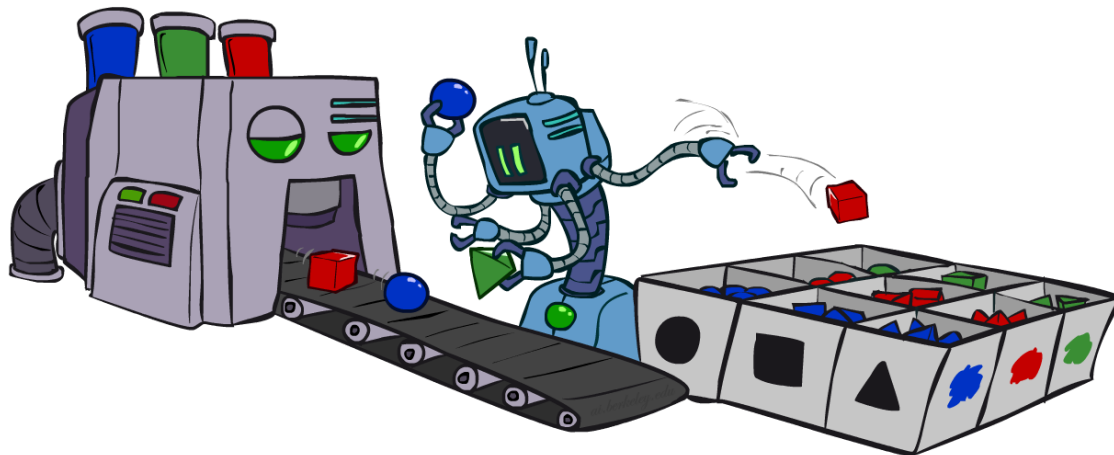
# Variable Elimination

- Interleave joining and marginalizing
- $d^k$  entries computed for a factor over  $k$  variables with domain sizes  $d$
- Ordering of elimination of hidden variables can affect size of factors generated
- Worst case: running time exponential in the size of the Bayes' net



# Sampling

- Goal: probability  $P$
- Basic idea
  - Draw  $N$  samples from a sampling distribution  $S$
  - Compute some quantity from the samples
  - Show this converges to the true probability  $P$
- Why sample?
  - Often very fast to get a decent approximate answer
  - The algorithms are very simple and general (easy to apply to fancy models)
  - They require very little memory ( $O(n)$ )



# Sampling from a discrete distribution

- Sampling from given distribution

- Step 1: Get sample  $u$  from uniform distribution over  $[0, 1)$ 
  - Random() in many programming languages
- Step 2: Convert this sample  $u$  into an outcome for the given distribution by associating each outcome  $x$  with a  $P(x)$ -sized sub-interval of  $[0,1)$

- Example

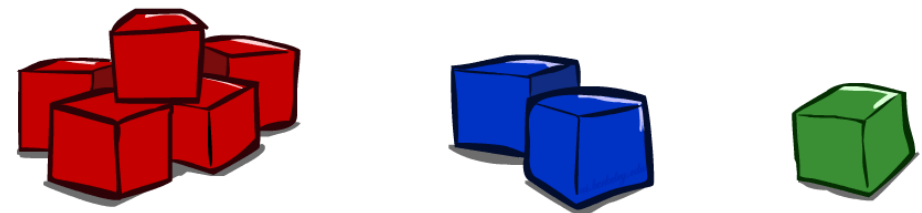
C	P(C)
red	0.6
green	0.1
blue	0.3

$$0 \leq u < 0.6, \rightarrow C = \text{red}$$

$$0.6 \leq u < 0.7, \rightarrow C = \text{green}$$

$$0.7 \leq u < 1, \rightarrow C = \text{blue}$$

- If random() returns  $u = 0.83$ , then our sample is  $C = \text{blue}$
- E.g, after sampling 8 times:



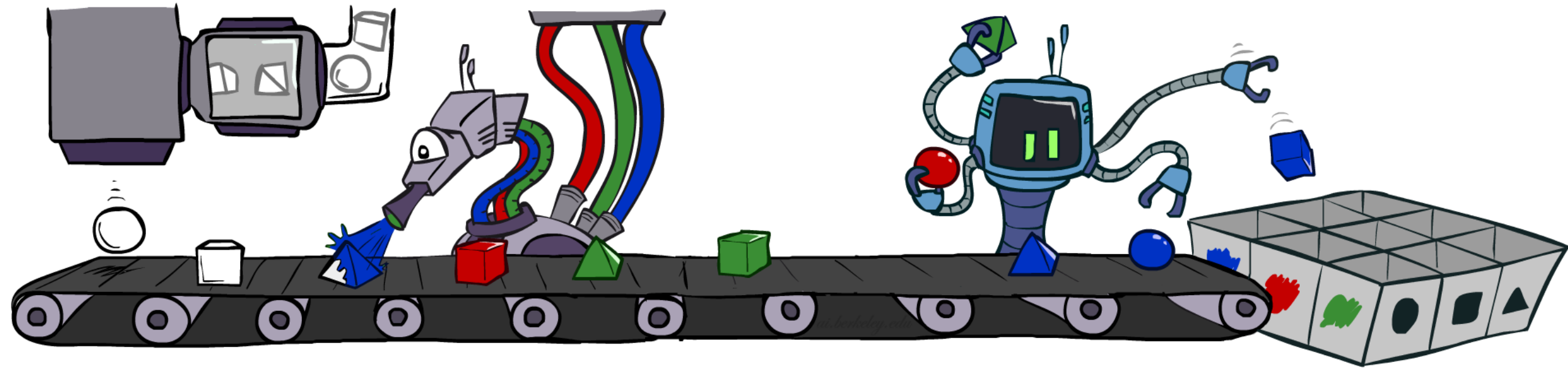
# Sampling in Bayes Nets

---

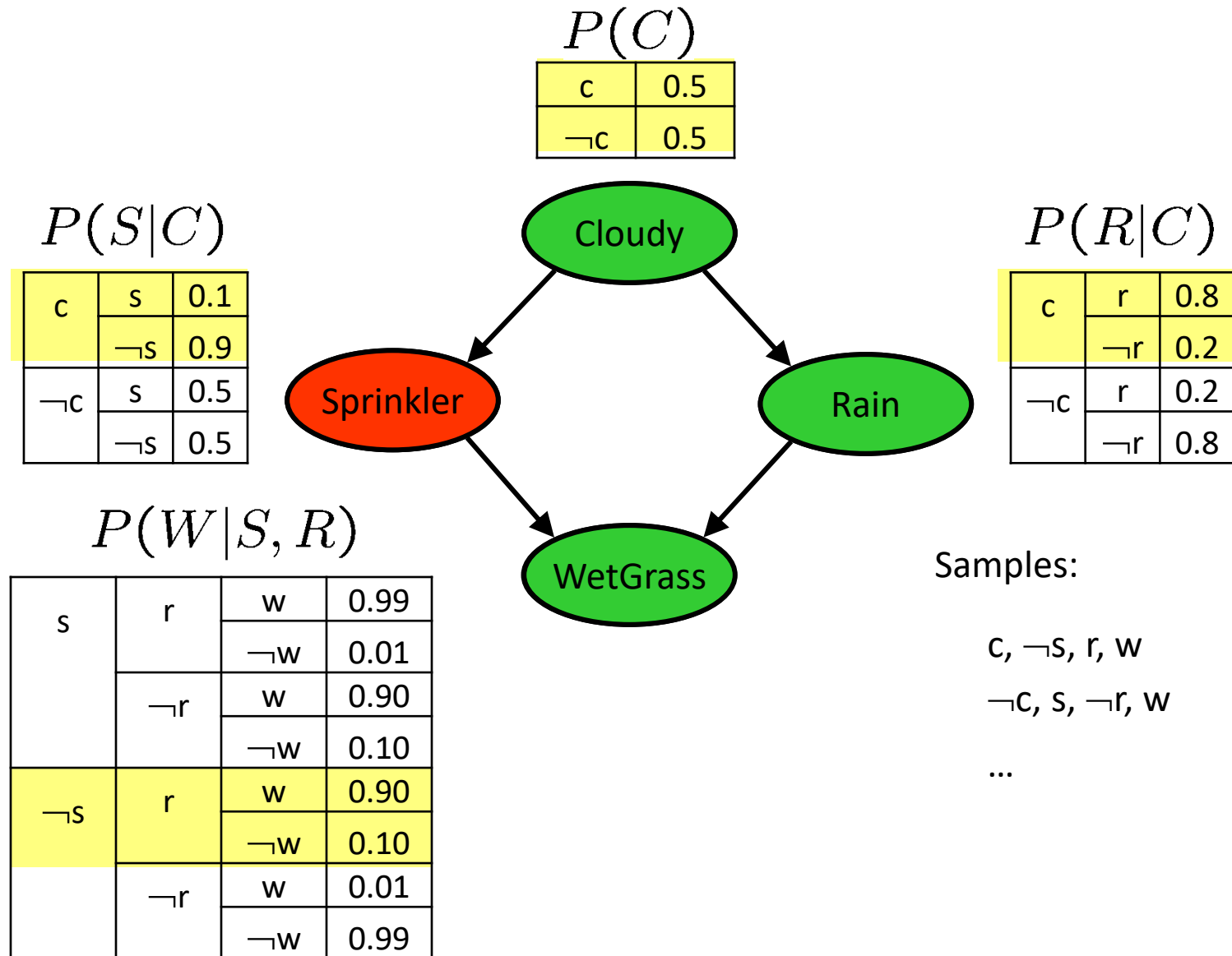
- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling



# Prior Sampling

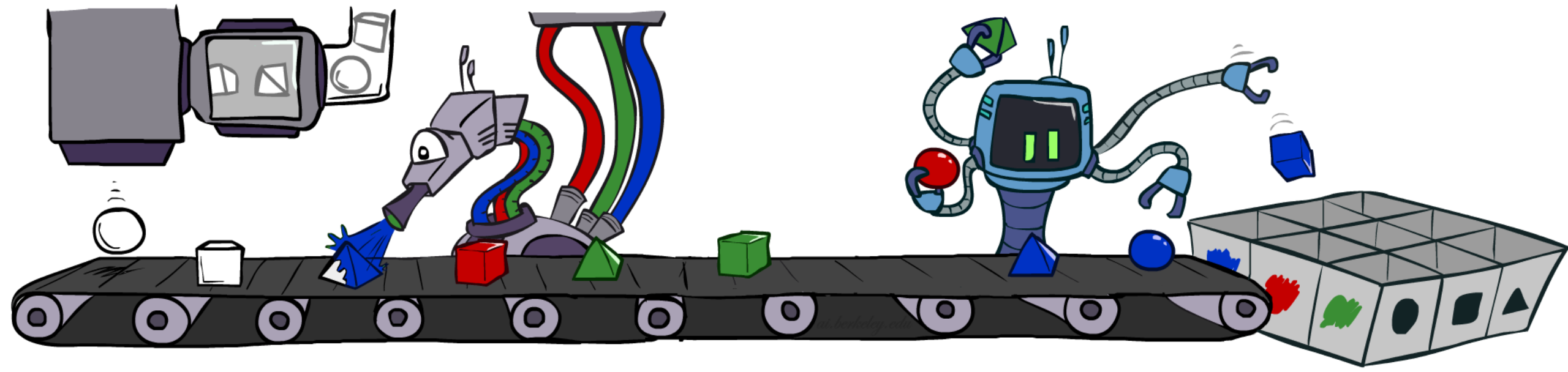


# Prior Sampling



# Prior Sampling

- For  $i=1, 2, \dots, n$  (in topological order)
  - Sample  $X_i$  from  $P(X_i \mid \text{parents}(X_i))$
- Return  $(x_1, x_2, \dots, x_n)$



# Using samples

- We'll get a bunch of samples from the BN:

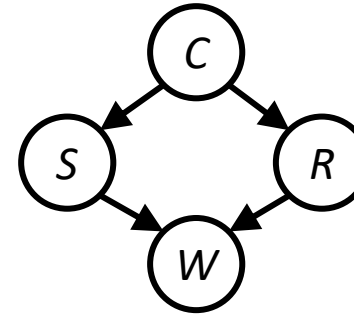
$C, \neg S, r, W$

$\neg C, S, r, W$

$\neg C, S, r, \neg W$

$C, \neg S, r, W$

$\neg C, \neg S, \neg r, W$



- If we want to know  $P(W)$ 
  - We have counts  $\langle w:4, \neg w:1 \rangle$
  - Normalize to get  $P(W) = \langle w:0.8, \neg w:0.2 \rangle$
  - This will get closer to the true distribution with more samples
- If we want to know  $P(C \mid r, w)$ 
  - Count  $(c, r, w)$  and  $(\neg c, r, w)$
  - Normalize to get  $P(C \mid r, w) = \langle c:0.67, \neg c:0.33 \rangle$

# Prior Sampling

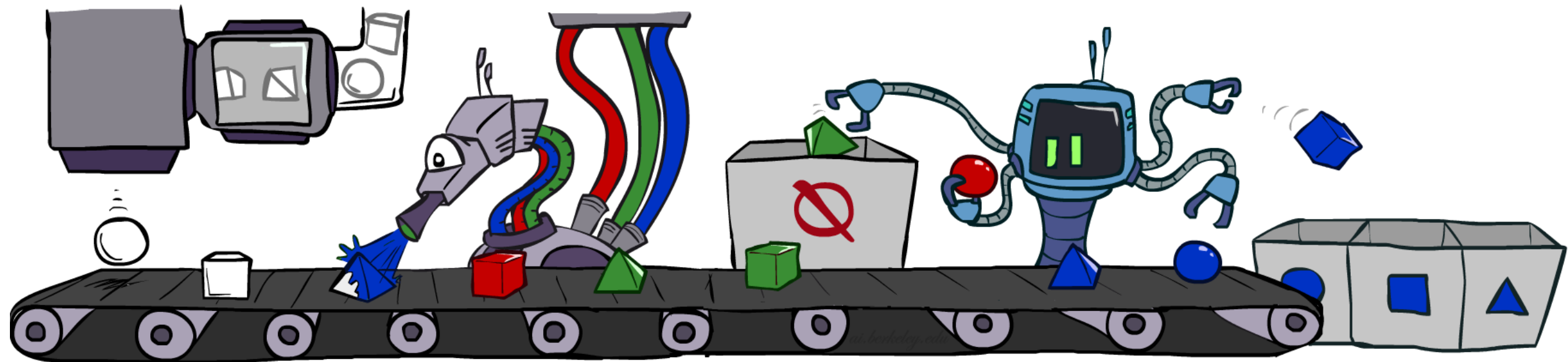
- This process generates samples with probability:

$$S_{PS}(x_1 \dots x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(X_i)) = P(x_1 \dots x_n)$$

...i.e. the BN's joint probability

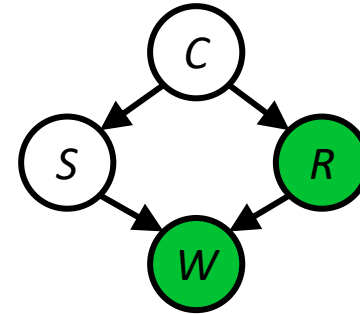
- Let the number of samples of an assignment be  $N_{PS}(x_1 \dots x_n)$
- So  $\hat{P}(x_1, \dots, x_n) = N_{PS}(x_1, \dots, x_n)/N$
- Then 
$$\begin{aligned} \lim_{N \rightarrow \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \rightarrow \infty} N_{PS}(x_1, \dots, x_n)/N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n) \end{aligned}$$
- I.e., the sampling procedure is **consistent**

# Rejection Sampling



# Rejection Sampling

- A simple modification of prior sampling for conditional probabilities
- Let's say we want  $P(C \mid r, w)$
- When generating a sample, reject it immediately if not  $R=\text{true}$ ,  $W=\text{true}$
- It is consistent for conditional probabilities (i.e., correct in the limit)



$C, \neg S, r, w$

~~$C, S, \neg r$~~

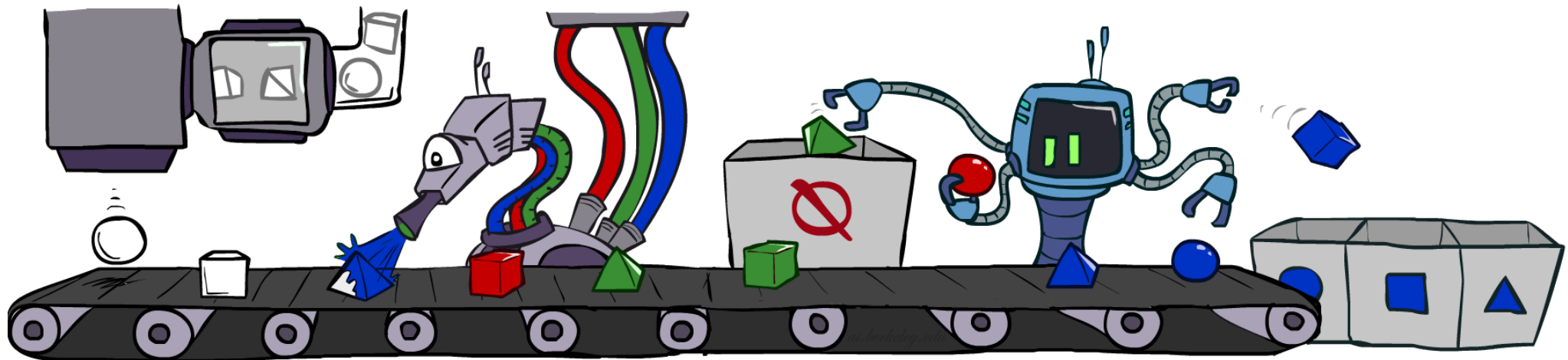
~~$\neg C, S, r, \neg w$~~

~~$C, \neg S, \neg r$~~

$\neg C, \neg S, r, w$

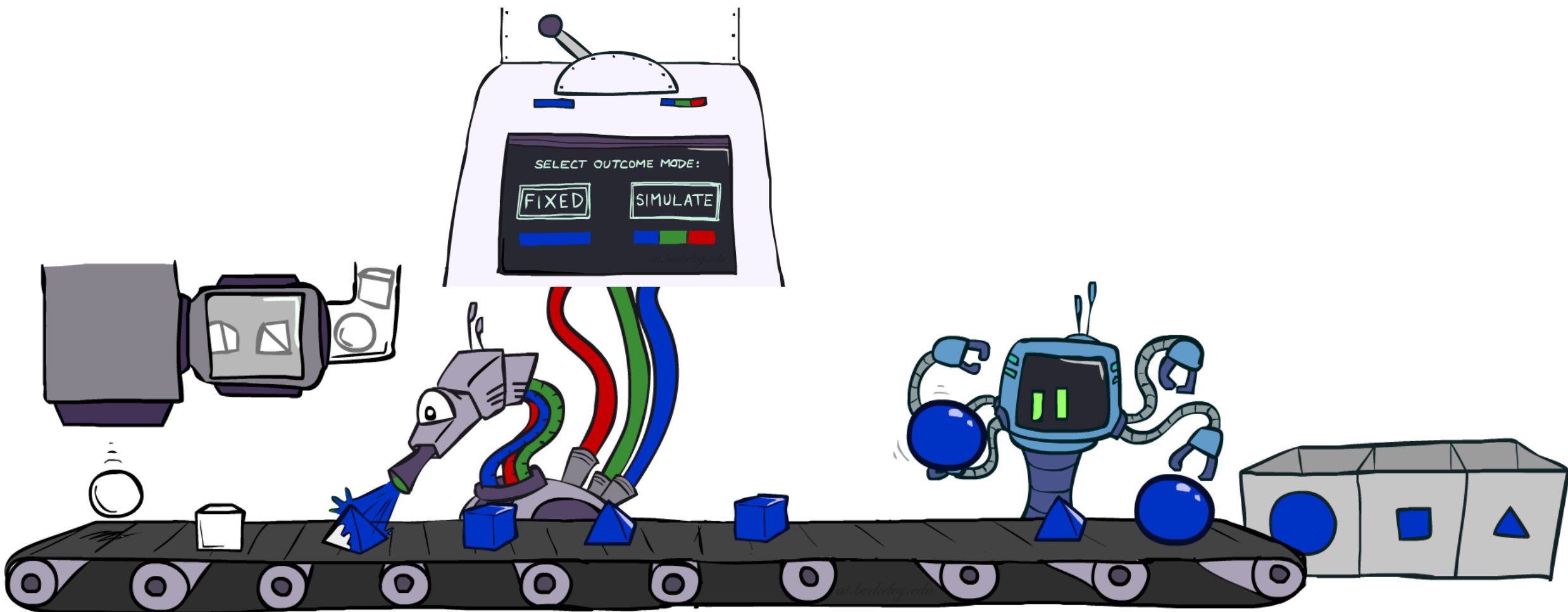
# Rejection Sampling

- Input: evidence  $e_1, \dots, e_k$
- For  $i=1, 2, \dots, n$ 
  - Sample  $x_i$  from  $P(x_i \mid \text{parents}(x_i))$
  - If  $x_i$  not consistent with evidence
    - Reject: Return, and no sample is generated in this cycle
- Return  $(x_1, x_2, \dots, x_n)$



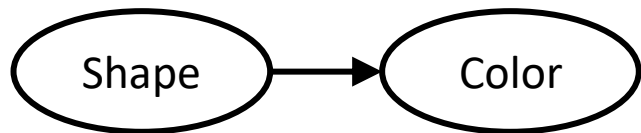


# Likelihood Weighting

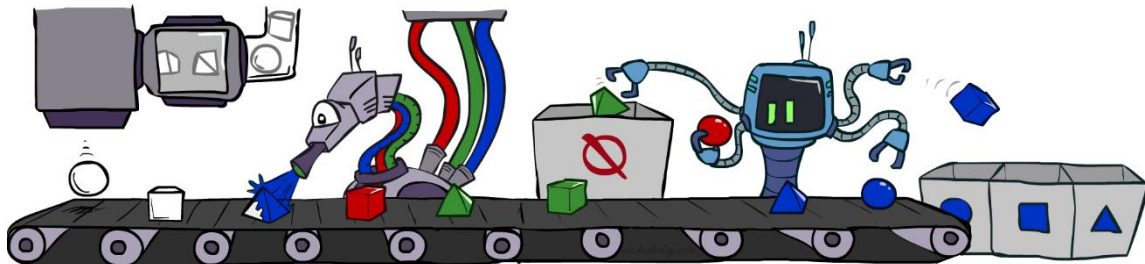


# Likelihood Weighting

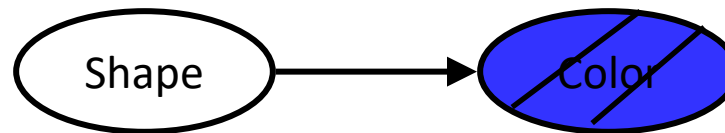
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider  $P(\text{Shape} \mid \text{Color}=\text{blue})$



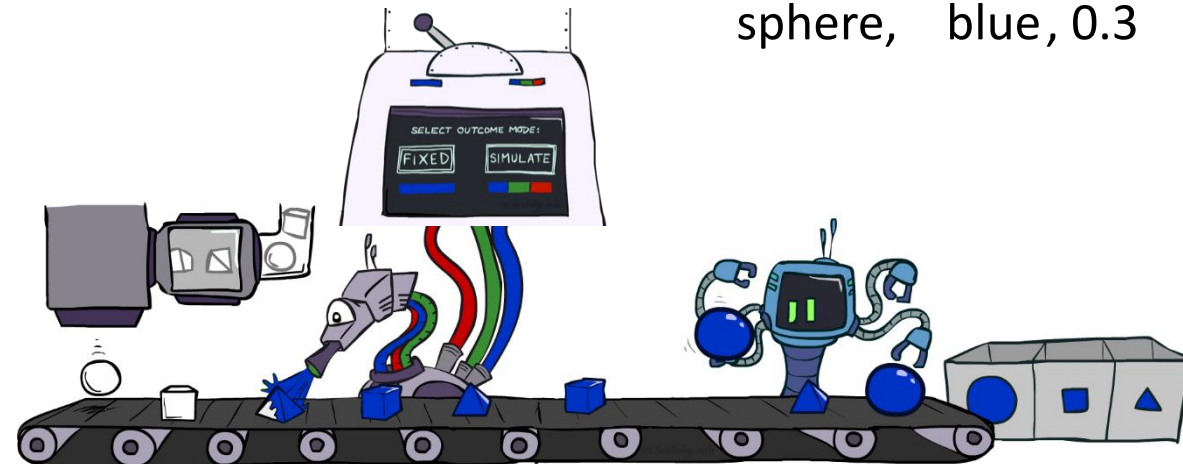
~~pyramid, green~~  
~~pyramid, red~~  
sphere, blue  
cube, red  
~~sphere, green~~



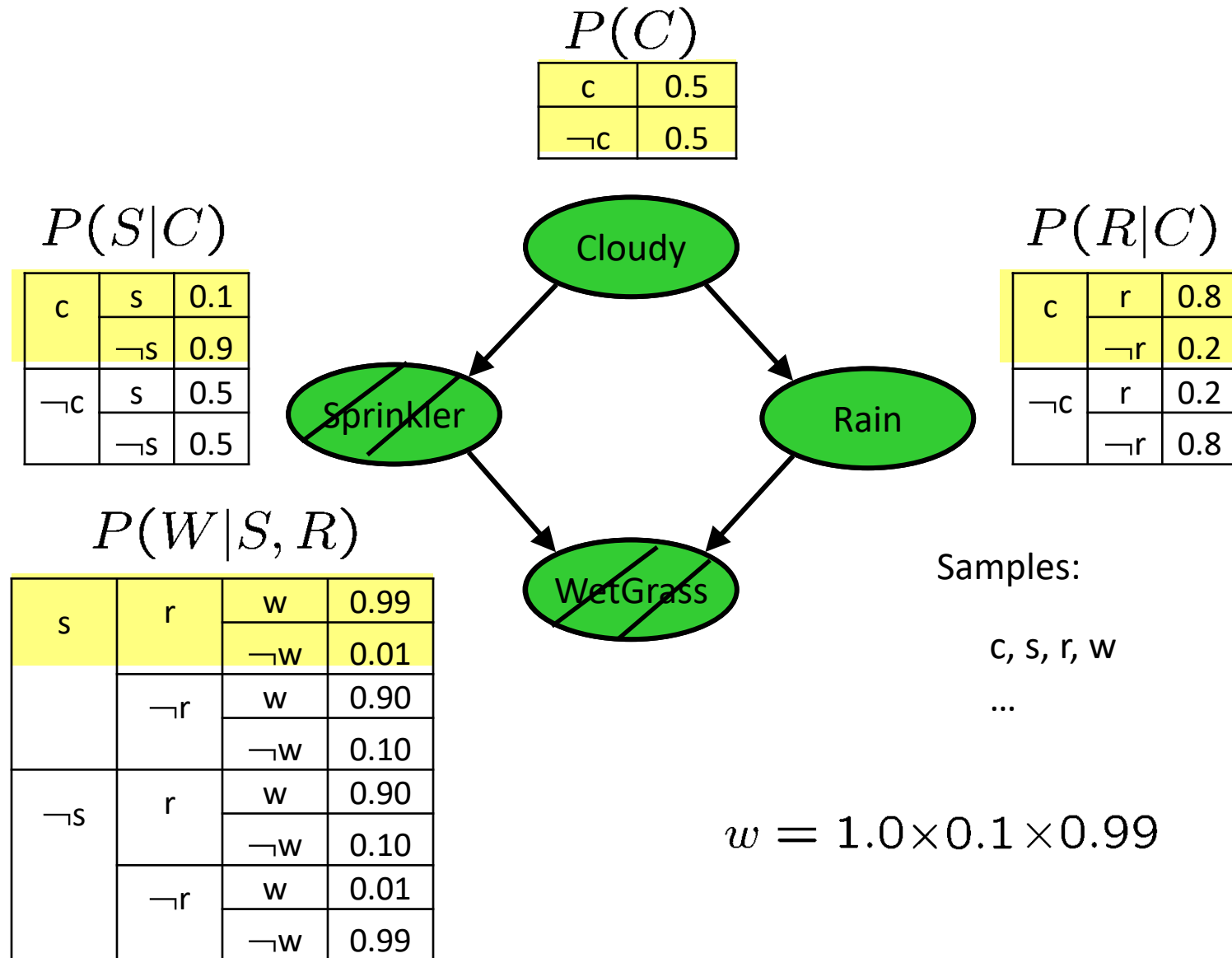
- Idea: fix evidence variables, sample the rest
  - Problem: sample distribution not consistent!
  - Solution: **weight** each sample by probability of evidence variables given parents



pyramid, blue, 0.4  
pyramid, blue, 0.4  
sphere, blue, 0.3  
cube, blue, 0.8  
sphere, blue, 0.3

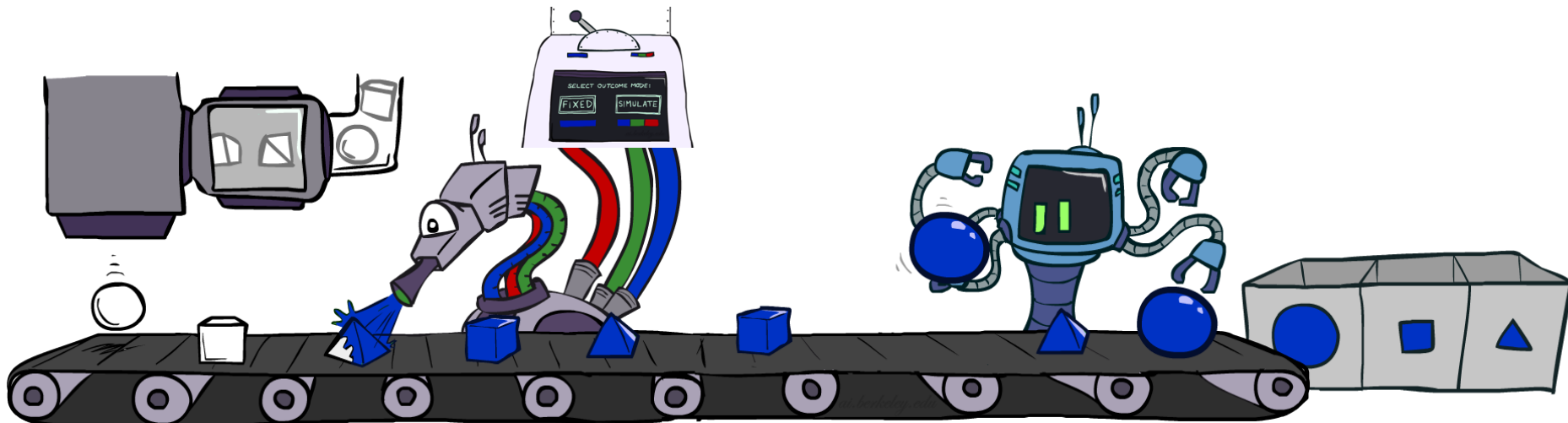


# Likelihood Weighting



# Likelihood Weighting

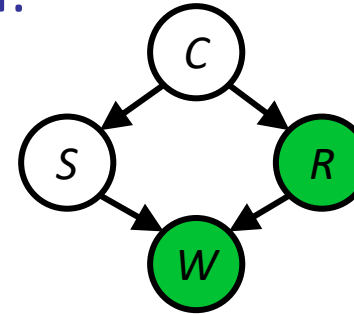
- Input: evidence  $e_1, \dots, e_k$
- $w = 1.0$
- for  $i=1, 2, \dots, n$ 
  - if  $X_i$  is an evidence variable
    - $x_i = \text{observed value}_i \text{ for } X_i$
    - Set  $w = w * P(x_i \mid \text{Parents}(X_i))$
  - else
    - Sample  $x_i$  from  $P(X_i \mid \text{Parents}(X_i))$
- return  $(x_1, x_2, \dots, x_n), w$



# Using samples

- We'll get a bunch of weighted samples from the BN:

$c, \neg s, r, w$	0.1
$c, s, r, w$	0.2
$\neg c, s, r, w$	0.3
$c, \neg s, r, w$	0.1
$\neg c, \neg s, r, w$	0.5



- If we want to know  $P(C | r, w)$ 
  - We have weight sums  $\langle (c, r, w): 0.4, (\neg c, r, w): 0.8 \rangle$
  - Normalize to get  $P(C | r, w) = \langle c: 0.33, \neg c: 0.67 \rangle$
  - This will get closer to the true distribution with more samples

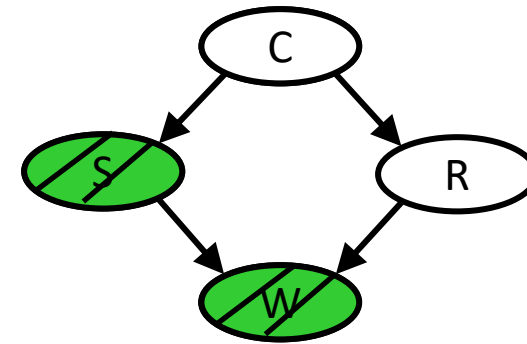
# Likelihood Weighting

- Sampling distribution ( $z$  is sampled and  $e$  is fixed evidence)

$$S_{WS}(z, e) = \prod_{i=1}^l P(z_i | \text{Parents}(Z_i))$$

- Now, samples have weights

$$w(z, e) = \prod_{i=1}^m P(e_i | \text{Parents}(E_i))$$

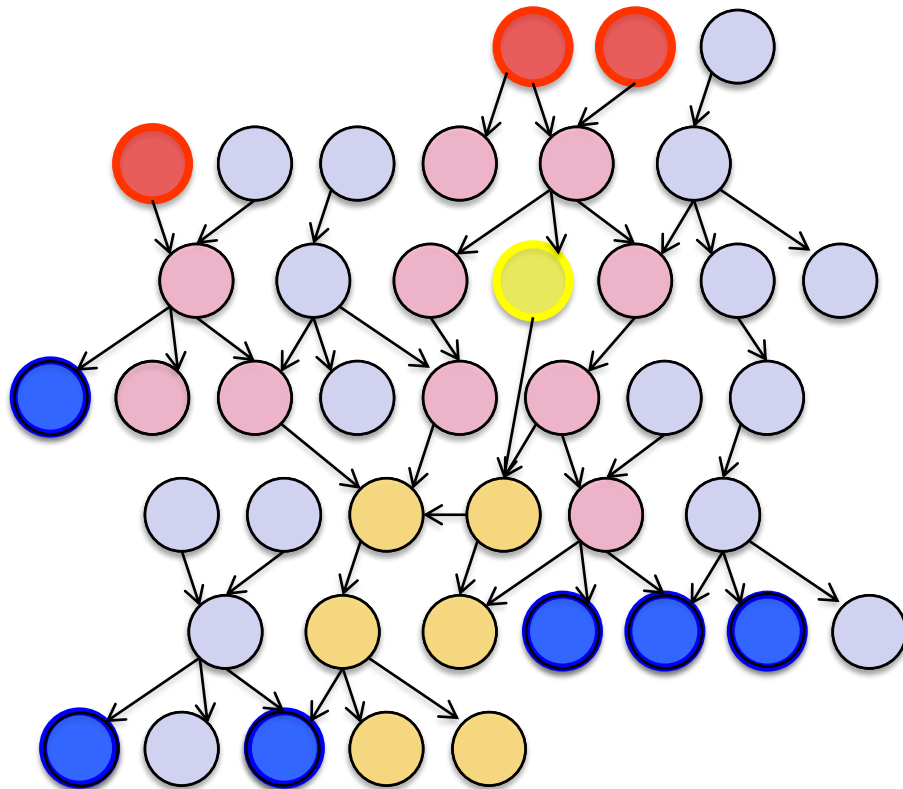


- Together, weighted sampling distribution is consistent

$$\begin{aligned} S_{WS}(z, e) \cdot w(z, e) &= \prod_{i=1}^l P(z_i | \text{Parents}(z_i)) \prod_{i=1}^m P(e_i | \text{Parents}(e_i)) \\ &= P(z, e) \end{aligned}$$

# Likelihood Weighting

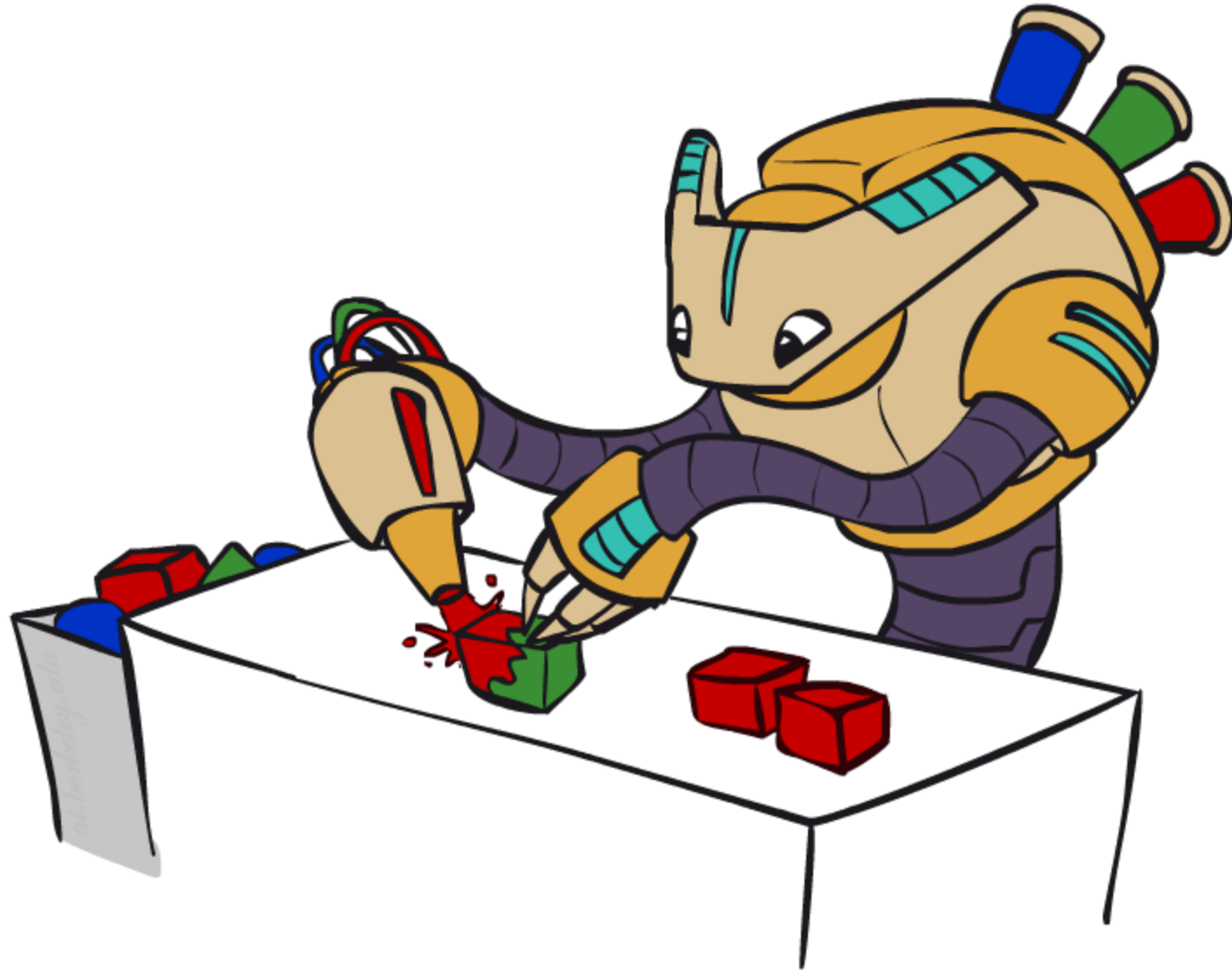
- Likelihood weighting is good
  - All samples are used
  - The values of *downstream* variables are influenced by *upstream* evidence



- Likelihood weighting still has weaknesses
  - The values of *upstream* variables are unaffected by *downstream* evidence
  - With many downstream evidence, we may
    - mostly get samples that are inconsistent with the evidence and thus have very small weights
    - get a few lucky samples with very large weights, which dominate the result
- We would like each variable to “see” *all* the evidence!

# Gibbs Sampling

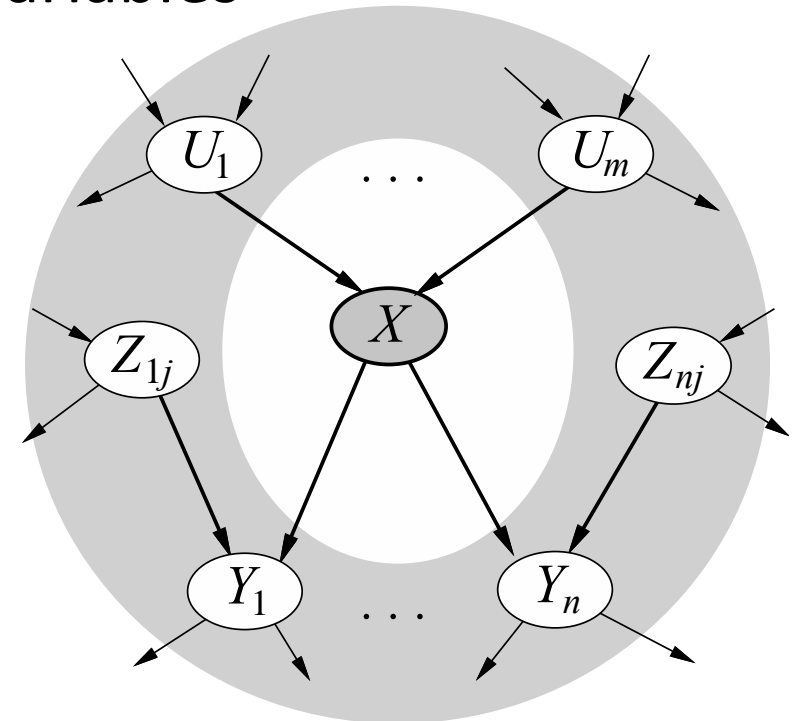
---





# Gibbs Sampling

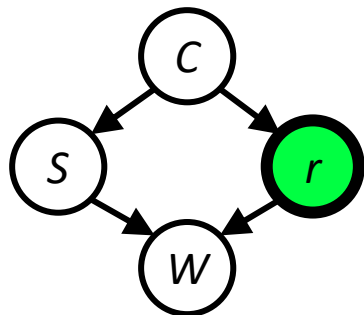
- Generate each sample by making a random change to the preceding sample
  - Evidence variables remain fixed. For each of the non-evidence variable, sample its value conditioned on all the other variables
  - $X_i' \sim P(X_i \mid x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$
  - In a Bayes net
$$P(X_i \mid x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$$
$$= P(X_i \mid \text{markov\_blanket}(X_i))$$
$$= \alpha P(X_i \mid u_1, \dots, u_m) \prod_j P(y_j \mid \text{parents}(Y_j))$$



# Gibbs Sampling Example: $P(S | r)$

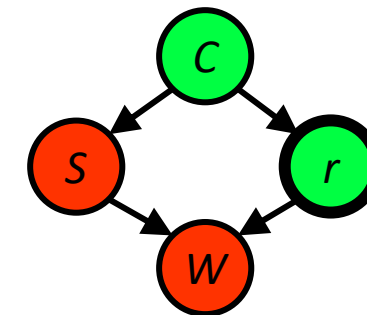
- Step 1: Fix evidence

- $R = \text{true}$



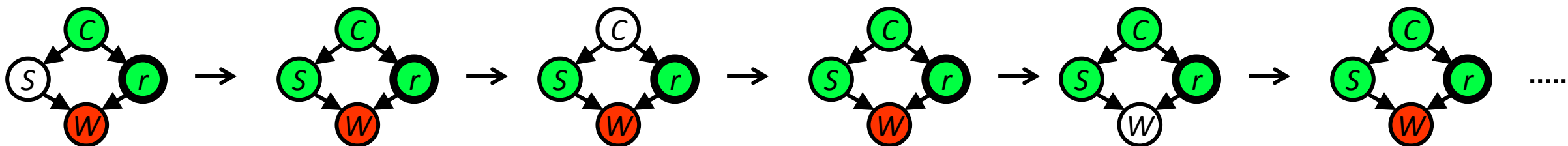
- Step 2: Initialize other variables

- Randomly



- Step 3: Repeat

- Choose an arbitrary non-evidence variable  $X$
- Resample  $X$  from  $P(X | \text{markov\_blanket}(X))$

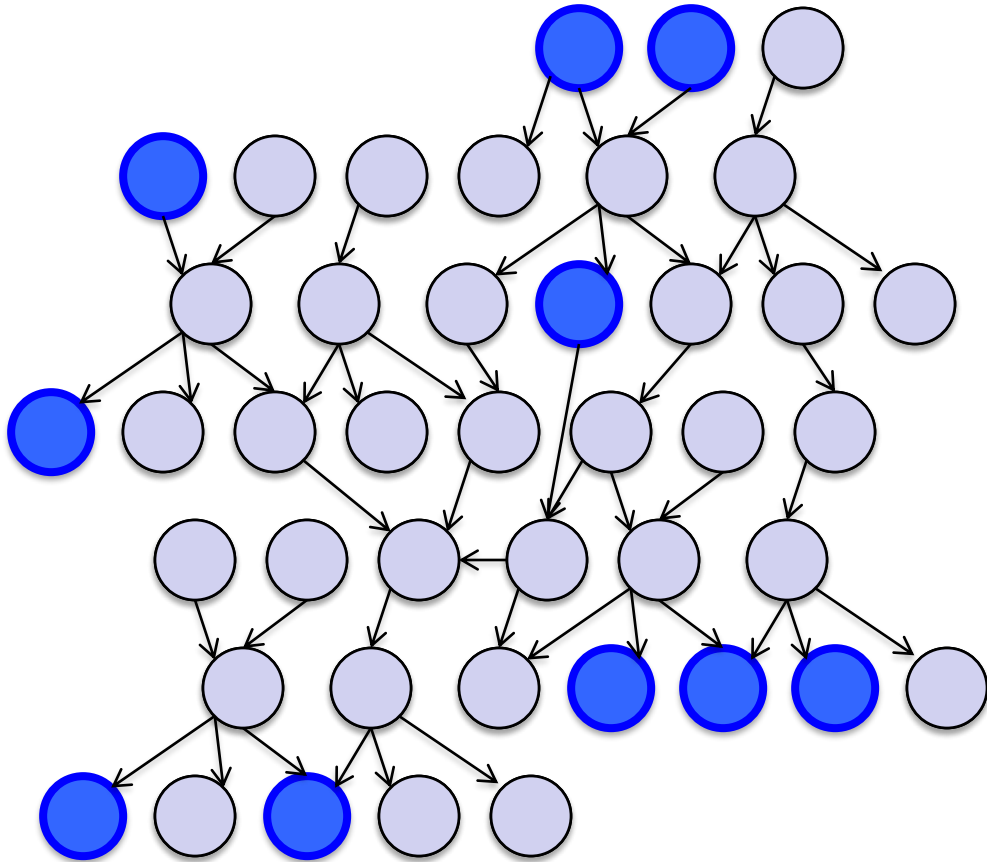


Sample  $S \sim P(S | c, r, \neg w)$

Sample  $C \sim P(C | s, r)$

Sample  $W \sim P(W | s, r)$

# Why doing this?



- Samples soon begin to reflect all the evidence in the network
- Eventually they are being drawn from the true posterior!
- Theorem: Gibbs sampling is consistent

# Why does it work? (see AIMA 14.5.2 for details)

- Suppose we run it for a long time and predict the probability of reaching any given state at time  $t$ :  $\pi_t(x_1, \dots, x_n)$  or  $\pi_t(\underline{x})$
- Each Gibbs sampling step (pick a variable, resample its value) applied to a state  $\underline{x}$  has a probability  $q(\underline{x}' \mid \underline{x})$  of reaching a next state  $\underline{x}'$
- So  $\pi_{t+1}(\underline{x}') = \sum_{\underline{x}} q(\underline{x}' \mid \underline{x}) \pi_t(\underline{x})$  or, in matrix/vector form  $\pi_{t+1} = Q\pi_t$
- When the process is in equilibrium  $\pi_{t+1} = \pi_t$  so  $Q\pi_t = \pi_t$
- This has a unique solution  $\pi_t = P(x_1, \dots, x_n \mid e_1, \dots, e_k)$
- So for large enough  $t$  the next sample will be drawn from the true posterior



- MCM  
som

- M  
w
- M



ing

walk”),

ino



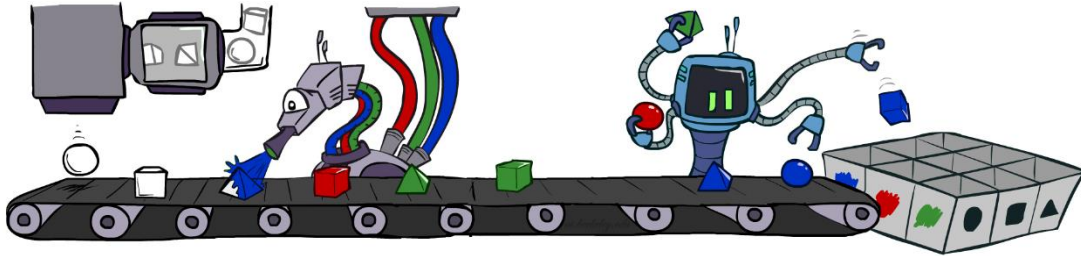
# Markov Chain Monte Carlo (MCMC)

---

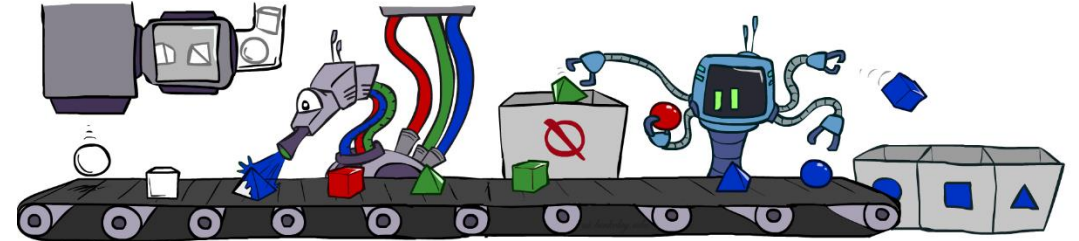
- MCMC is a family of randomized algorithms for approximating some quantity of interest over a very large state space
  - Markov chain = a sequence of randomly chosen states (“random walk”), where each state is chosen conditioned on the previous state
  - ~~Monte Carlo = a very expensive city in Monaco with a famous casino~~
  - Monte Carlo = an algorithm (usually based on sampling) that is likely to find a correct answer
- MCMC = sampling by constructing a Markov chain
- Gibbs, Metropolis-Hastings, Hamiltonian, Slice, etc.

# Summary

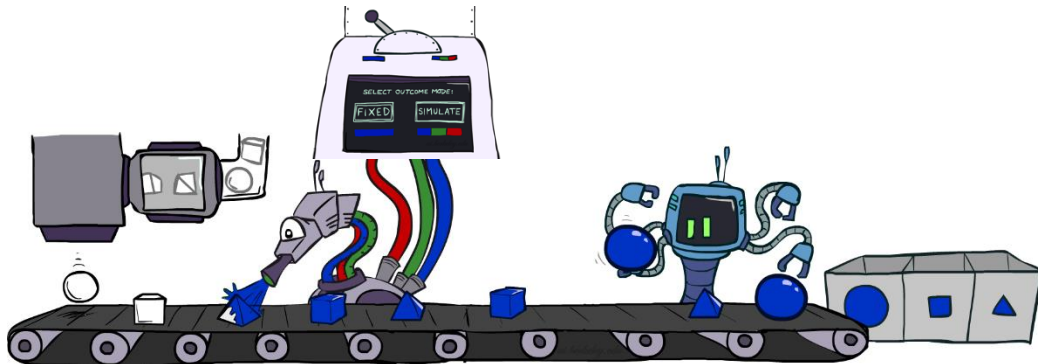
- Prior Sampling  $P$



- Rejection Sampling  $P(Q | e)$



- Likelihood Weighting  $P(Q | e)$



- Gibbs Sampling  $P(Q | e)$

