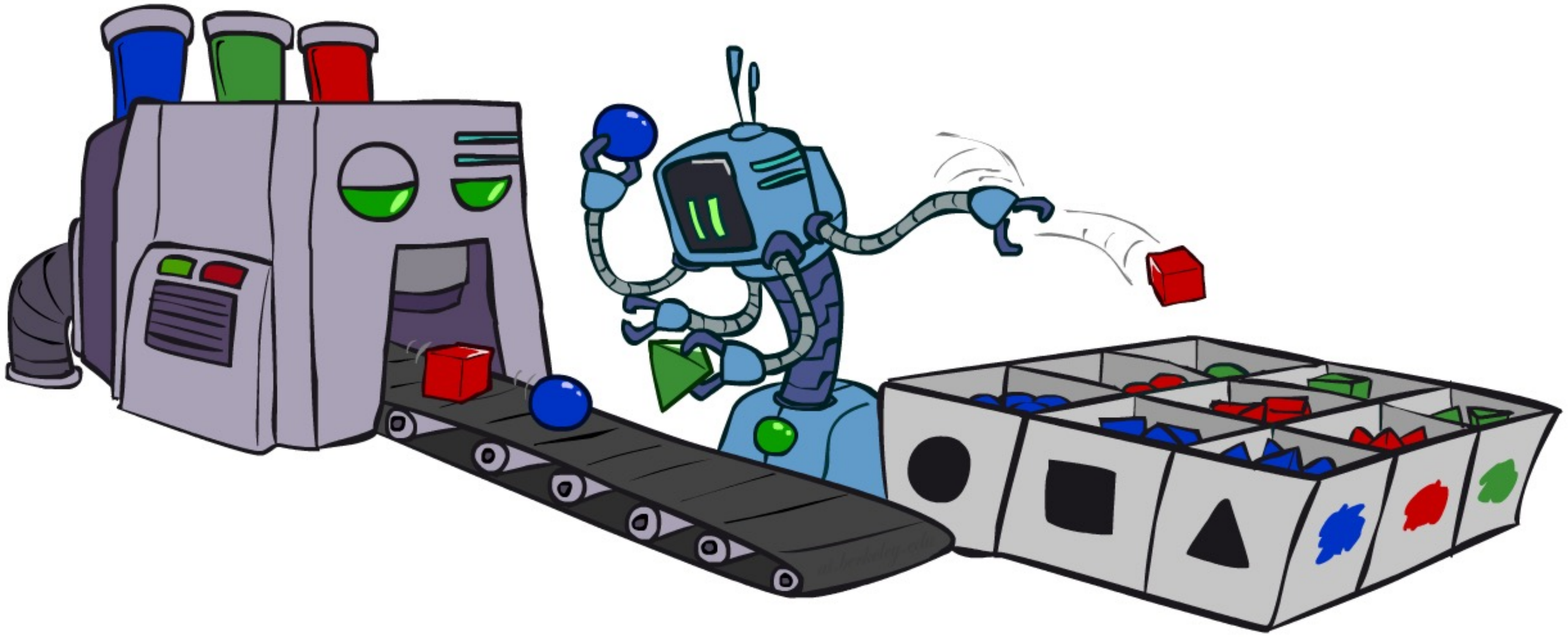
Ve492: Introduction to Artificial Intelligence

Bayesian Networks: Sampling



Paul Weng

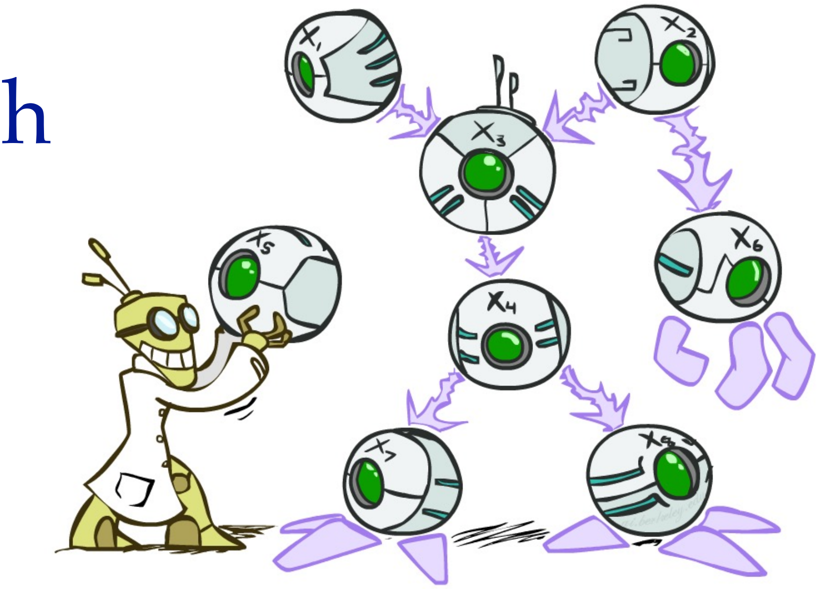
UM-SJTU Joint Institute

Slides adapted from http://ai.berkeley.edu, AIMA, UM, CMU

Bayes’ Net

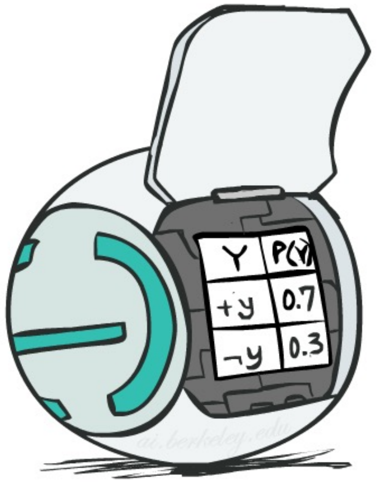
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents’ values

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



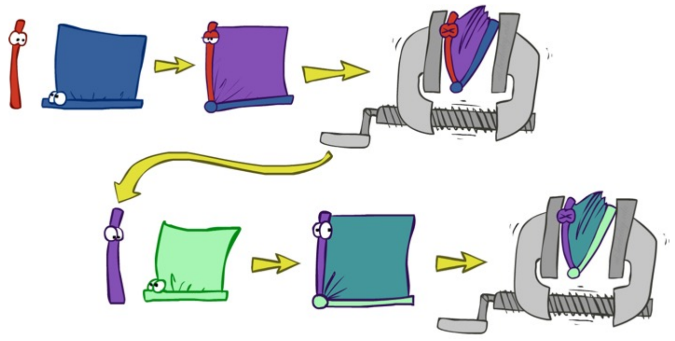
- Bayes’ nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

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

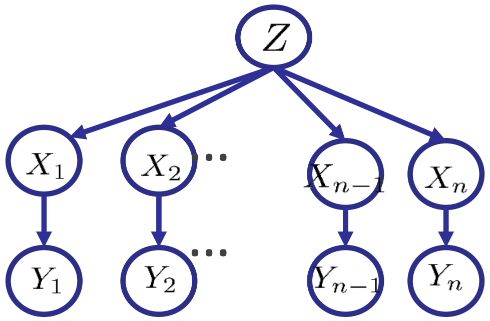


Variable Elimination

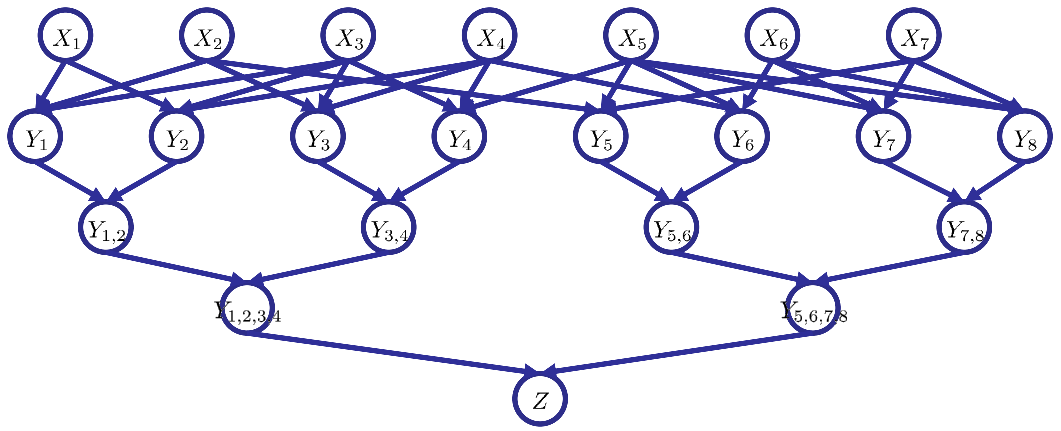
- Interleave joining and marginalizing



- 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

Worst Case Complexity?

- CSP:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

$P(X_i = 0) = P(X_i = 1) = 0.5$

$Y_1 = X_1 \vee X_2 \vee \neg X_3$

$\ldots$

$Y_8 = \neg X_5 \vee X_6 \vee X_7$

$Y_{1,2} = Y_1 \wedge Y_2$

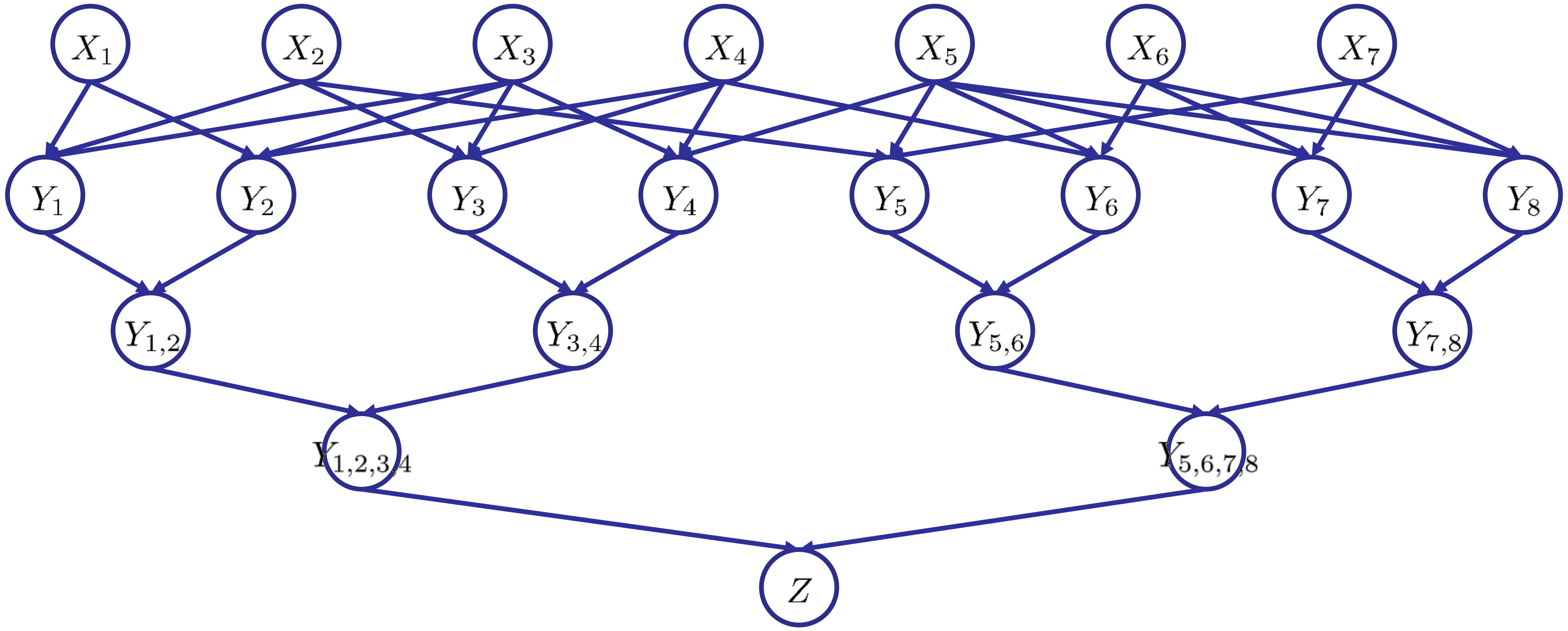
$\ldots$

$Y_{7,8} = Y_7 \wedge Y_8$

$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$

$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$

$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$



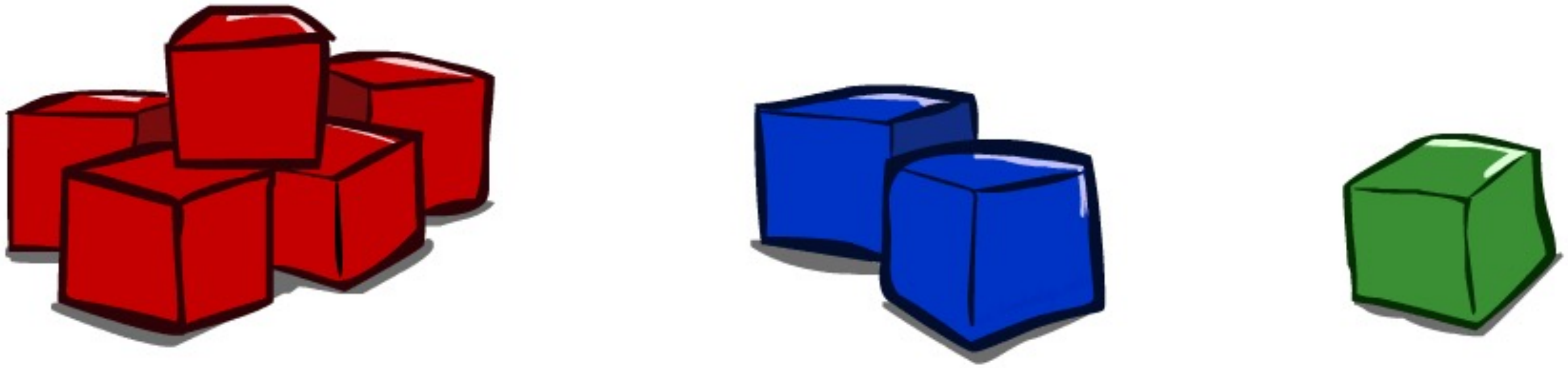
- If we can answer P(z) equal to zero or not, we answered whether the 3-SAT problem has a solution.

- Hence inference in Bayes' nets is NP-hard. No known efficient probabilistic inference in general.

Bayes’ Nets

- ✓ Representation
- ✓ Conditional Independences
- Probabilistic Inference
  - ✓ Enumeration (exact, exponential complexity)
  - ✓ Variable elimination (exact, worst-case exponential complexity, often better)
  - ✓ Probabilistic inference is NP-complete
  - Approximate inference (sampling)

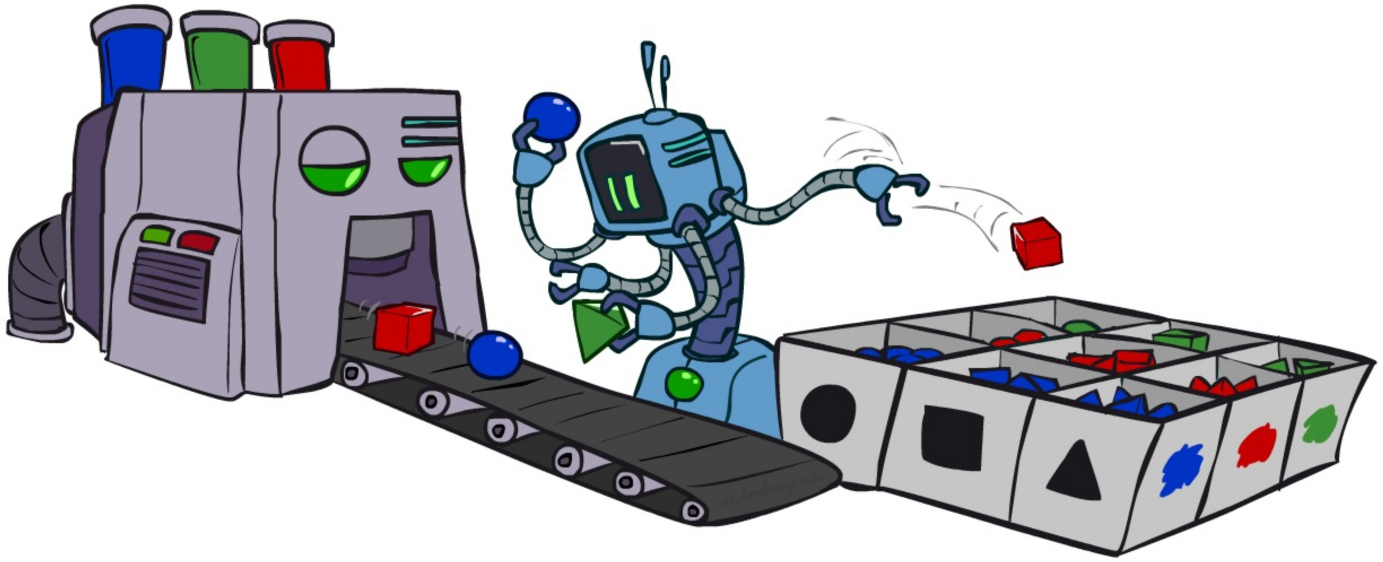
Approximate Inference: Sampling



Sampling

- Basic idea
  - Draw N samples from a **sampling distribution** S
  - Compute an approximate posterior probability
  - 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))
  - They can be applied to large models, whereas exact algorithms blow up



Example

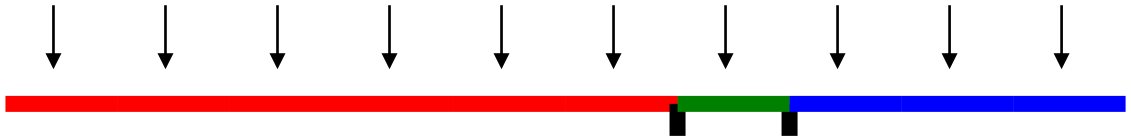
- Suppose you have two agent programs A and B for Monopoly
- What is the probability that A wins?

- Method 1:
  - Let s be a sequence of dice rolls and Chance and Community Chest cards
  - Given s, the outcome $V(s)$ is determined (1 for a win, 0 for a loss)
  - Probability that A wins is $\sum_s P(s)V(s)$
  - Problem: infinite number of sequences s !

- Method 2:
  - Sample N sequences from $P(s)$, play N games (maybe 100)
  - Probability that A wins is roughly $\frac{1}{N}\sum_i V(s_i)$, i.e., fraction of wins in the sample

Sampling Basics: Discrete (Categorical) Distribution

- Sampling from given distribution
  - **Step 1**: Get sample *u* from uniform distribution over [0, 1)
    - E.g., random() in python
  - **Step 2**: Convert this sample *u* into an outcome for the given distribution by having each outcome associated with a sub-interval of [0,1) with sub-interval size equal to probability of the outcome

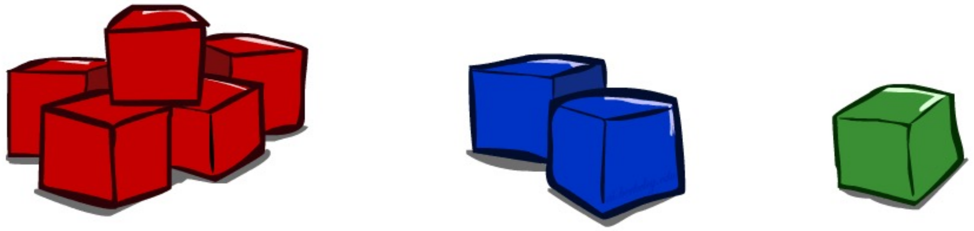


- Example

C	P(C)
red	0.6
green	0.1
blue	0.3

$$0 \leq u < 0.6, \rightarrow C = red$$
$$0.6 \leq u < 0.7, \rightarrow C = green$$
$$0.7 \leq u < 1, \rightarrow C = blue$$

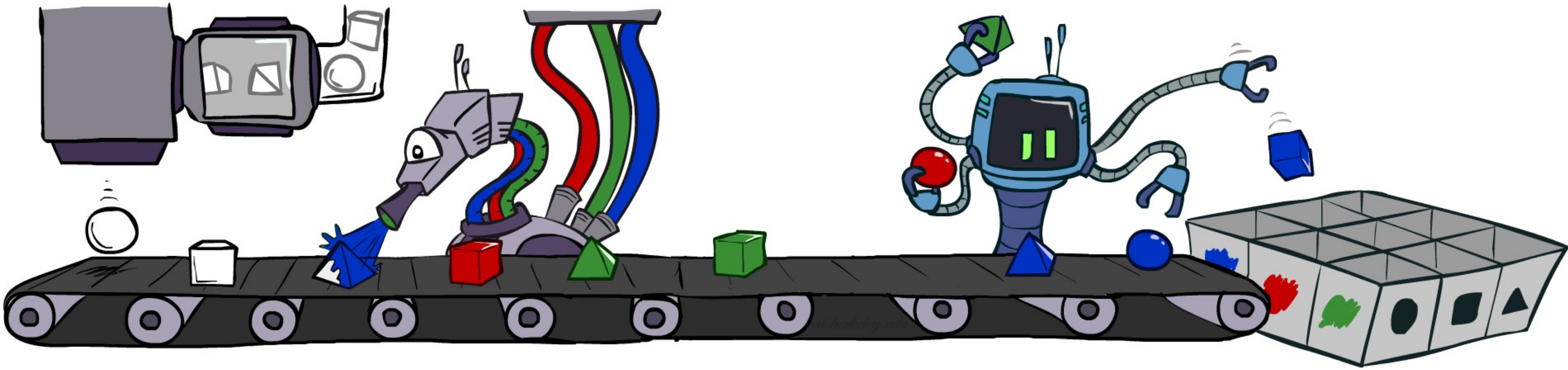
- If random() returns $u = 0.83$, then our sample is C = blue
- E.g., after sampling 8 times:



Sampling in Bayes' Nets

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

Prior Sampling



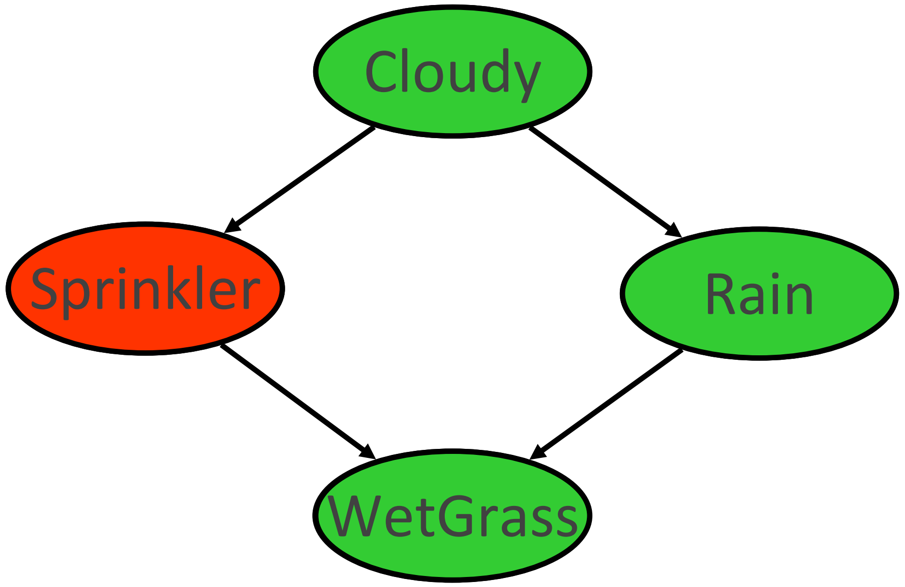
Prior Sampling

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, -s, +r, +w

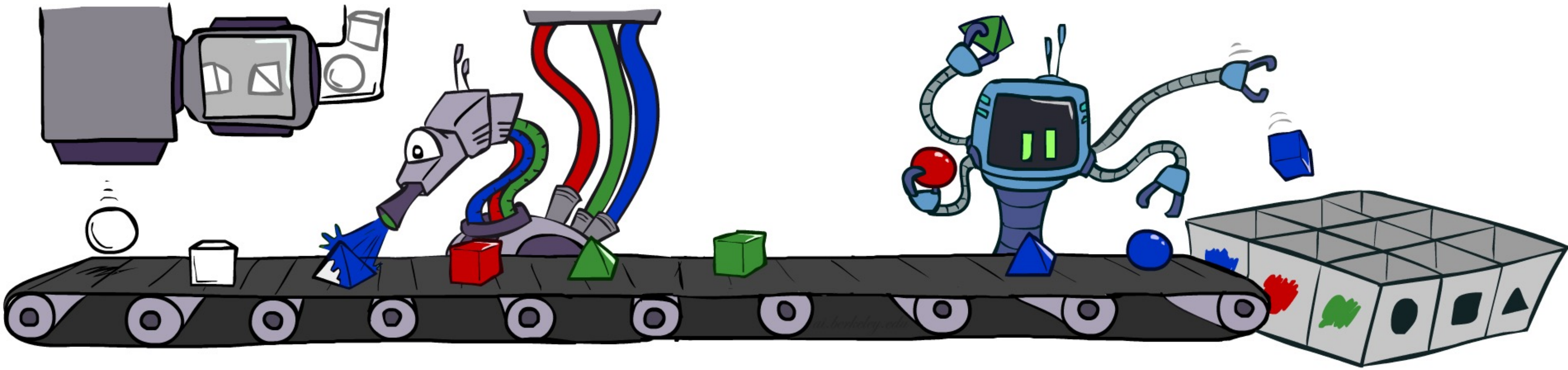
-c, +s, -r, +w

…

Prior Sampling

```
For i=1, 2, …, n
    Sample $x_i$ from $P(X_i \mid Parents(X_i))$

Return $(x_1, x_2, \ldots, x_n)$
```



Prior Sampling

- This process generates samples with probability:

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

  i.e., the BN's joint probability

- Let the number of samples of an event be $N_{PS}(x_1 \ldots x_n)$

- Then $$\begin{aligned} \lim_{N\to\infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

- i.e., the sampling procedure is consistent

Example

- Assume we have a bunch of samples from a BN:

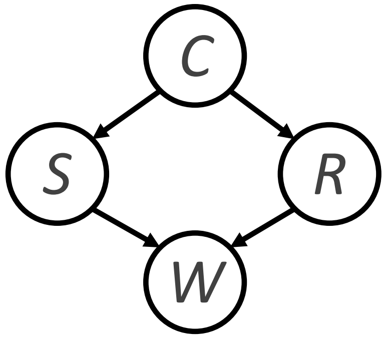
  +c, -s, +r, +w

  +c, +s, +r, +w

  -c, +s, +r, -w

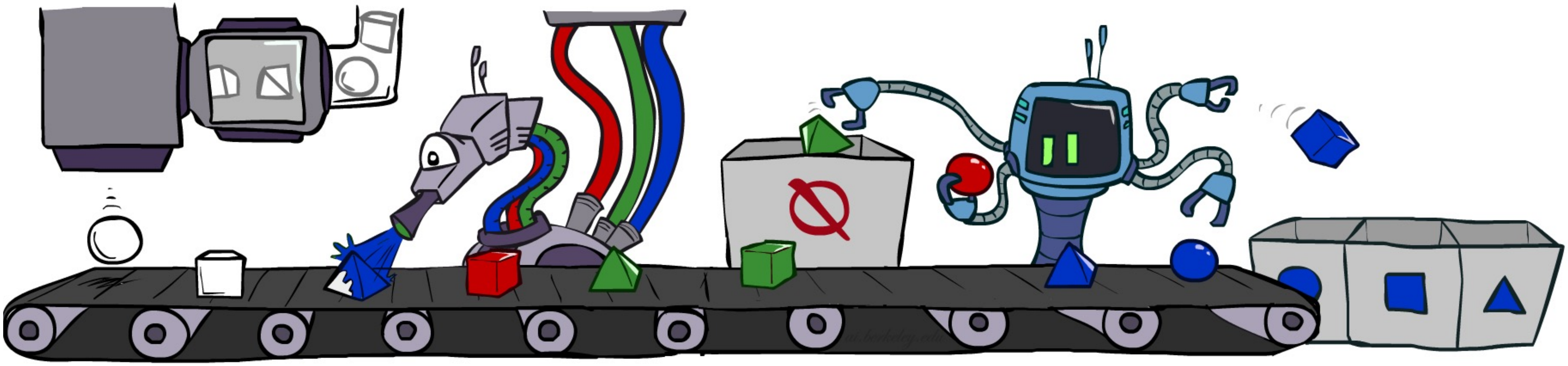
  +c, -s, +r, +w

  -c, -s, -r, +w



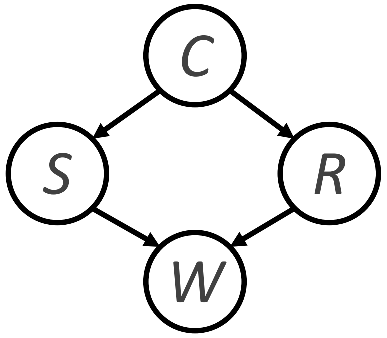
- How to estimate P(W)?
  - Count realizations of W <+w:4, -w:1>
  - Normalize to estimate P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
- General approach: can estimate any (conditional) probability
  - P(C | +w)? P(C | +r, +w)? P(C | -r, -w)?
  - Fast: can use fewer samples if less time (what's the drawback?)

Rejection Sampling



Rejection Sampling

- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go



- Let's say we want P(C | +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)

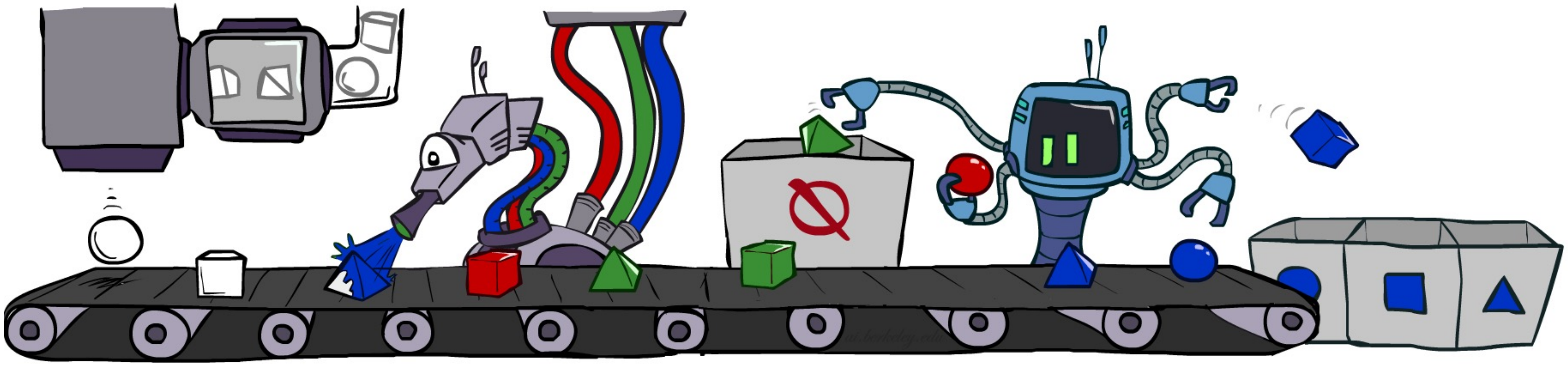
+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

Rejection Sampling

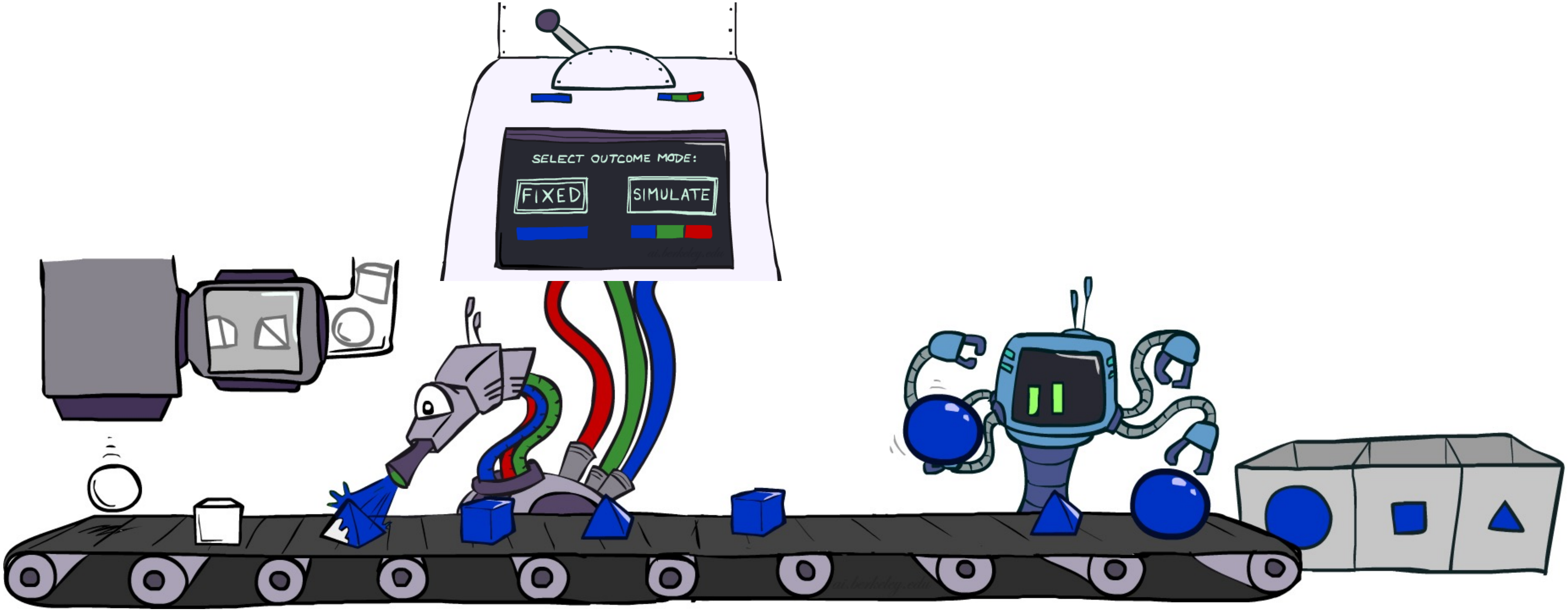
```
IN: evidence instantiation
For i=1, 2, …, n
    Sample $x_i$ from $P(X_i \mid Parents(X_i))$

    If $x_i$ not consistent with evidence
        Reject: Return, and no sample is generated in this cycle

Return $(x_1, x_2, …, x_n)$
```



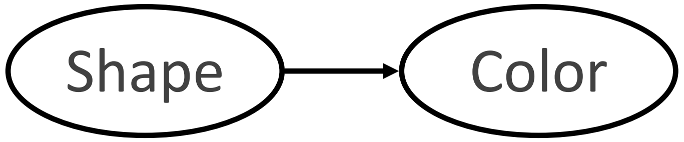
Likelihood Weighting



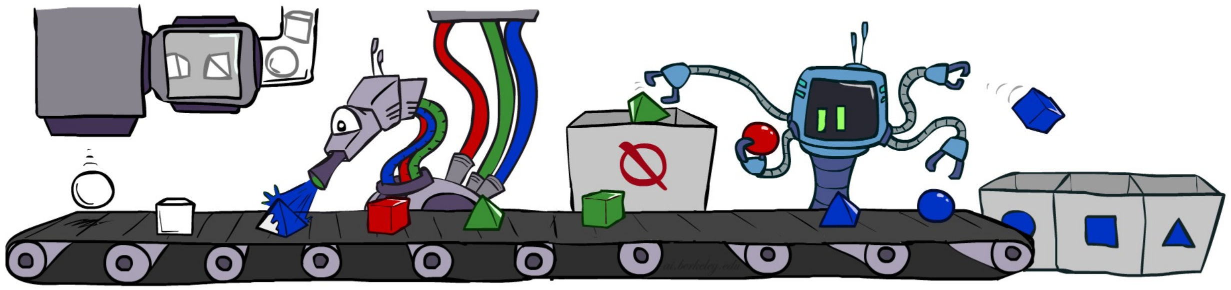
Likelihood Weighting

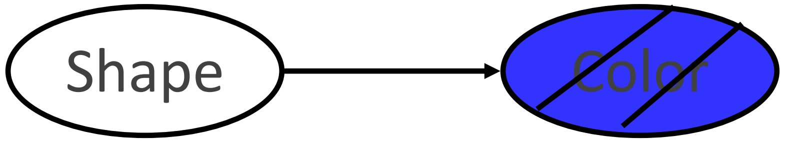
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P(Shape | blue)

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

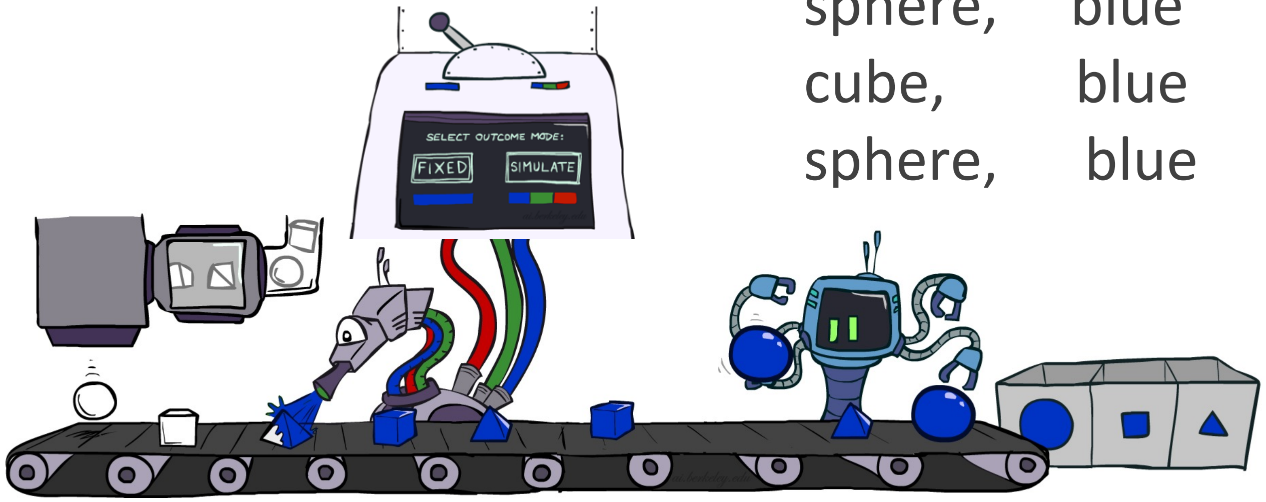


~~pyramid,~~	~~green~~
~~pyramid,~~	~~red~~
sphere,	blue
~~cube,~~	~~red~~
~~sphere,~~	~~green~~





pyramid,	blue
pyramid,	blue
sphere,	blue
cube,	blue
sphere,	blue



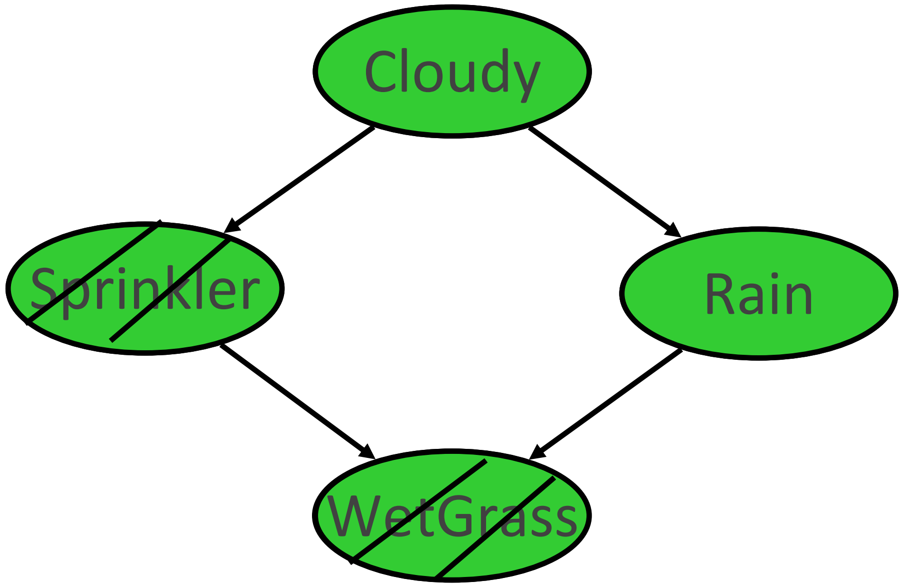
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

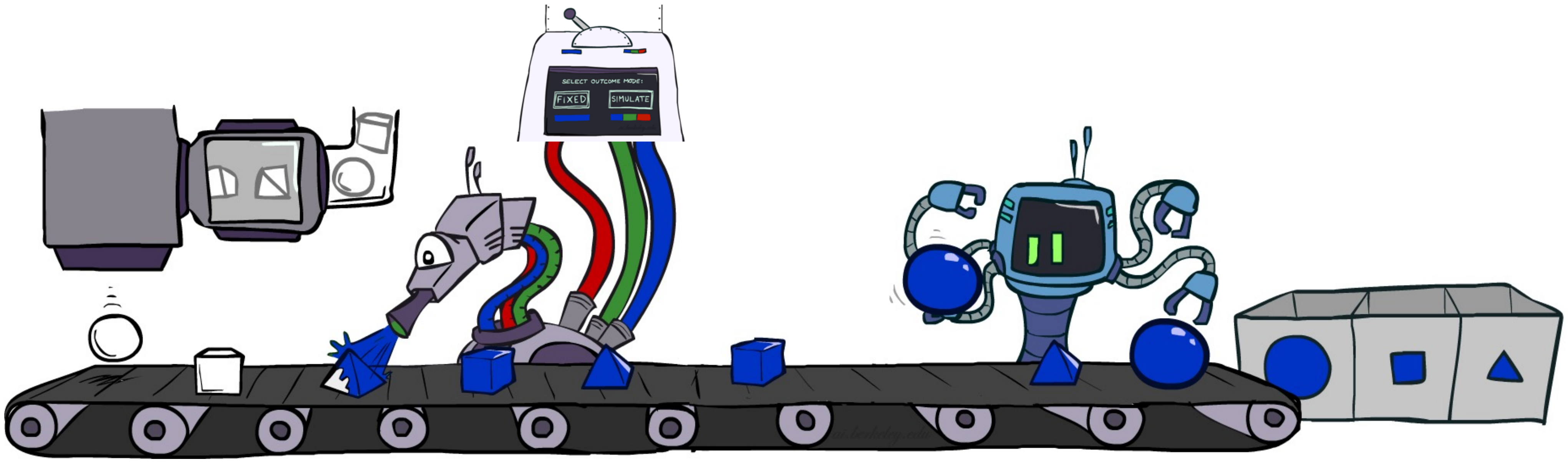
+c, +s, +r, +w

…

$$w = 1.0 \times 0.1 \times 0.99$$

Likelihood Weighting

```
IN: evidence instantiation
w = 1.0
for i=1, 2, …, n
    if $X_i$ is an evidence variable
        $x_i$ = observation for $X_i$
        Set $w = w * P(x_i \mid Parents(X_i))$
    else
        Sample $x_i$ from $P(X_i \mid Parents(X_i))$
return $(x_1, x_2, \ldots, x_n)$, w
```

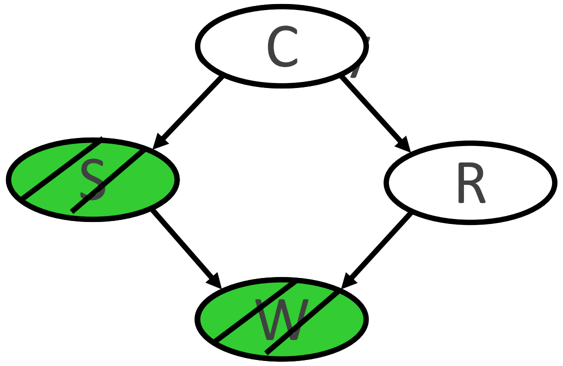


Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

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

- Now, samples have weights



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

- Together, weighted sampling distribution is consistent

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

$$= P(\mathbf{z}, \mathrm{e})$$

Quiz: Likelihood Weighting

- Two identical samples from likelihood weighted sampling will have the same exact weights.

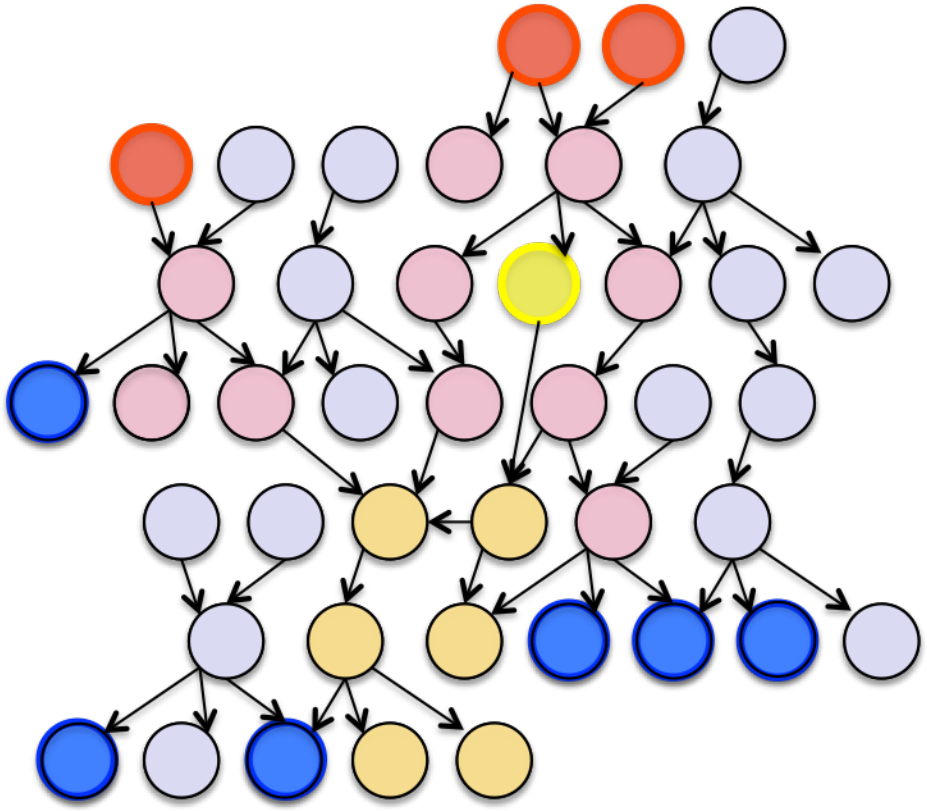
  A. True

  B. False

  C. It depends

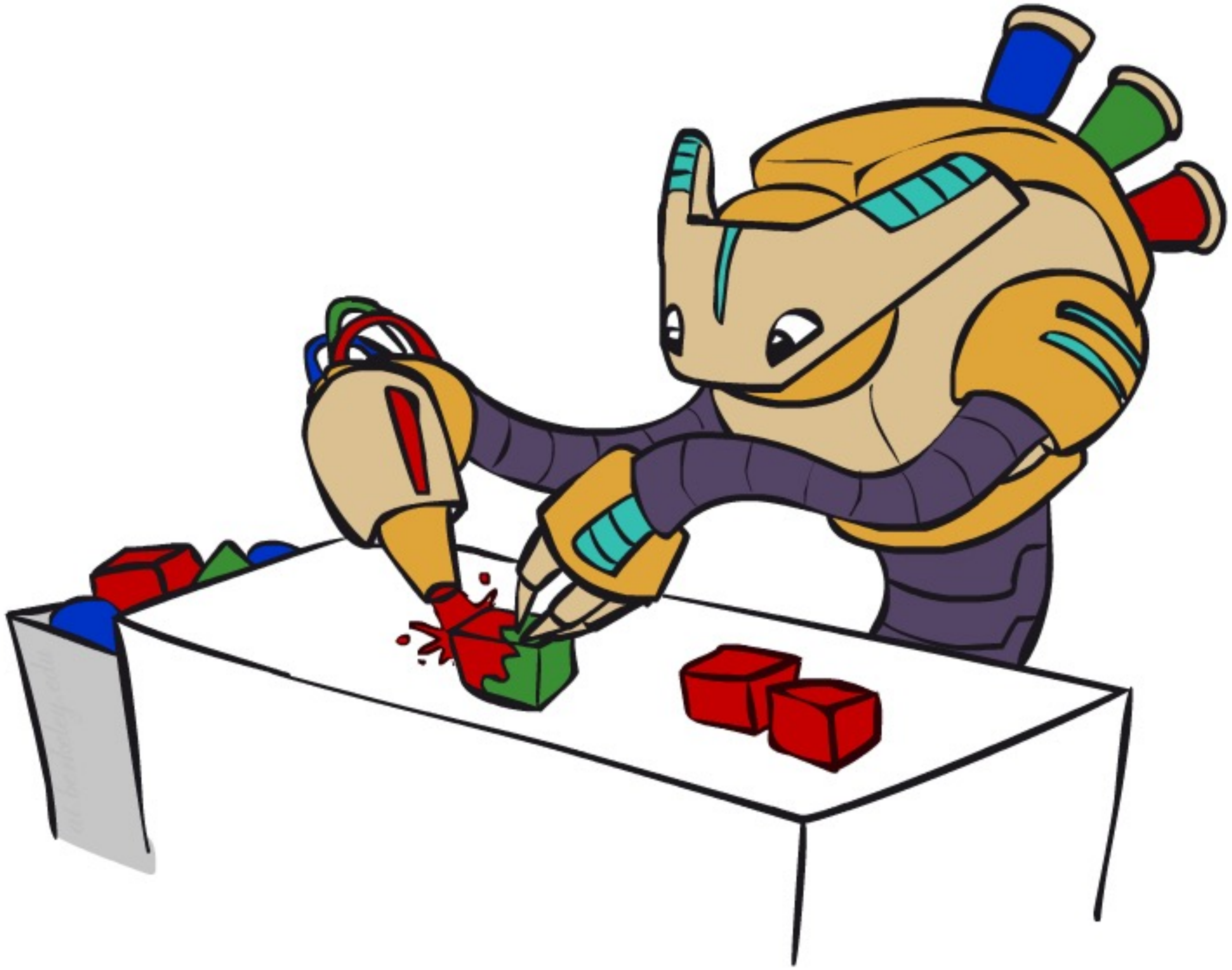
Likelihood Weighting

- 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
    - E.g., suppose evidence is a video of a traffic accident
  - With evidence in k leaf nodes, weights will be O(2^{-k})
  - With high probability, one lucky sample will have much larger weight than the others, dominating the result
- We would like each variable to "see" all the evidence!

Gibbs Sampling



Gibbs Sampling

- **Procedure**: keep track of a full instantiation $x_1, x_2, \ldots, x_n$.
  1. Start with an arbitrary instantiation consistent with the evidence.
  2. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed.
  3. Keep repeating this for a long time.

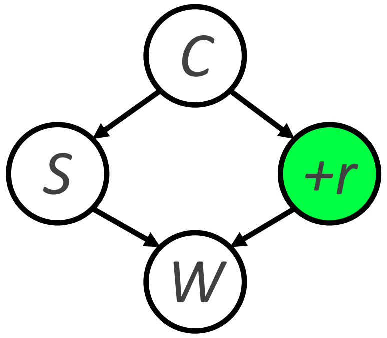
- **Property**: in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution

- **Rationale**: both upstream and downstream variables condition on evidence.

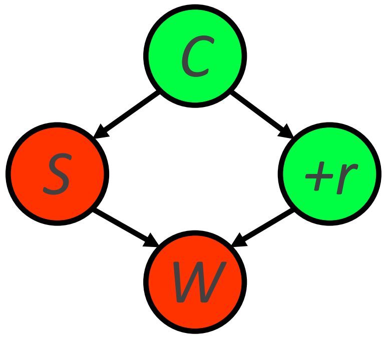
- **In contrast**: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many "effective" samples were obtained, so want high weight.

Gibbs Sampling Example: P(S | +r)

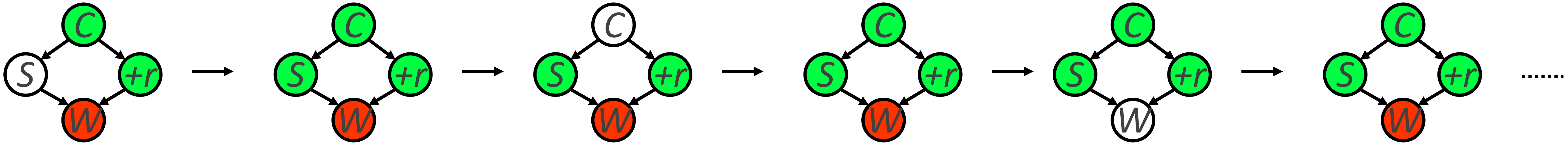
- Step 1: Fix evidence
  - R = +r



- Step 2: Initialize other variables
  - Randomly



- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P(X | all other variables)



Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

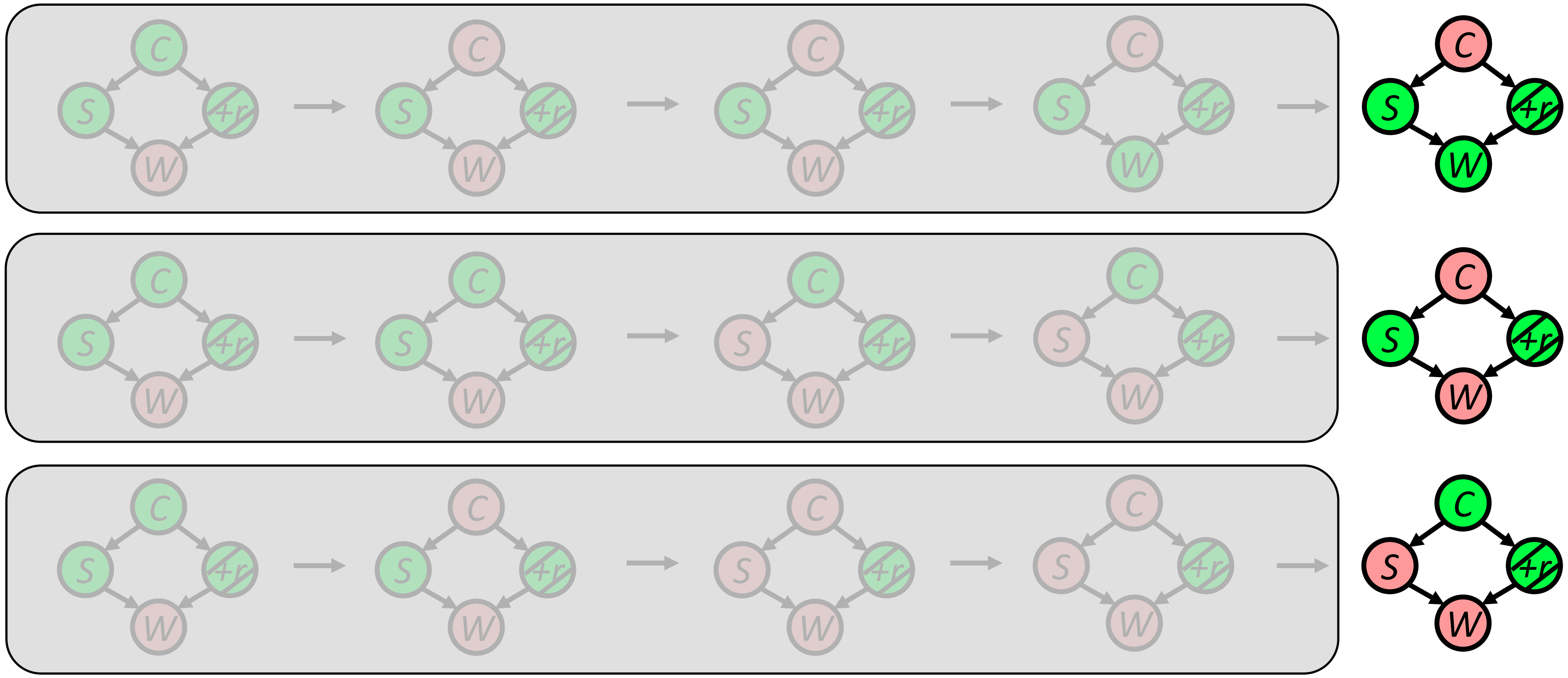
Gibbs Sampling Example: P(S | +r)

- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P(X | all other variables)
- Keep only the last sample from each iteration:

1.

2.

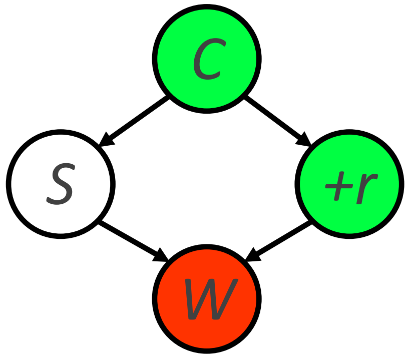
3.



Efficient Resampling of One Variable

- Sample from P(S | +c, +r, -w)

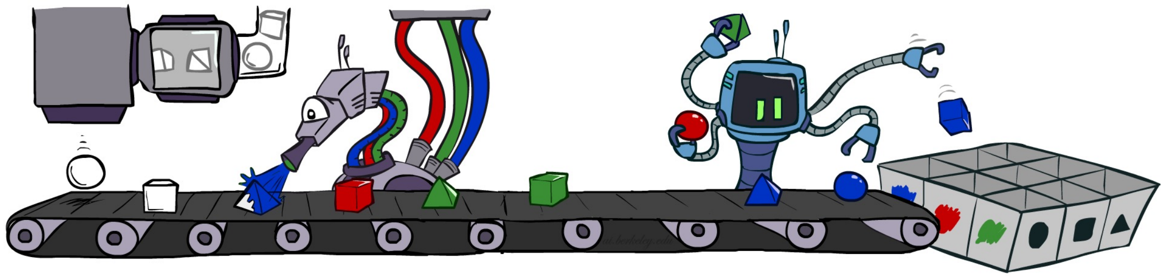
$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{P(+c)P(+r|+c)\sum_s P(s|+c)P(-w|s,+r)} \\
&= \frac{P(S|+c)P(-w|S,+r)}{\sum_s P(s|+c)P(-w|s,+r)}
\end{aligned}
$$



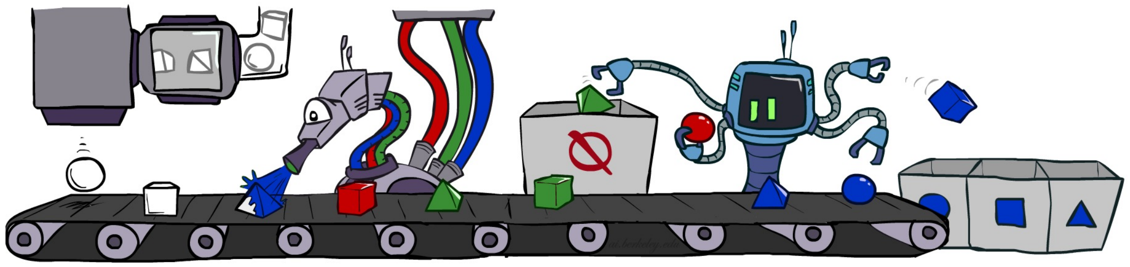
- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs that have resampled variable need to be considered, and joined together

Bayes' Net Sampling Summary

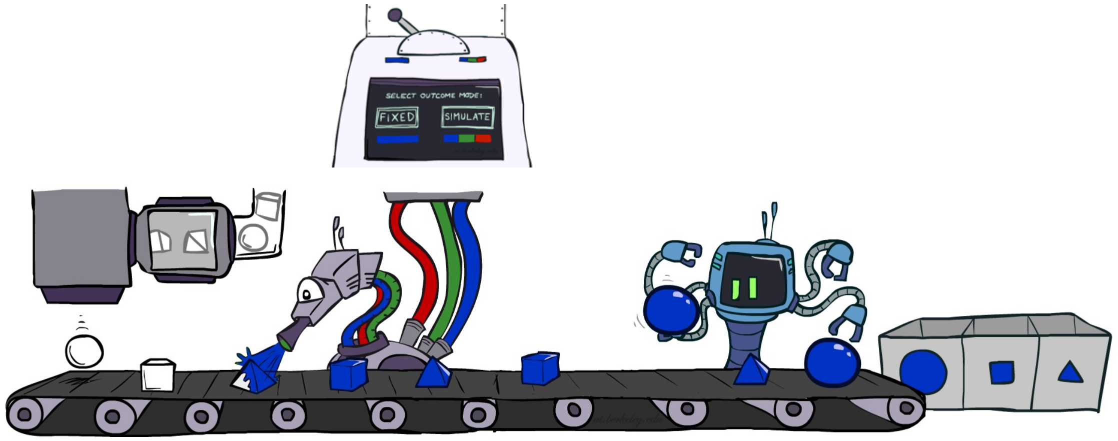
- Prior Sampling P



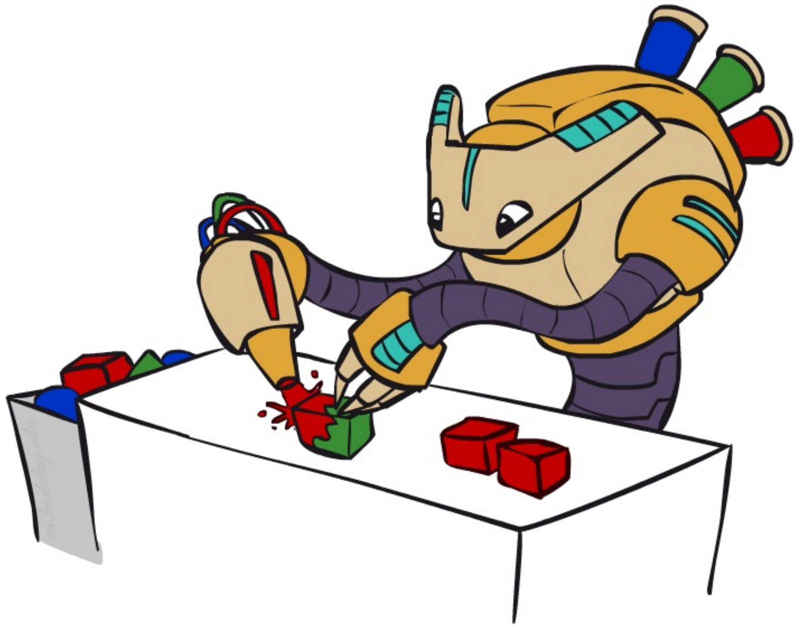
- Rejection Sampling P(Q | e)



- Likelihood Weighting P(Q | e)



- Gibbs Sampling P(Q | e)



Further Reading on Gibbs Sampling*

- Gibbs sampling produces sample from the query distribution P(Q | e) in limit of re-sampling infinitely often

- Gibbs sampling is a special case of more general methods called Markov chain Monte Carlo (MCMC) methods

  - Metropolis-Hastings is one of the more famous MCMC methods (in fact, Gibbs sampling is a special case of Metropolis-Hastings)

- You may read about Monte Carlo methods – they're just sampling