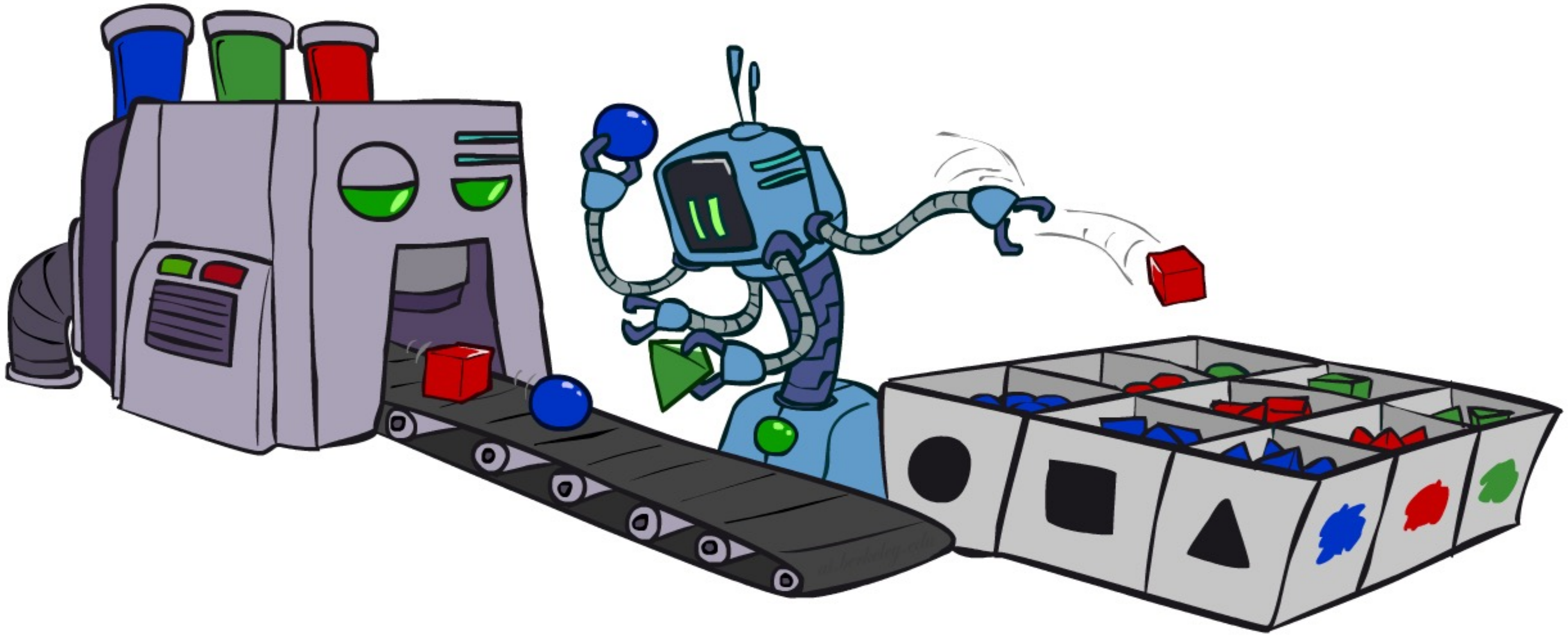
CS 3317: Artificial Intelligence

Bayes Nets: Sampling

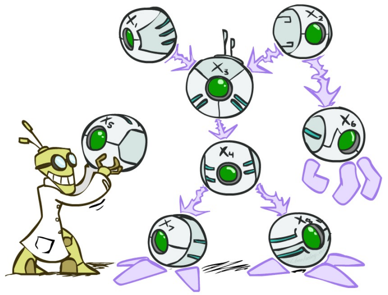


Instructor: Panpan Cai

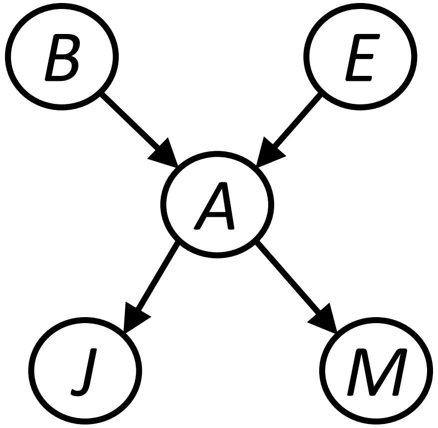
[Slides adapted from UC Berkeley CS188]

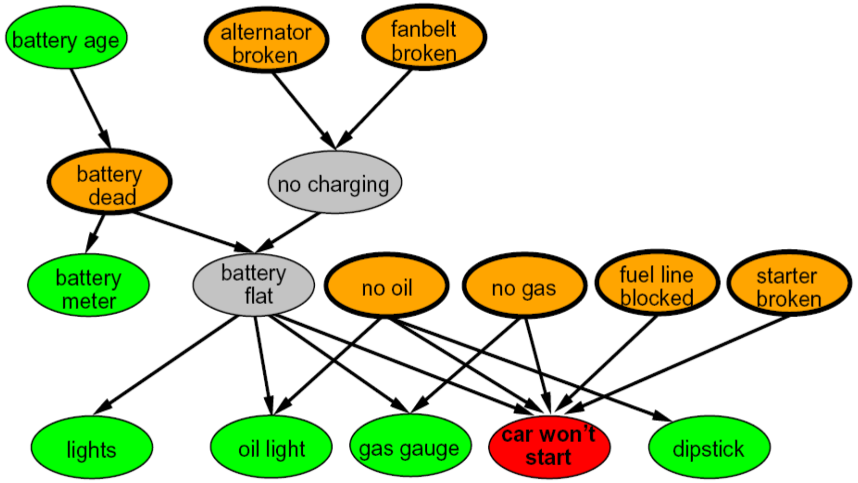
Bayes Nets: Graphical Model

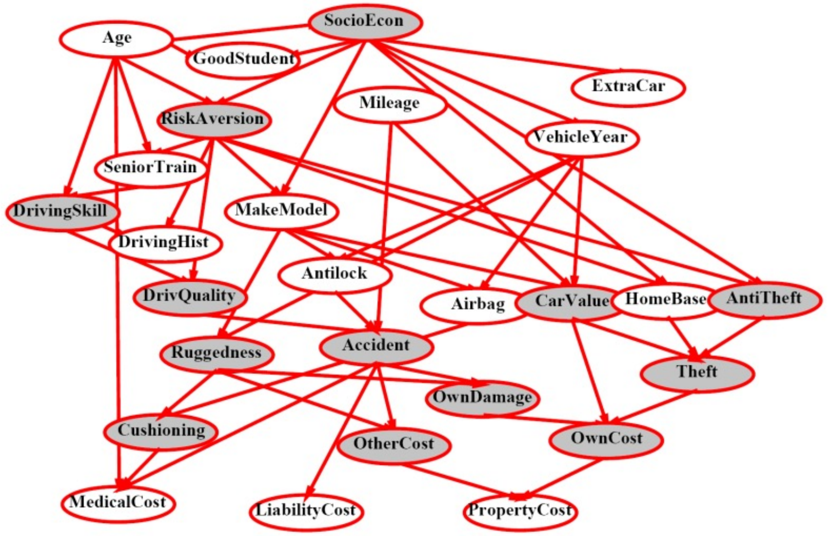
- A directed acyclic graph (DAG) consisting of random variables
  - A graphical model built from experience/data



- Examples:



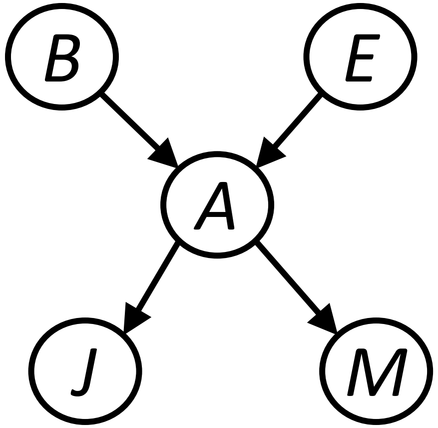




Bayes Nets: Local CPTs

- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

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



- Examples:

B	P(B)
+b	0.001
-b	0.999

E	P(E)
+e	0.002
-e	0.998

A	J	P(J|A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

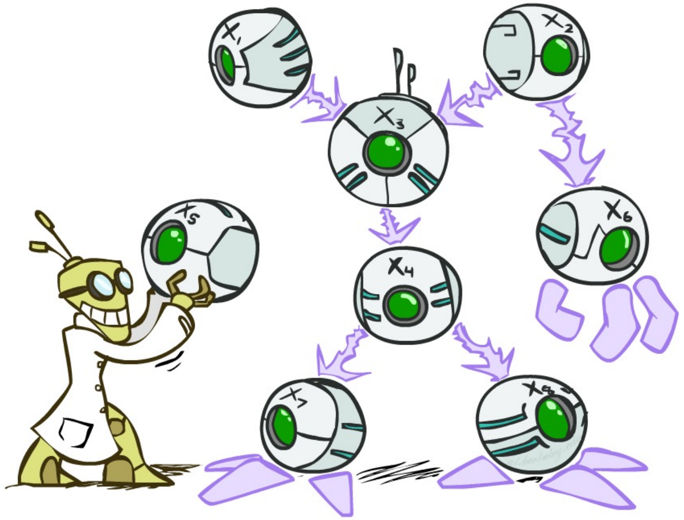
A	M	P(M|A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

Bayes Nets: Probabilities and Independences

- 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:

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



- The following conditional independence assumptions are made immediately, when given a BN graph:

$$P(x_i|x_1, \ldots x_{i-1}) = P(x_i|parents(X_i))$$

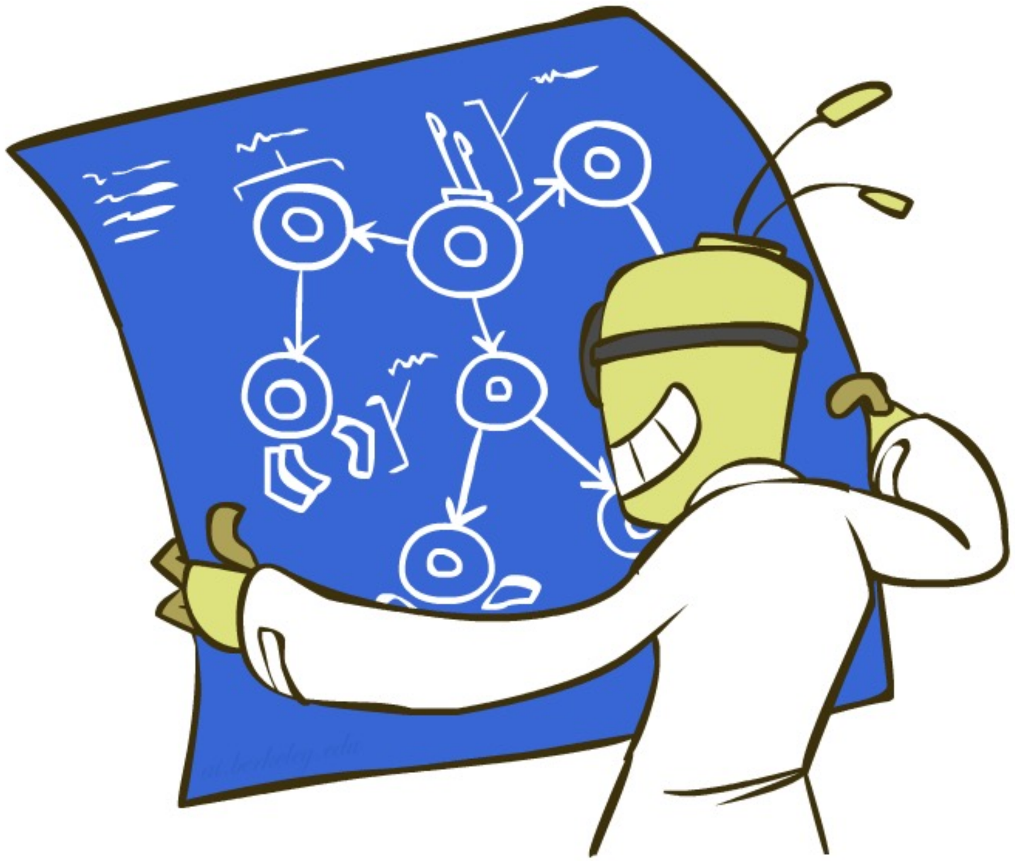
  - X_i is independent of all non-parent variables coming before it, given its parents

“Read” All Independences

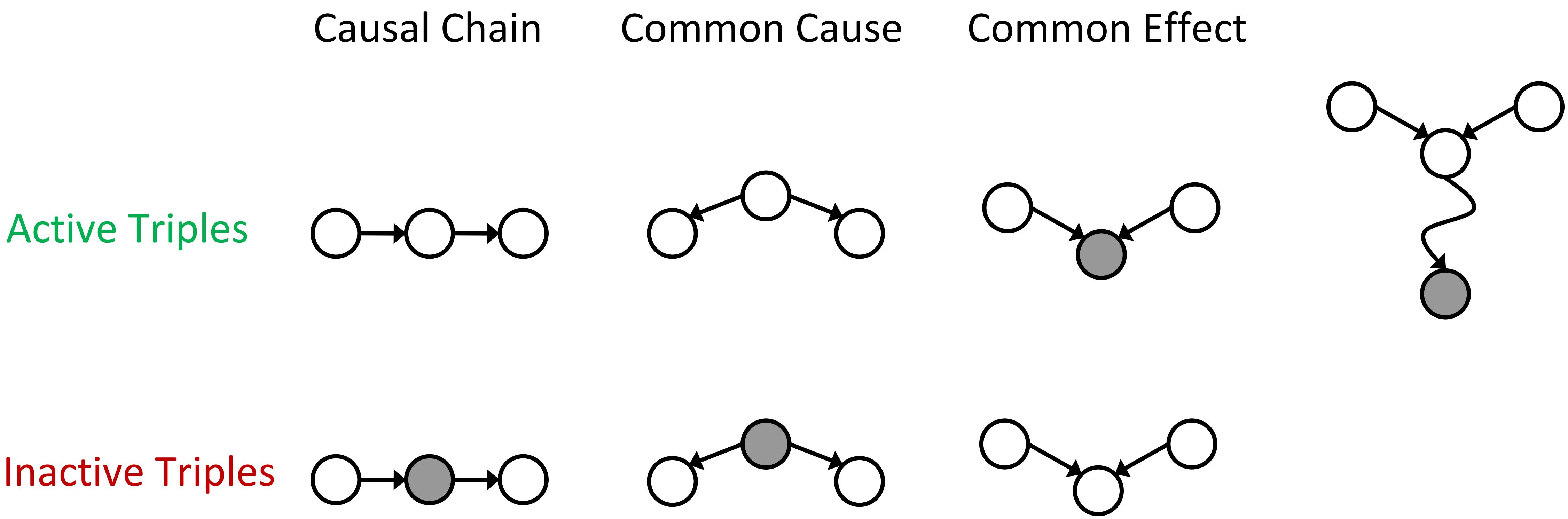
- Given a Bayes net structure, can run d-separation algorithm to read a *complete* list of conditional independences of the form:

$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

- This list determines the set of probability distributions that a BN graph can represent

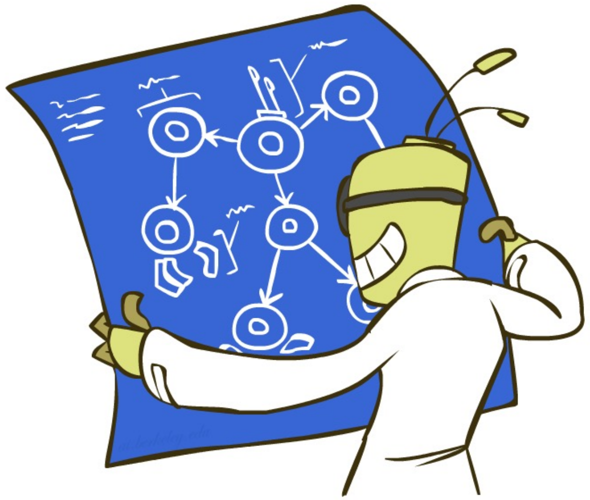


D-Separation: Canonical Cases



D-Separation: Algorithm

- Query: $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$?



- Check all *undirected* paths between X_i and X_j
  - If one or more path active, independence broken

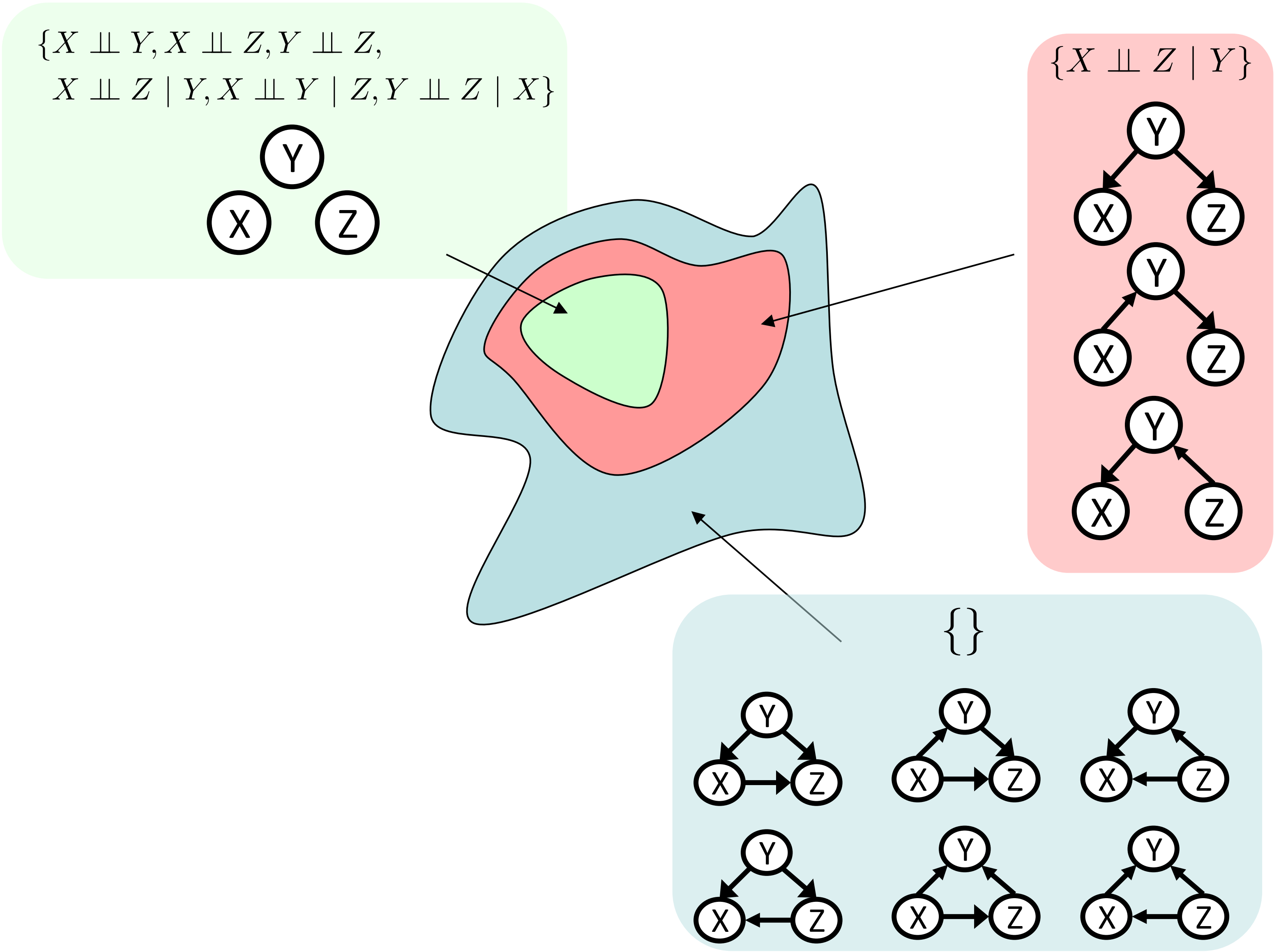
$$X_i \not\perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

  - Otherwise (i.e. if all paths are inactive), independence guaranteed

$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

Topology Limits Distributions

- Given some graph topology G, only a subset of joint distributions can be expressed
- Adding arcs increases the set of distributions, but at a cost of increased complexity
- Full conditioning can encode any distribution



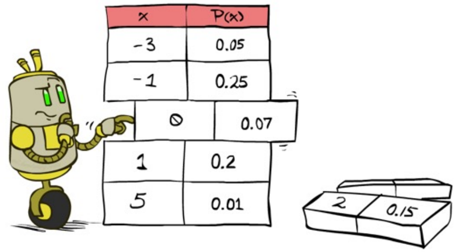
Bayes Nets

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

Exact Inference: Inference by Enumeration

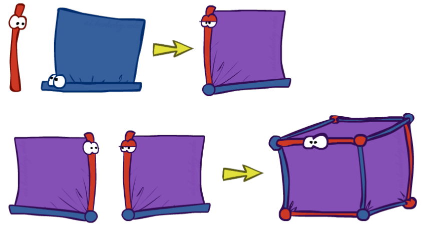
- Query: $P(Q|E_1 = e_1, \dots E_k = e_k)$

- Start with initial factors:
  - Local CPTs (but instantiated by evidence)



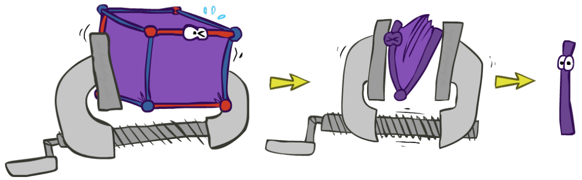
- Join all factors to get a complete joint

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



- Sum out all hidden variables

$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} P(Q, h_1 \dots h_r, e_1 \dots e_k)$$

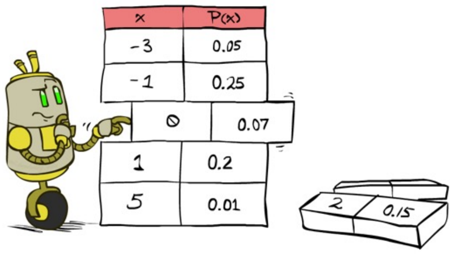


- Normalize

$$\times \frac{1}{Z}$$

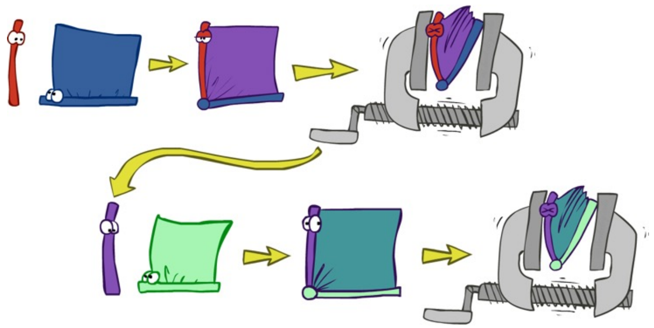
Exact Inference: Variable Elimination

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$



- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H



- Join all remaining factors and normalize

$$\times \frac{1}{Z}$$

Worst-Case Complexity: NP Hard!

- CSP: assign variables so that the sentence is true

Clause need be true

$$(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_3 \lor \neg x_4) \land (x_2 \lor \neg x_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor \neg x_5) \land (x_2 \lor x_5 \lor x_7) \land (x_4 \lor x_5 \lor x_6) \land (\neg x_5 \lor x_6 \lor \neg x_7) \land (\neg x_5 \lor \neg x_6 \lor x_7)$$

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

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

$$\ldots$$

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

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

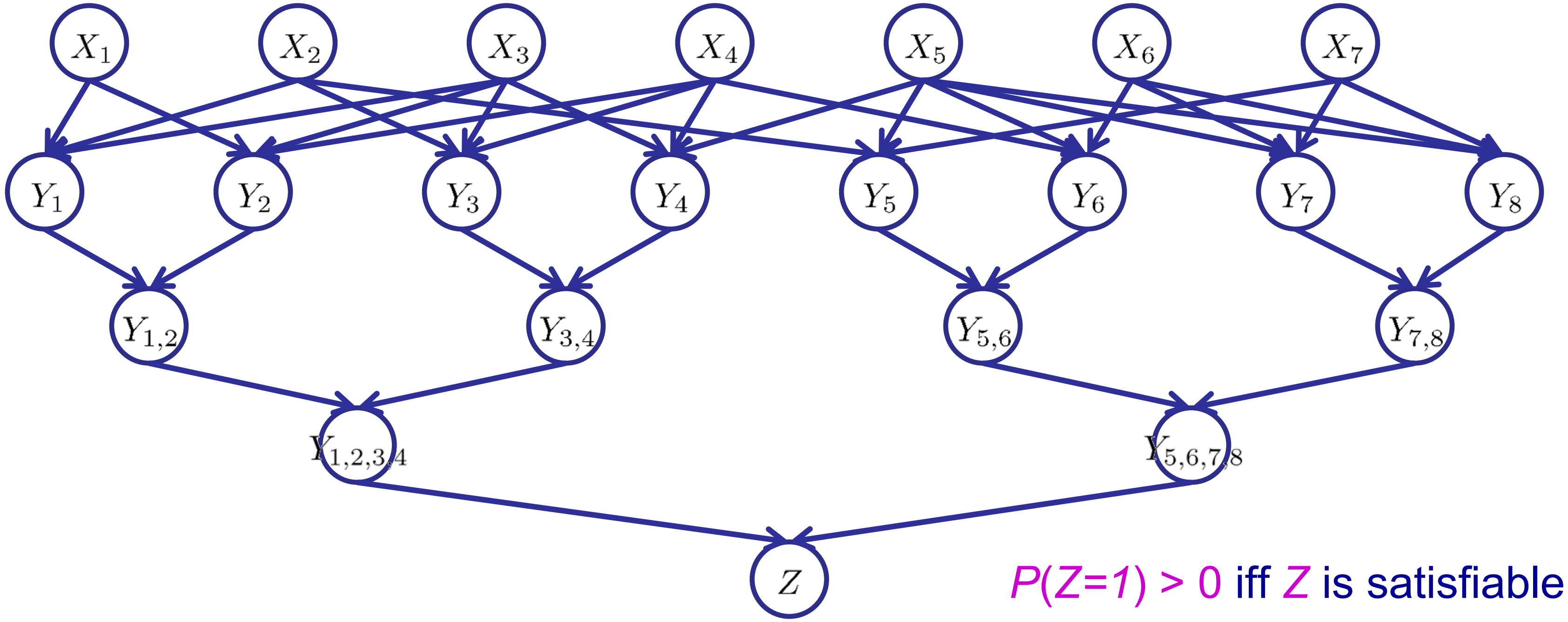
$$\ldots$$

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

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

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

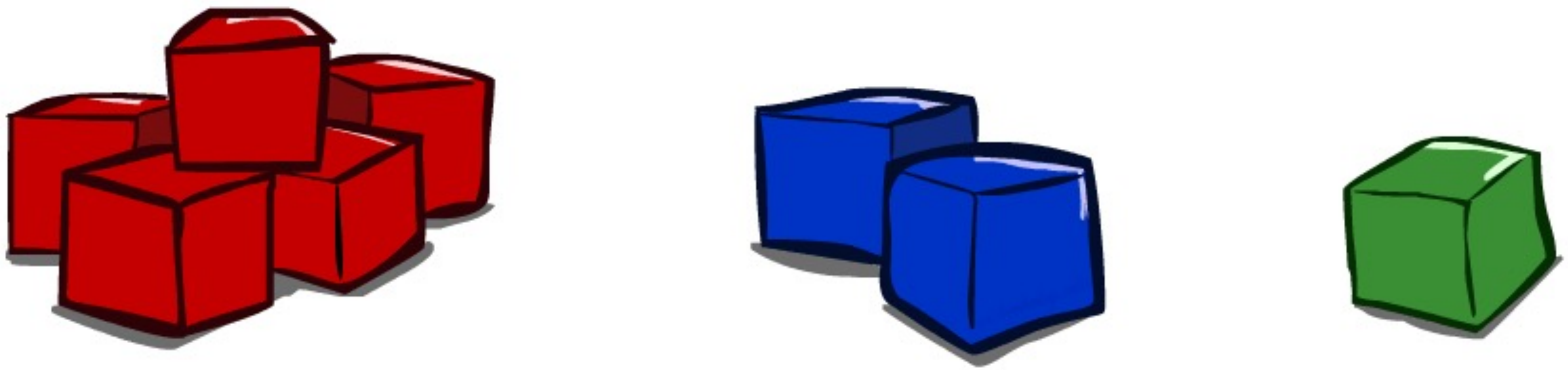
$$Z = Y_{1,2,3,4} \land Y_{5,6,7,8}$$



P(Z=1) > 0 iff *Z* is satisfiable

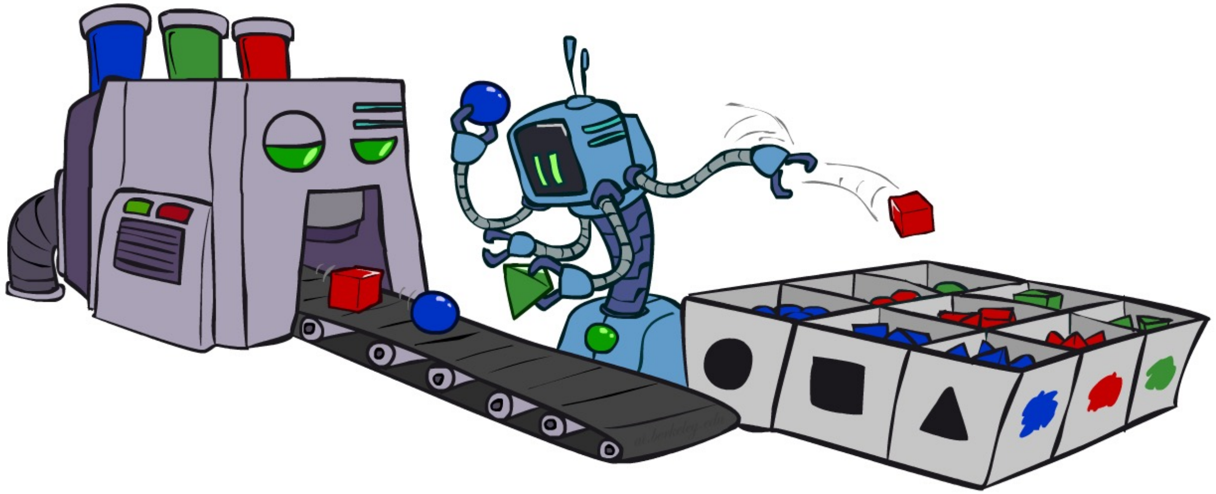
- If we can answer whether P(z) equal to zero or not, we answered whether the 3-SAT problem has a solution (which is NP hard!)
- Hence probabilistic inference in Bayes nets is NP-hard.
  - No known efficient (exact) solution *in general*.

Approximate Inference: Sampling



Sampling

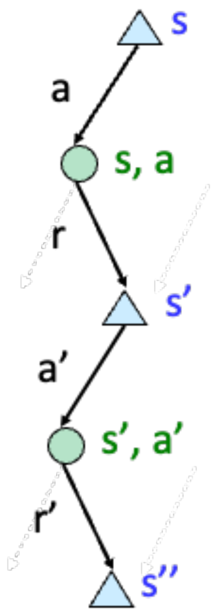
- Sampling is just repeated random simulation
  - X is a random variable with an underlying distribution $P(X)$
  - Treat the model $P(X)$ as a simulator, where each simulation generates a sample x_i
  - In the limit, the *frequency* of seeing a particular value x converges to the *probability* of x



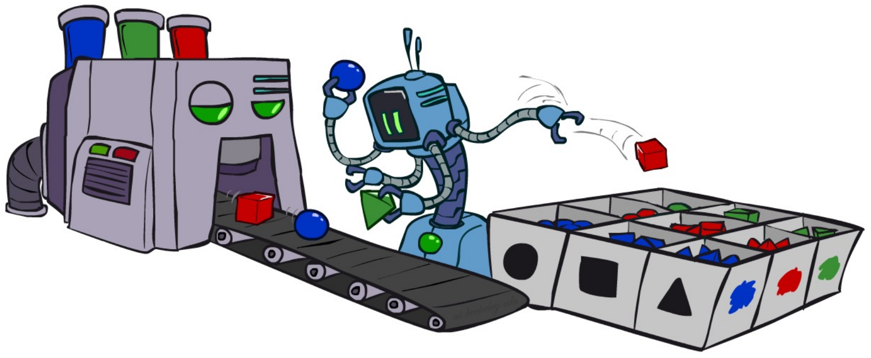
- The simulator can also host:
  - Joint distributions
  - Conditional distributions
  - Bayes Nets (with evidences or not)
  - Or any *black-box* random process…

Sampling

- Why sample?
  - *Learning*: estimate a distribution that you don't know
    - *Model-based RL: sample transitions, learn models*
    - *Q-learning: sample transitions, learn Q-functions*



  - *Speedup inference*: estimating using samples is faster than computing the exact answer (e.g. with variable elimination)
    - *This lecture!*



How to Sample?

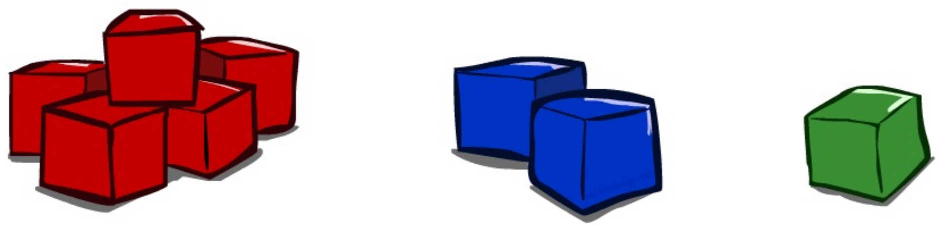
- Sampling from given distribution
  - Step 1: Get sample u from uniform distribution over [0, 1)
    - E.g. random() in python
  - Step 2: Convert u into an outcome
    - Associate each outcome with a *sub-interval* of [0,1) with *size* equal to *probability* of the outcome

- Example

C	P(C)
red	0.6
green	0.1
blue	0.3

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

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

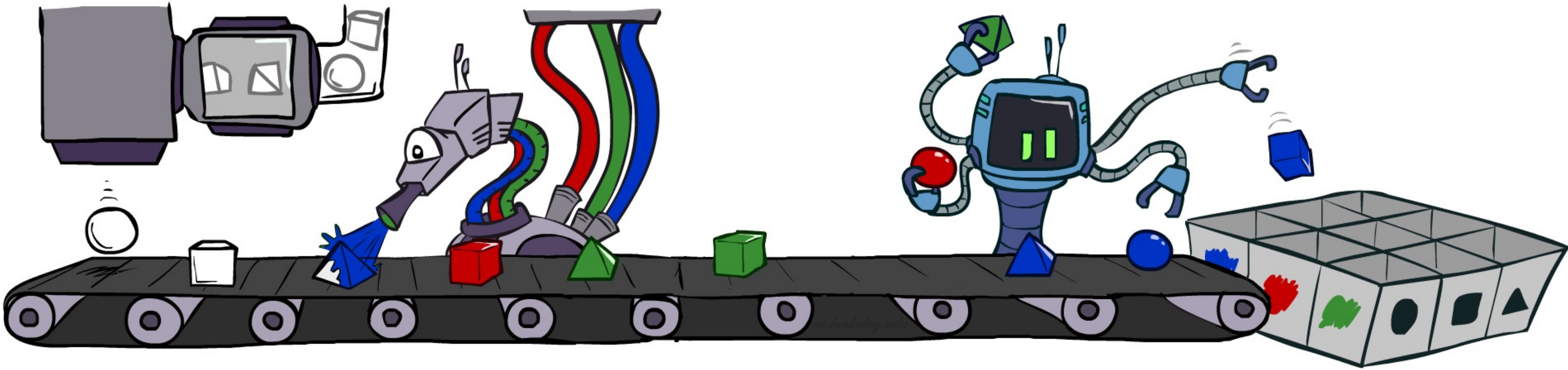


Sampling in Bayes Nets

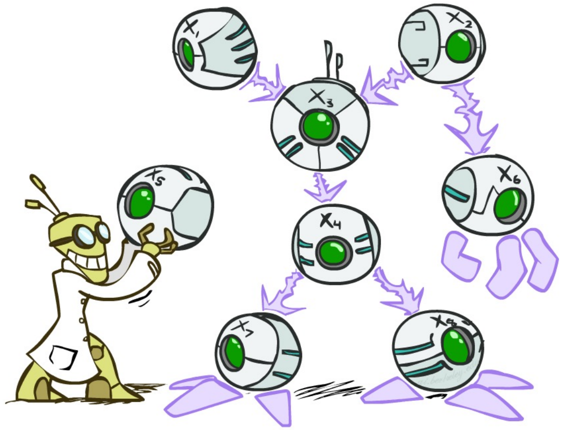
- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling

Prior Sampling

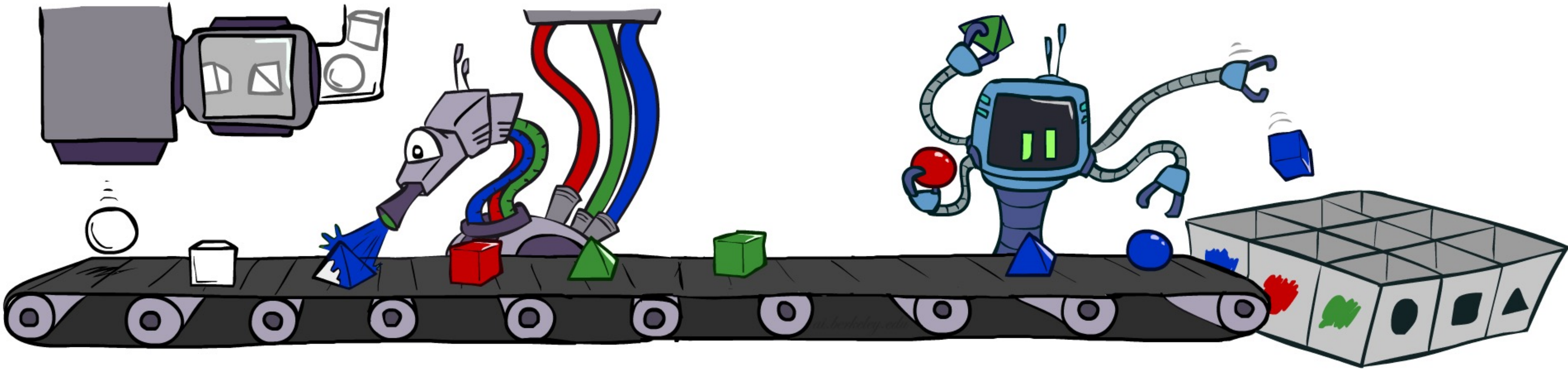
"Shaping machine" *"Coloring machine"* *"Sorting machine"*



Prior Sampling



- For i = 1, 2, …, n in topological order
  - Sample x_i from $P(X_i \mid Parents(X_i))$
- Return $(x_1, x_2, \ldots, x_n)$



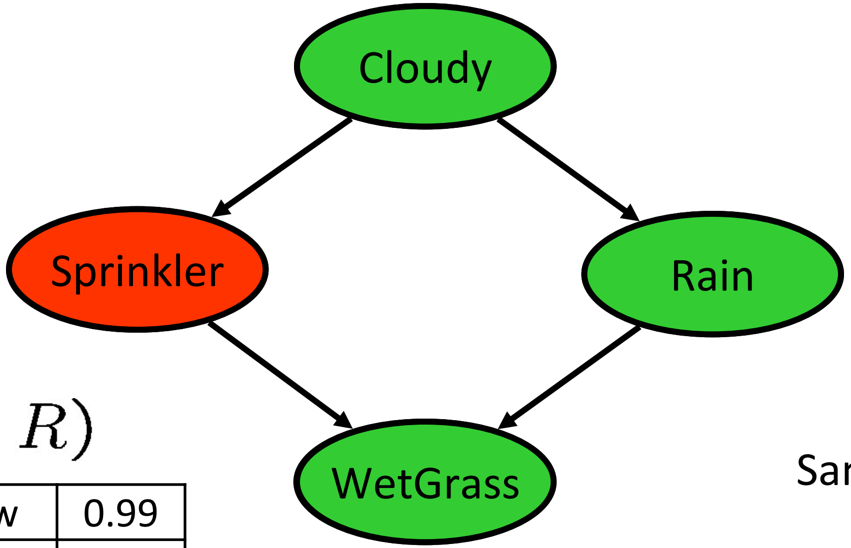
Example: Prior Sampling

$P(C)$

+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

- This process generates samples with probability:

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

Probability of independent events happening together

Product of probability of each event

BN's joint probability

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

- Then

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

Converge

- I.e., the sampling procedure is consistent

Example: Prior Sampling

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

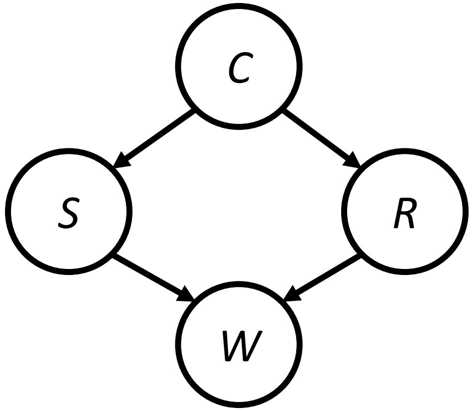
  +c, -s, +r, +w

  +c, +s, +r, +w

  -c, +s, +r, -w

  +c, -s, +r, +w

  -c, -s, -r, +w



- If we want to know P(W)
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
- Can estimate anything else, too
  - P(C | +w)? P(C | +r, +w)?
  - What about P(C | -r, -w)?
- Can also use this to estimate expected value of a function E[f(X)] - *Monte Carlo Estimation*

Connect: Inference by Enumeration

- Query: $P(Q|E_1 = e_1, \dots E_k = e_k)$

- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

- Join all factors to get a complete joint

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

- Sum out all hidden variables

$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} P(Q, h_1 \dots h_r, e_1 \dots e_k)$$

- Normalize

2. Select only relevant samples

1. Sample complete values

3. Counting (sum up weights of samples)

4. Normalize

Problem of Prior Sampling

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

  +c, -s, +r, +w

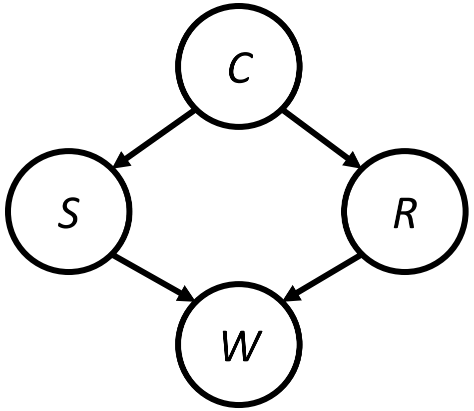
  +c, +s, +r, +w

  -c, +s, +r, -w

  +c, -s, +r, +w

  -c, -s, -r, +w

  Hosting all samples here!

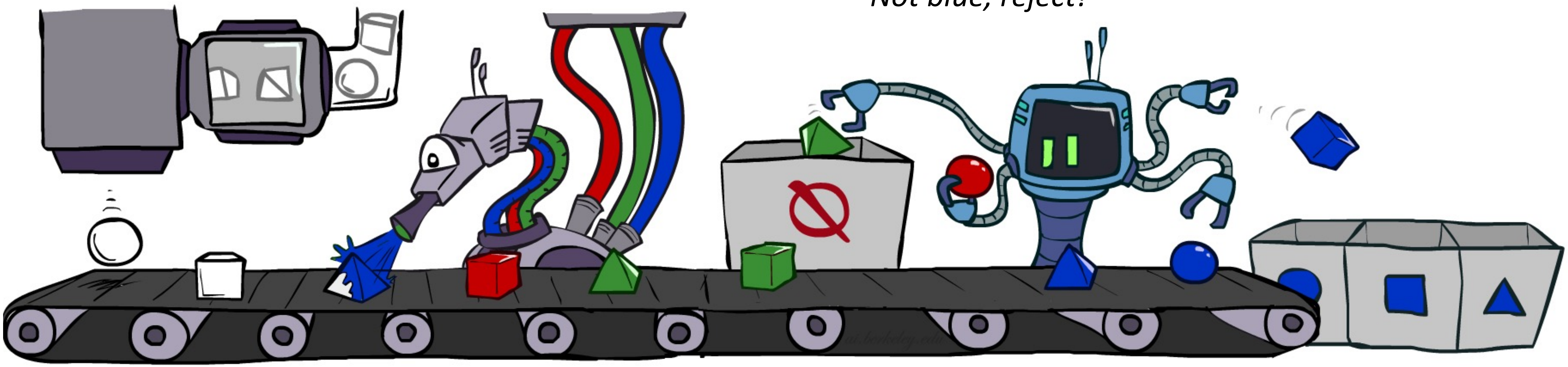


- If we want to know P(W)
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
- Can estimate anything else, too
  - P(C | +w)? P(C | +r, +w)?
  - What about P(C | -r, -w)?

- *Many samples are not consistent with evidence*
- *What we need are just numbers, e.g.,* $N_{PS}(+c \mid +r, +w)$

Rejection Sampling

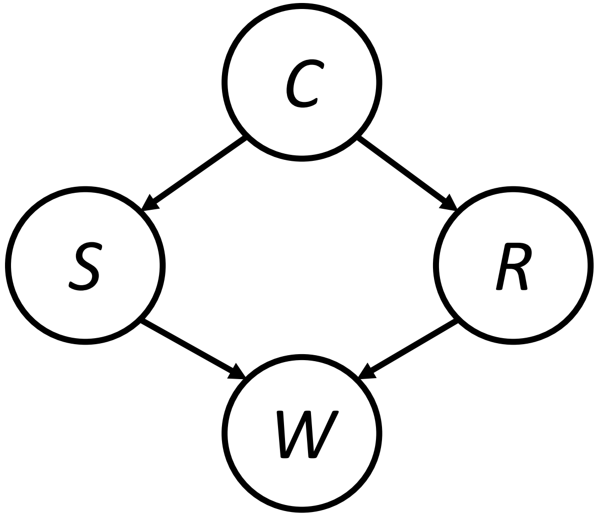
"Not blue, reject!"



Rejection Sampling

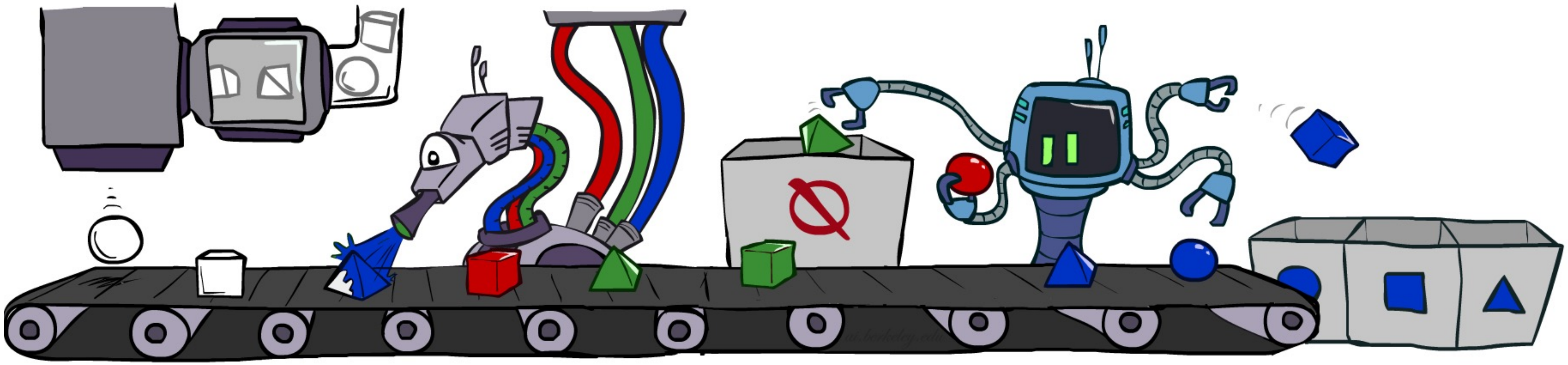
- Let's say we want P(C)
  - Nothing to reject, just track counts of c as we go (no need to store samples)

- Let's say we want P(C | +s)
  - Count c, but reject samples which don't have S=+s
  - This is called *rejection sampling*
  - Benefit: we can filter out samples early!
  - Property: also *consistent* for conditional probabilities (i.e., correct in the limit)



Rejection Sampling

- Input: evidence instantiation
- For $i = 1, 2, \ldots, n$ in topological order
  - Sample x_i from $P(X_i \mid Parents(X_i))$
  - If x_i not consistent with evidence
    - Return (*reject: no sample is generated in this cycle)*
- Return $(x_1, x_2, \ldots, x_n)$



Problem of Rejection Sampling

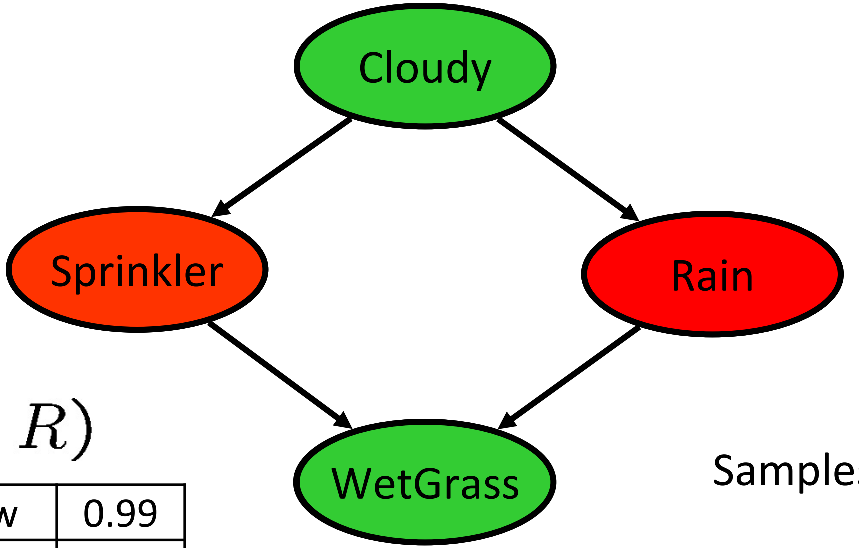
- Let's say we want P(+w|+c,-s,-r)

$P(C)$

+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

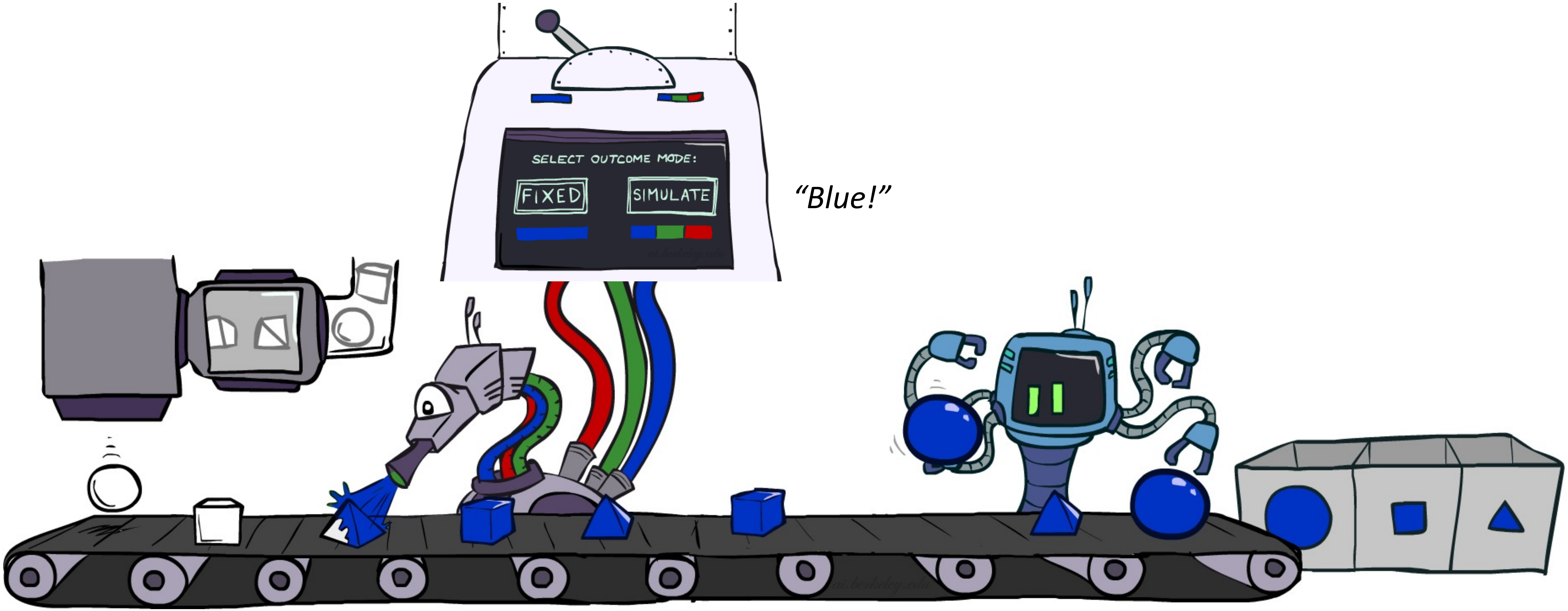
+c, +s, +r, -w

…

How many samples to see the first +c, -s, -r, +w?

Likelihood Weighting

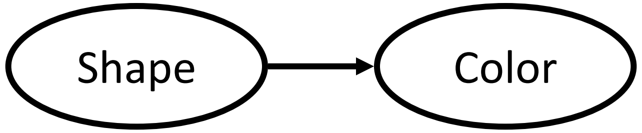
“Hacked machine”



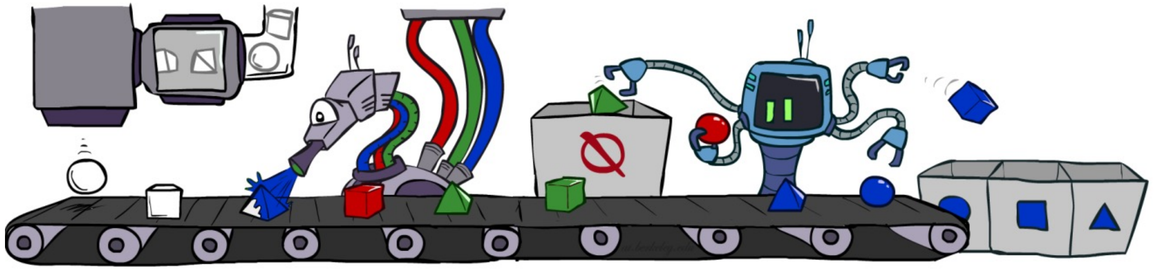
“Blue!”

Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Consider P(Shape | blue)

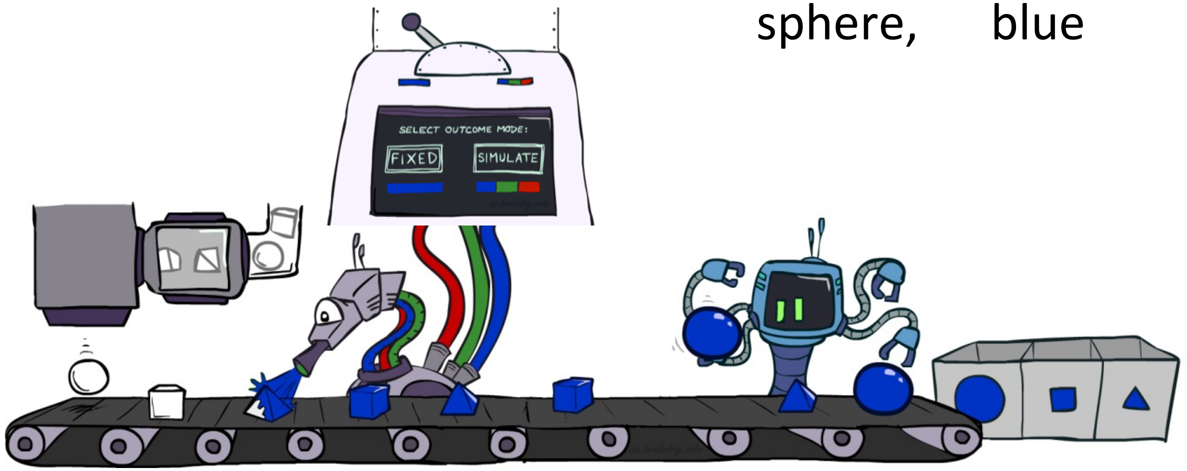


~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~



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

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



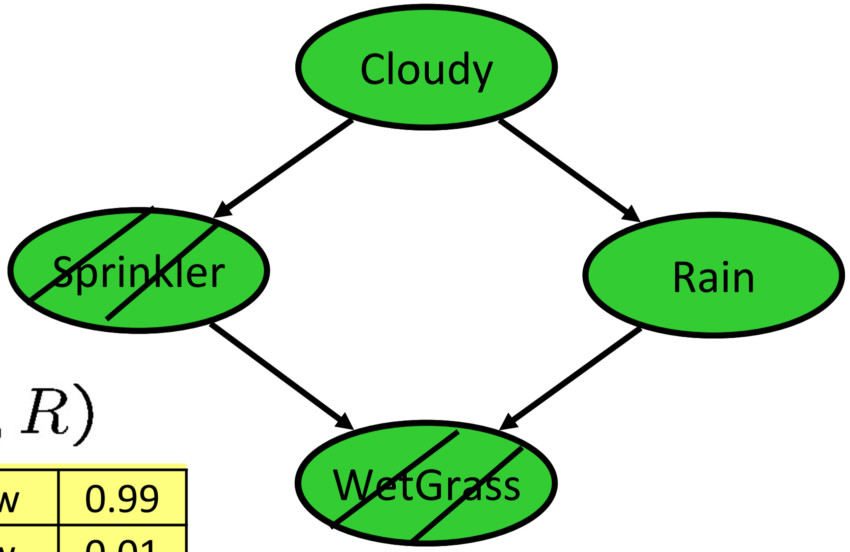
Likelihood Weighting

$P(C)$

+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

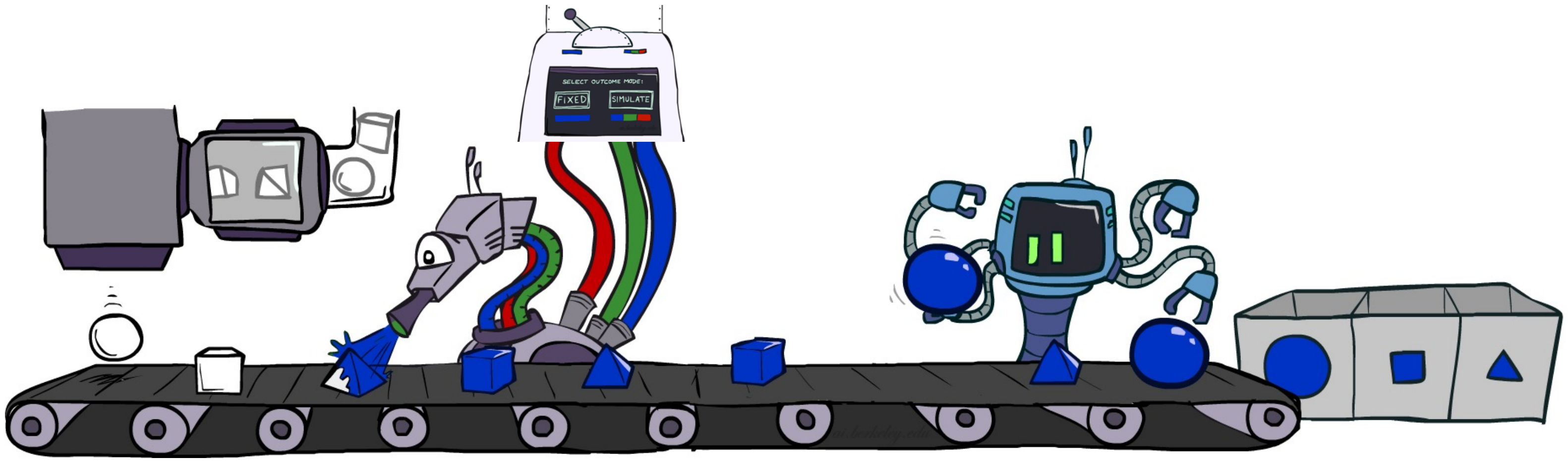
Samples:	Weights:
+c, +s, +r, +w	w = 1.0 x
-c, +s, -r, +w	w = 1.0 x 0.5 x 0.90
…	

Likelihood Weighting

- Input: evidence instantiation
- $w = 1.0$
- for $i = 1, 2, \ldots, n$ in topological order
  - if X_i is an evidence variable
    - X_i = observation 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 $(x_1, x_2, \ldots, x_n)$, w

Counting

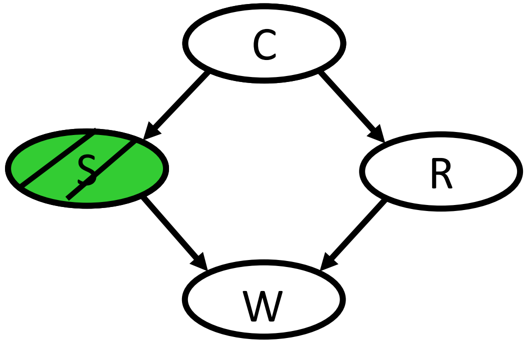
Summing up weights



Likelihood Weighting

- Sampling distribution if z sampled and evidence e fixed

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



- Now, correct with weights

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

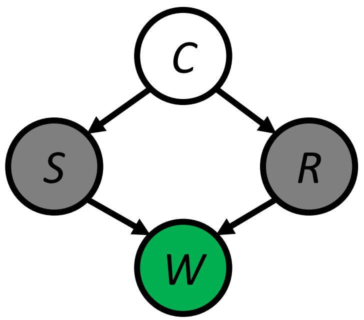
- Combining together, weighted sampling distribution becomes *consistent*

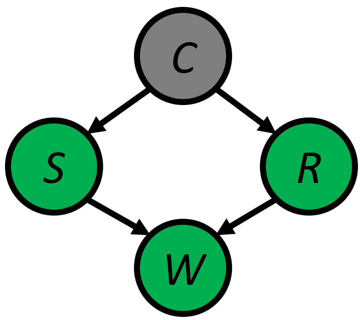
$$S_{\text{WS}}(z,e) \cdot w(z,e) = \prod_{i=1}^{l} P(z_i|\text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i|\text{Parents}(e_i))$$

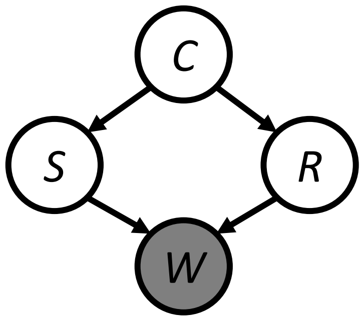
$$= P(\mathbf{z},\mathbf{e})$$

Likelihood Weighting

- Likelihood weighting is helpful
  - We have taken evidence into account as we generate the sample
    - *E.g. here, W's value will be picked based on fixed evidence values of S, R*
  - More of our samples reflect the state of the world suggested by the evidence
- Likelihood weighting doesn't solve all our problems
  - Fixation of evidence influences the choice of *downstream* variables, but not *upstream* ones
    - *W is more likely to match the evidence c*
    - *C isn't more likely to match the evidence w*
- We would like to consider the influence of fixed evidence when sampling all variables (Gibbs sampling!)







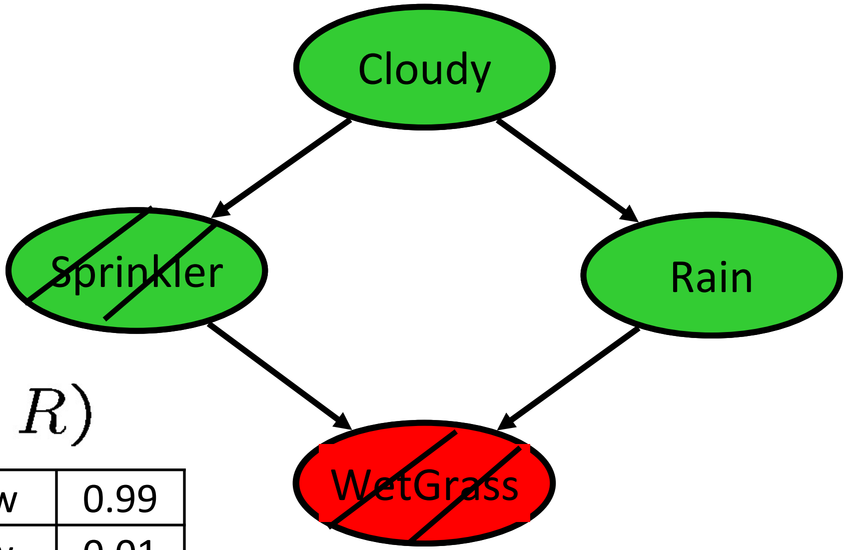
Problem of Likelihood Weighting

$P(C)$

+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: Weights:

+c, +s, +r, -w w = 1.0 x 0.1 x 0.01 = 0.001

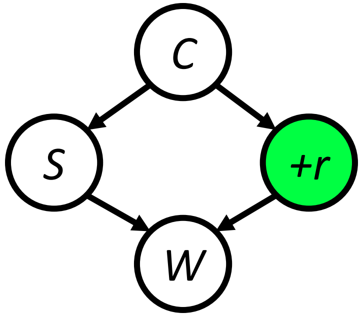
Small weights!

Gibbs Sampling

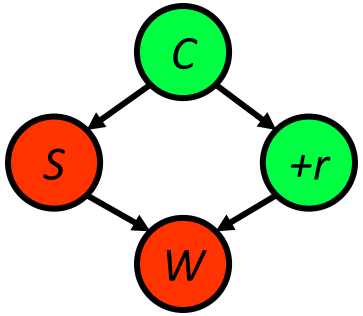
"Iterative update!"

Gibbs Sampling Example: P(S | +r)

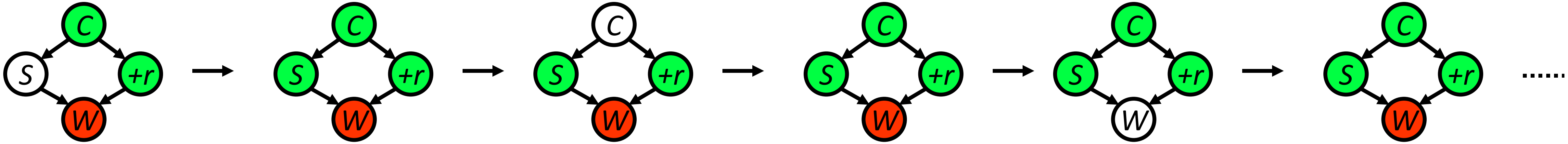
- Step 1: Fix evidence
  - R = +r



- Step 2: Initialize other variables
  - Randomly



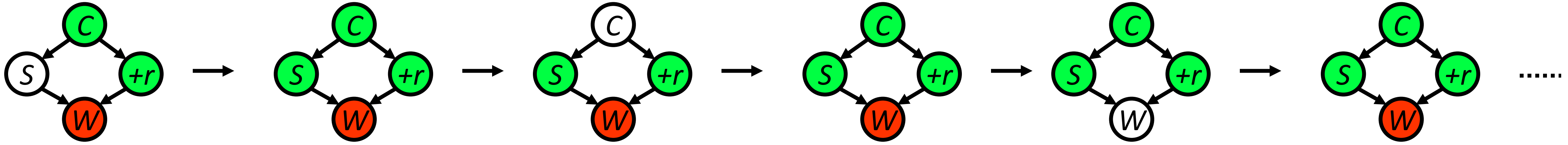
- Steps 3: Repeat
  - Choose a non-evidence variable X
  - *Resample* X from P(X | all other variables)*



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

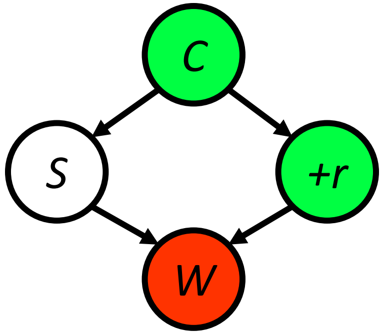
Gibbs Sampling

- ***Procedure***: *keep track of a full instantiation $x_1, x_2, \ldots, x_n$.*
  - *Start with an arbitrary instantiation consistent with the evidence*
  - *Resample one (non-evidence) variable at a time, conditioned on all the rest*
  - *Keep repeating this for a long time*
- ***Property***: *in the limit (repeated unlimited times), the resulting set of samples reflect the correct distribution (i.e. $P(Q|e)$).*



Resampling of One Variable

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

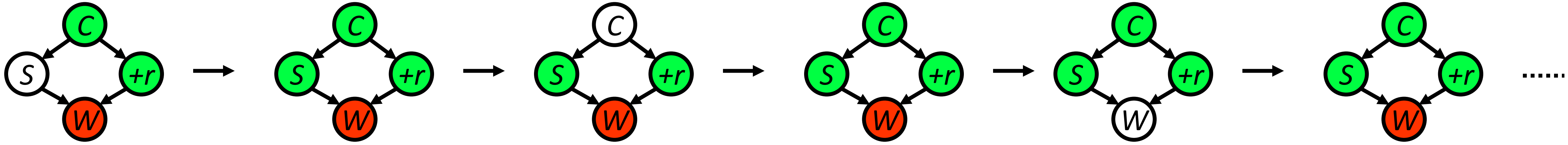


$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{P(+c)P(+r|+c)\sum_s P(s|+c)P(-w|s,+r)} \\
&= \frac{P(S|+c)P(-w|S,+r)}{\sum_s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

- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs with *the resampled variable* need to be considered
  - Join them together and normalize

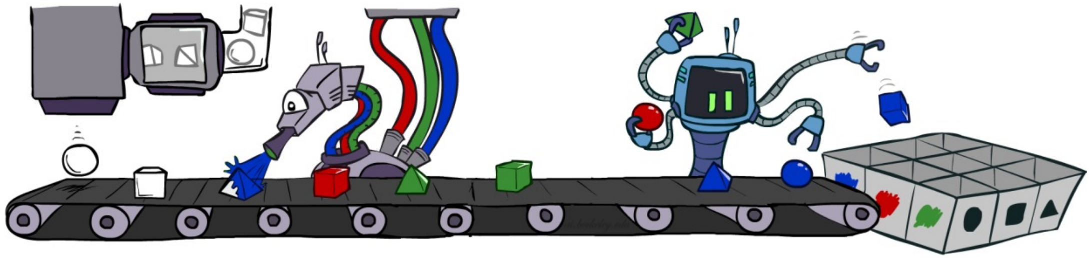
Gibbs Sampling

- ***Rationale***: pick both *upstream* and *downstream* variables condition on evidence
  - All samples match the evidence, and are equally "effective" (with same weights)
  - Avoided small weights in likelihood weighting!

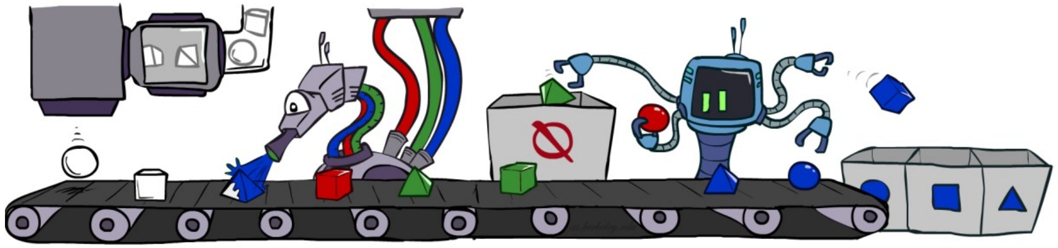


Bayes Net Sampling Summary

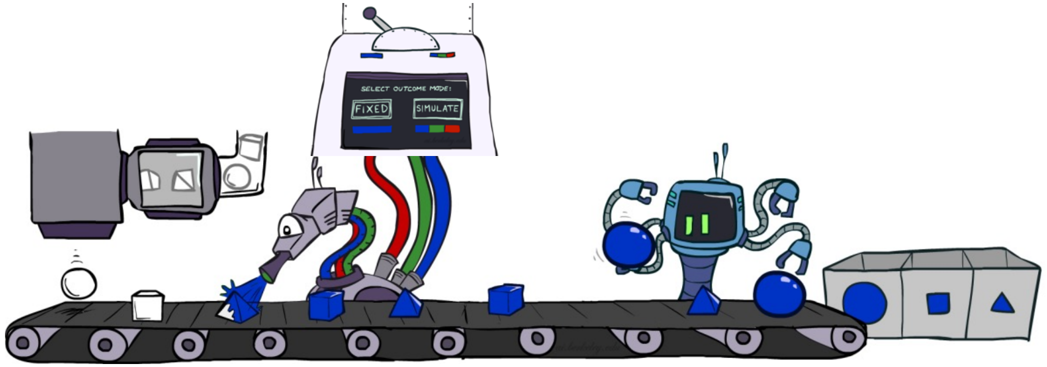
- Prior Sampling P(Q)



- Rejection Sampling P(Q | e)



- Likelihood Weighting P(Q | e)



- Gibbs Sampling P(Q | e)