CS 7180: Behavioral Modeling and Decision-making in AI

Bayesian Networks for Dynamic and/or Relational Domains

Prof. Amy Sliva

October 12, 2012

World is not only uncertain, it is dynamic

- Beliefs, observations, and relationships are not static
  - Diabetic blood sugar and insulin levels
  - Economic activity of a nation
  - Tracking vehicle location

- Represent world as a series of snapshots or **time slices**
  - Temporal state-space model—keep track of value of evidence and outcome variables at each time slice
  - **State-variable** representation

- Assume time is **bounded discrete instants**
  - **Step size** depends on the domain (i.e., hour vs. day)
  - Fixed interval between time slices is fixed (represent by integers)
  - Starts at time $t = 0$

Representing the state at a time slice

- Two types of state variables
- X_t = **Unobserved** random variables at time ***t***
  $Rain_t$, $BloodSugar_t$, $StomachContents_t$, $QualityOfLife_t$

- E_t = **Observed** evidence variables at time ***t***
  $Umbrella_t$, $MeasuredBloodSugar_t$, GDP_t

  $E_t = e_t$ is the actual observation at time t

- Assume evidence starts arriving at time $t = 1$
  - Represent domain by sequences of state variables and evidence
    $R_0, R_1, R_2, \ldots$ and $E_1, E_2, E_3, \ldots$
    $X_{a:b}$ denotes variables from X_a to X_b

Representing state changes over time

- How can we reason about states over time?
  - Leverage structural features of Bayesian networks
  - What are parents?

- **Transition model**—how the world evolves over time
  - Probability of the state variables given the previous values

    $$P(X_t \mid X_{0:t-1})$$

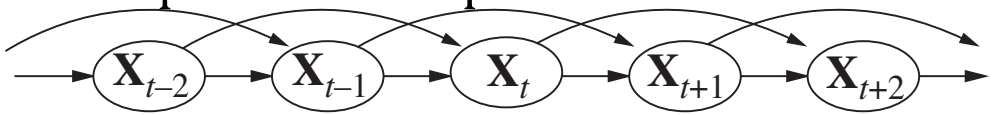
    unbounded in size as t increases
- Assume **stationary process** for transition
  - Process of change governed by rules that do not themselves change
    $P(X_t \mid X_{0:t-1})$ the same for all t—no need to recompute at time slice

- **Observation model**—how evidence is sensed over time

Markov process for transitions and observations

- **Markov assumption**
  - Current state depends on only *a finite fixed number* of previous states
  - Future conditionally independent of the past given a subset of previous states
- **Markov process** (or Markov Chain)
  - First-order Markov process—depends only on the previous state
    $P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-1})$

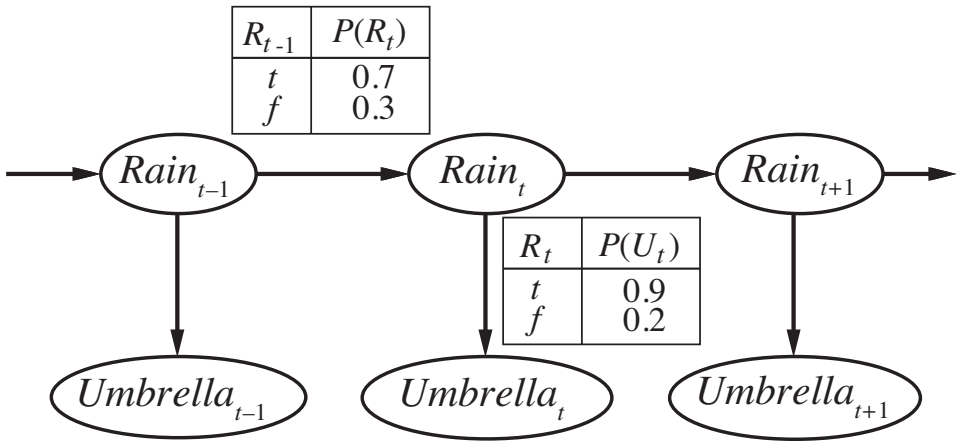
  - Second-order Markov process—depends on the previous two states
    $P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-2}, X_{t-1})$



- **Sensor Markov assumption**
  - Evidence depends only on current state
    $P(E_t \mid X_{0:t-1}, E_{0:t-1}) = P(E_t \mid X_t)$

Bayesian network with temporal model

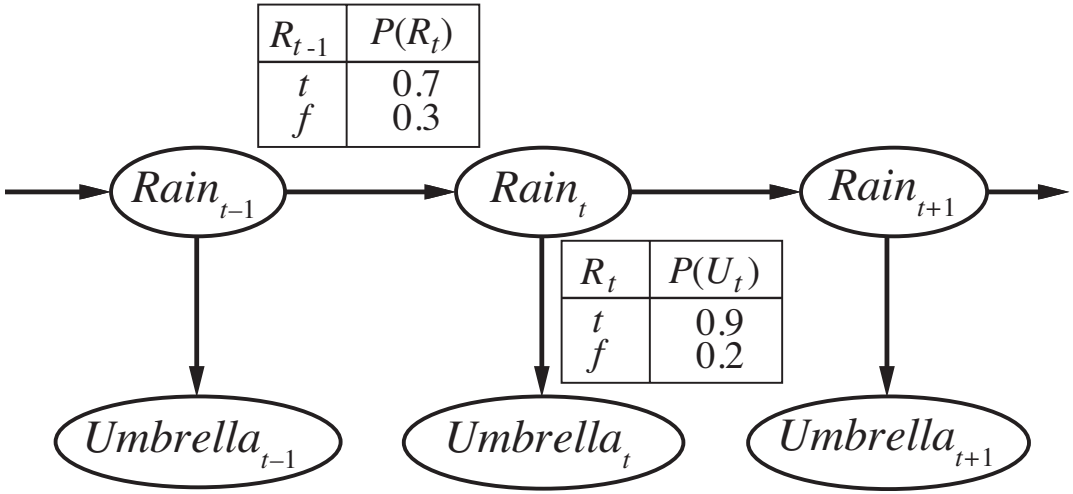
- $Rain_t$ = it is raining at time t
- $Umbrella_t$ = our friend is carrying an umbrella at time t
- **Dynamic Bayesian network**—more to come!



- Start with **prior probability** distribution $\mathrm{P}(X_0)$ at time $t = 0$
- **Joint distribution** over all variables in the network
  $\mathbf{P}(X_{0:t}, E_{1:t}) = \mathrm{P}(X_0) \prod_{\mathrm{i}\,=\,1 \text{ to } \mathrm{t}} \mathrm{P}(X_i \mid X_{i-1})\, \mathrm{P}(E_i \mid X_i)$

First-order Markov assumption unrealistic?

- First-order Markov not exactly true in real world!
  - *Rain* only depends on if it rained yesterday…?



- Improving **accuracy** of the model
  - Increase order of Markov process
  - Increase set of variables—additional information and relationships
    $Temperature_t$, $BarametricPressure_t$

Inference in temporal models

- Common reasoning patterns through temporal model

- **Filtering**: $P(X_t \mid e_{1:t})$—computing the **belief state** given evidence sequence to facilitate rational decision-making
- **Prediction**: $P(X_{t+k} \mid e_{1:t})$, $k > 0$—compute the posterior probability of **future** state given the evidence sequence
- **Smoothing**: $P(X_k \mid e_{1:t})$, $0 \leq k \leq 1$—compute the probability of **past** state given the evidence sequence
- Most likely **explanation**: $\text{argmax}_{x1:t} P(x_{1:t} \mid e_{1:t})$—sequence of states most likely to have generated the observations

- **Learning**—learning the structure and probabilities from data using **expectation maximization** (EM)

Inference in temporal models

- Common reasoning patterns through temporal model

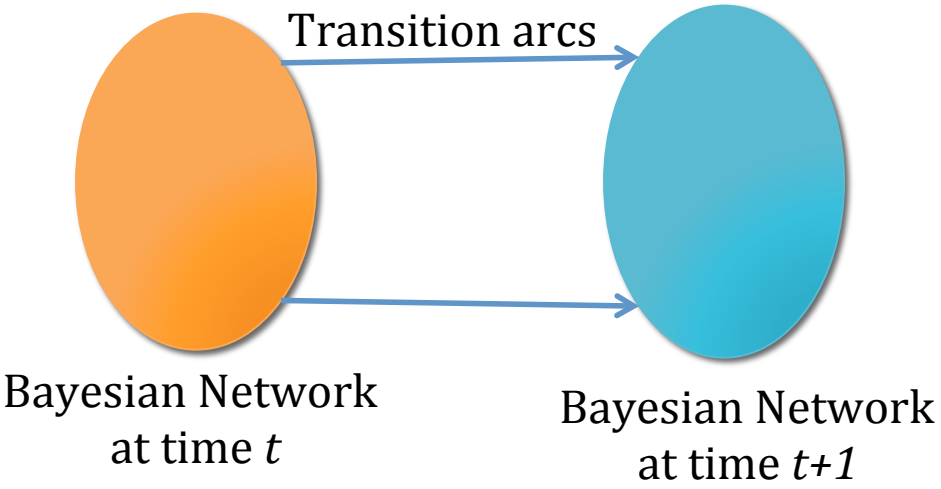
- **Filtering**: $P(X_t \mid e_{1:t})$—computing the **belief state** given evidence sequence to facilitate rational decision-making
- **Prediction**: $P(X_{t+k} \mid e_{1:t})$, $k > 0$—compute the posterior probability of **future** state given the evidence sequence
- **Smoothing**: $P(X_k \mid e_{1:t})$, $0 \leq k \leq 1$—compute the probability of **past** state given the evidence sequence
- Most likely **explanation**: $\text{argmax}_{x1:t} P(x_{1:t} \mid e_{1:t})$—sequence of states most likely to have generated the observations

- **Learning**—learning the structure and probabilities from data using **expectation maximization** (EM)

Dynamic Bayesian networks (DBNs)

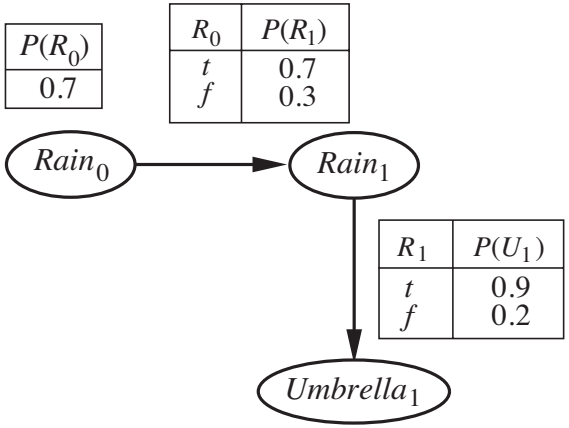
- Bayesian network representing temporal probability model
  - Stationary, Markov process of state transitions

- Includes prior distribution $P(X_0)$, transition model $P(X_t \mid X_{t-1})$, and observation model $P(E_i \mid X_i)$
  - Depends on topology between time slices
- Connection between Bayesian network at time *t* and *t+1*



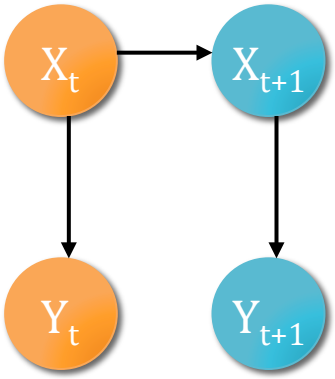
Basic approach to DBNs

- **Copy** the state and evidence from one time slice to the next
  - Only specify for first time slice and replicate for all the others

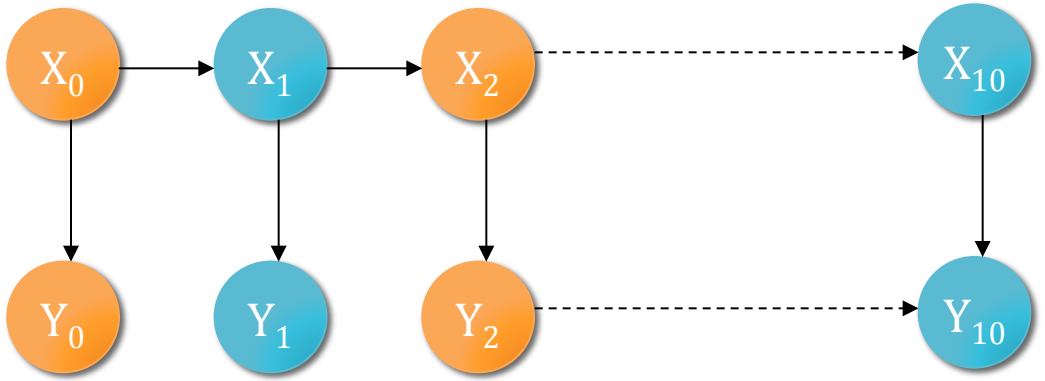


- Process called **unrolling** the DBN

One slice DBN

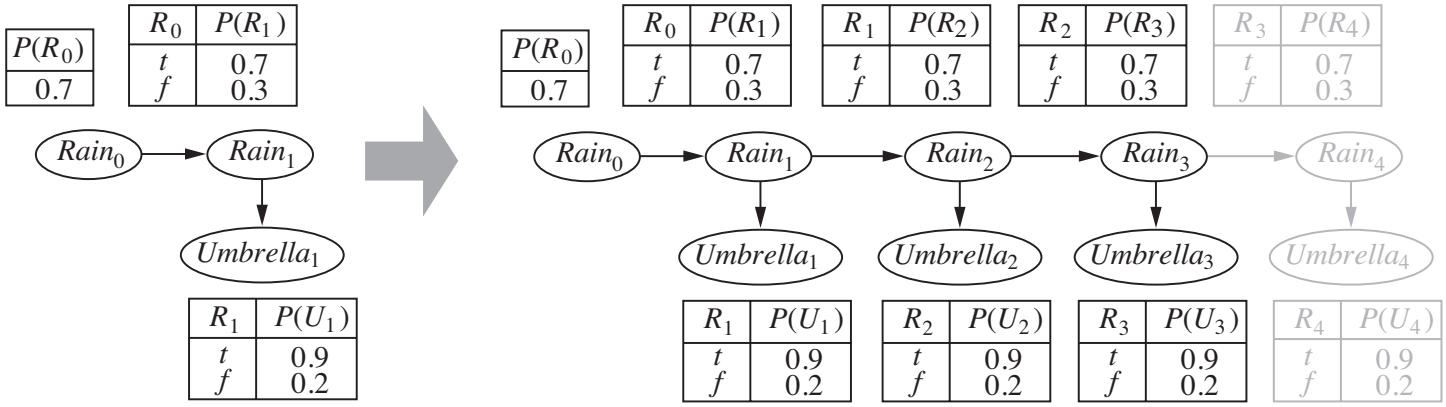


Unrolled for time $t = 0$ to $t = 10$



Exact inference in DBNs

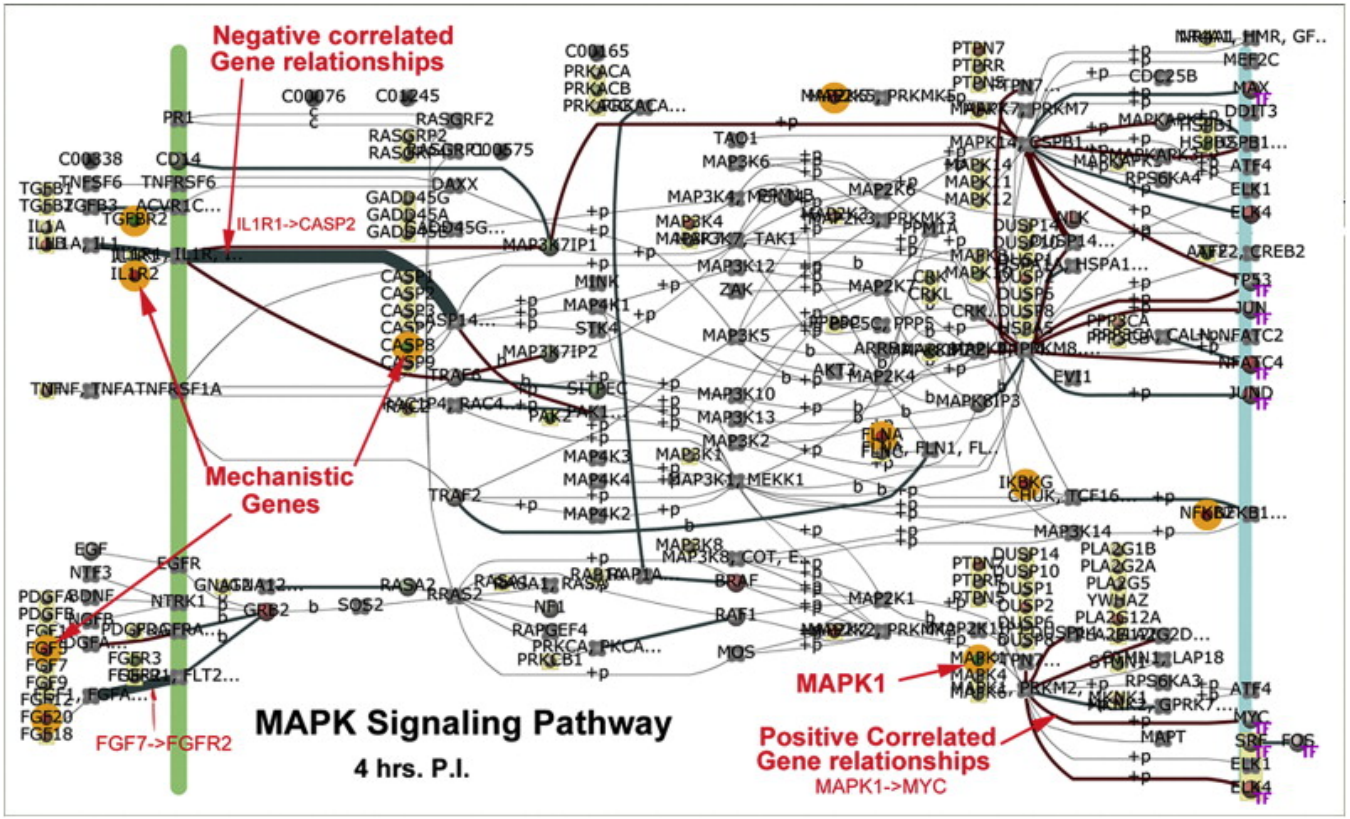
- Naïve approach—**unroll** the whole network and apply any exact Bayesian reasoning algorithm



- The inference cost for each update grows with t
- Use **variable elimination** to "sum out" previous time slices
  - Keep at most two slices in memory at a time
- Still exponential in number of state variables
- Need approximations!

Unrolling intractable in real-world models

- Pathways, biological processes, cellular components, and molecular components that change with growing bacterial infection



Approximation in DBNs using particle filtering

- **Filtering**: $P(X_t \mid e_{1:t})$—computing the **belief state** given evidence sequence to facilitate rational decision-making

- Filtering algorithm **maintains** current state and **updates** with new evidence, rather than looking at entire sequence

$$P(X_{t+1} \mid e_{1:t+1}) = f(e_{t+1}, P(X_t \mid e_{1:t}))$$

  - **Recursive estimation**

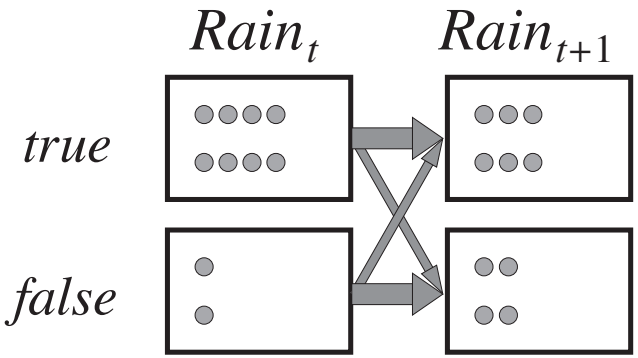
- Particle filtering for importance sampling
  - Focus the samples (particles) on high-probability regions
  - Throw away samples with low weights according to the observations and **replicate** those with high weights
  - Population of samples representative of reality

Particle filtering algorithms

- Sample N initial states from $P(X_0)$
- **Update** cycle for each time step:
  1. Propagate sample forward using Markov transition model $P(X_{t+1} \mid X_t)$
  2. Weight sample by likelihood of new evidence using observation model $P(e_{t+1} \mid x_{t+1})$
  3. **Resample** N new samples from the current population—probability of selection proportional to weight
     - New samples are unweighted

Using particle filtering

- N = 10 samples at each time slice

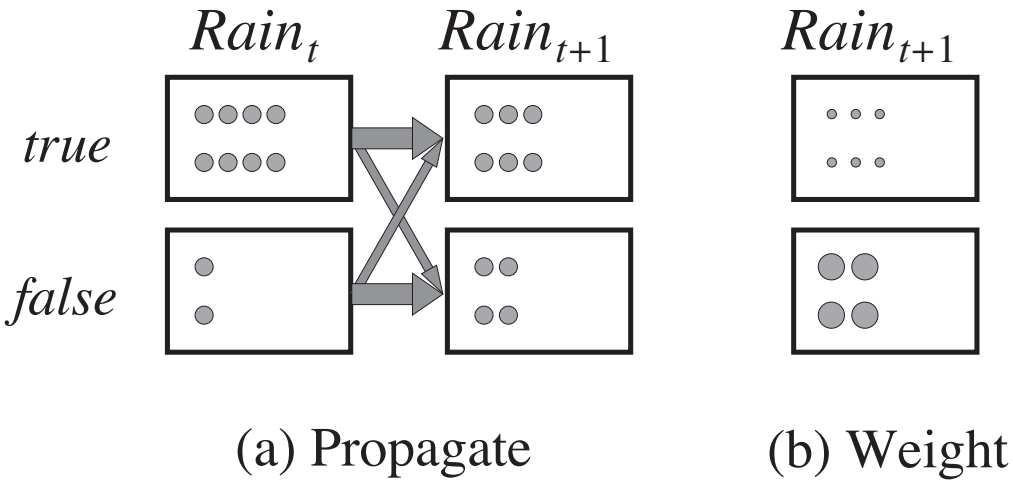


(a) Propagate

- Time t 8 samples indicate *Rain* is true, and 2 false
- Use **transition model** to propagate to *t+1*—sample $Rain_{t+1}$ from CPT conditioned on $Rain_t$
- Time *t+1* 6 samples indicate *Rain* is true, and 4 false

Using particle filtering

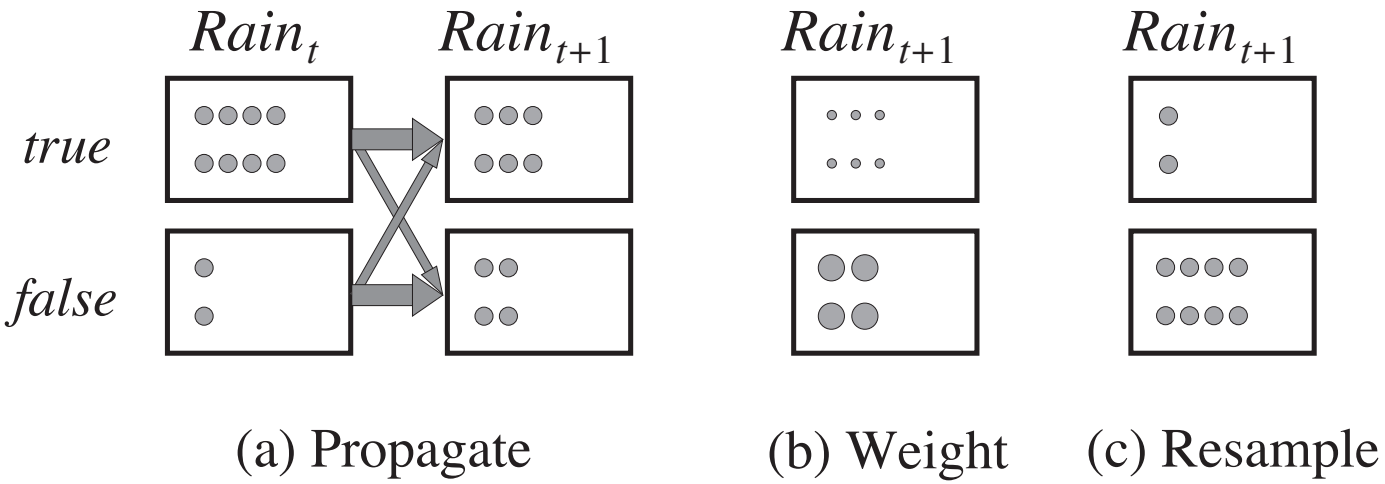
- N = 10 samples at each time slice



- At time *t+1*, observation is ¬*Umbrella*
- Use this evidence to weight the sample just generated

Using particle filtering

- N = 10 samples at each time slice



- Generate refined set of 10 samples
  - Weighted random selection from current set
  - 2 samples indicate rain, 8 no rain
- Now **propagate** this tuned sample to time *t+2*

Analysis of particle filtering

- Consistent estimation—converges to exact probabilities as $N \rightarrow \infty$
- **Resampling** allows us to **refine** likelihood weighting—throw out small weights and focus on large ones
- Drawback of particle filtering
  - Inefficient in **high-dimensional** spaces
    (Variance becomes so large)

- Solution
  - ***Rao-Balckwellization***—sample a subset of variables allowing the remainder to be integrated out exactly
    - Estimates have lower variance

Rao-Blackwellized particle filtering

- How can we reduce the number of particles (samples) needed to achieve the same accuracy?
- Sample **subset** of the variables allowing remainder to be integrated out exactly
  - Results in estimates having lower variance
- **Partition** the state variables at time t s.t.

$$X_t = (R_t, V_t)$$

where

$$P(R_{0:t}, V_{0:t} \mid E_{1:t}) = P(V_{0:t} \mid R_{0:t}, E_{1:t})\, P(R_{0:t} \mid E_{1:t})$$

- Assume we can tractably compute $P(V_{0:t} \mid R_{0:t}, E_{1:t})$
- Just focus on estimating probability from lower dimension space

$$P(R_{0:t} \mid E_{1:t}) = \frac{P(E_t \mid E_{1:t-1}, R_{0:t}) P(R_t \mid R_{t-1})\, P(R_{0:t-1} \mid E_{1:t-1})}{P(E_t \mid E_{t-1})}$$

Rao-Blackwellised particle filtering

- How can we reduce the number of particles (samples) needed to achieve the same accuracy?
- Sample **subset** of the variables allowing remainder to be integrated out exactly
  - Results in estimates having lower variance
- **Partition** the state variables at time t s.t.

$$X_t = (R_t, V_t)$$

where

$$P(R_{0:t}, V_{0:t} \mid E_{1:t}) = P(V_{0:t} \mid R_{0:t}, E_{1:t}) P(R_{0:t} \mid E_{1:t})$$

Only sample this!

- Assume we can tractably compute $P(V_{0:t} \mid R_{0:t}, E_{1:t})$
- Just focus on estimating probability from lower dimension space

$$P(R_{0:t} \mid E_{1:t}) = \frac{P(E_t \mid E_{1:t-1}, R_{0:t}) P(R_t \mid R_{t-1}) P(R_{0:t-1} \mid E_{1:t-1})}{P(E_t \mid E_{t-1})}$$

Rao-Blackwellised particle filtering

- How can we reduce the number of particles (samples) needed to achieve the same accuracy?
- Sample **subset** of the variables allowing remainder to be integrated out exactly
  - Results in estimates having lower variance
- **Partition** the state variables at time t s.t.

$$X_t = (R_t, V_t)$$

where

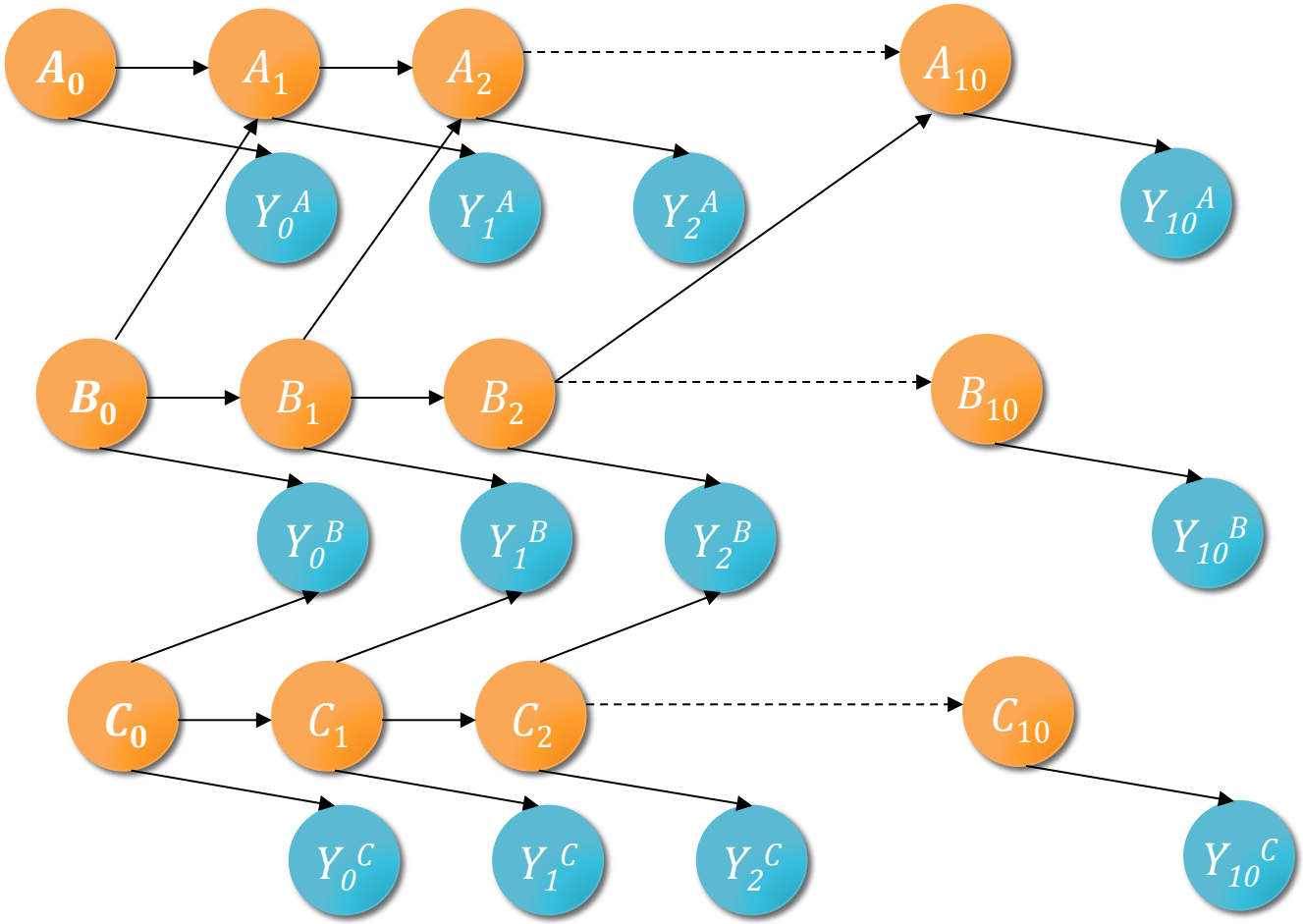
$$P(R_{0:t}, V_{0:t} \mid E_{1:t}) = P(V_{0:t} \mid R_{0:t}, E_{1:t}) P(R_{0:t} \mid E_{1:t})$$

The rest of the values conditionally independent given the sample and evidence

- Assume we can tractably compute $P(V_{0:t} \mid R_{0:t}, E_{1:t})$
- Just focus on estimating probability from lower dimension space

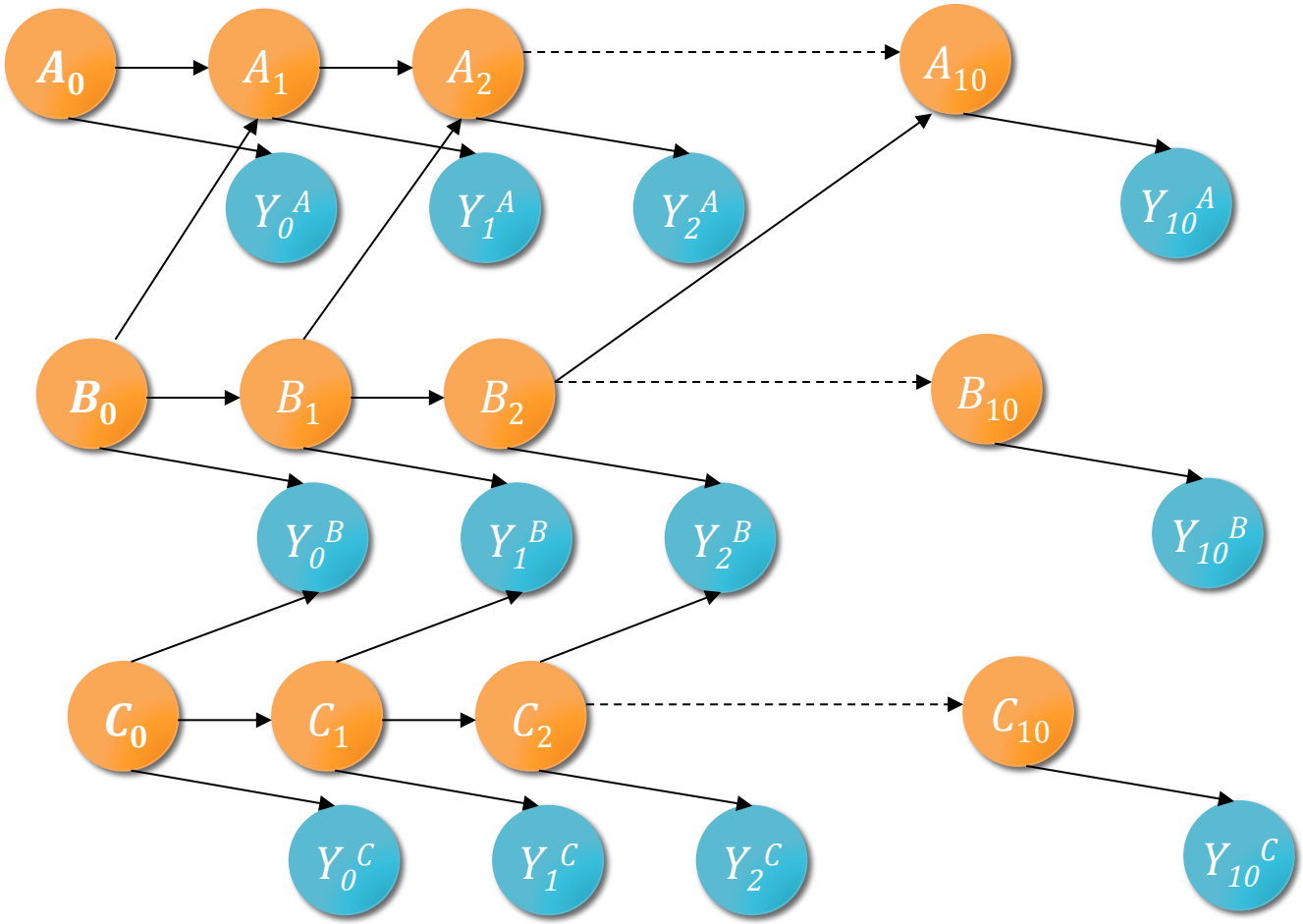
$$P(R_{0:t} \mid E_{1:t}) = \frac{P(E_t \mid E_{1:t-1}, R_{0:t}) P(R_t \mid R_{t-1}) P(R_{0:t-1} \mid E_{1:t-1})}{P(E_t \mid E_{t-1})}$$

Approximate inference with fewer samples



Goal: compute joint filtering distribution $\mathrm{P}(A_t, B_t, C_t \mid E_{1:t})$

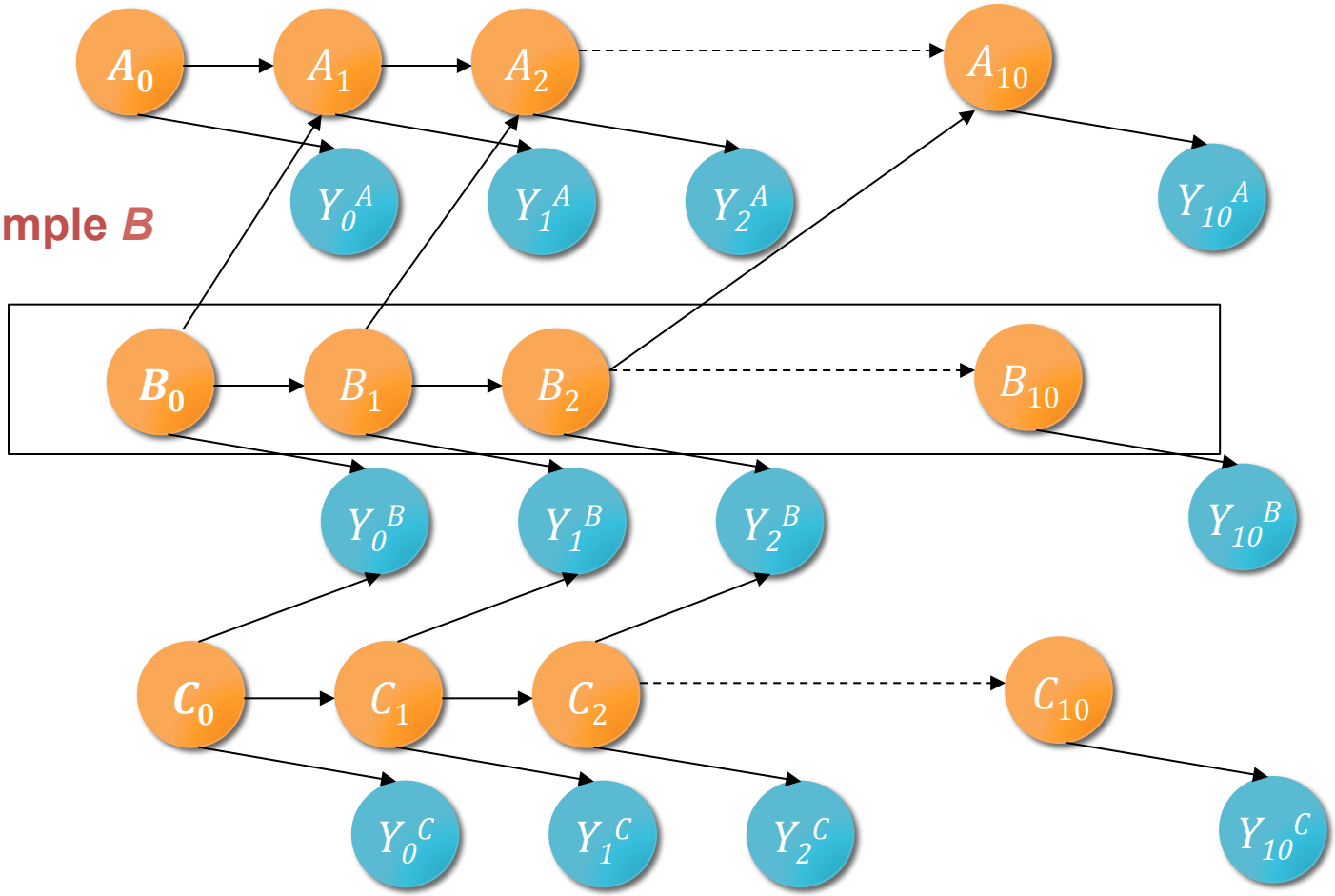
Approximate inference with fewer samples



$$P(A_t, B_t, C_t \mid Y_{1:t}) = P(A_{1:t}, C_{1:t} \mid Y_{1:t}, B_{1:t})\, P(B_{1:t} \mid Y_{1:t})$$
$$= P(A_{1:} \mid Y_{1:t}^{A}, B_{1:t-1})\, P(C_{1:} \mid Y_{1:t}^{C}, B_{1:t-1})\, P(B_{1:t} Y_{1:t})$$

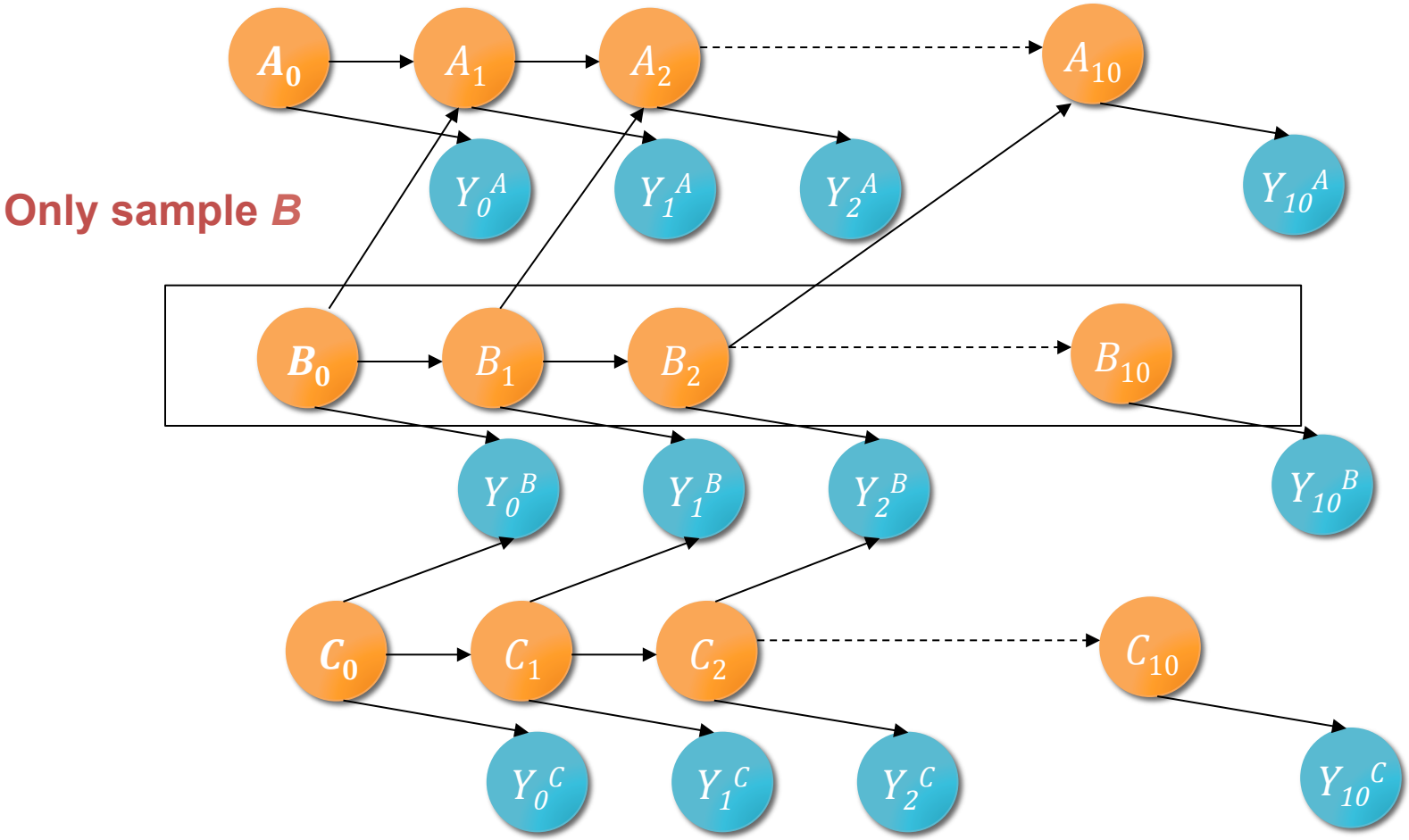
Approximate inference with fewer samples

Only sample *B*



$$P(A_t, B_t, C_t \mid Y_{1:t}) = P(A_{1:t}, C_{1:t} \mid Y_{1:t}, B_{1:t})\, P(B_{1:t} \mid Y_{1:t})$$
$$= P(A_{1:} \mid Y_{1:t}^{A}, B_{1:t-1})\, P(C_{1:} \mid Y_{1:t}^{C}, B_{1:t-1})\, P(B_{1:t} Y_{1:t})$$

Approximate inference with fewer samples



Where do we get these partitions?

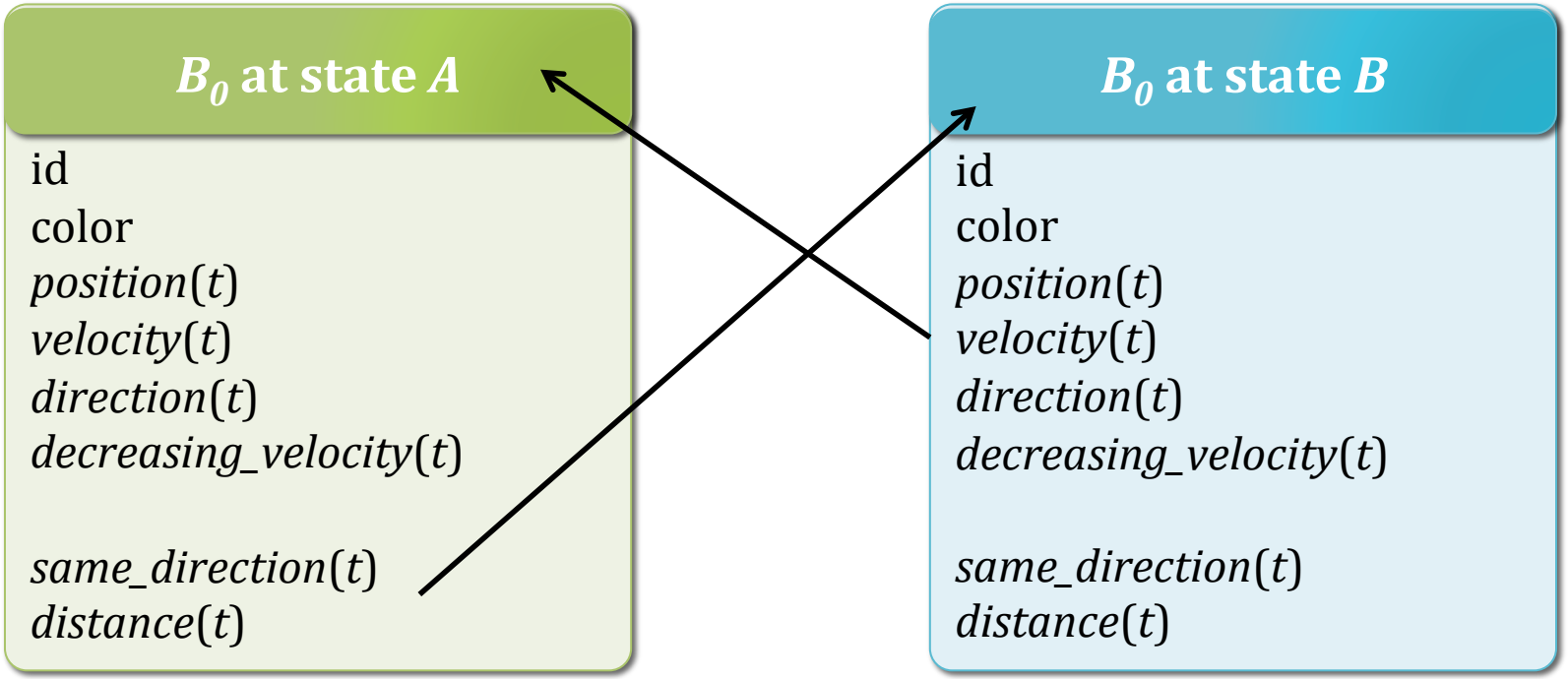
Typically domain or application specific.

Limitations of only using random variables

- DBNs extend traditional Bayesian networks
  - Facilitate probabilistic reasoning over time
- Knowledge representation is still not very expressive
  - Random variables are essentially **propositions**, with same drawbacks
  - How do we express relationships and properties of objects?
    - Exhaustively represent all possible objects and relations among them
    - Intractable in real-world relational domains
- Incorporate **first-order logic** into DBNs
  - **Relational Dynamic Bayesian Networks**

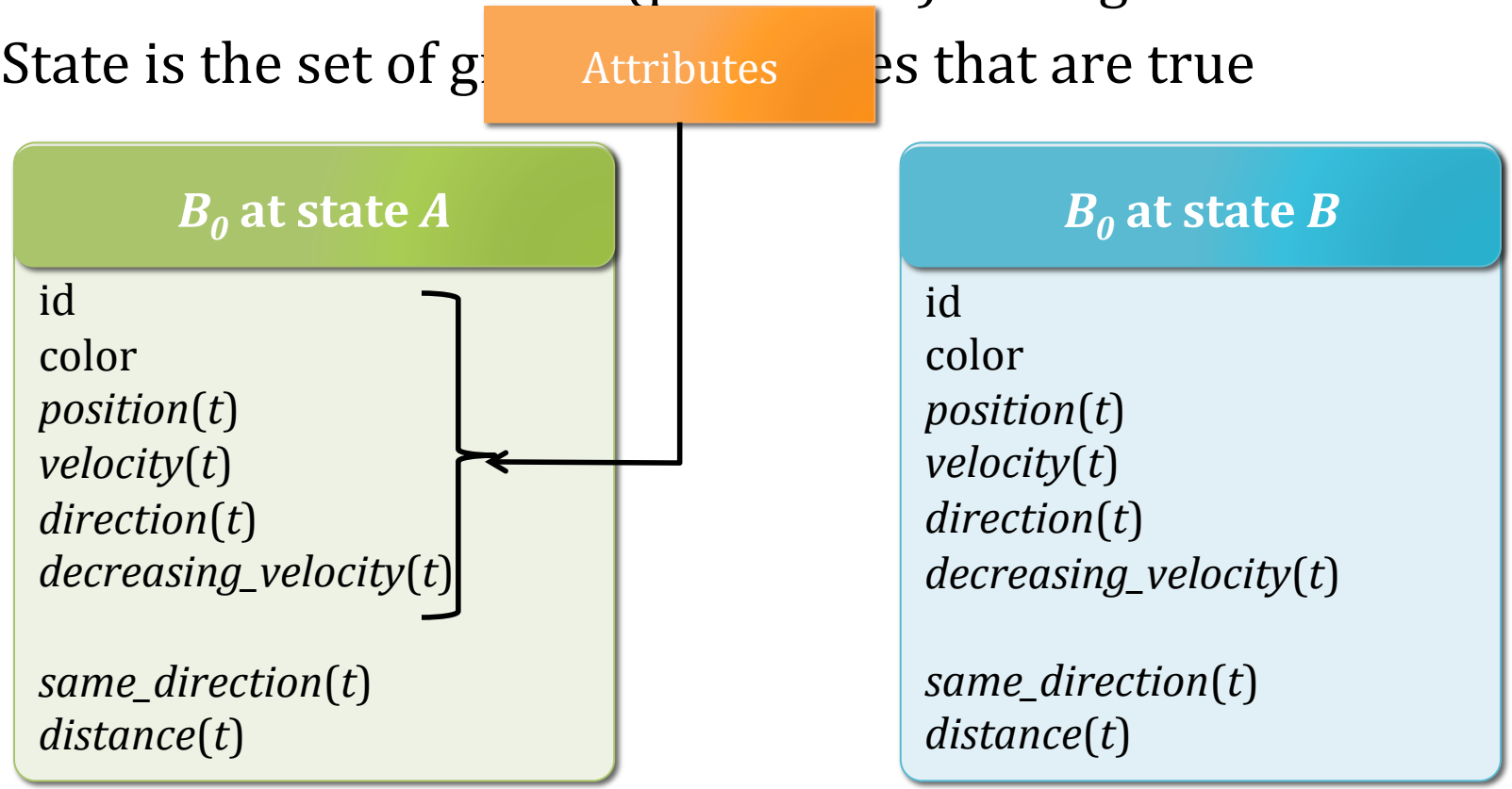
Dynamic Relational domains

- Set of **objects** (constants, variables, functions) and **attributes** or **relations** (predicates) among them
- State is the set of ground predicates that are true



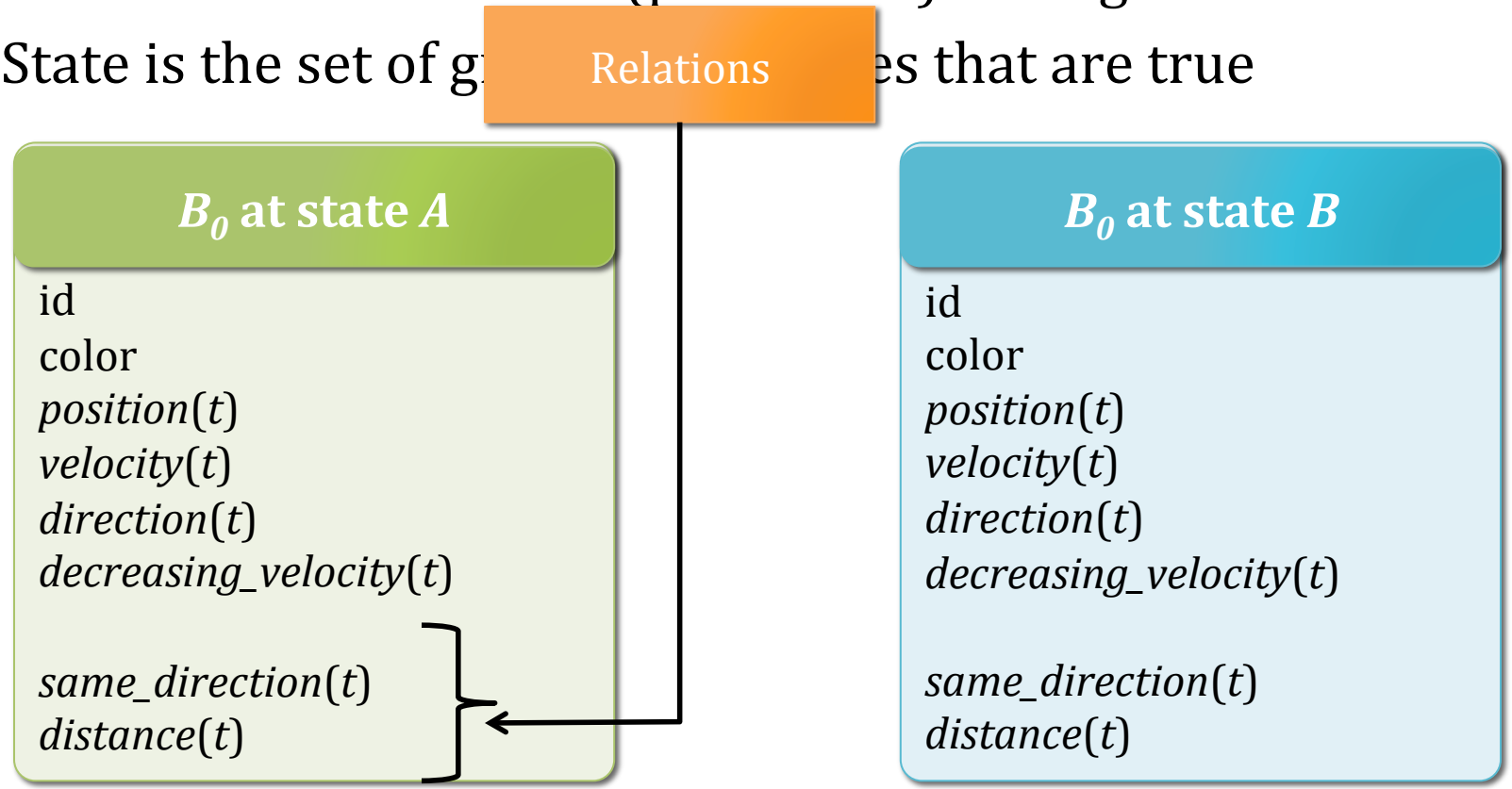
Relational domains

- Set of **objects** (constants, variables, functions) and **attributes** or **relations** (predicates) among them
- State is the set of g… …es that are true



Relational domains

- Set of **objects** (constants, variables, functions) and **attributes** or **relations** (predicates) among them
- State is the set of g[illegible]es that are true



Relational Bayesian Network (RBN)

- Syntax
  - Set of nodes—one for each FOL predicate
  - DAG—directed acyclic graph
  - Conditional distribution for each node given its parents

- Now do not have to instantiate all ground atoms and use propositional Bayesian network
  - Represent **general relationships** between objects

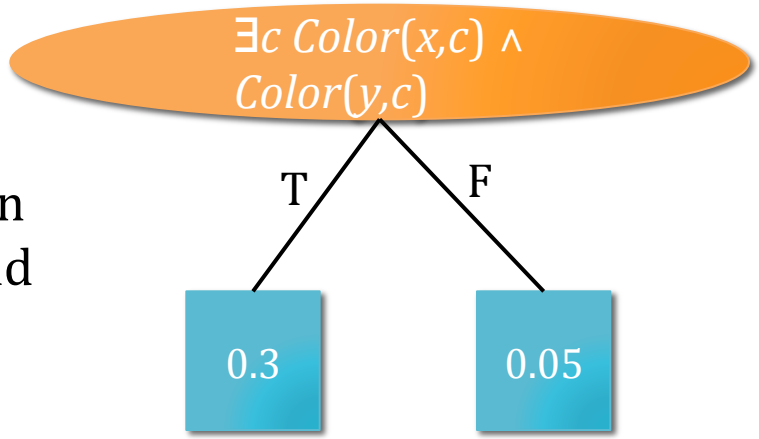
- To ensure no cycles in the RBN predicates must be **ordered**

Conditional model for each node

- For each node conditional distribution determined by relational information

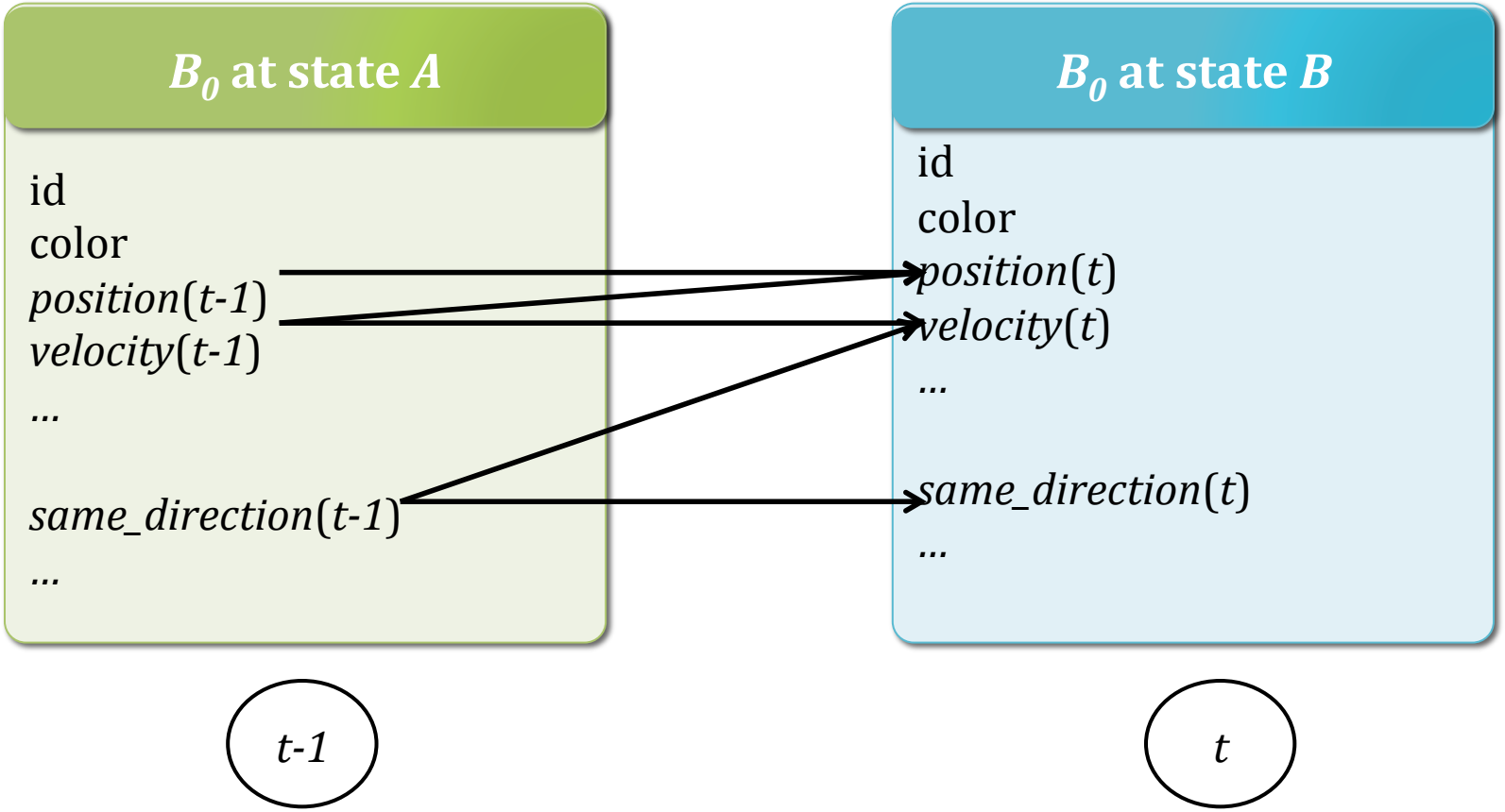
- **First-order probability tree**
  - Conditional model of ground node given its parents
  - Store FOPT at each node rather than conditional probability tables

- Construction
  - Interior node: first-order formula F_n on parent predicates—will make the child either true or false
  - Leaves: probability distribution



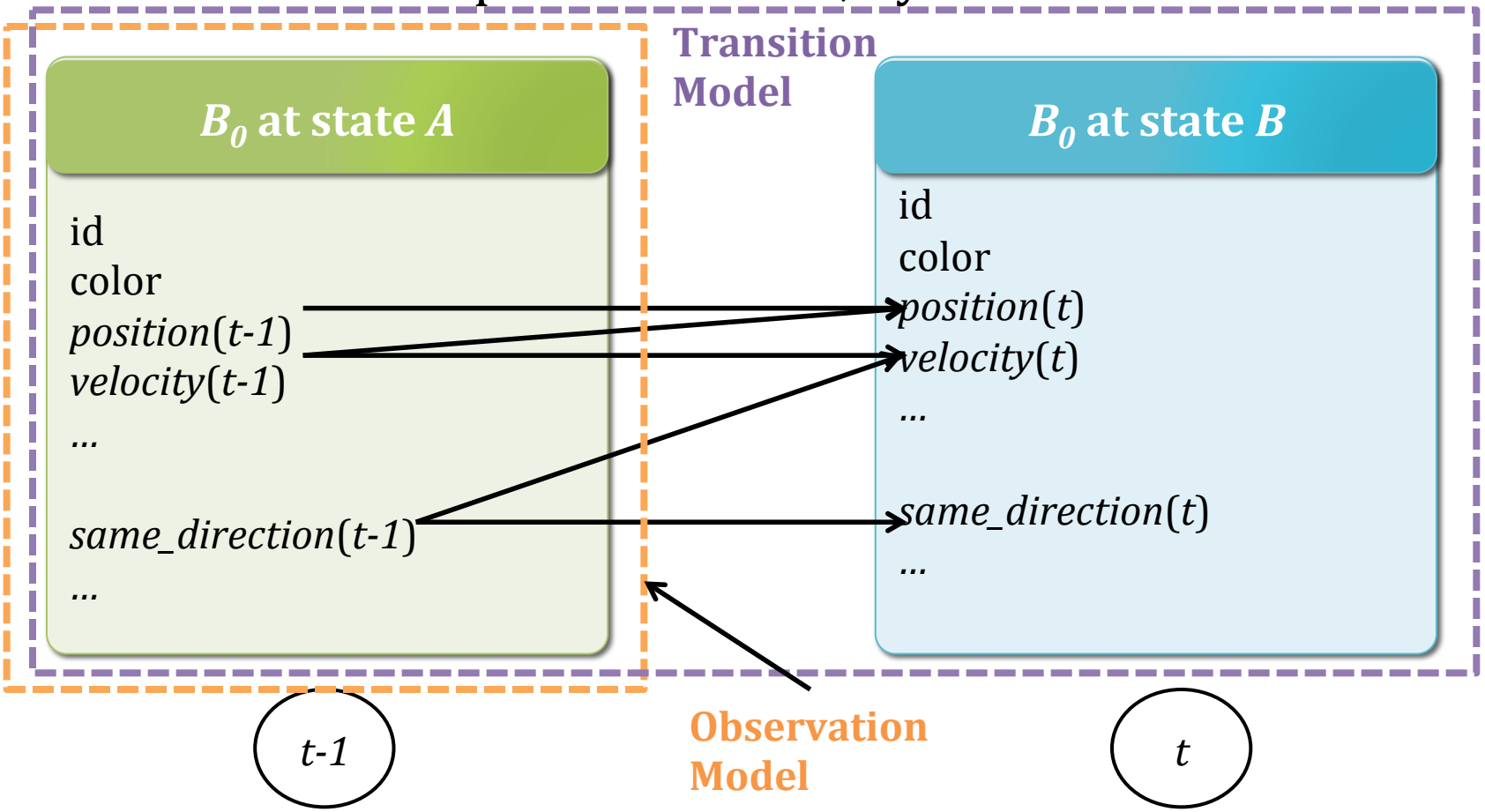
Relational Dynamic Bayesian Network (RDBN)

- Infeasible to use exact DBN on all ground predicates
- Extend RBN for explicit relational, dynamic network



Relational Dynamic Bayesian Network (RDBN)

- Infeasible to use exact DBN on all ground predicates
- Extend RBN for explicit relational, dynamic network



Transition model is first-order Markov

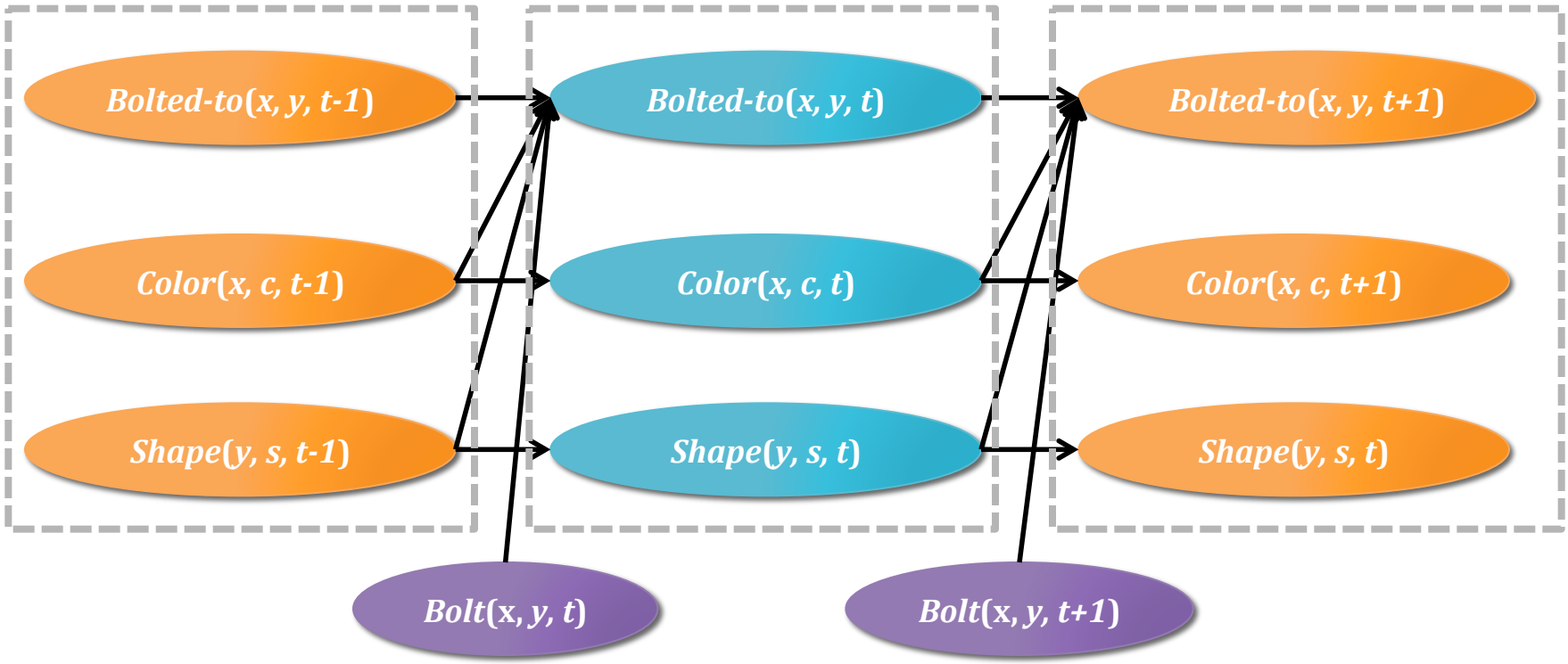
- Predicates at time *t* depend only on those at *t-1*

- Create node at *t* for every ground predicate
  - Use conditional model (FOPT) based on grounding at the node
  - Number of **ground predicates** (per slice!) is $O(N^k)$ where N is the size of the domain raised to the arity k of the predicate
    - Domain size can be tens of thousands or more!

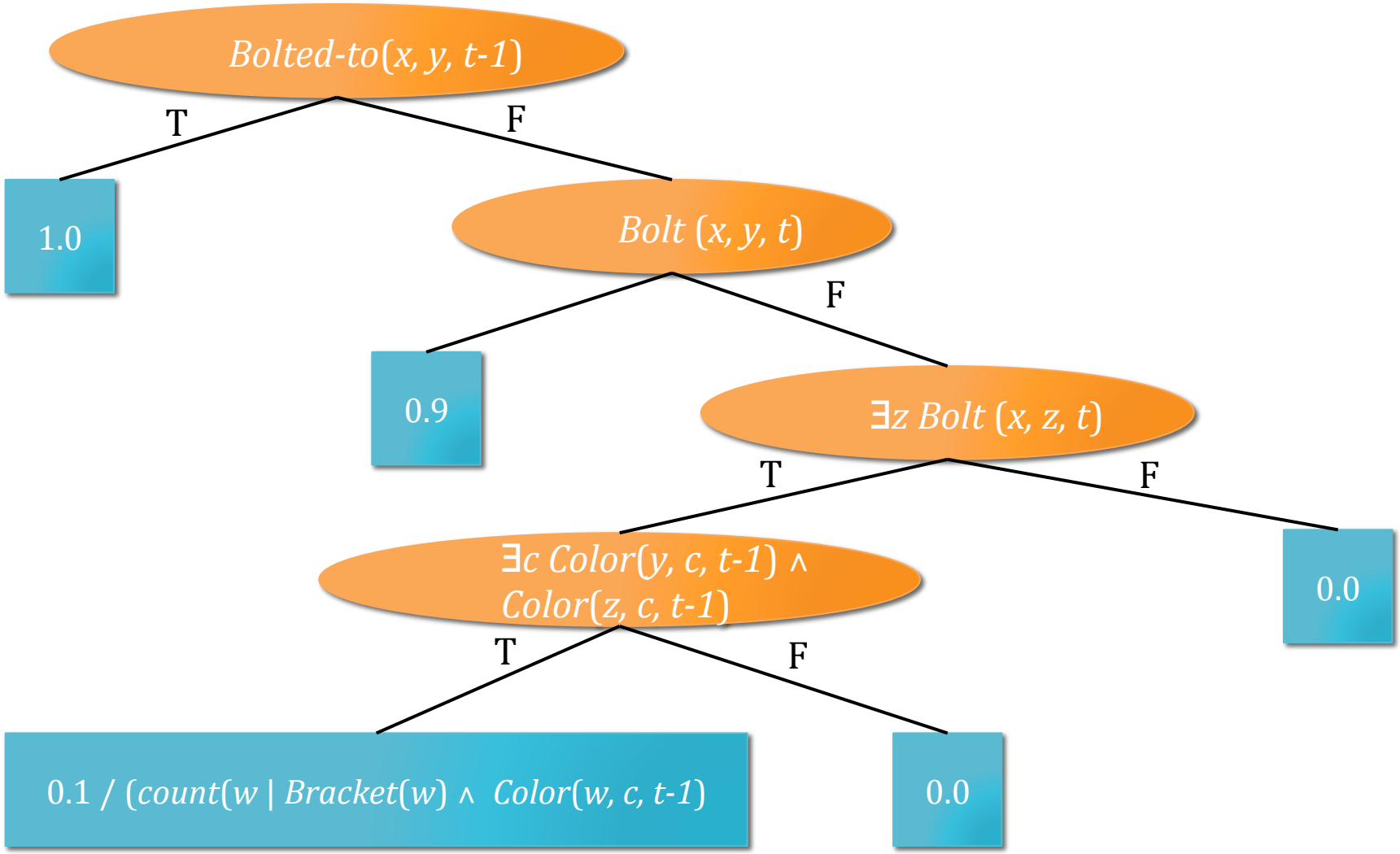
- Assume **one action** performed per time slice

Example of RDBN

- Factory assembly domain
  - Plates, brackets, etc. welded and bolted together
  - Plates and brackets have attributes such as size, shape, and color



First order probability tree for RDBN



Inference in RDBN using FOL properties

- DBN inference on ground version
  - Exact inference completely intractable!
  - Particle filtering will sample poorly because of high variance in large domains
- "Lifted" versions of the existing algorithms makes use of FOL structure

- Identify two categories of predicates
  - **Complex**—if the domain size is "large"
    - *Bolted-To*(*x*,*y*,*t*) where items *x* and *y* are components (i.e., plates and brackets) that can be bolted together in manufacturing
  - **Simple** otherwise
    - *Color*(*x*,*c*,*t*) where the number of possible colors *c* in this application is small
  - Largeness of the domain depends on the application

Rao-Blackwellization in RDBNs

- Partition using simple and complex predicates (well, and make some assumptions…)

- **Assumption 1**: Uncertain complex predicates do not appear in the RDBN as the parents of other predicates
  - All parents of unobserved complex predicates are simple or known

- **Assumption 2**: For any object *o* there is at most one other object *o'* s.t. the ground predicate $R(o, o', t)$ is true and one *o''* s.t. $R(o'', o, t)$ is true
  - Objects in a relation are mutually exclusive

Conditional independence gives partitions

- Complex predicates **independent of each other**
  - Conditioned on simple predicates and known evidence (i.e., parents)

- Simple predicates **independent of unknown complex** ones
  - Given known evidence

- Rao-Blackwell partitions of (unknown) predicates *P* at *t:*

$$P_t = (Complex_t, Simple_t)$$

so

$$P(Simple_{0:t}, Complex_{0:t} \mid E_{1:t}) = P(Complex_{0:t} \mid Simple_{0:t}, E_{1:t})\, P(Simple_{0:t} \mid E_{1:t})$$

Conditional independence gives partitions

- Complex predicates **independent of each other**
  - Conditioned on simple predicates and known evidence (i.e., parents)
- Simple predicates **independent of unknown complex** ones
  - Given known evidence
- Rao-Blackwell partitions of (unknown) predicates *P* at *t:*

$$P_t = (Complex_t, Simple_t)$$

so

$$P(Simple_{0:t}, Complex_{0:t} \mid E_{1:t}) = P(Complex_{0:t} \mid Simple_{0:t}, E_{1:t}) P(Simple_{0:t} \mid E_{1:t})$$

Sample the simple predicates

Compute the complex predicates

Efficiency of Rao-Blackwellization

- Rao-Blackwellized particle filtering better than standard algorithm
- Domains with large numbers of objects and relations are still complex even with Rao-Blackwellization

- Leverage context or domain specific independence to improve efficiency
  - Group related objects o and o' that give rise to $R(o,o',t)$ into **abstractions**
  - Disjoint sets $A_{R1}, A_{R2}, \ldots, A_{rm}$ s.t. two pairs of objects (o_i,o_j'), (o_k,o_l') in A_R iff $Pr(o_i,o_j',t) = Pr(o_k,o_l')$
  - Specify abstractions with **FOL formulas**
  - Maintain conditional probabilities for abstractions rather than pairs
- Abstractions **improve performance** by factor of 30 to 70

DBNs and RDBNs are not the only way

- Several approaches to handling time and uncertainty depending on the task
  - Markov decision process
  - Hidden Markov models

- DBNs are **generalizations** of many of these other systems
  - Can be even more effective when domain knowledge allows additional conditional independence assumptions