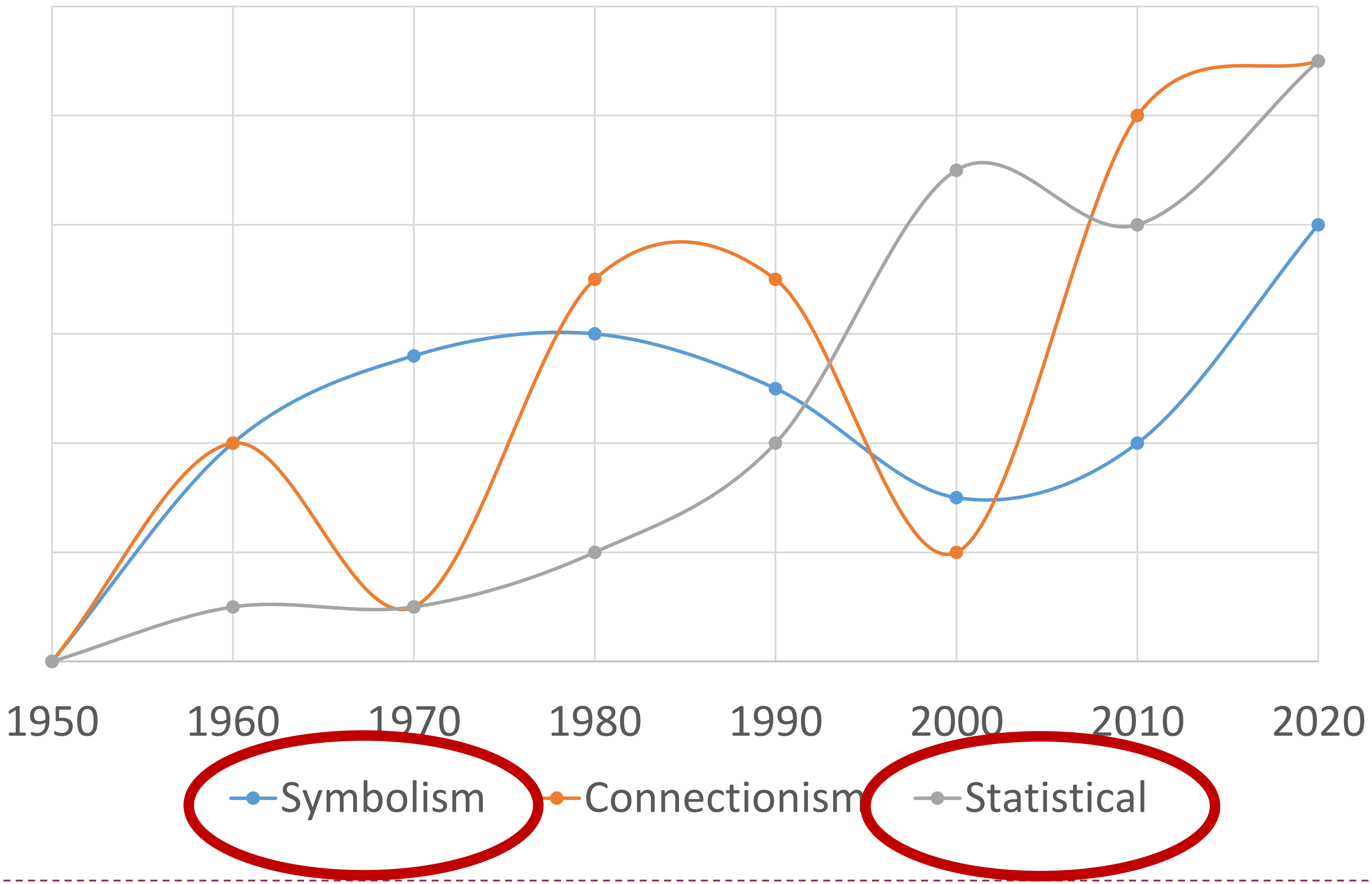
Three types of (strong) AI approaches



Probabilistic Logics

AIMA 14.6

Additional materials

[Adapted from slides by Lise Getoor at UC Santa Cruz and Pedro Domingos at U Washington]

Additional reference materials

- L. Getoor and B. Taskar (eds.), Introduction to Statistical Relational Learning, 2007. Cambridge, MA: MIT Press.
  - Ch 5: Probabilistic Relational Models
  - Ch 12: Markov Logic

Logics vs. Probabilistic Models

- Symbolic logics
  - FOL is very expressive
    - relations between objects, quantifiers
  - But it cannot model uncertainty

- Probabilistic Models
  - BN/MN model uncertainty in a concise manner
  - But limited in expressiveness
    - BN/MN is essentially propositional

Probabilistic Logics

- Goal
  - Combine (subsets of) logic and probability into a single language
- A.k.a. Statistical Relational Learning

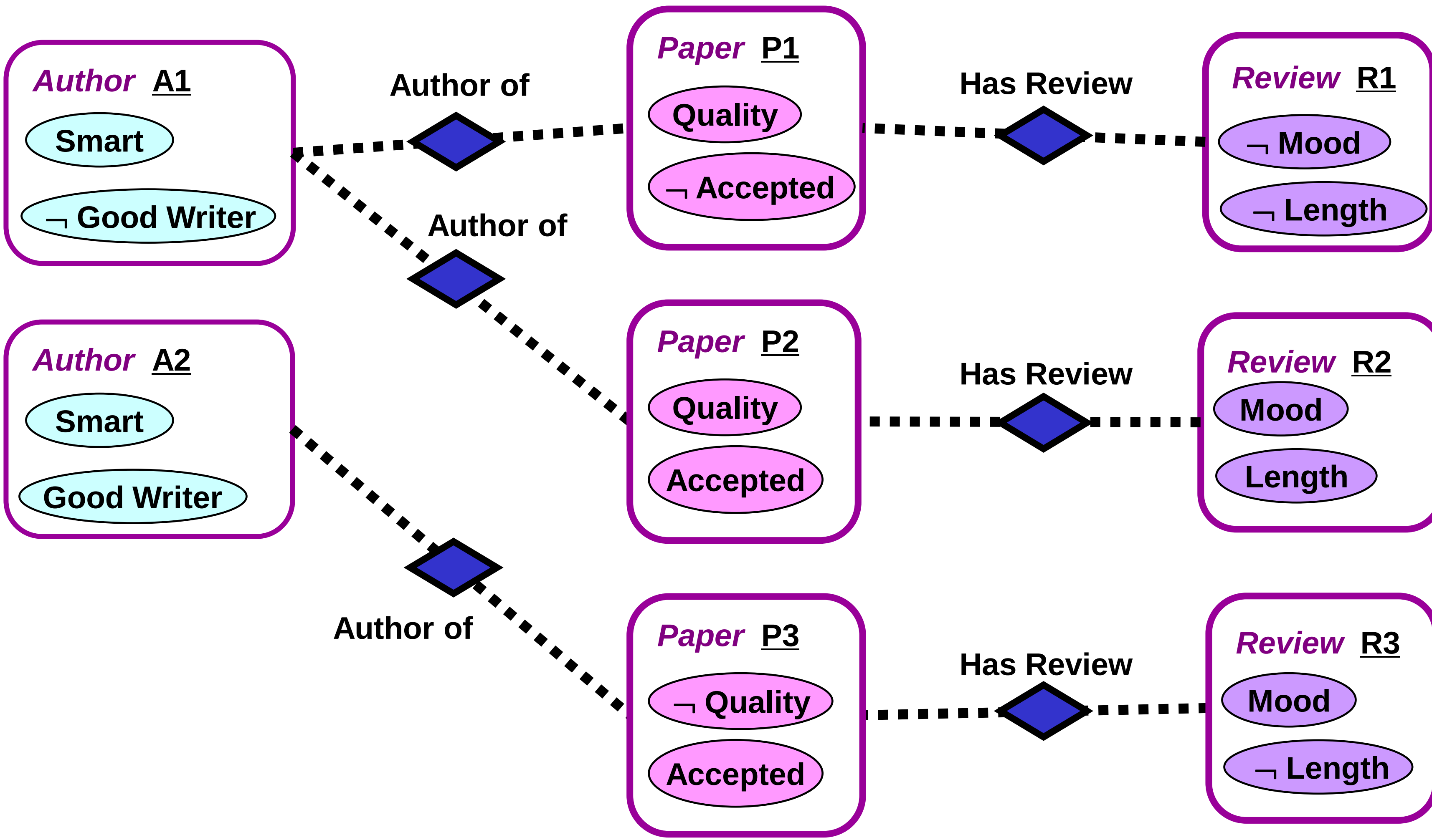
- Lots of approaches. We will cover two of them:
  - Probabilistic Relational Models
  - Markov Logic

Probabilistic Relational Models

Probabilistic Relational Models

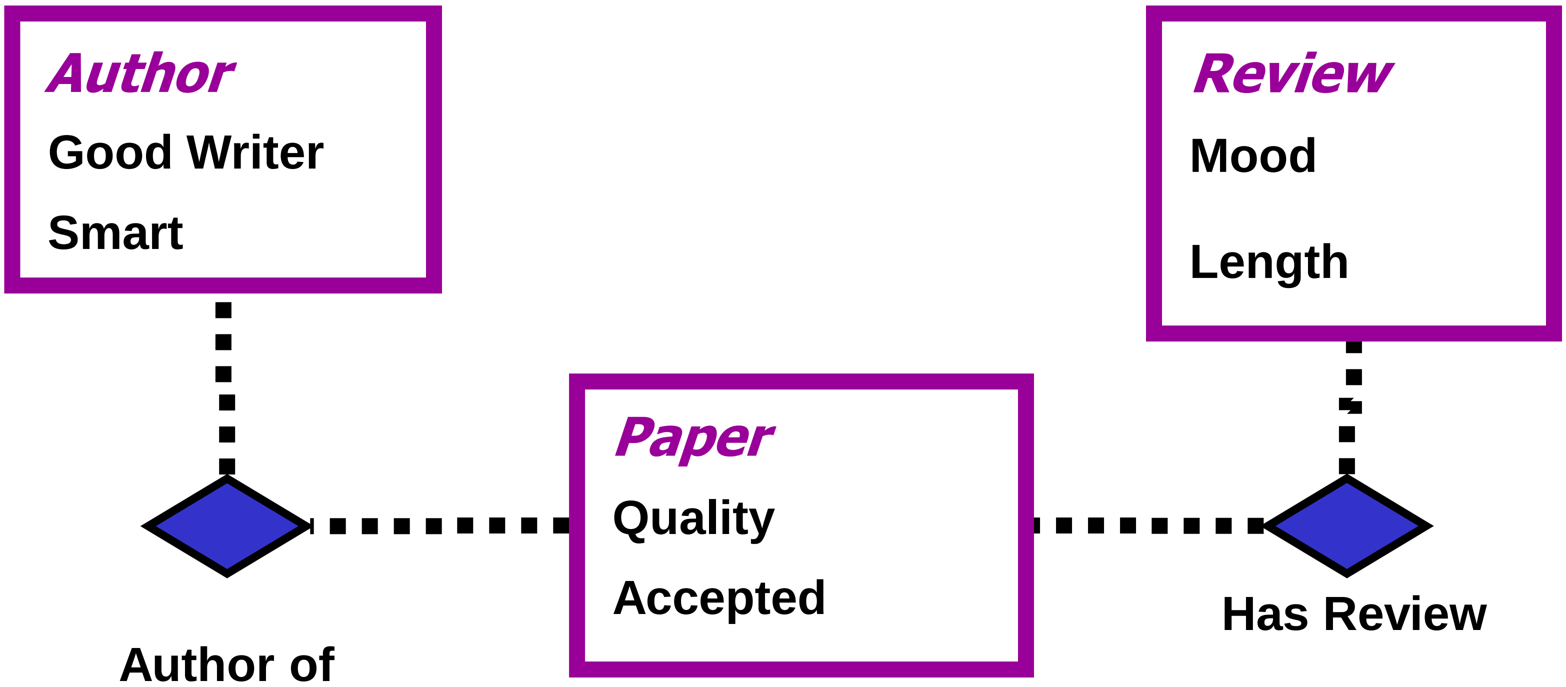
- Logical language
  - Frame (typed relational knowledge)
    - A subclass of FOL
- Probabilistic language
  - Bayes nets

Typed relational knowledge



Typed relational knowledge

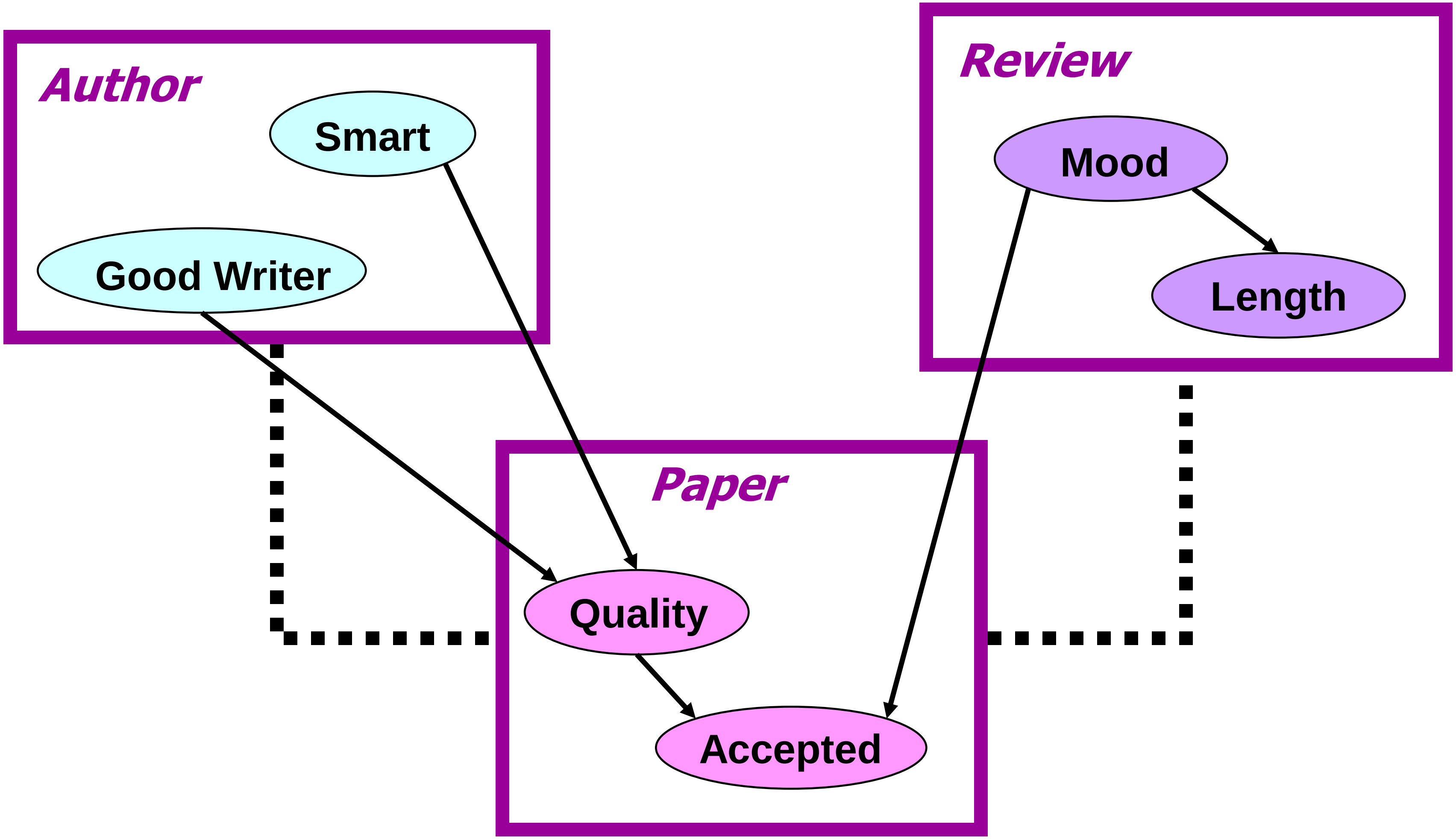
Why is this a subclass of FOL?



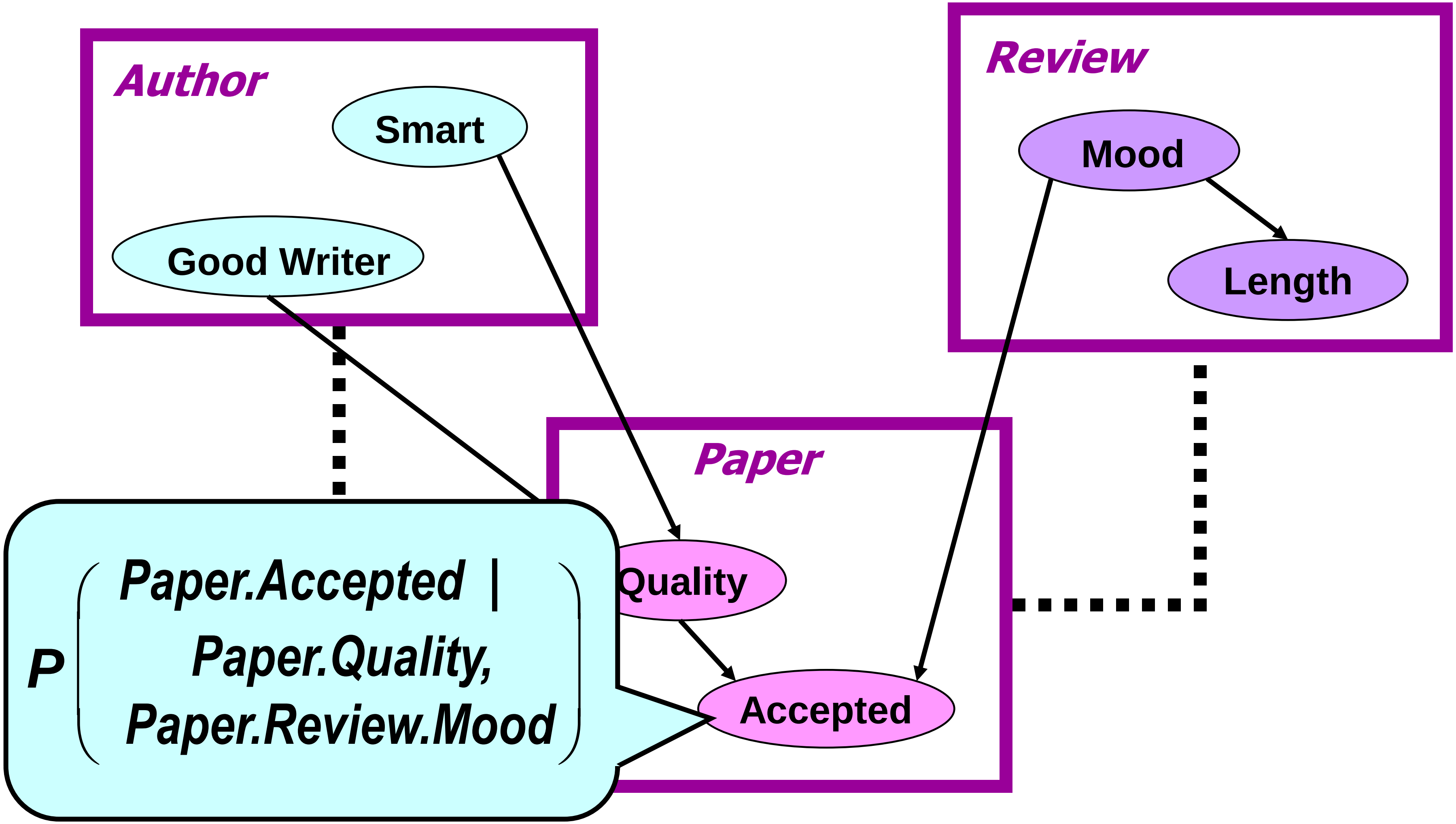
Ontology / Schema

- The types of objects and their valid relations and attributes

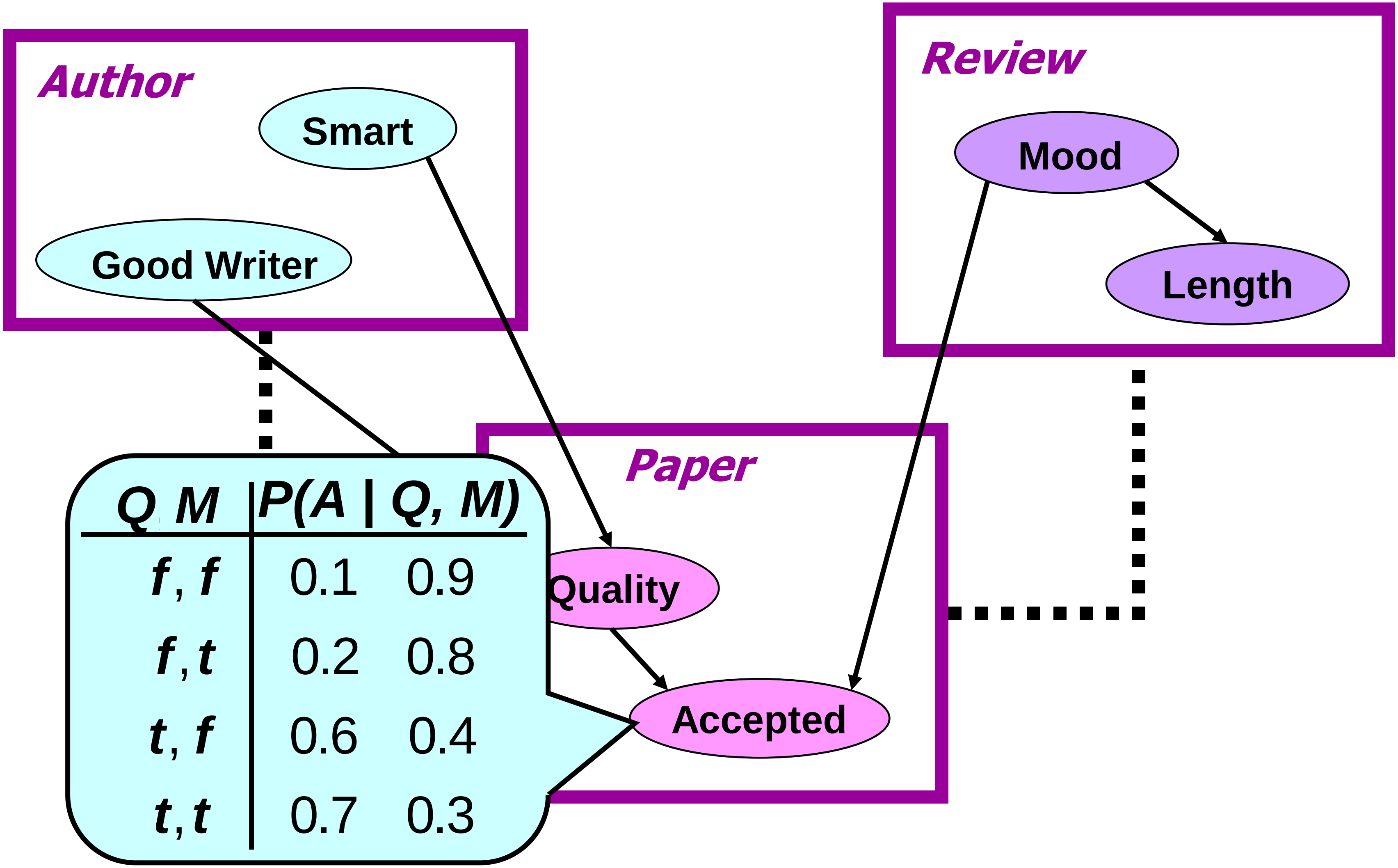
Probabilistic Relational Model



Probabilistic Relational Model

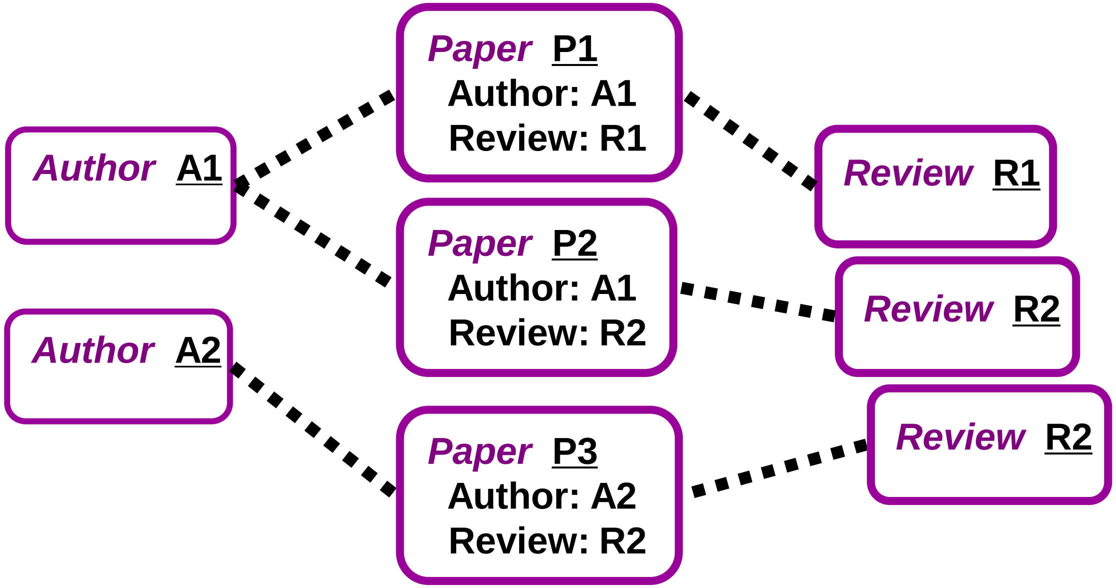


Probabilistic Relational Model



Q, M	P(A | Q, M)	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

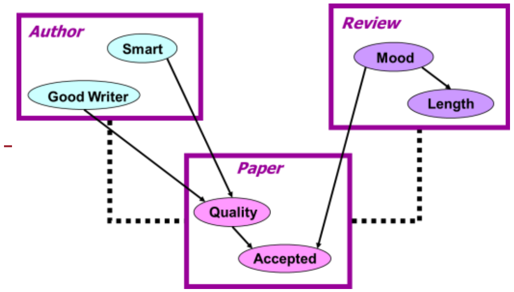
Relational Skeleton

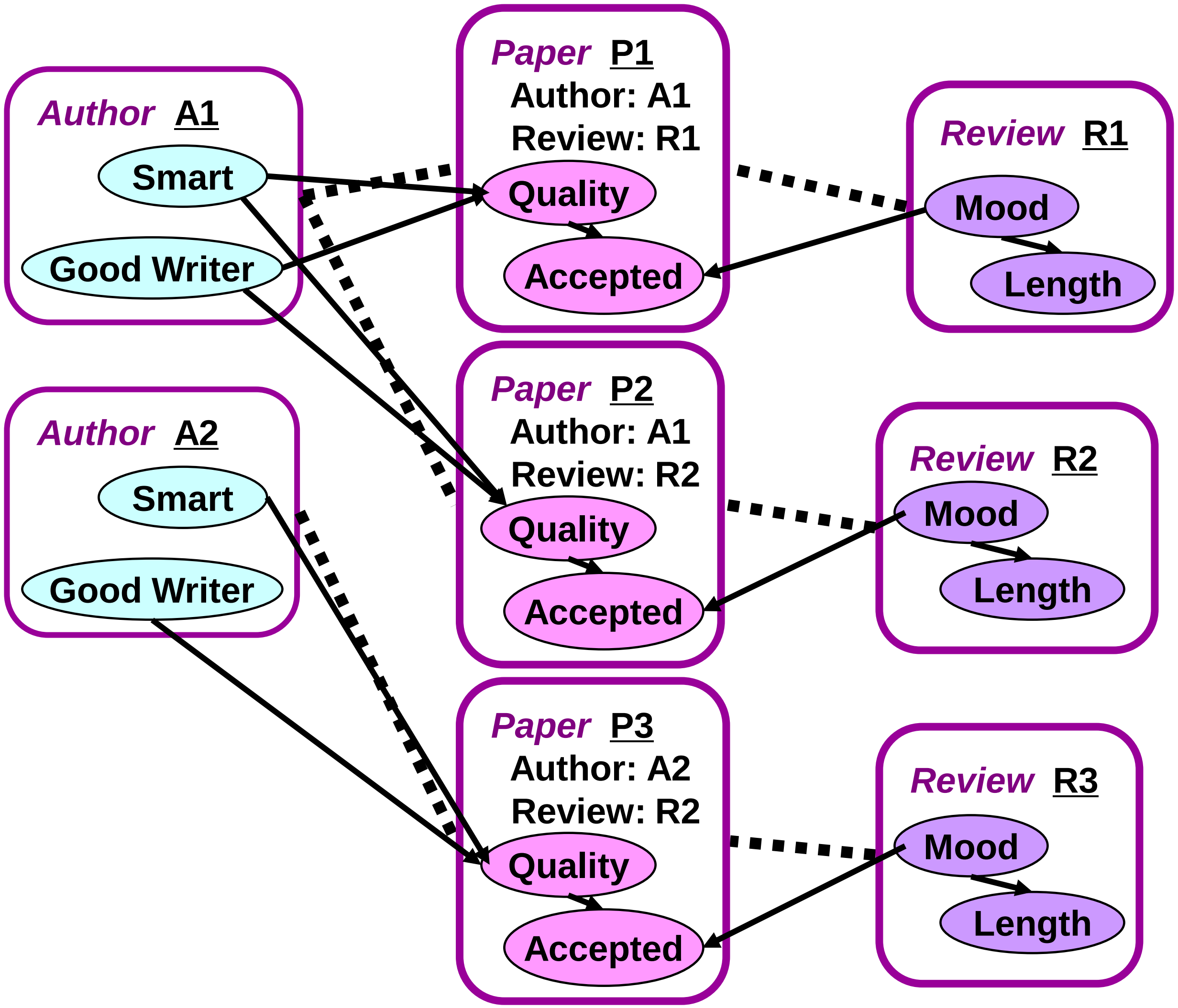


Fixed relational skeleton σ:

- set of objects in each class
- relations between them
- attribute values unknown

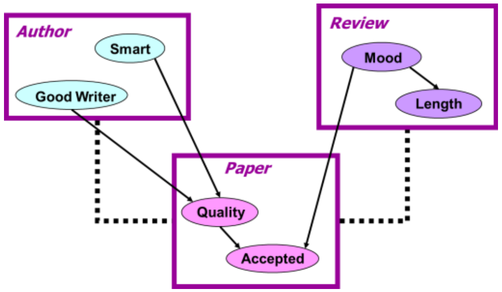
PRM with Attribute Uncertainty

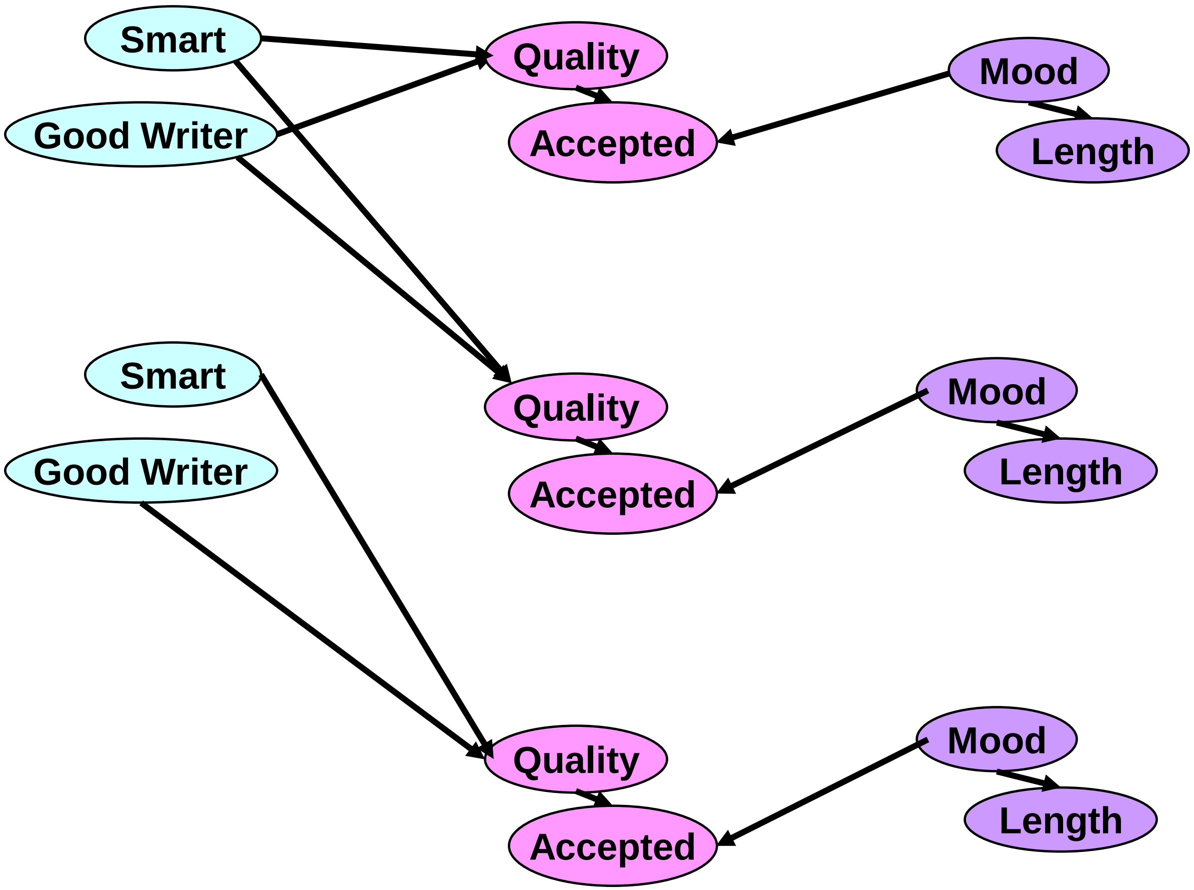




PRM unrolled wrt. the relational skeleton produces a BN that models the distribution over instantiations of attributes.

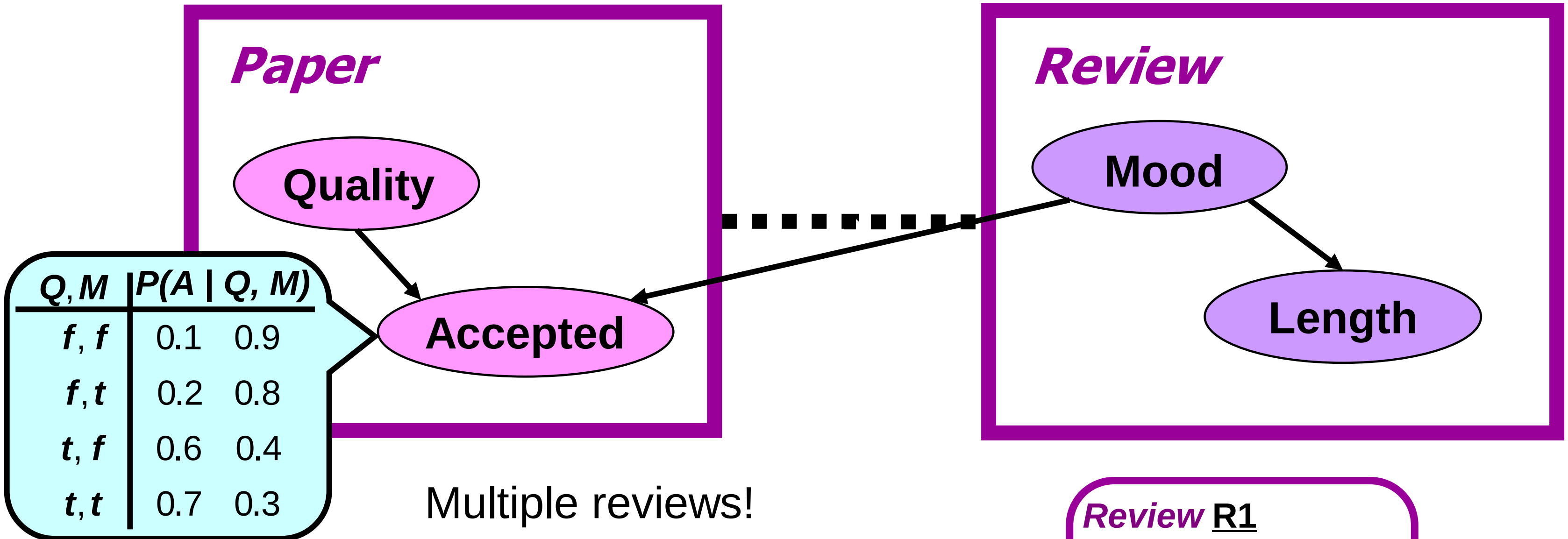
PRM with Attribute Uncertainty





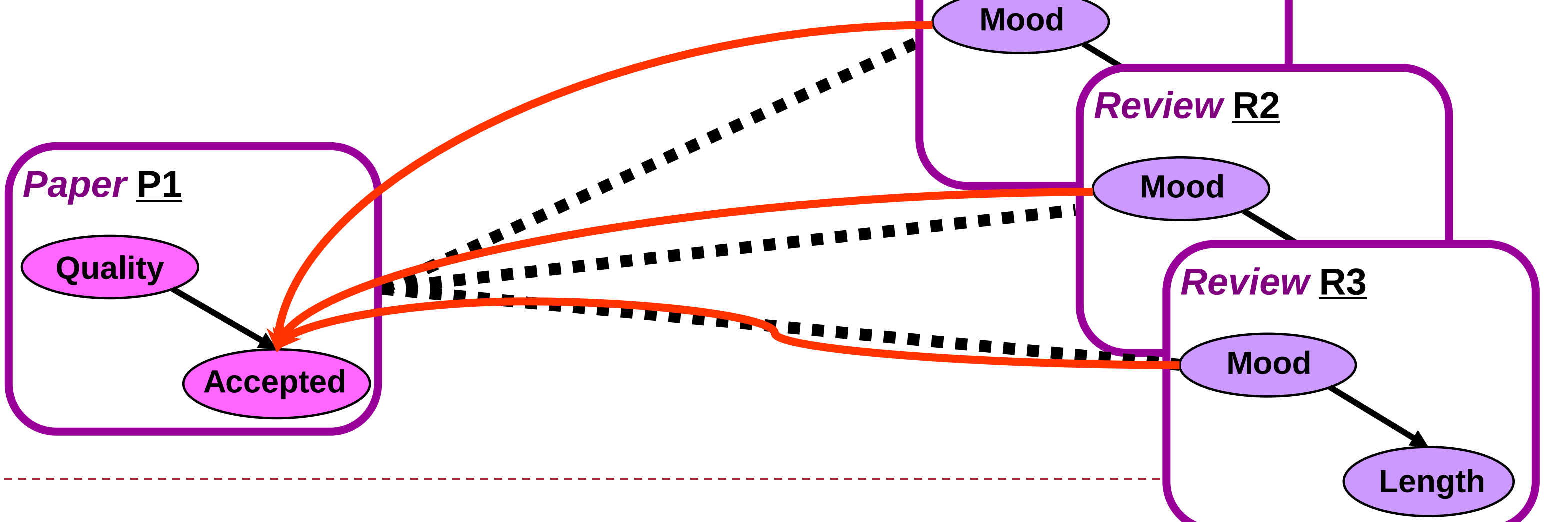
PRM unrolled wrt. the relational skeleton produces a BN that models the distribution over instantiations of attributes.

PRM: Aggregate Dependencies

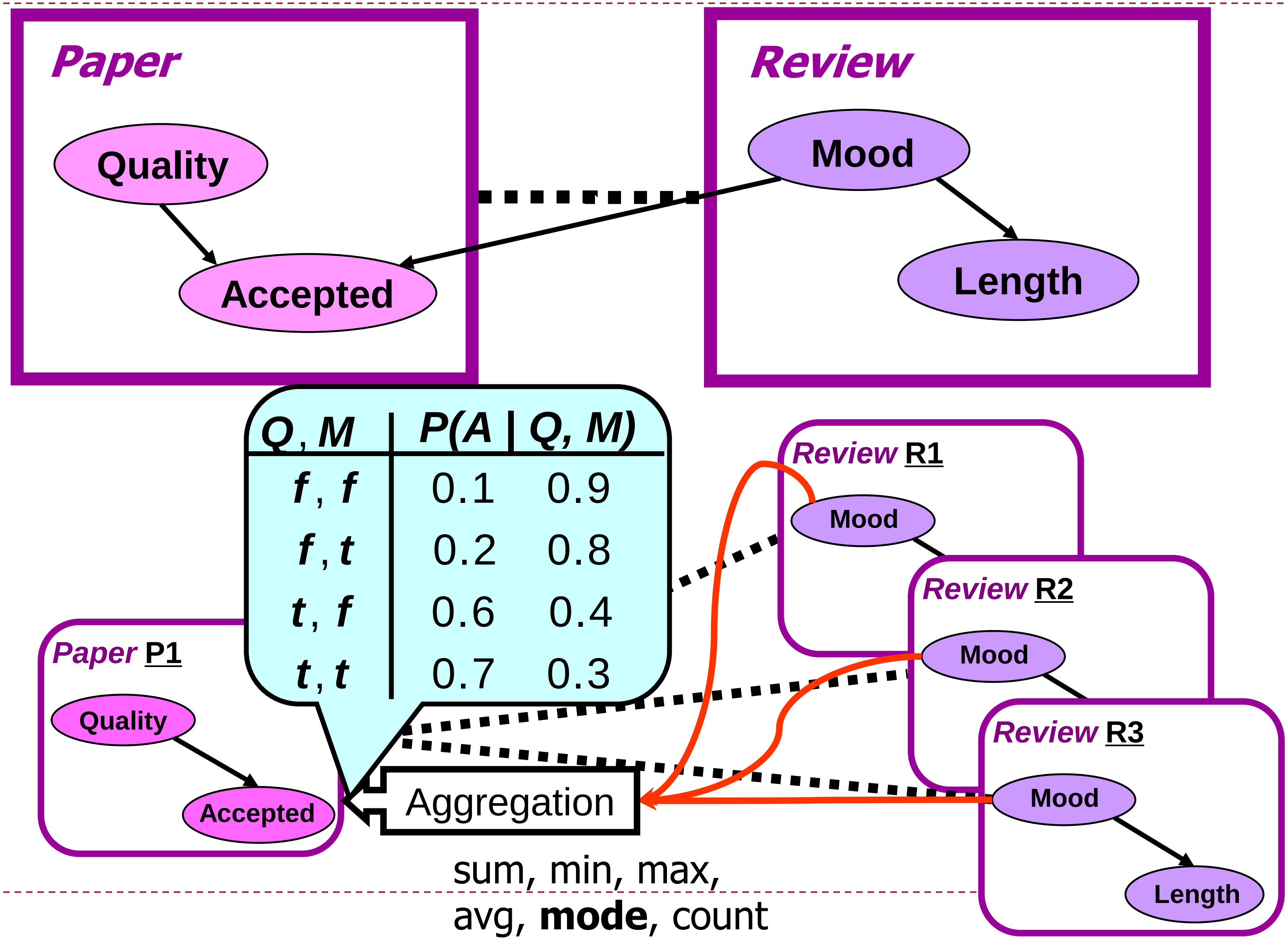


Q, M	P(A | Q, M)	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

Multiple reviews!
Number of reviews not fixed!

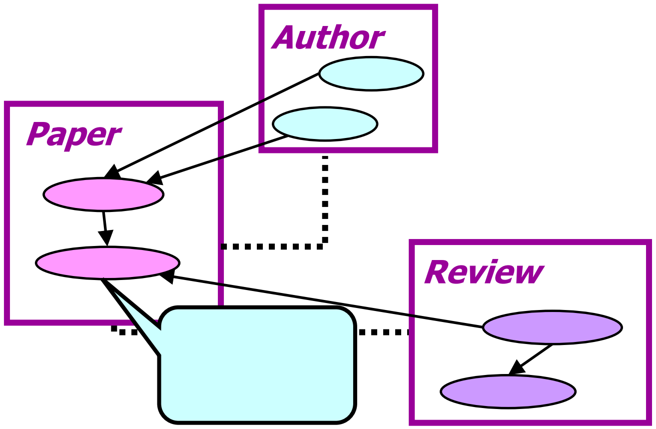


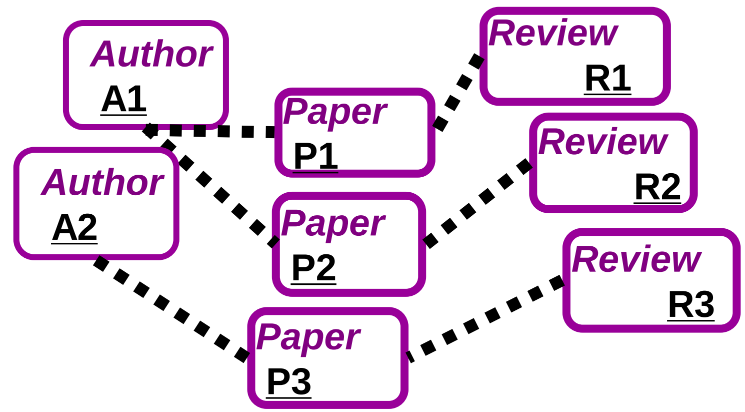
PRM: Aggregate Dependencies



Q, *M*	*P(A | Q, M)*	
f*, *f	0.1	0.9
f*, *t	0.2	0.8
t*, *f	0.6	0.4
t*, *t	0.7	0.3

PRM with Attribute Uncertainty





PRM (S,Θ) + relational skeleton (σ) =

probability distribution over instantiations of attributes I:

$$P(\mathrm{I} \mid \sigma, S, \Theta) = \prod_{x \in \sigma} \prod_{x.A} P(x.A \mid parents_{S,\sigma}(x.A))$$

Objects Attributes

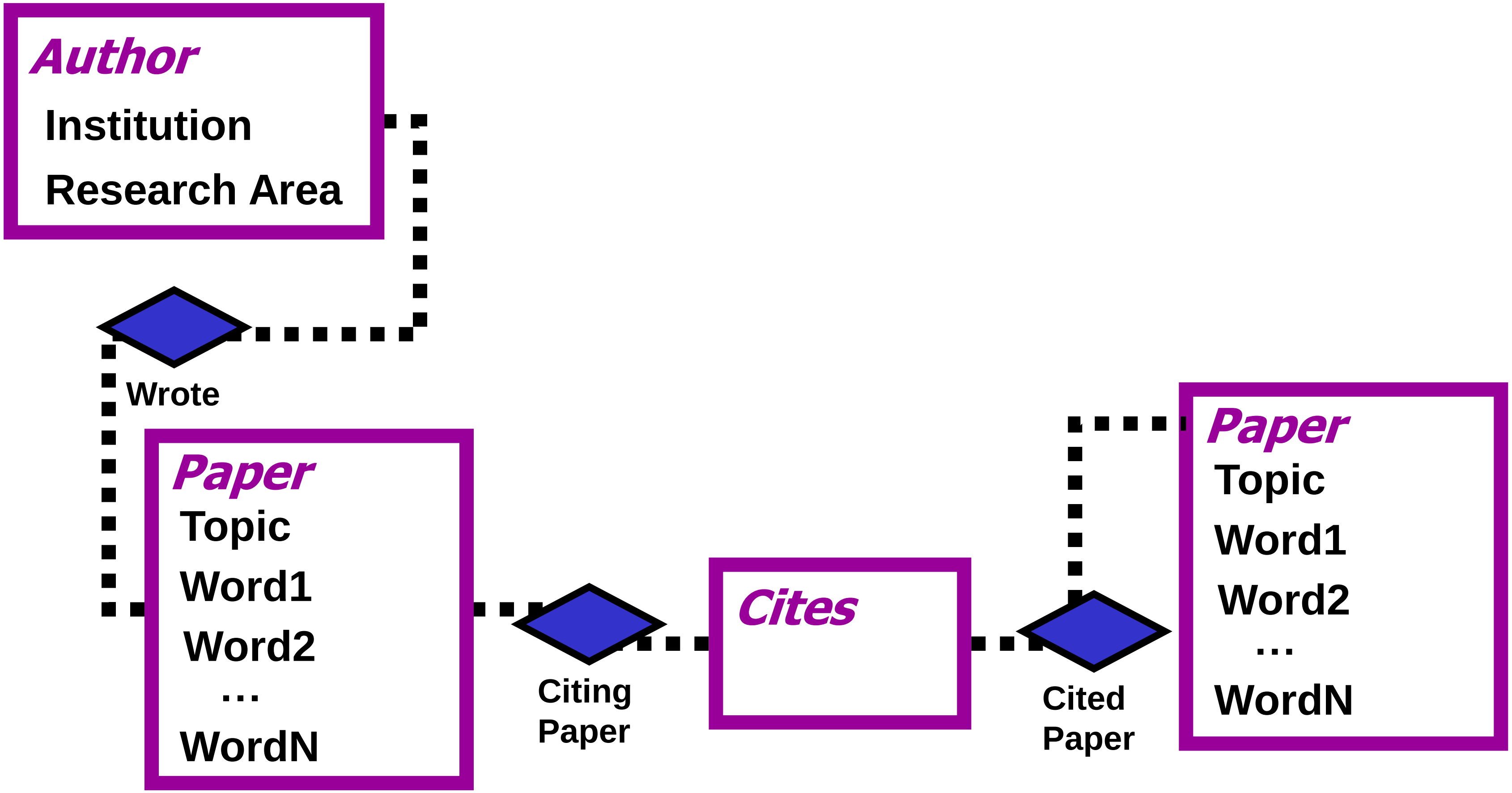
Structural Uncertainty

- PRM with AU only well-defined when the relational skeleton is known
- What if we are uncertain about the relational structure?
  - How many objects does an object relate to?
  - Which object is an object related to?
  - Does an object actually exist?
  - Are two objects identical?

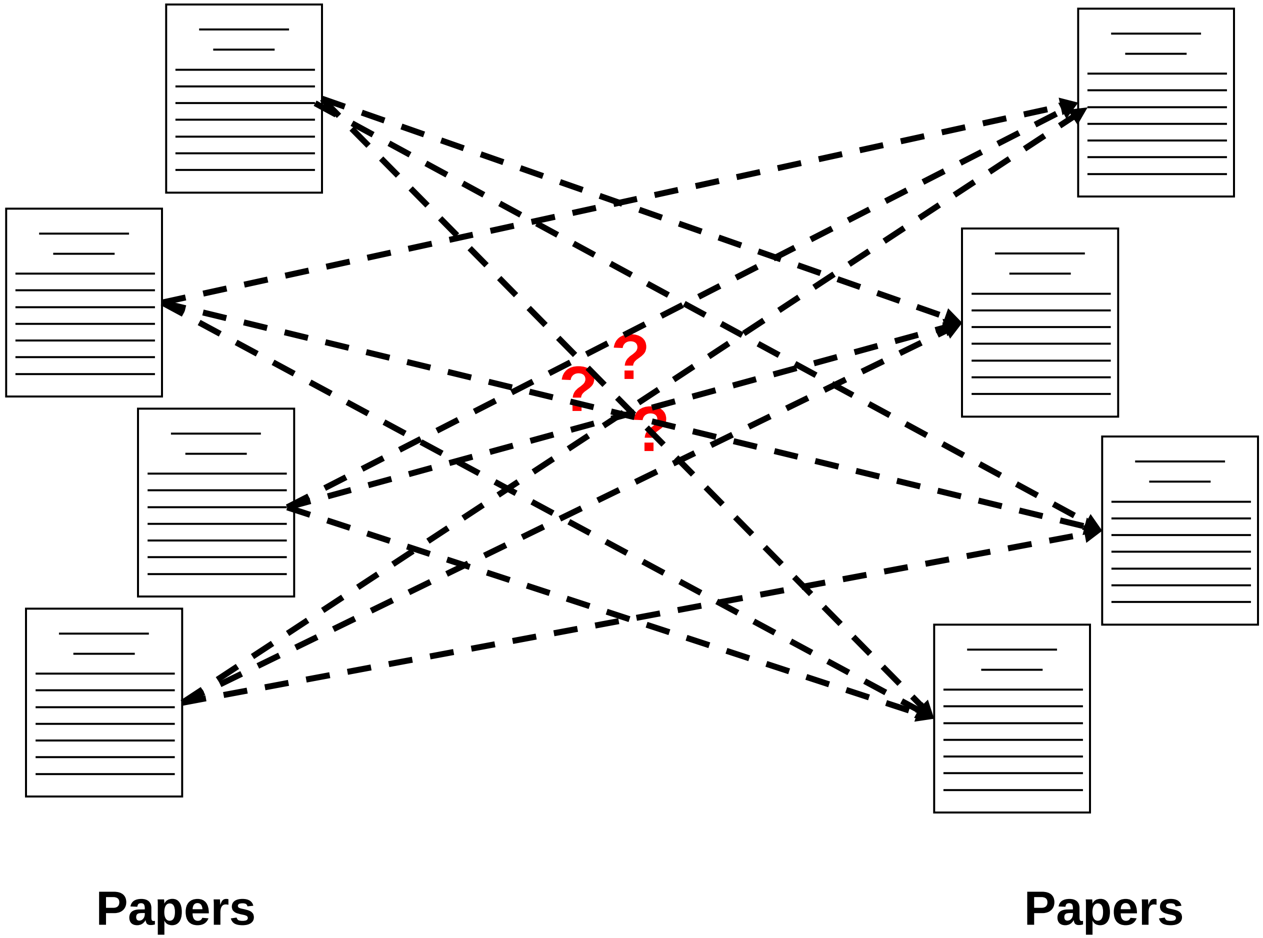
Structural Uncertainty

- Need probabilistic models that capture structural uncertainty
- Types of SU:
  - Existence uncertainty
  - Reference uncertainty
  - Number uncertainty
  - Type uncertainty
  - Identity uncertainty

Citation Schema



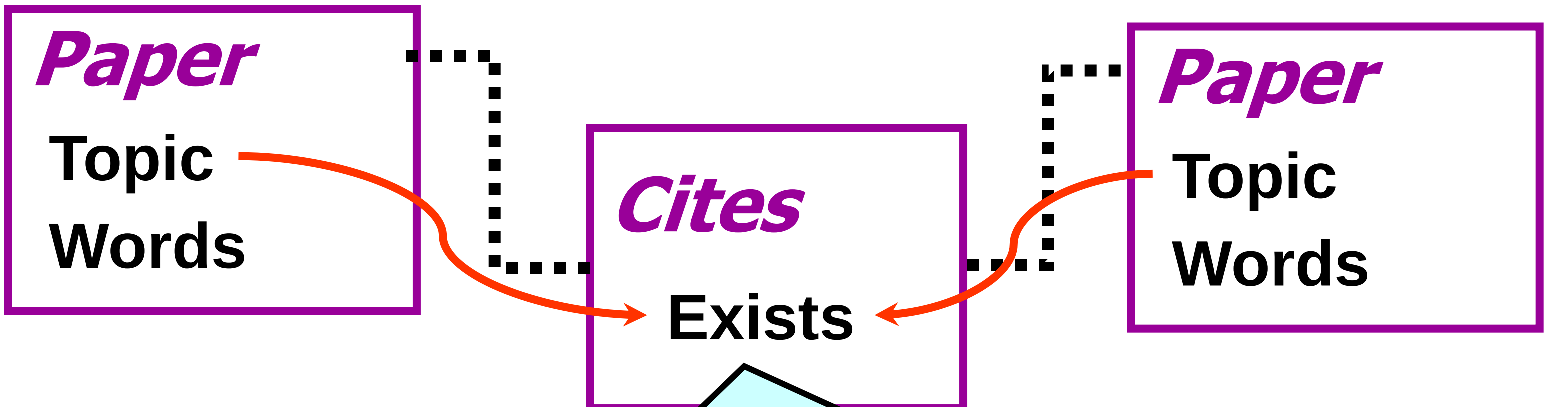
Existence Uncertainty



PRM with Existence Uncertainty

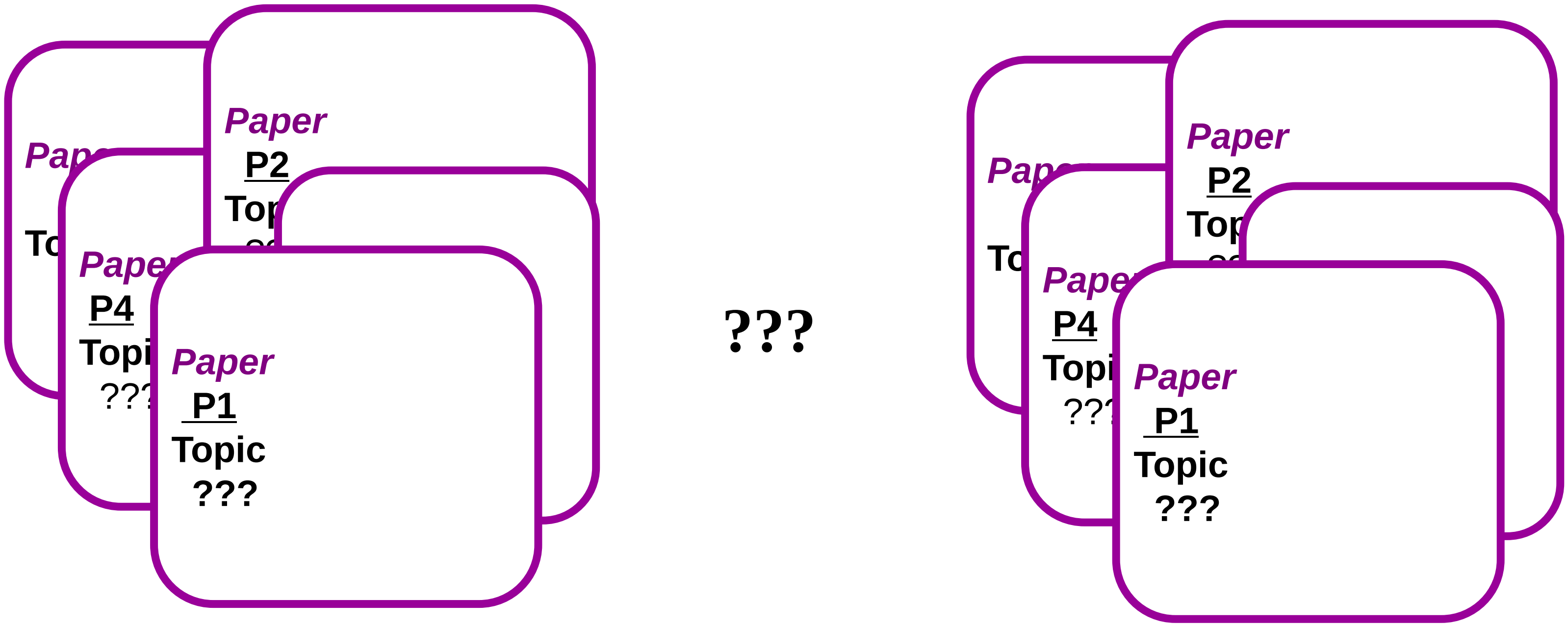
Introduce the **Exists** attribute for ***Cites***

PRM with Existence Uncertainty



Citer.Topic	Cited.Topic	*False*	*True*
Theory	*Theory*	0.995	0.005
Theory	*AI*	0.999	0.001
AI	*Theory*	0.997	0.003
AI	*AI*	0.992	0.008

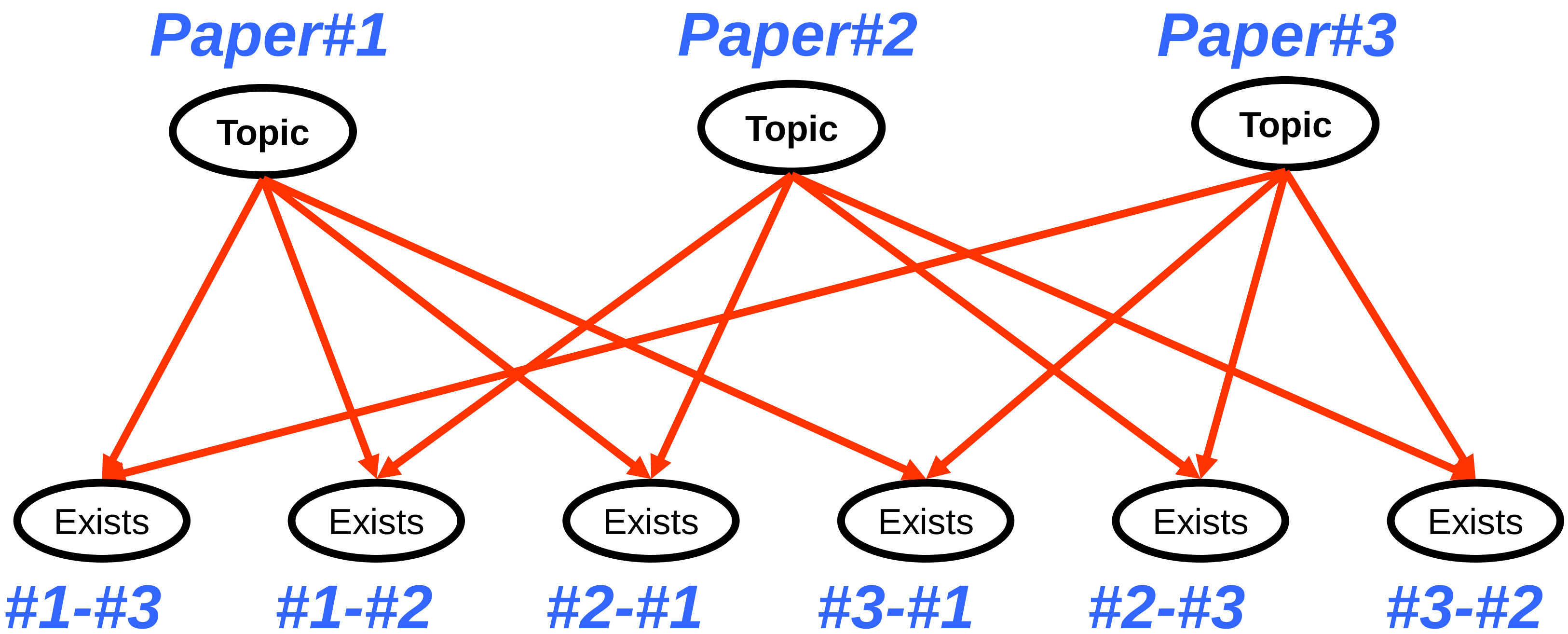
Object skeleton



Fixed object skeleton σ:

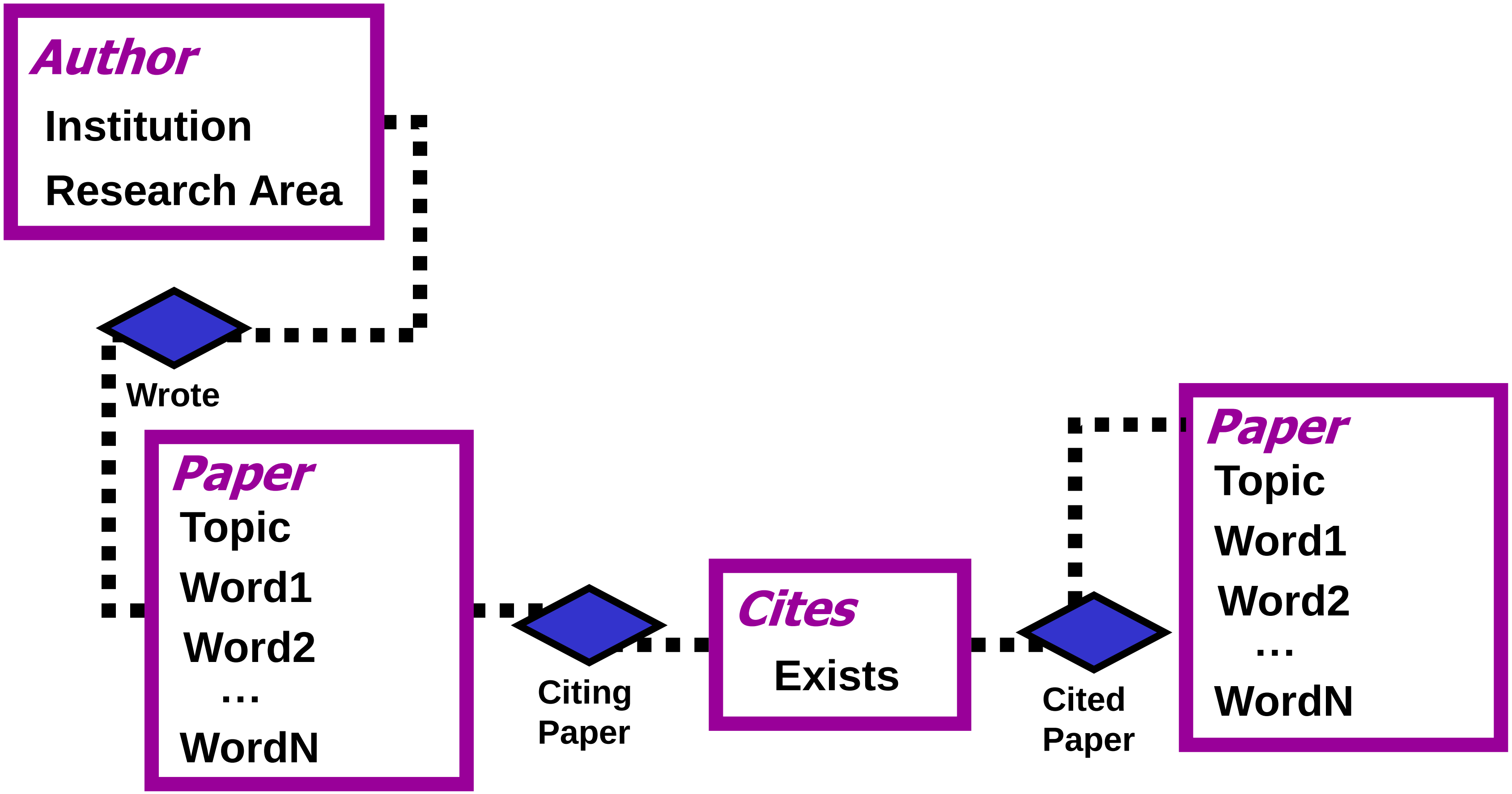
- set of objects in each class
- unknown relations between them
- unknown attribute values

PRM with Existence Uncertainty

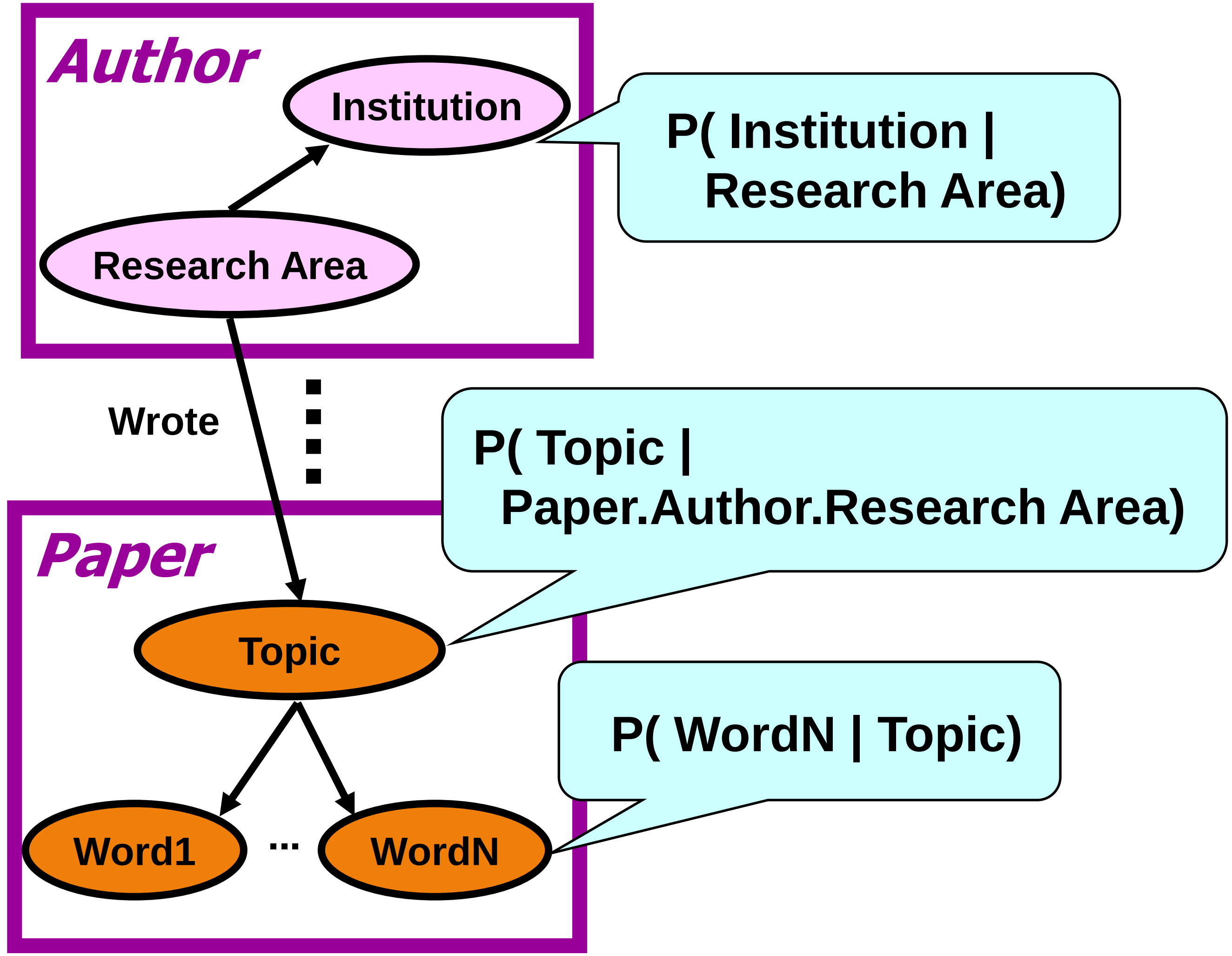


PRM w/ EU unrolled wrt. the object skeleton produces a BN

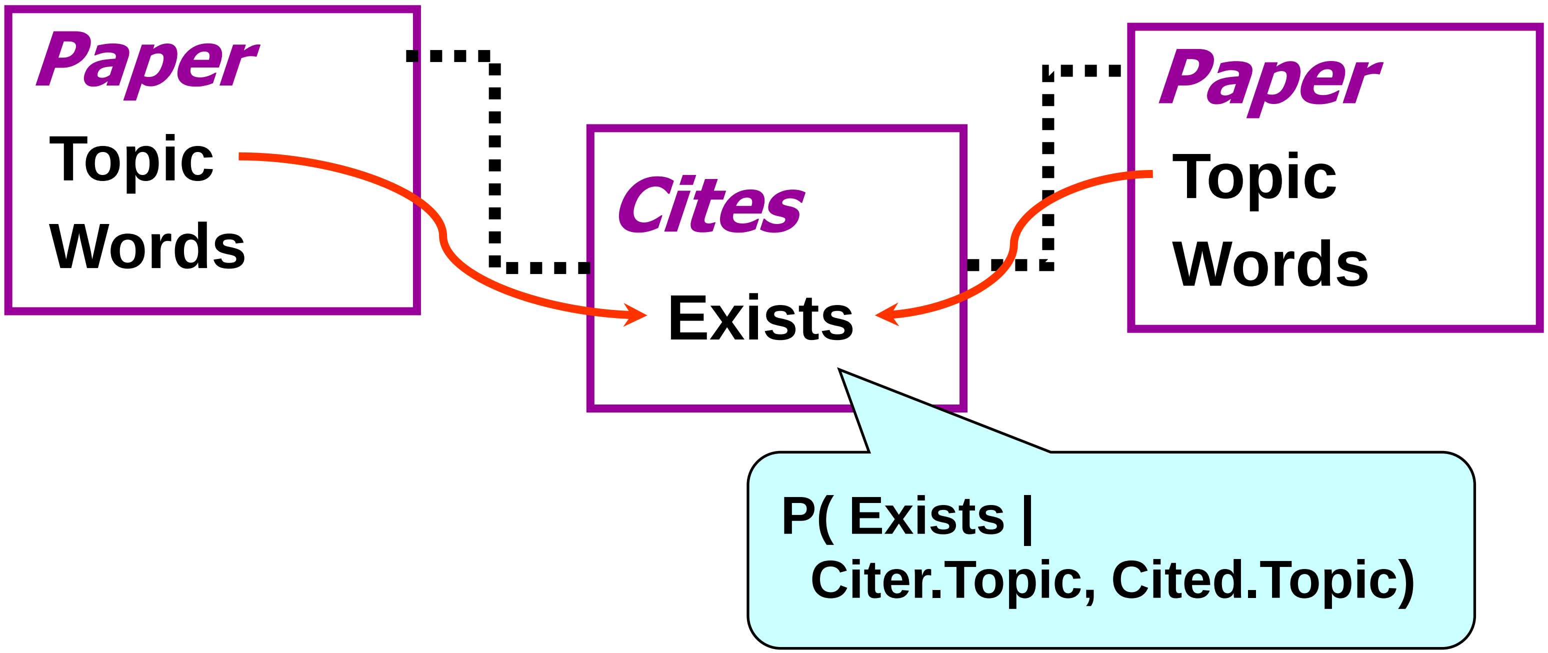
A more complicated example



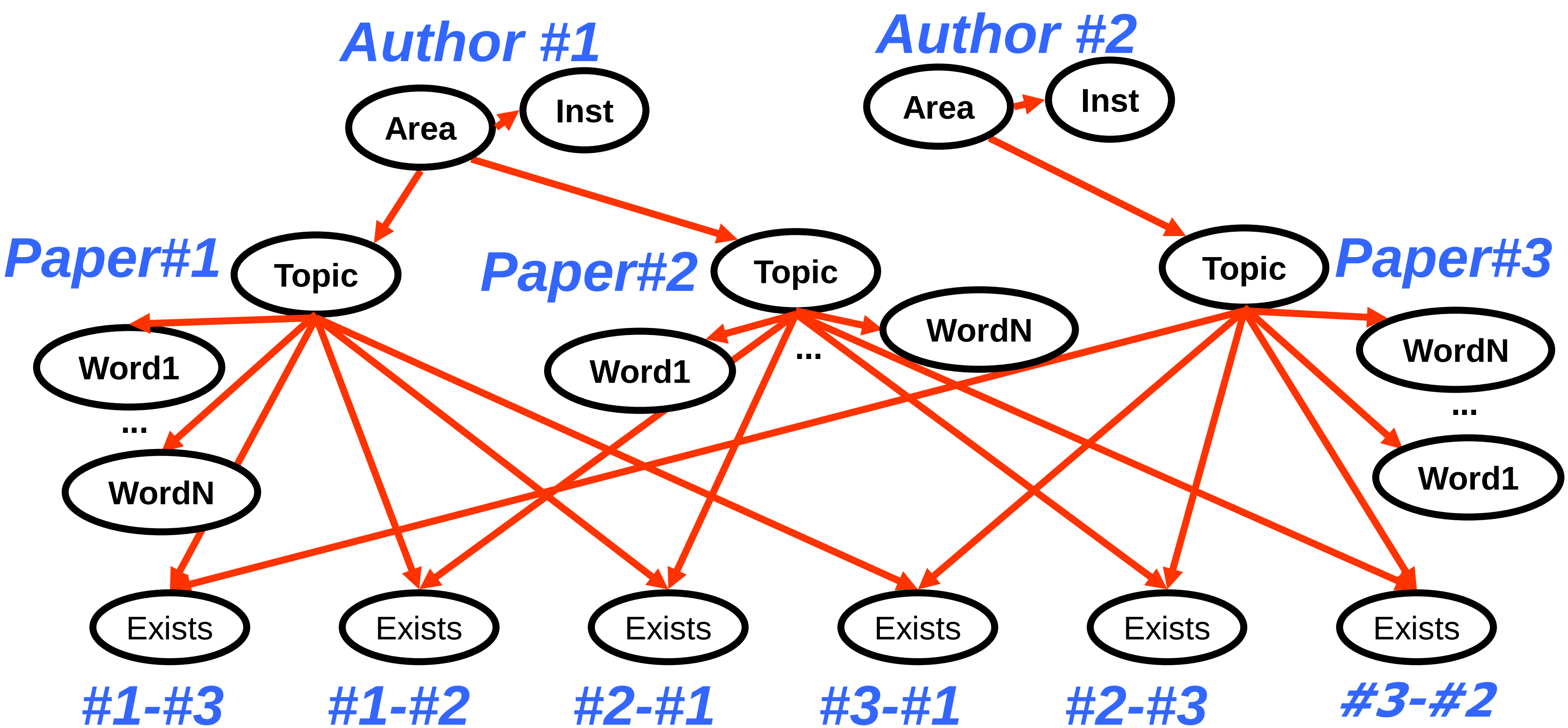
PRM with Attribute Uncertainty



PRM with Existence Uncertainty



PRM with Existence Uncertainty



Markov Logic

Markov Logic

- Logical language
  - First-order logic
- Probabilistic language
  - Markov networks

Review: Markov networks

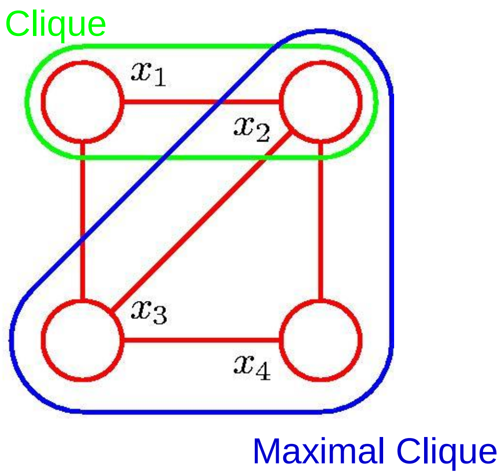
- A Markov network (or Markov random field) encodes a joint distribution with an undirected graph

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

where $\psi_C(\mathbf{x}_C)$ is the potential over clique C and

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

is the normalization coefficient.



Markov Logic: Intuition

- A logical KB is a set of **hard constraints** on the set of possible worlds
  - If a world violates a formula, it becomes impossible
- Let's make them **soft constraints**: When a world violates a formula, it becomes less probable, not impossible
- Give each formula a **weight** (Higher weight $\Rightarrow$ Stronger constraint)

$$\mathrm{P(world)} \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

Markov Logic: Definition

- A Markov Logic Network (MLN) is a set of pairs (F, w) where
  - F is a formula in first-order logic
  - w is a real number
- Together with a set of constants, it defines a Markov network with
  - One node for each grounding of each predicate in the MLN
    - This is *propositionalization* (remember?)
  - One clique for each grounding of each formula F in the MLN, with the potential being:
    - exp(w) for node assignments that satisfy F
    - 1 otherwise

Example: Friends & Smokers

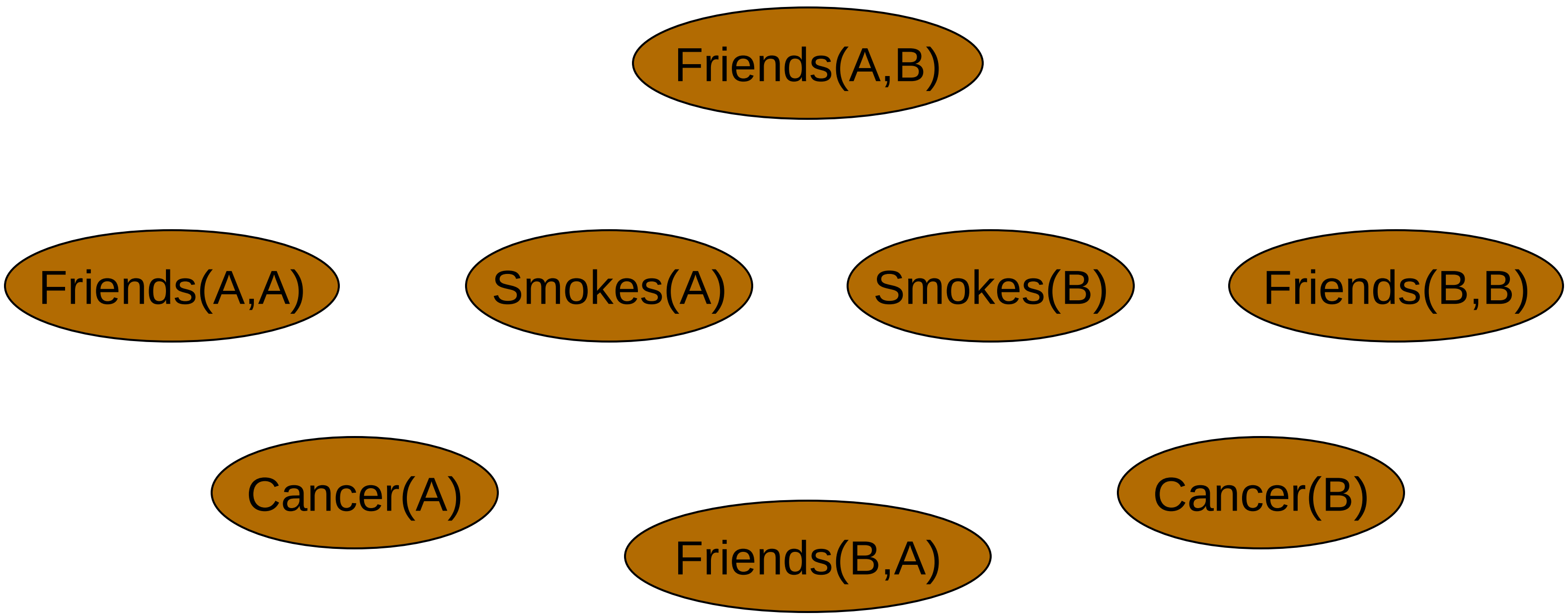
Smoking causes cancer.

Friends have similar smoking habits.

Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$

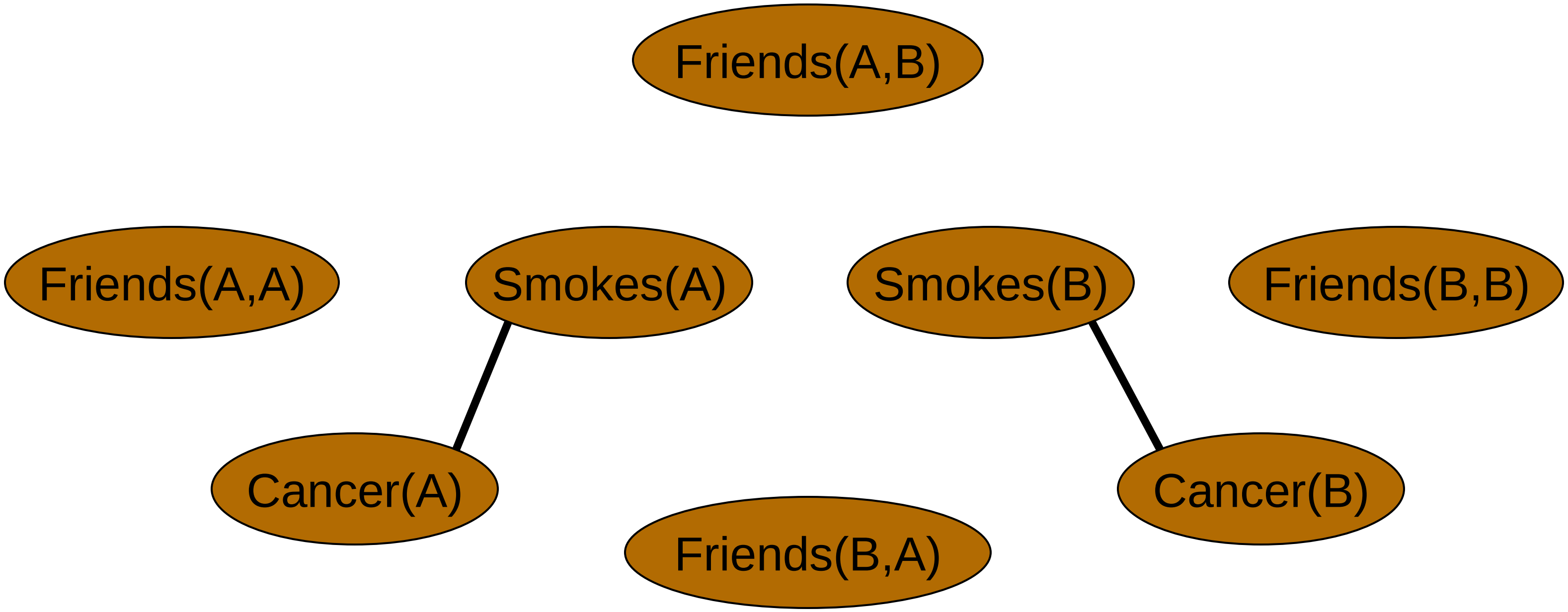
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$

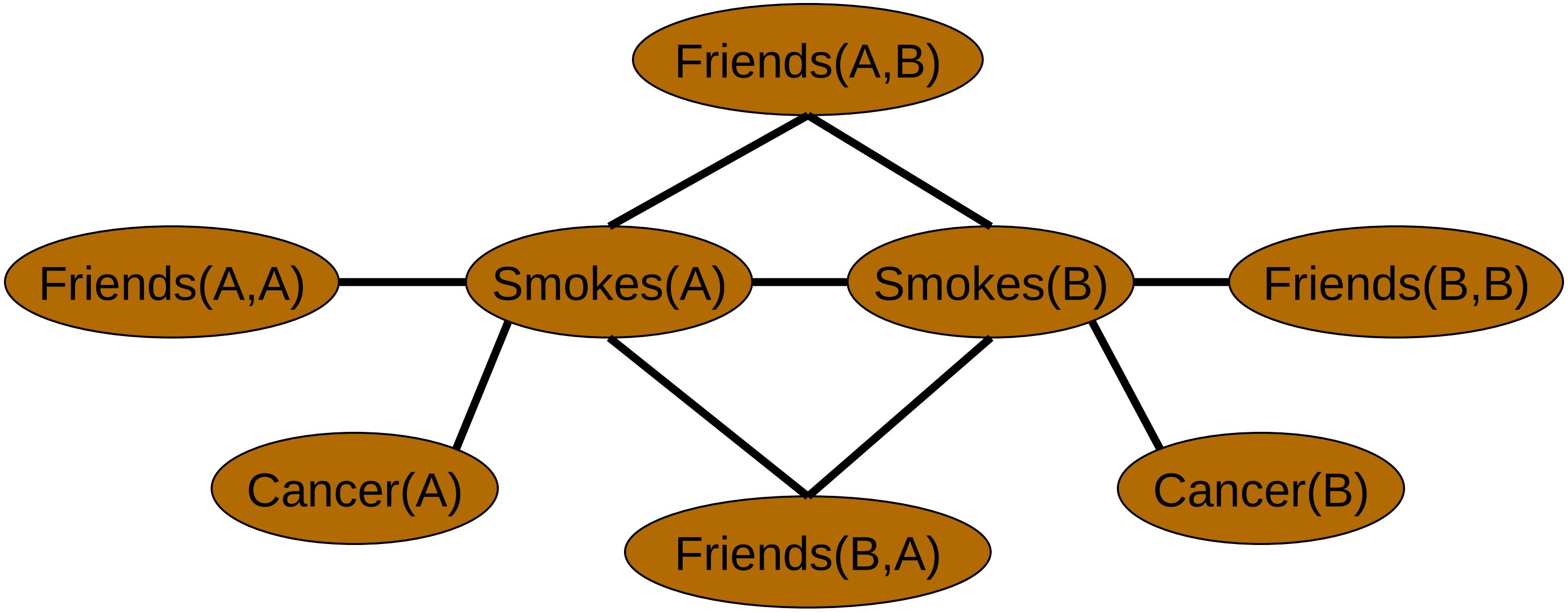
Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)

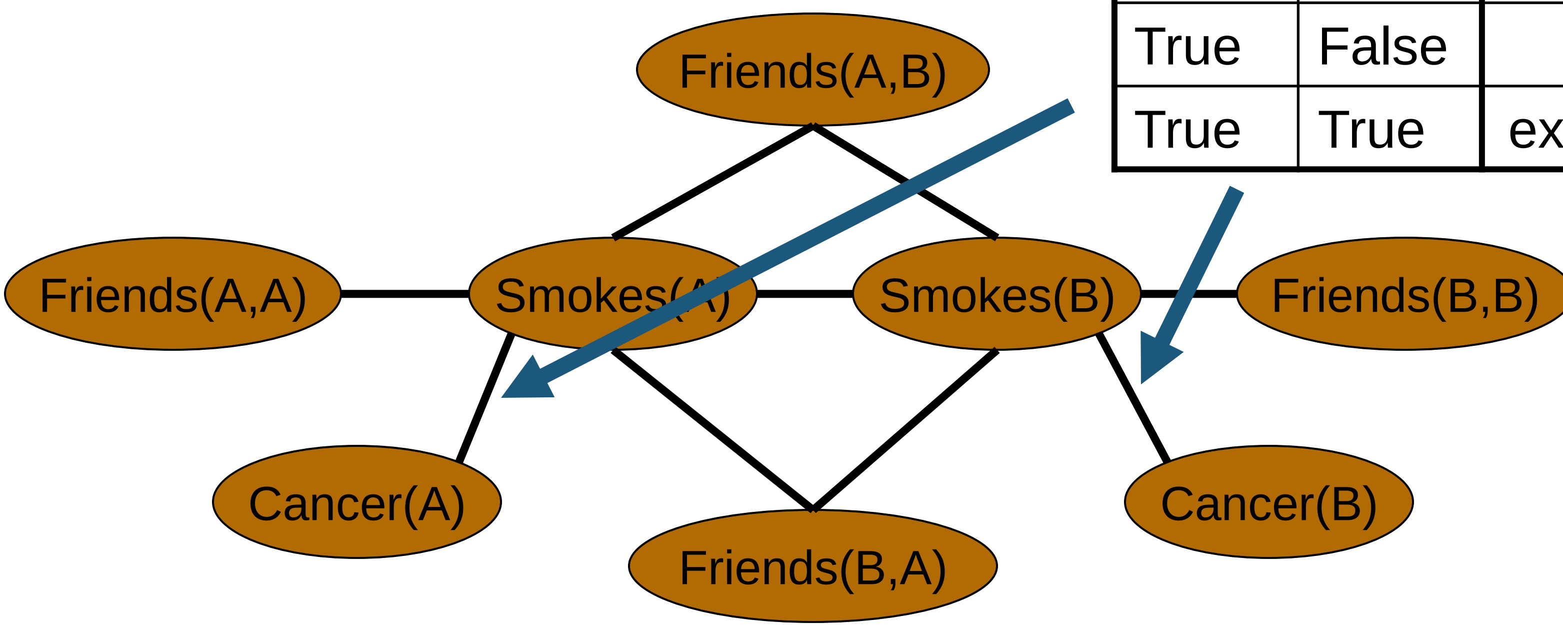


Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \big(Smokes(x) \Leftarrow$

S	C	Φ(S,C)
False	False	exp(1.5)
False	True	exp(1.5)
True	False	1
True	True	exp(1.5)

Two constants: **Anna** (A) and **Bob** (B)



Markov Logic Networks

- MLN is **template** for ground Markov nets
- Probability of a world x:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i \boxed{w_i} \boxed{n_i(x)} \right)$$

Weight of formula *i*

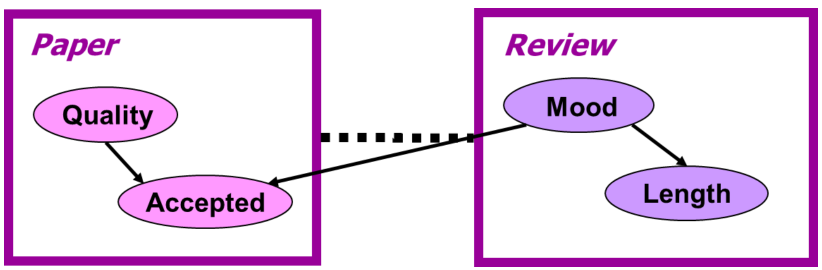
No. of true groundings of formula *i* in *x*

Relation to First-Order Logic

- Infinite weights $\Longrightarrow$ First-order logic
  - P(x)>0 iff. x satisfies KB
- Markov logic allows contradictions between formulas

Relation to PRM

- MLN is More general and flexible than PRM
- In principle, a PRM can be converted into a MLN by writing a formula for each entry of each CPT and setting the weight to be the logarithm of the conditional probability



Q, M	**P(A | Q, M)**	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

$\log 0.1$: $\forall x,y\ hasReview(x,y) \wedge Q(x,f) \wedge M(y,f) \wedge A(x,t)$

$\log 0.9$: $\forall x,y\ hasReview(x,y) \wedge Q(x,f) \wedge M(y,f) \wedge A(x,f)$

$\log 0.2$: $\forall x,y\ hasReview(x,y) \wedge Q(x,f) \wedge M(y,t) \wedge A(x,t)$

……

Software of MLN

- Alchemy
  - https://alchemy.cs.washington.edu/

Inference

Inference

- A naive approach
  - Unroll the model to a BN or MN and run inference algorithms (such as VE)
  - Problem: the BN/MN may be very large and highly interconnected

- Lifted inference
  - Lots of repeated structures in the unrolled model $\Rightarrow$ repeated computation in inference
  - Group similar random variables at the FOL level and handle them at the same time

Summary

- Probabilistic Relational Models
  - Logical language: Frame
  - Probabilistic language: Bayes nets
  - Bayes net template for object classes
  - Object’s attrs. can depend on attrs. of related objs.
- Markov Logic
  - Logical language: First-order logic
  - Probabilistic language: Markov networks
  - Syntax: First-order formulas with weights
  - Semantics: Templates for Markov net cliques