Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 34

Dec, 2, 2015

Slide source: from David Page (MIT) (which were from From Lise Getoor, Nir Friedman, Daphne Koller, and Avi Pfeffer) and from Lise Getoor

Lecture Overview

- **Recap Motivation and Representation for Probabilistic Relational Models (PRMs)**
  - Full Relational Schema and its Instances
  - Relational Skeleton and its Completion Instances
- Probabilistic Model of PRMs
  - Dependency Structure
  - Parameters

How PRMs extend BNs?

1. PRMs conceptually extend BNs to allow the specification of a probability model for *classes* of objects rather than a fixed set of simple attributes

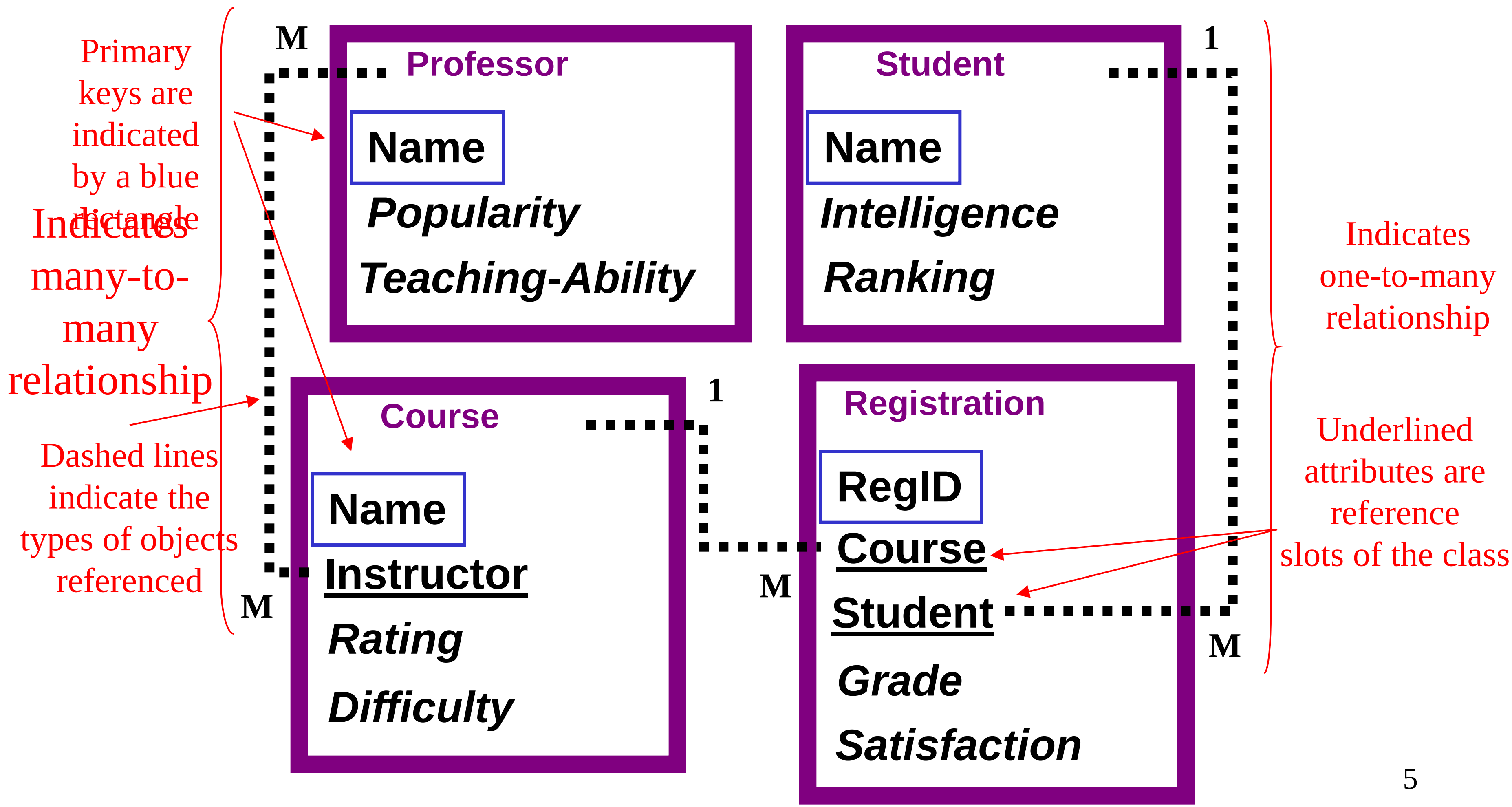
2. PRMs also allow properties of an entity to depend probabilistically on properties of other *related entities*

Mapping PRMs from Relational Models

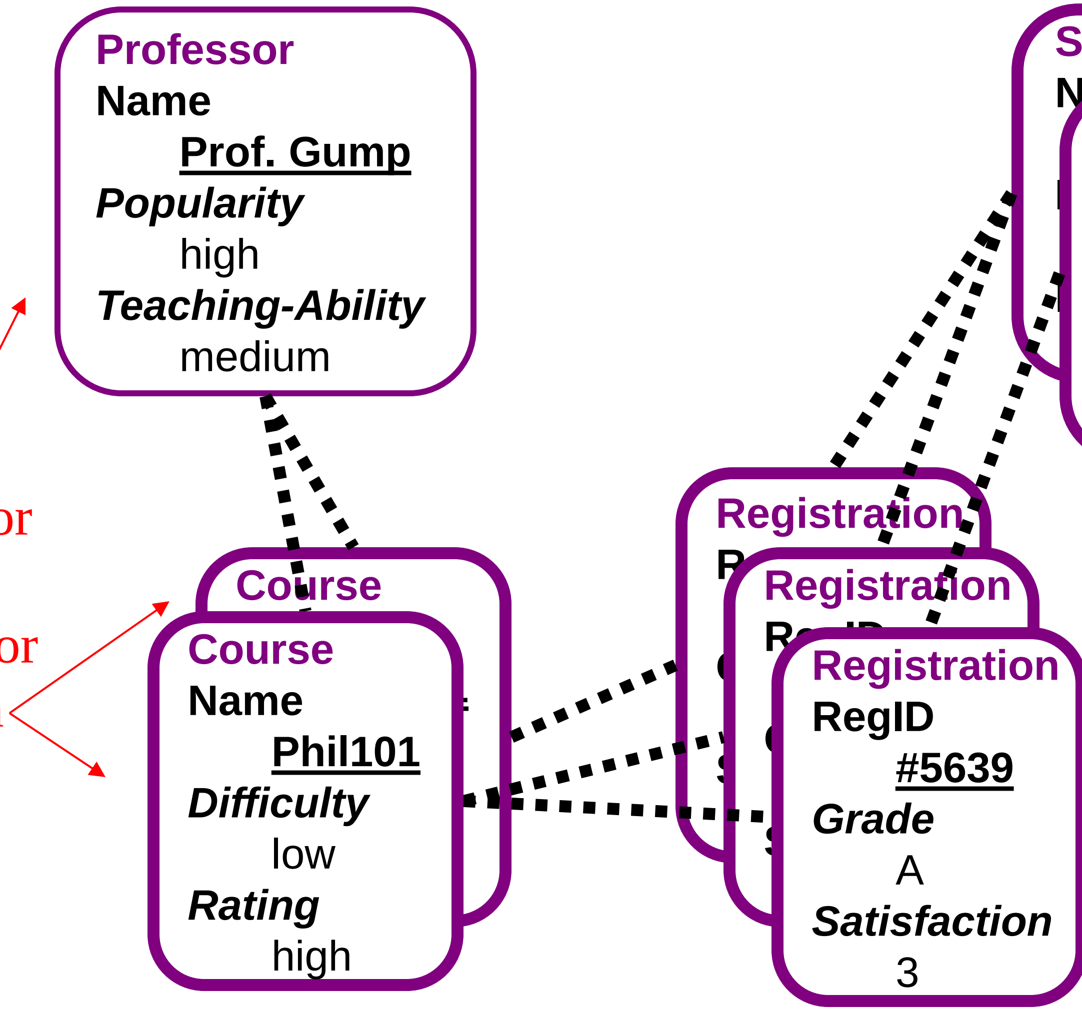
- The representation of PRMs is a direct mapping from that of relational databases

- A relational model consists of a set of *classes* $X_1,\ldots,X_n$ and a set of *relations* $R_1,\ldots,R_m$, where each relation R_i is typed

University Domain Example - Full Relational Schema



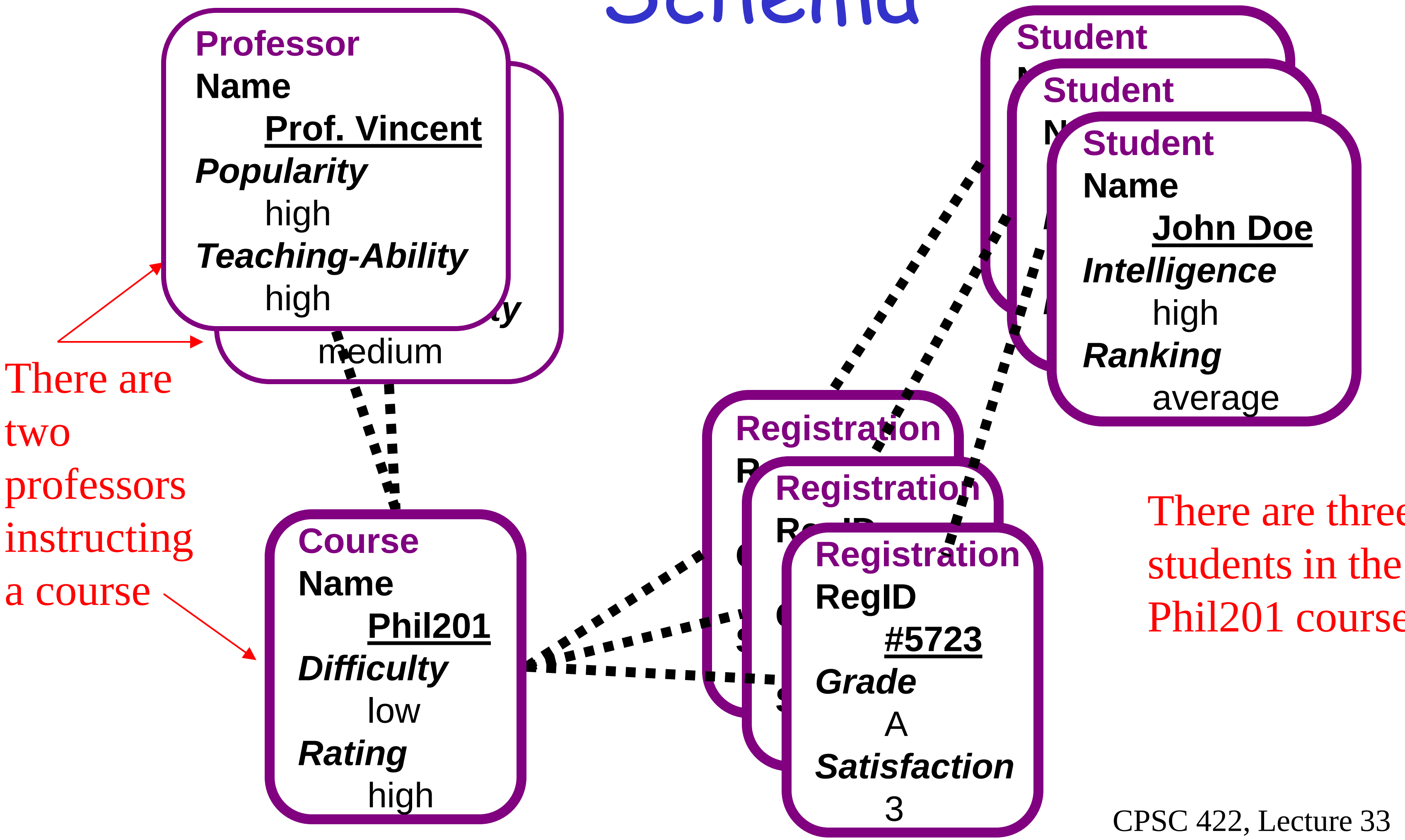
University Domain Example - An Instance of the Schema



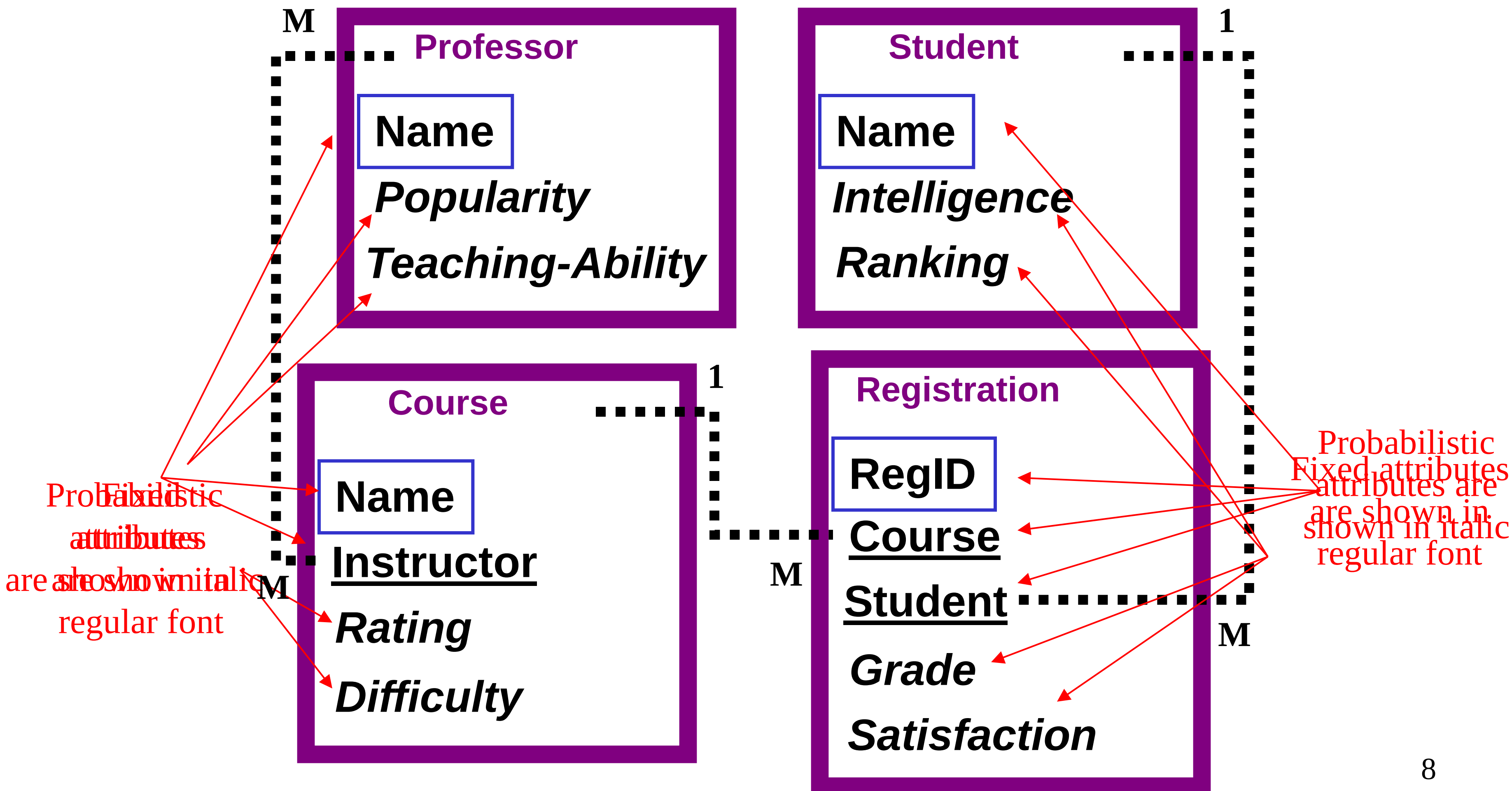
One professor is the instructor for both courses

Jane Doe is registered for only one course, Phil101, while the other student is registered for both courses

University Domain Example - Another Instance of the Schema



University Domain Example - fixed vs. probabilistic attributes

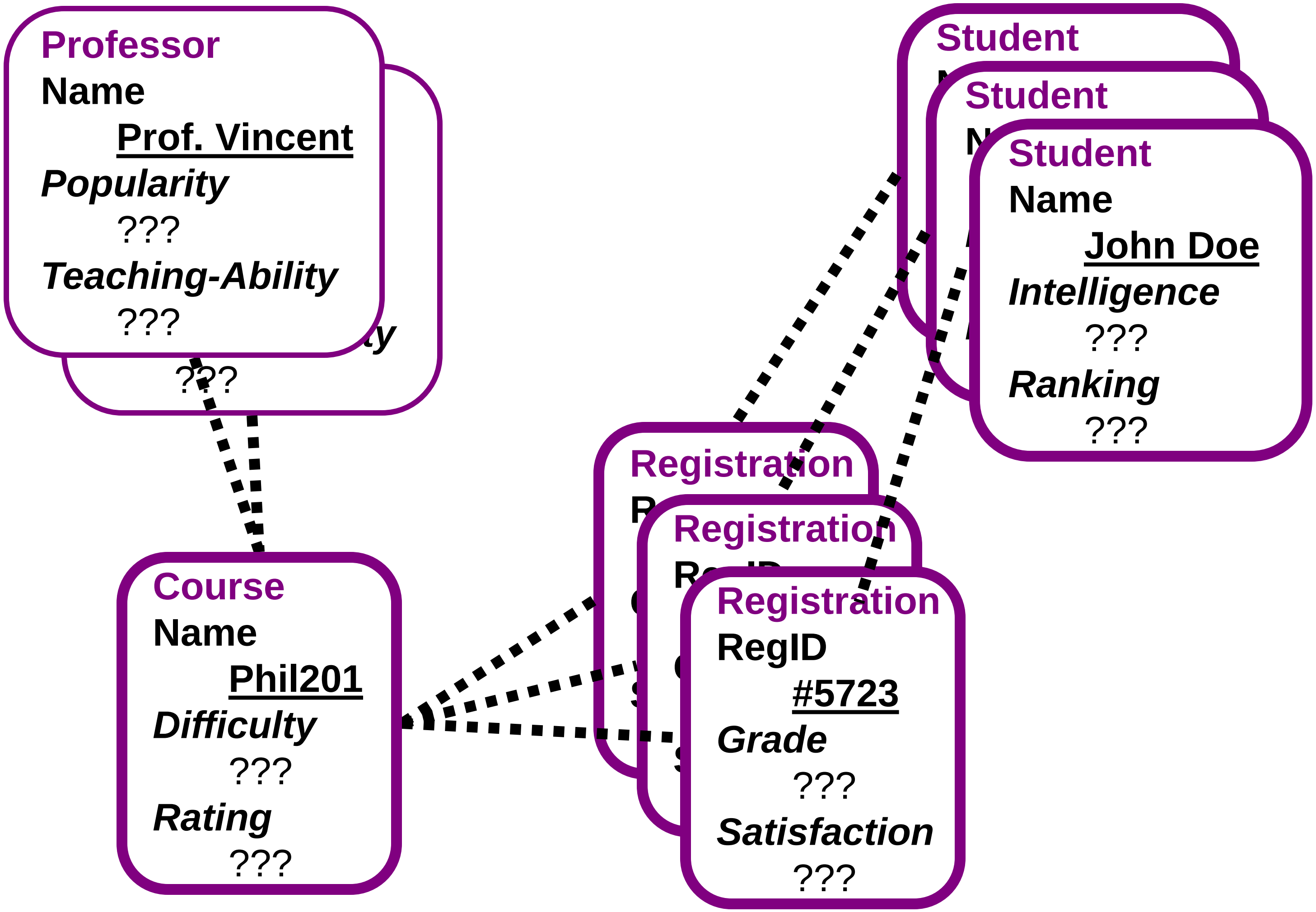


PRM Semantics: Skeleton Structure

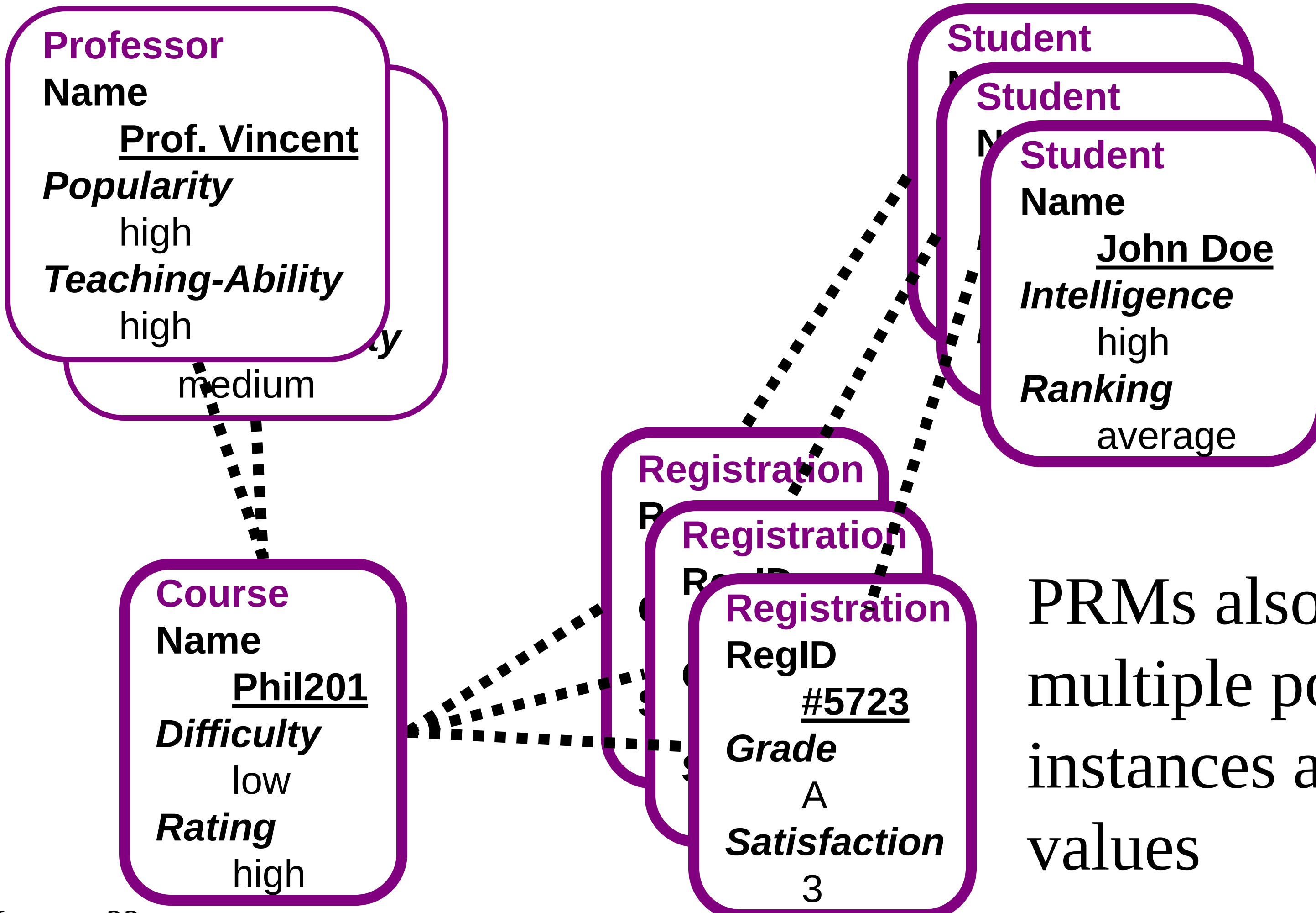
- A *skeleton structure* σ of a relational schema is a partial specification of an instance of the schema. It specifies
  - set of objects for each class,
  - values of the fixed attributes of these objects,
  - relations that hold between the objects
- The values of probabilistic attributes are left unspecified

- A *completion* I of the skeleton structure σ extends the skeleton by also specifying the values of the probabilistic attributes

University Domain Example – Relational Skeleton



University Domain Example – The Completion Instance *I*



PRMs also allow multiple possible instances and values

Lecture Overview

- Recap Motivation and Representation for Probabilistic Relational Models (PRMs)
  - Full Relational Schema and its Instances
  - Relational Skeleton and its Completion Instances
- **Probabilistic Model of PRMs**
  - Dependency Structure
  - Parameters

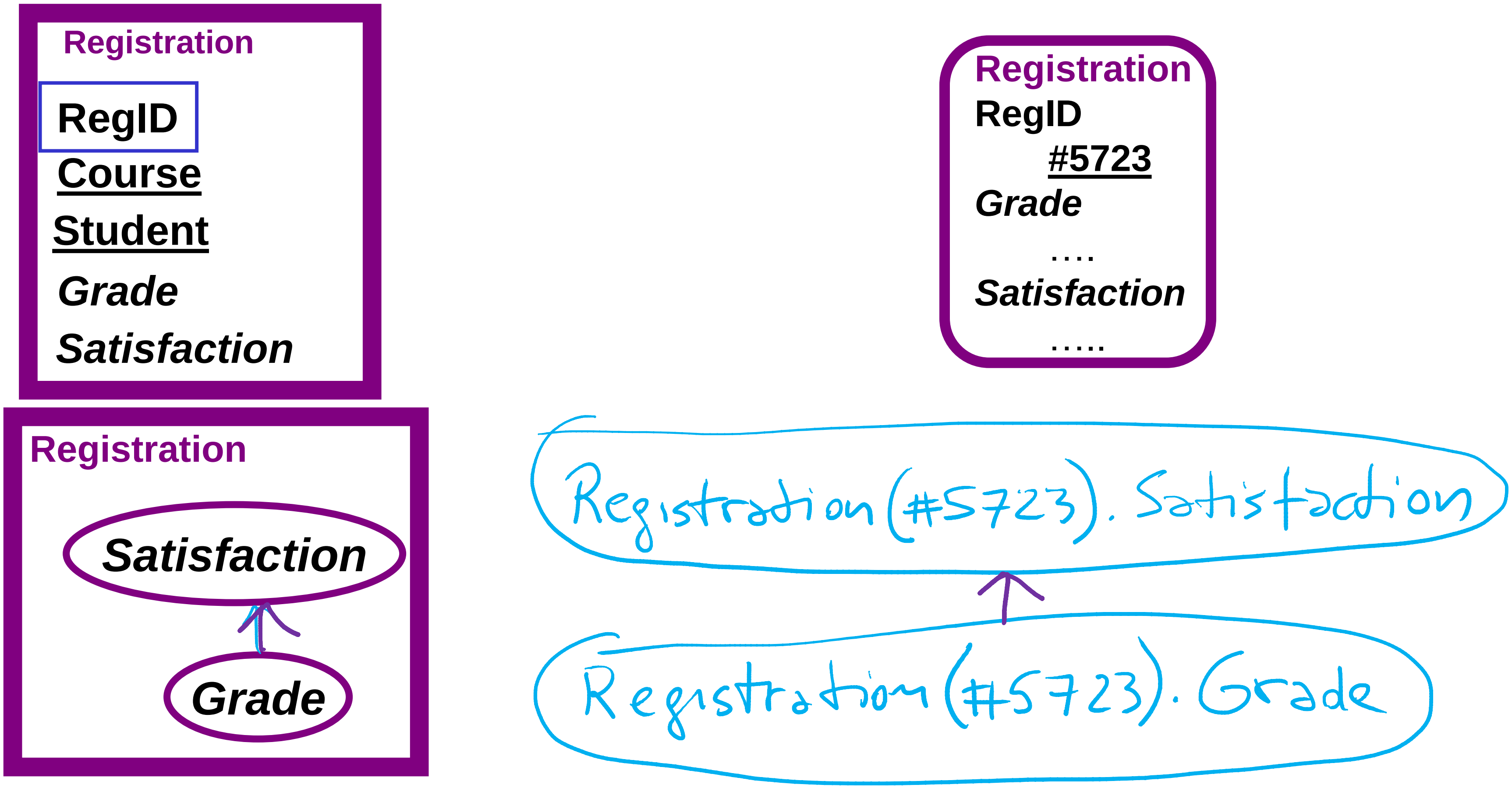
PRMs: Probabilistic Model

- **The probabilistic model consists of two components:**
  - **the qualitative dependency structure, S**
  - **the parameters associated with it, θ_S**

- **The dependency structure is defined by associating with each attribute *X.A* a set of *parents* Pa(X.*A*); parents are attributes that are "direct influences" on *X.A*. This dependency holds for any object of class *X***

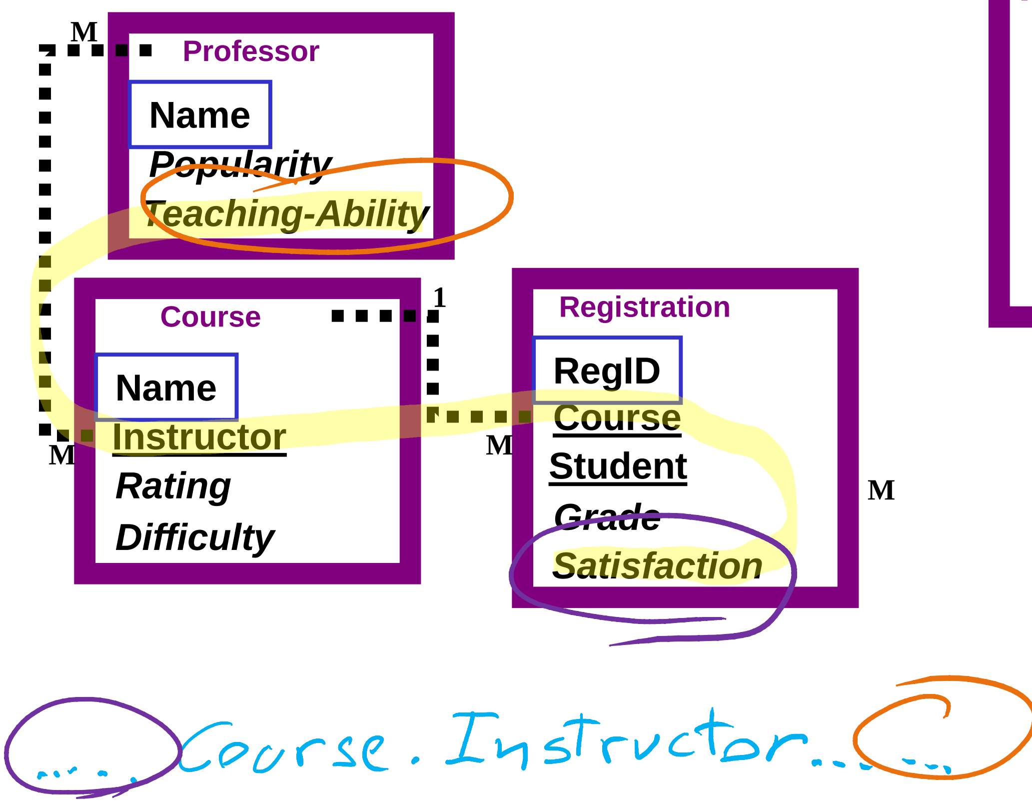
Dependencies within a class

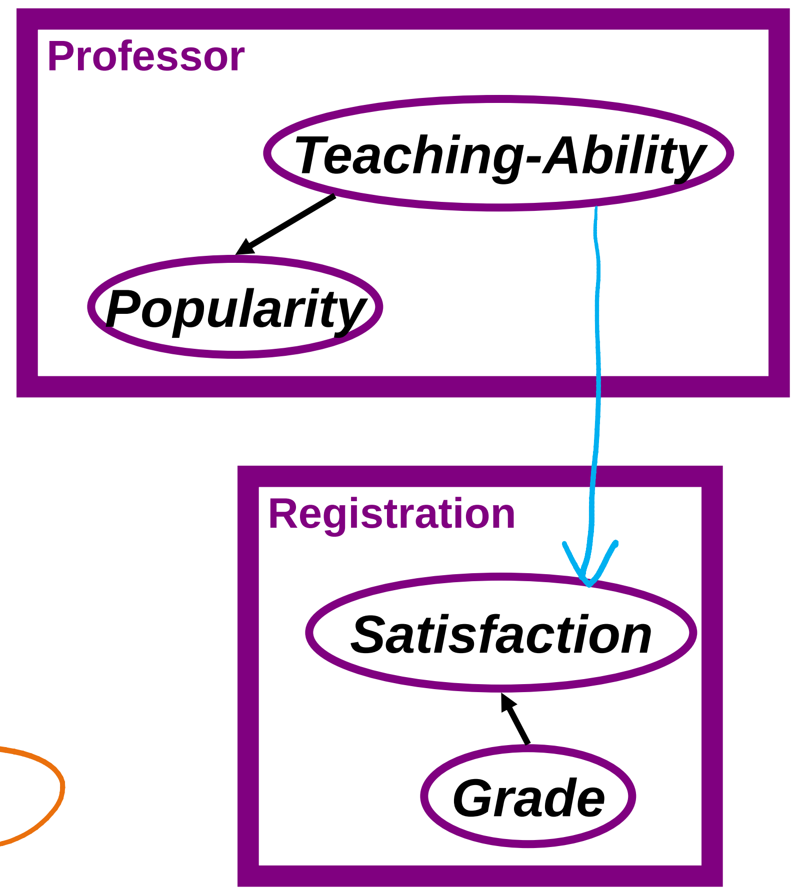
The prob. attribute *X.A* can depend on another probabilistic attribute *B* of *X*. This induces a corresponding dependency for individual objects



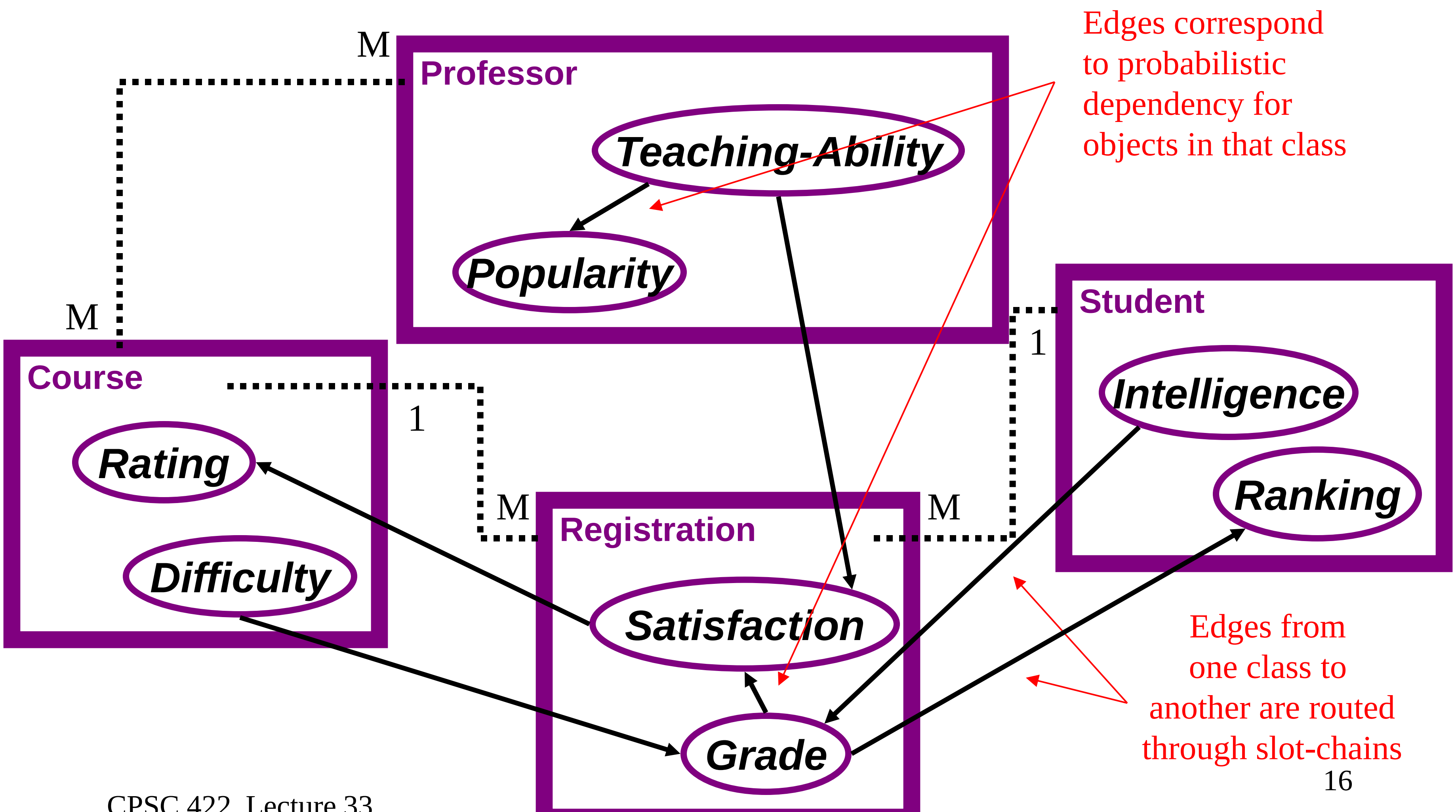
Dependencies across classes

- The attribute *X.A* can also depend on attributes of related objects *X.τ.B*, where *τ* is a slot chain

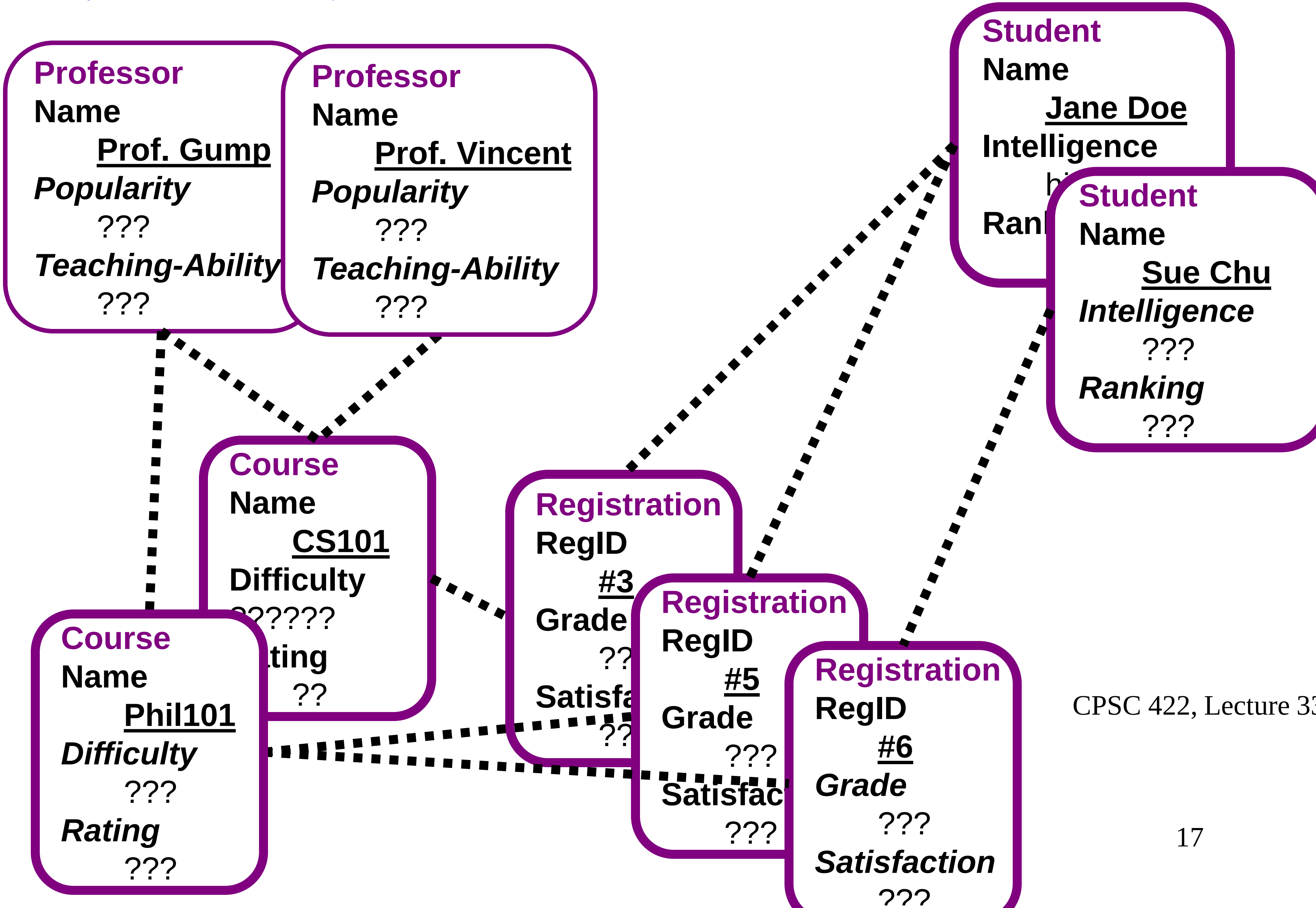


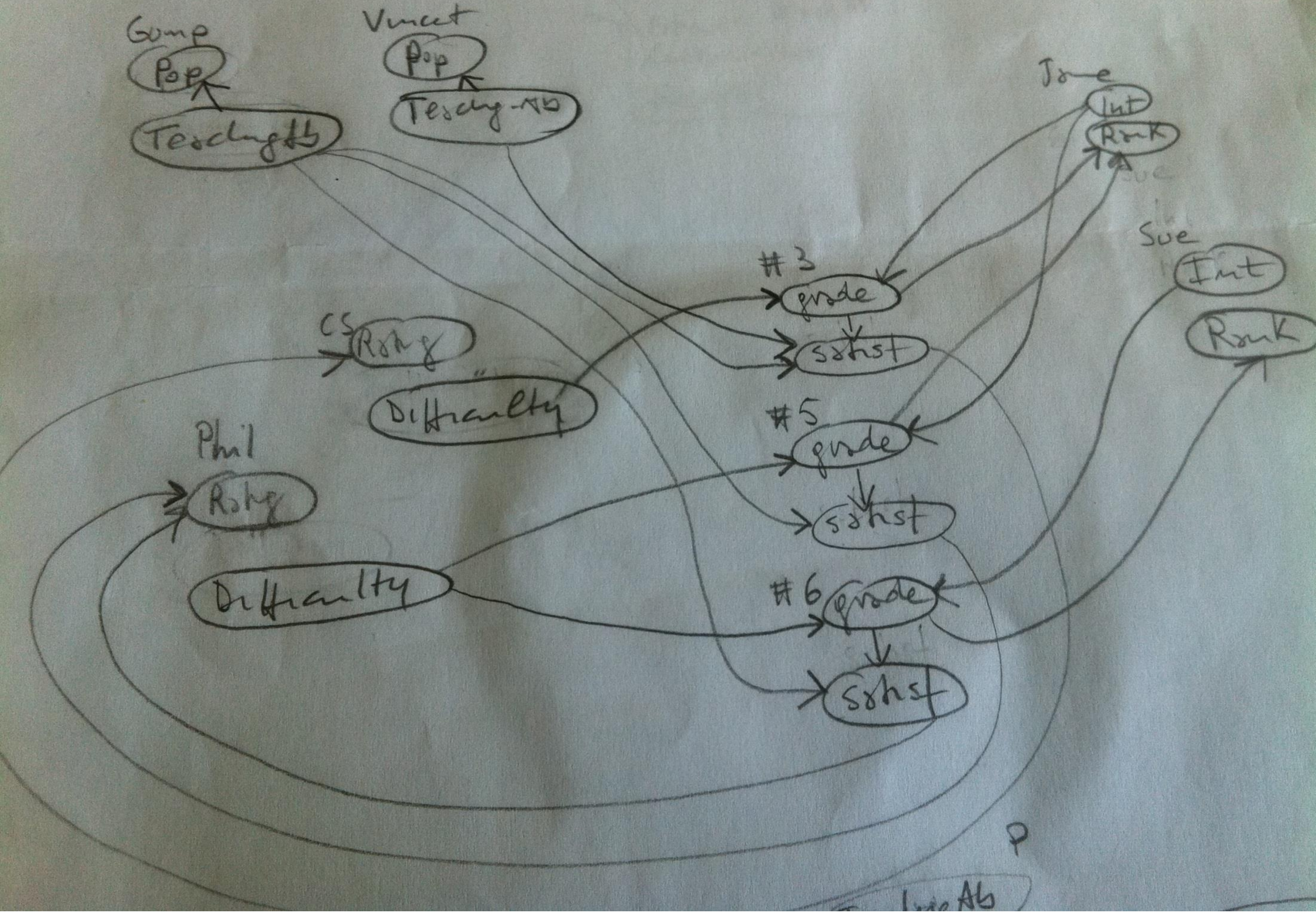


Possible PRM Dependency Structure for the University Domain



Let's derive the Corresponding "grounded" Dependency Structure for this Skeleton



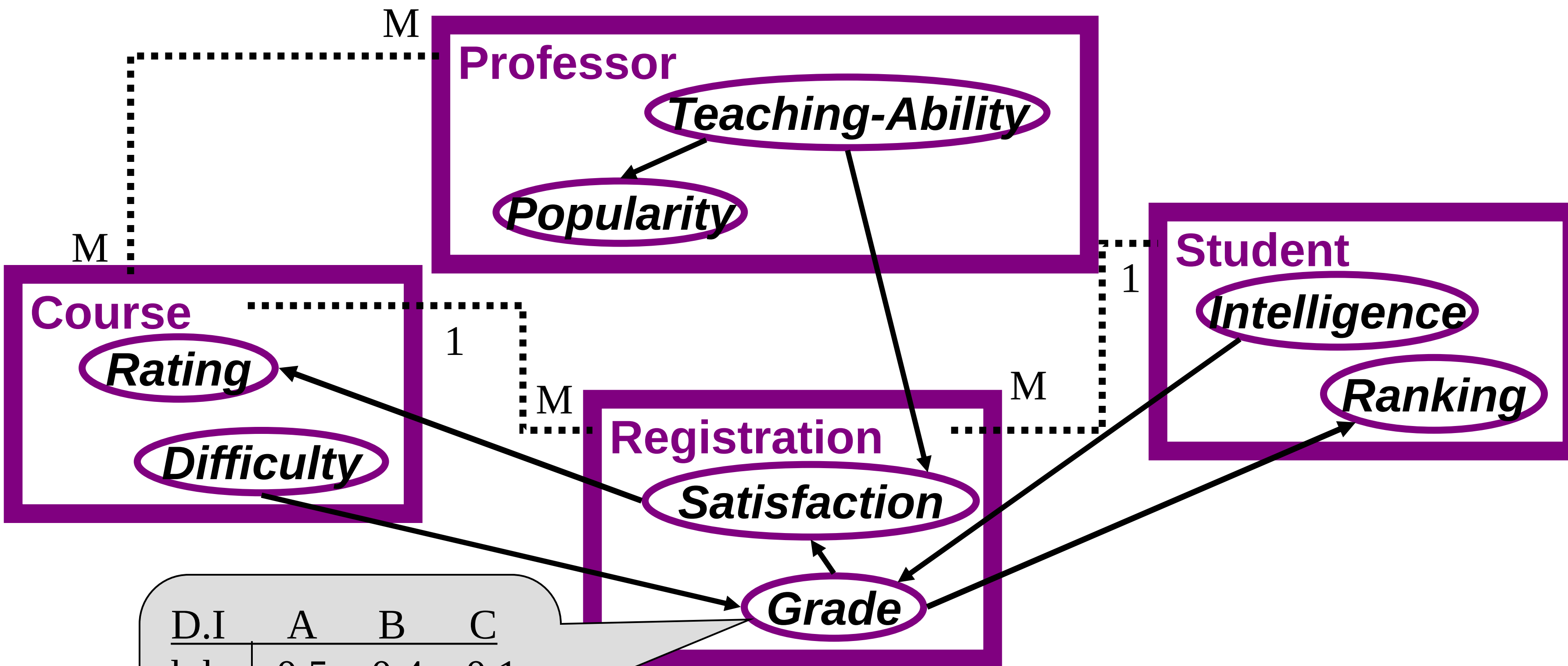


Gump
Pop
Vincent
Pop
Jane
Int
Rank
Sue
Int
Rank
#3
grade
CS
Rating
Difficulty
#5
grade
Phil
Rating
Difficulty
#6
grade
P

Parameters of PRMs

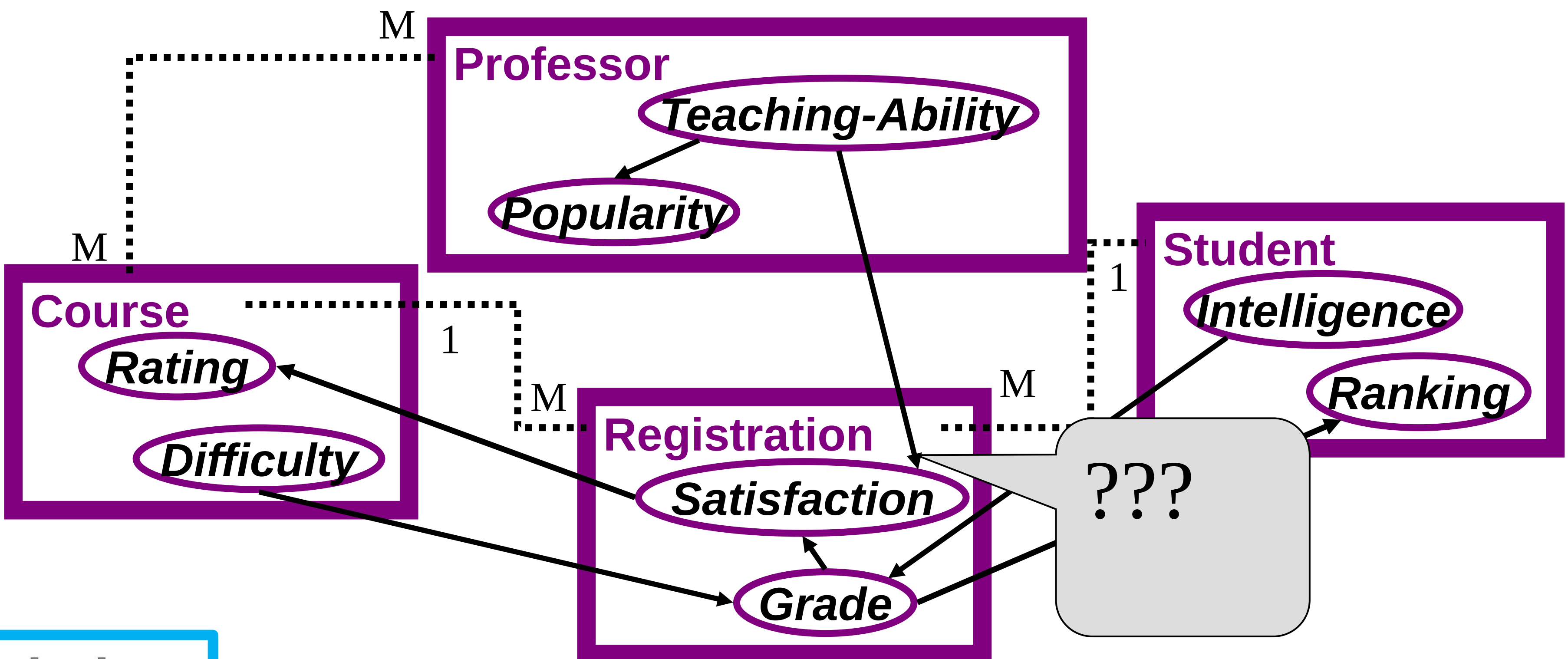
- **A PRM contains a *conditional probability distribution* (CPD) *P*(*X.A*|Pa(*X.A*)) for each attribute *X.A* of each class**
- **More precisely, let U be the set of parents of X.A. For each tuple of values u ∈ *V*(U), the CPD specifies a distribution *P*(*X.A*|u) over *V*(*X.A*). The parameters in all of these CPDs comprise θ_S**

Now, what are the parameters θ_S



D.I	A	B	C
h,h	0.5	0.4	0.1
h,l	0.1	0.5	0.4
l,h	0.8	0.1	0.1
l,l	0.3	0.6	0.1

Problem with some parameters θ_S



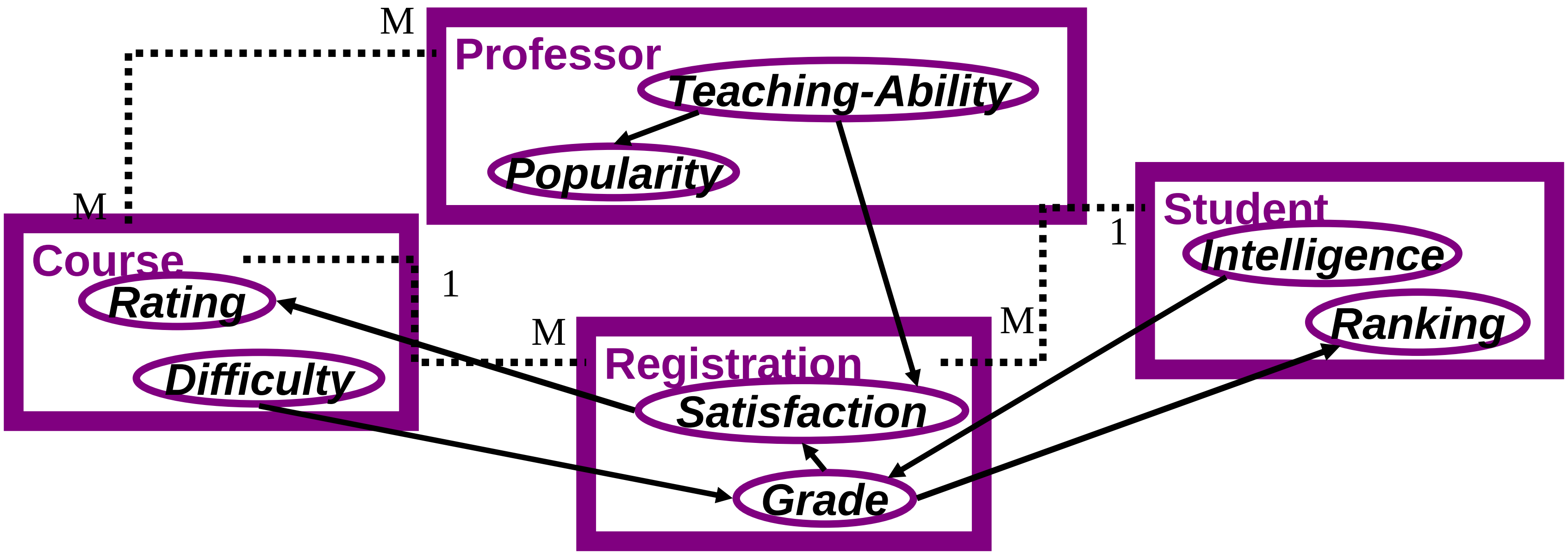
A. too many parents

B. variable # of parents

C. Too few parents

D. another problem

Problem with some parameters θ_S



When the slot chain τ (e.g. Course. Instructor) is not guaranteed to be single-valued, we must specify the probabilistic dependence of

- $x.A$ Registration. Satisfaction
- on the set $\{y.B: y \in x.\tau\}$

The Teaching-Ability of all the profs who are instructors of the Course

How to specify cond. Prob. When # of parents can vary?

- The notion of *aggregation* from database theory gives us the tool to address this issue; i.e., *x.a* will depend probabilistically on some aggregate property of this set

Aggregation in PRMs

Examples of aggregation are:

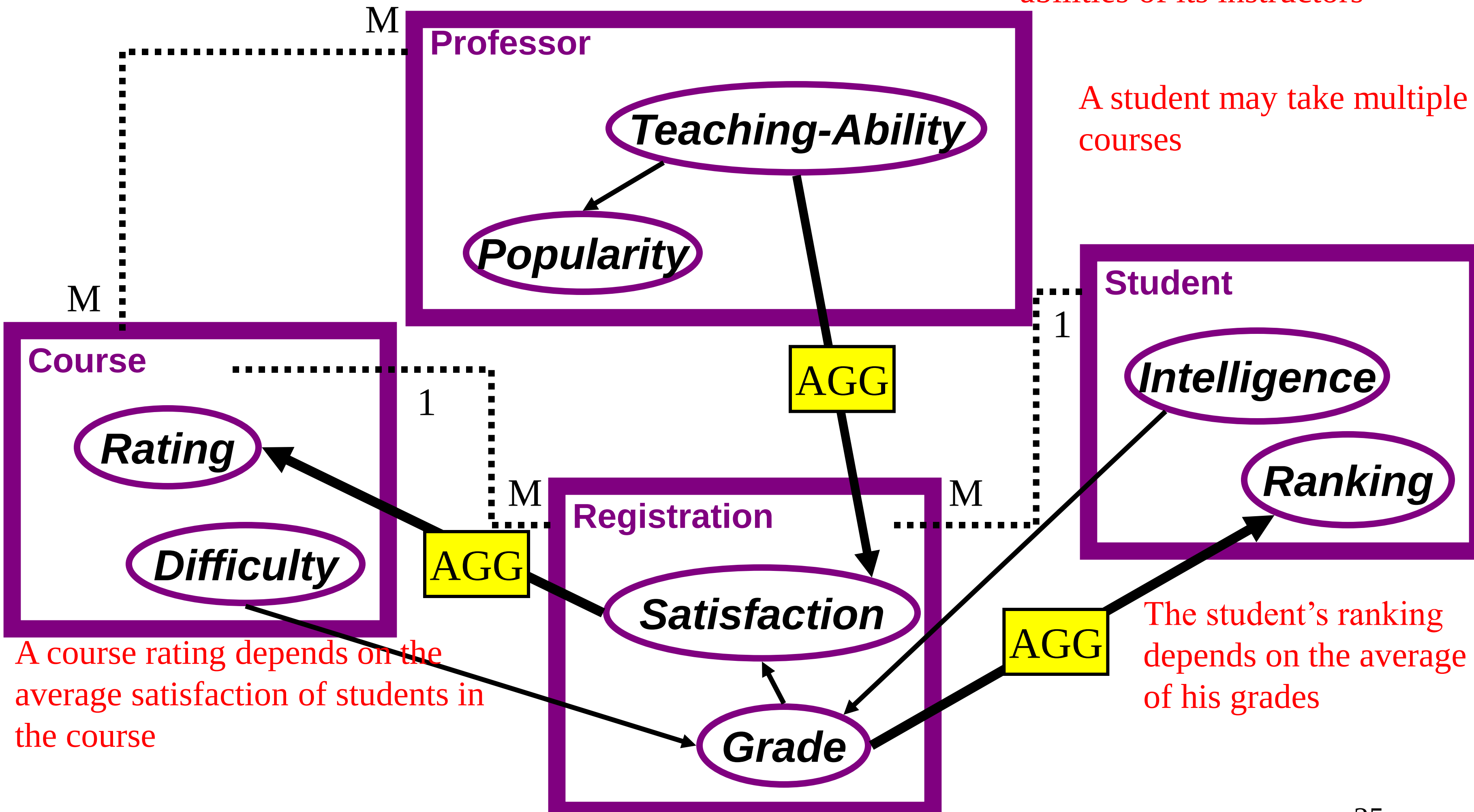
- the **mode** of the set (most frequently occurring value);
- **mean** value of the set (if values are numerical);
- **median**, **maximum**, or **minimum** (if values are ordered);
- **cardinality** of the set; etc.

PRM Dependency Structure with aggregations

The same course can be taught by multiple profs

A course satisfaction depends on the teaching abilities of its instructors

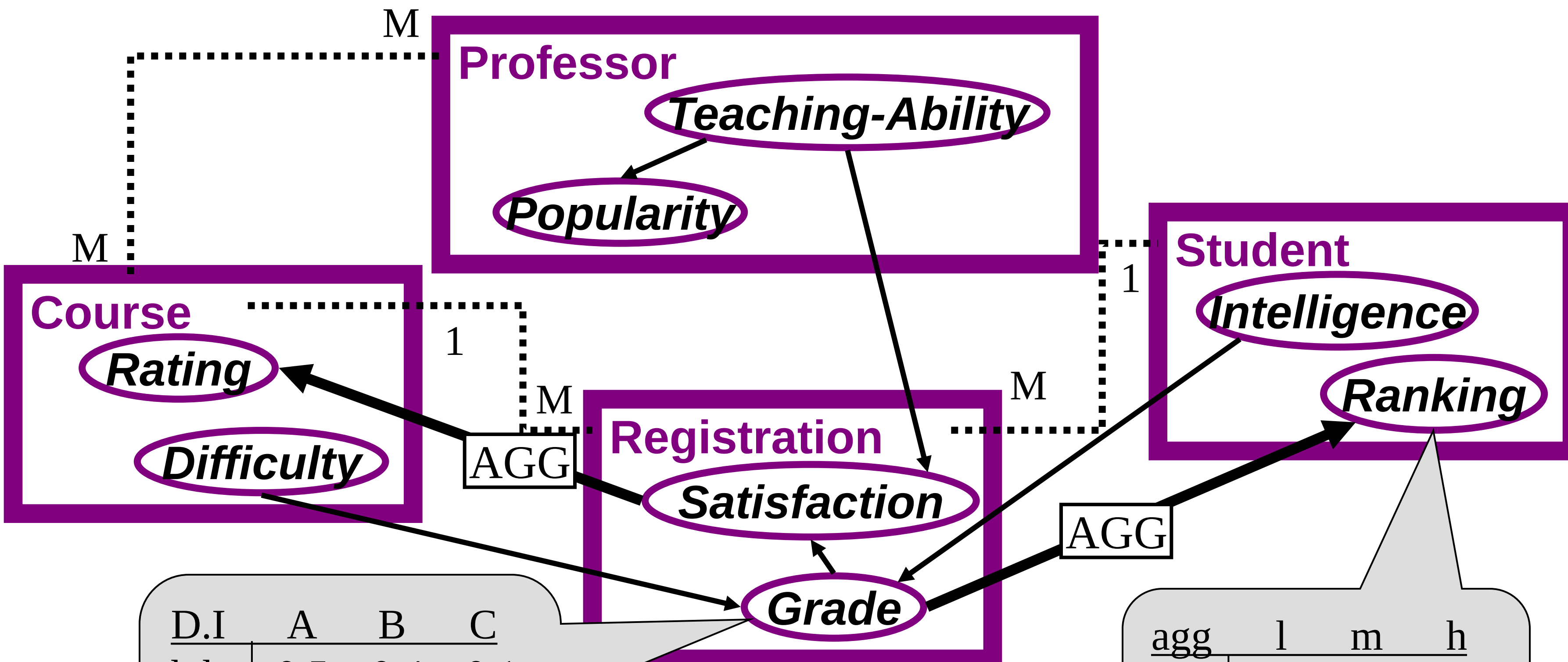
A student may take multiple courses



The student's ranking depends on the average of his grades

A course rating depends on the average satisfaction of students in the course

CPDs in PRMs



D.I	A	B	C
h,h	0.5	0.4	0.1
h,l	0.1	0.5	0.4
l,h	0.8	0.1	0.1
l,l	0.3	0.6	0.1

agg	l	m	h
A	0.1	0.2	0.7
B	0.2	0.4	0.4
C	0.6	0.3	0.1

JPD in PRMs

- Given a **skeleton structure** σ for our schema, we can apply these **local conditional probabilities** to define a **JPD** (joint probability distribution) **over all completions** of the skeleton

- Note that the objects and relations between objects in a skeleton are always specified by σ, hence **we are disallowing uncertainty over the relational structure of the model**

Parameter Sharing / CPTs reuse, where else?

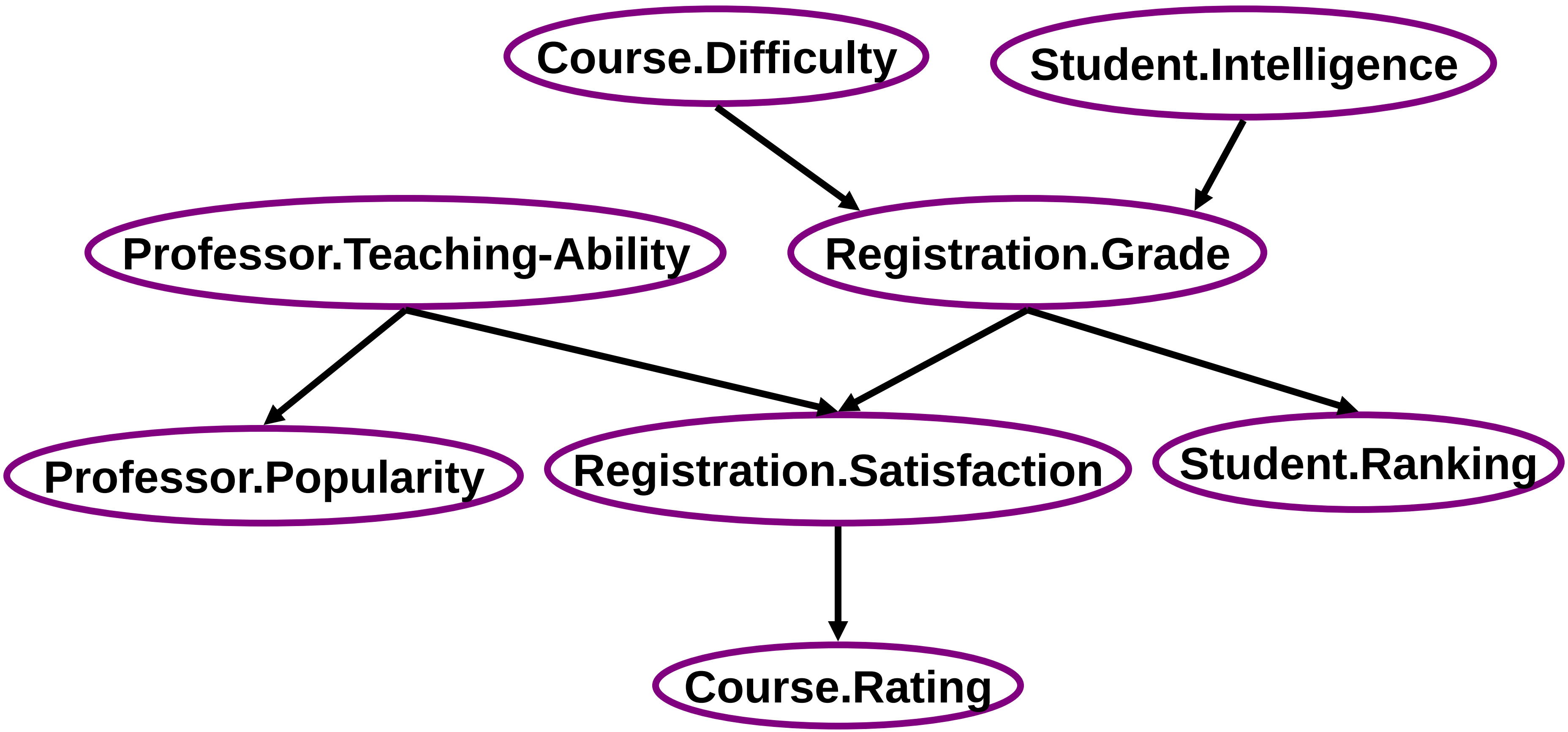
- **Temporal Models**
- **Because of the stationary assumption!**

Final Issue….

- To define a coherent probabilistic model, we **must ensure that our probabilistic dependencies are…..**

- Acyclic!

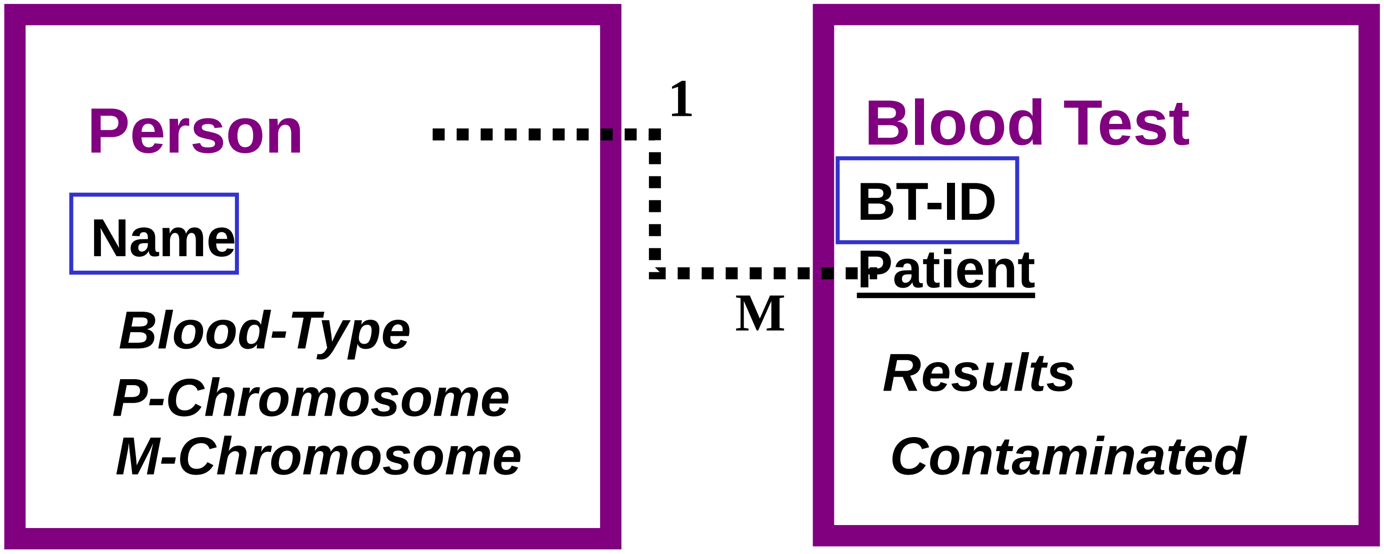
Class Dependency Graph for the University Domain



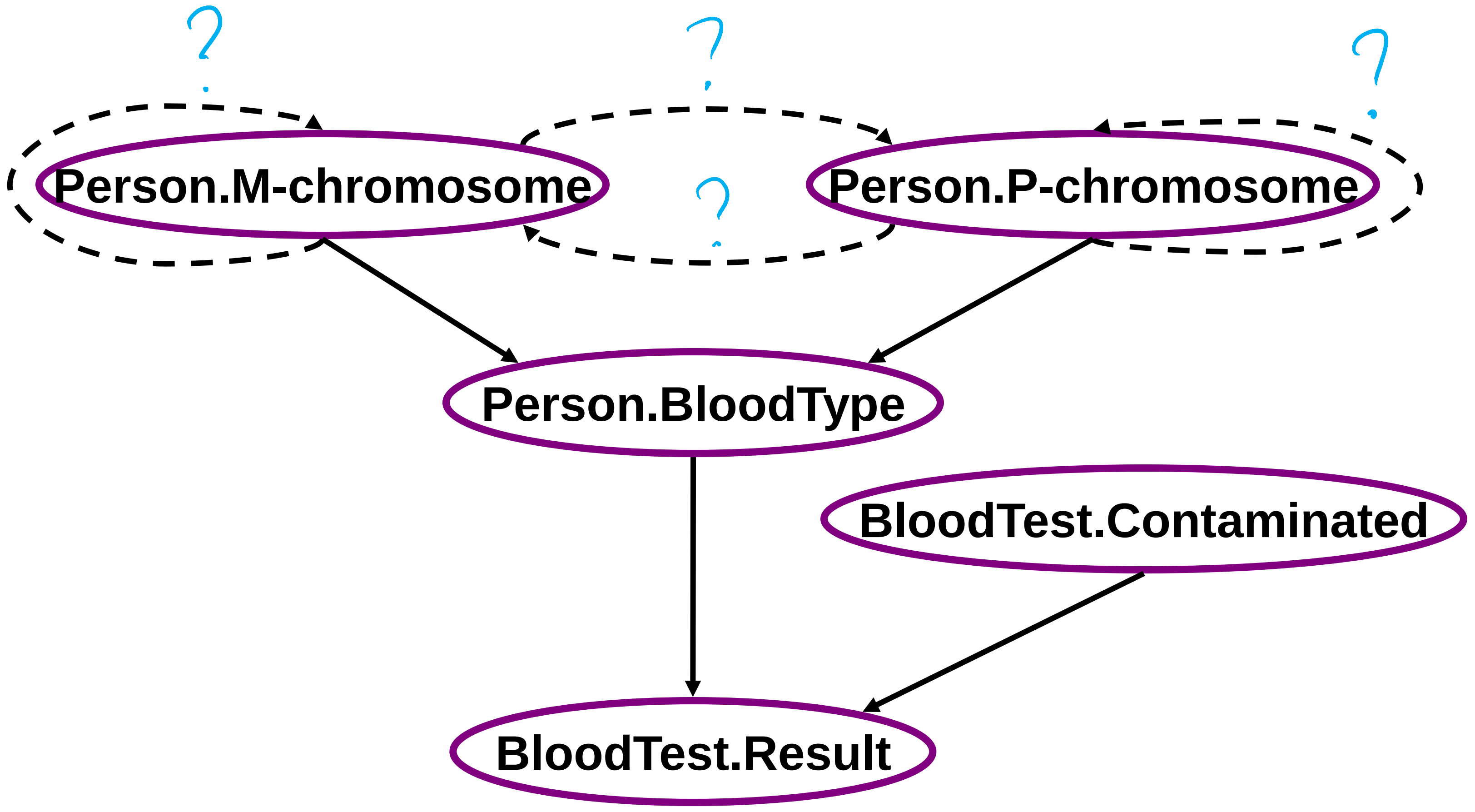
Ensuring Acyclic Dependencies

- In general, however, a cycle in the class dependency graph does not imply that all skeletons induce cyclic dependencies
- A model **may appear to be cyclic at the class level**, however, this cyclicity is always resolved at the level of individual objects
- The ability to guarantee that the cyclicity is resolved relies on **some prior knowledge about the domain**. The user can specify that certain slots are ***guaranteed acyclic***

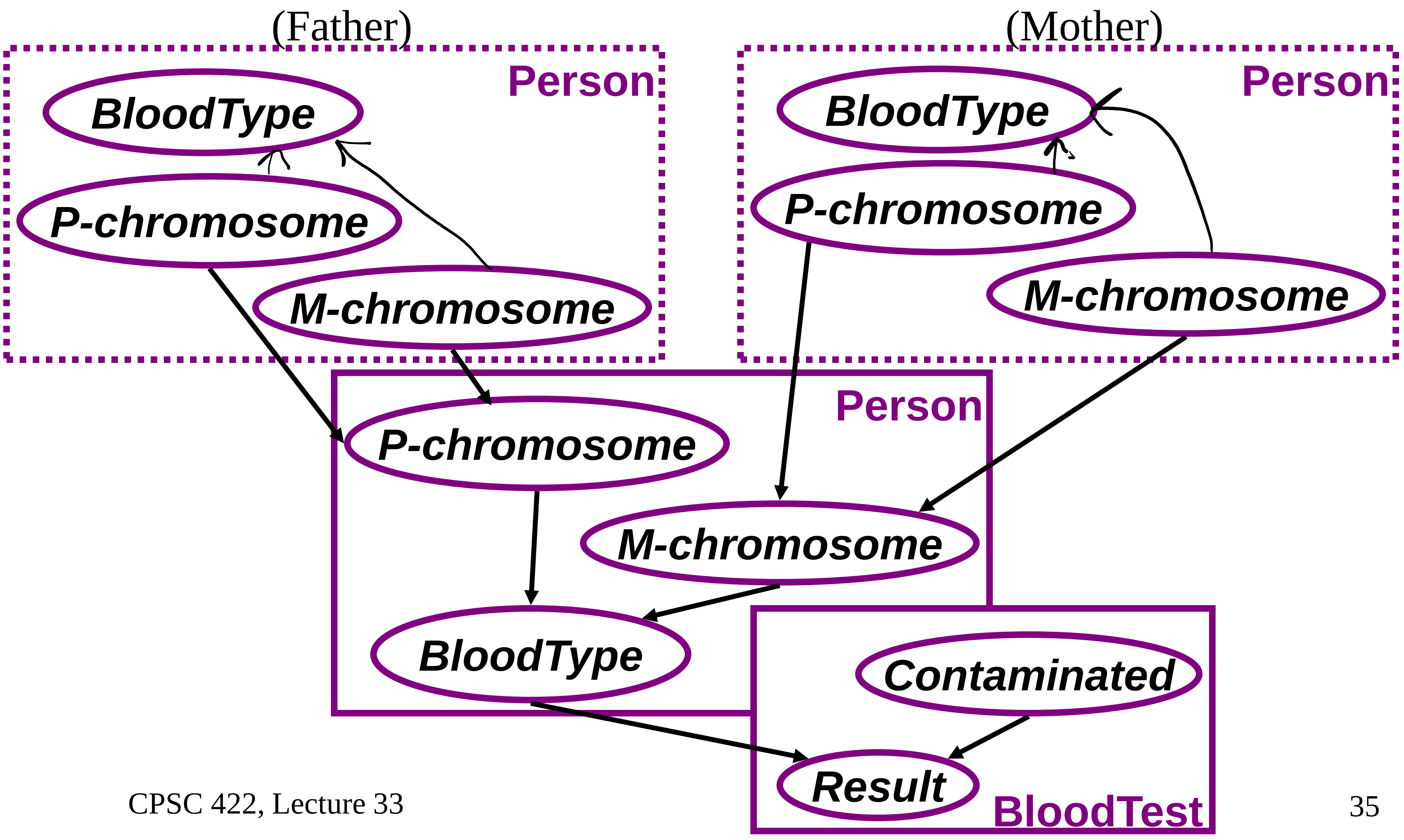
Relational Schema for the Genetics Domain



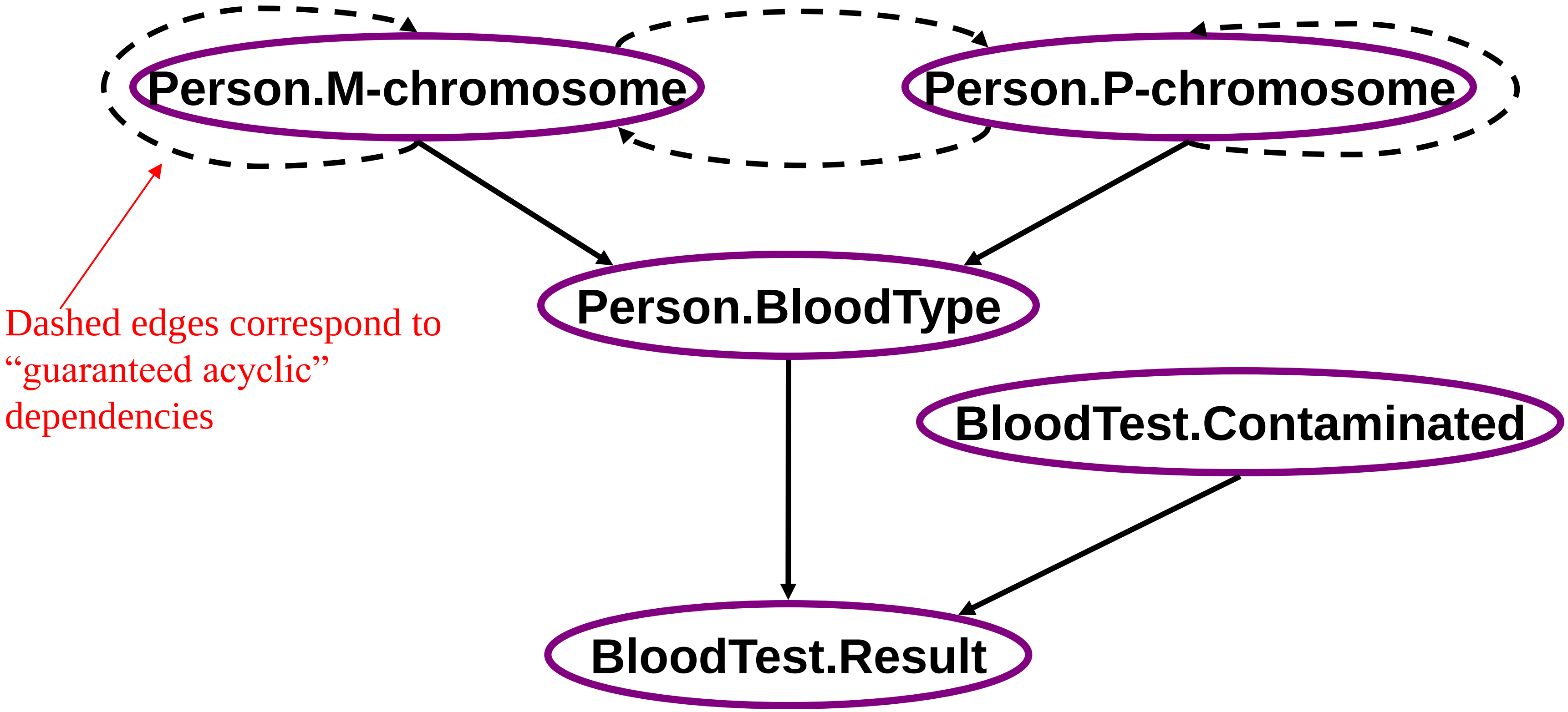
Dependency Graph for Genetics Domain



PRM for the Genetics Domain



Dependency Graph for Genetics Domain



Learning Goals for today's class

You can:

- **Build the grounded Bnet, given a Relational Skeleton, a dependency structure, and the corresponding parameters**
- **Define and apply guaranteed acyclicity**

422 big picture: Where are we?

StarAI (statistical relational AI)

Hybrid: Det +Sto

Prob CFG

Prob Relational Models

Markov Logics

	Deterministic	Stochastic
Query	*Logics* *First Order Logics* *Ontologies* • Full Resolution • SAT	*Belief Nets* Approx. : Gibbs *Markov Chains and HMMs* Forward, Viterbi…. Approx. : Particle Filtering *Undirected Graphical Models* *Markov Networks* *Conditional Random Fields*
Planning		*Markov Decision Processes and Partially Observable MDP* • Value Iteration • Approx. Inference *Reinforcement Learning*

Applications of AI

Representation

Reasoning Technique

Last class on Fri

- Beyond 322/422 (ML + grad courses)
- Watson….
- Final Exam

Fill out on-line Teaching Evaluation