Intelligent Systems (AI-2)
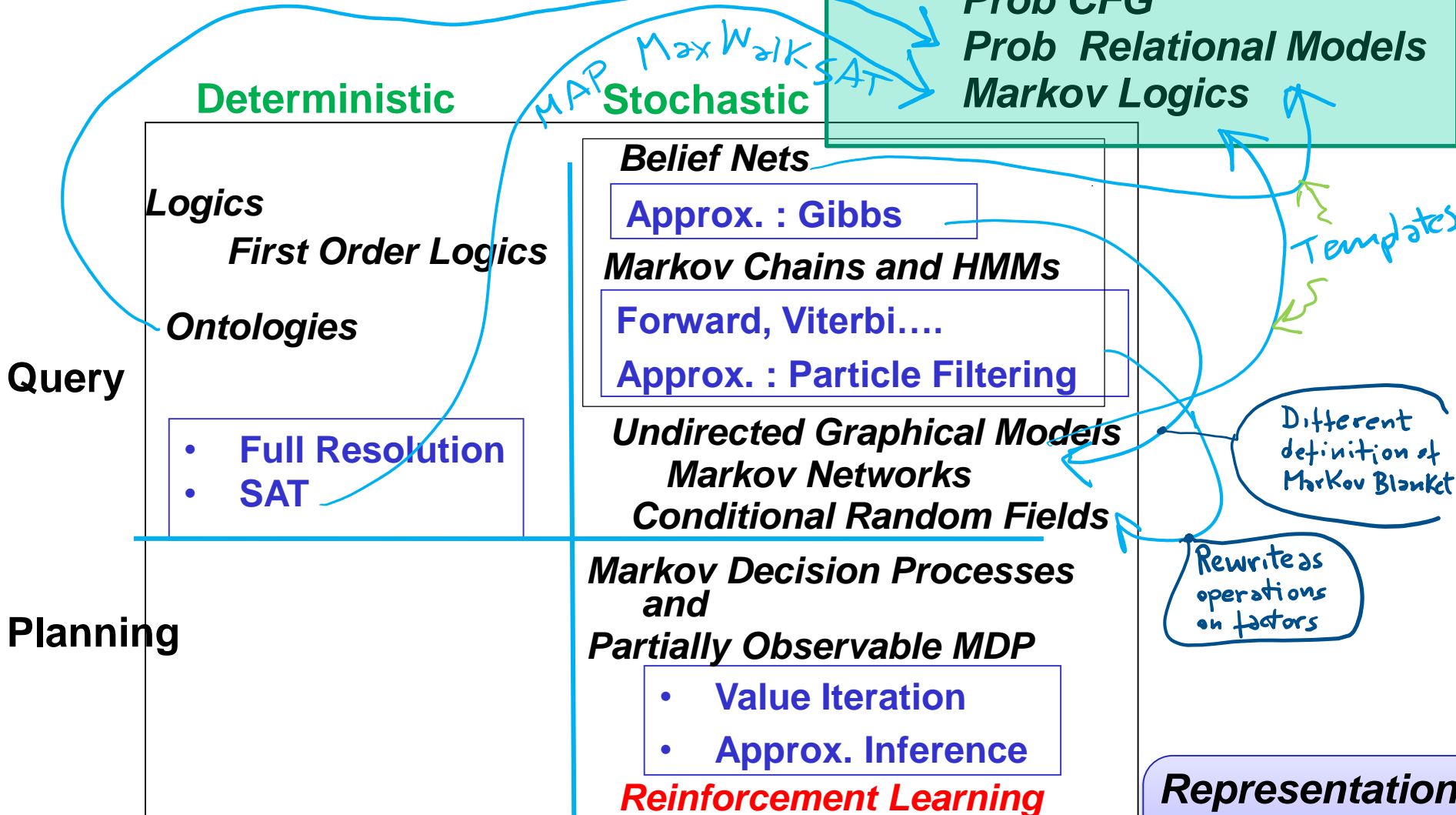
Computer Science cpsc422, Lecture 33

Apr, 27, 2021

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

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*

MAP Max Walk SAT

Templates

Different definition of Markov Blanket

*Rewrite as operations on factors*

Applications of AI

Representation

Reasoning Technique

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**

Registration

RegID
Course
Student
Grade
Satisfaction

Registration

Satisfaction

Grade

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**
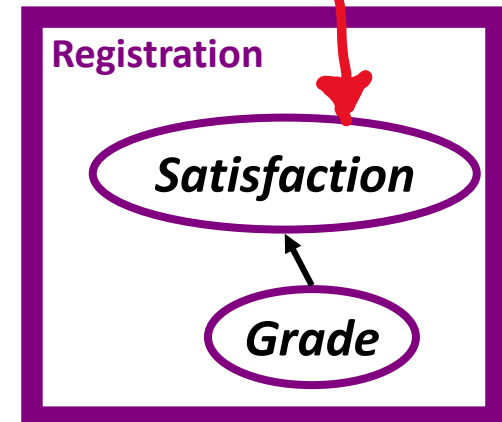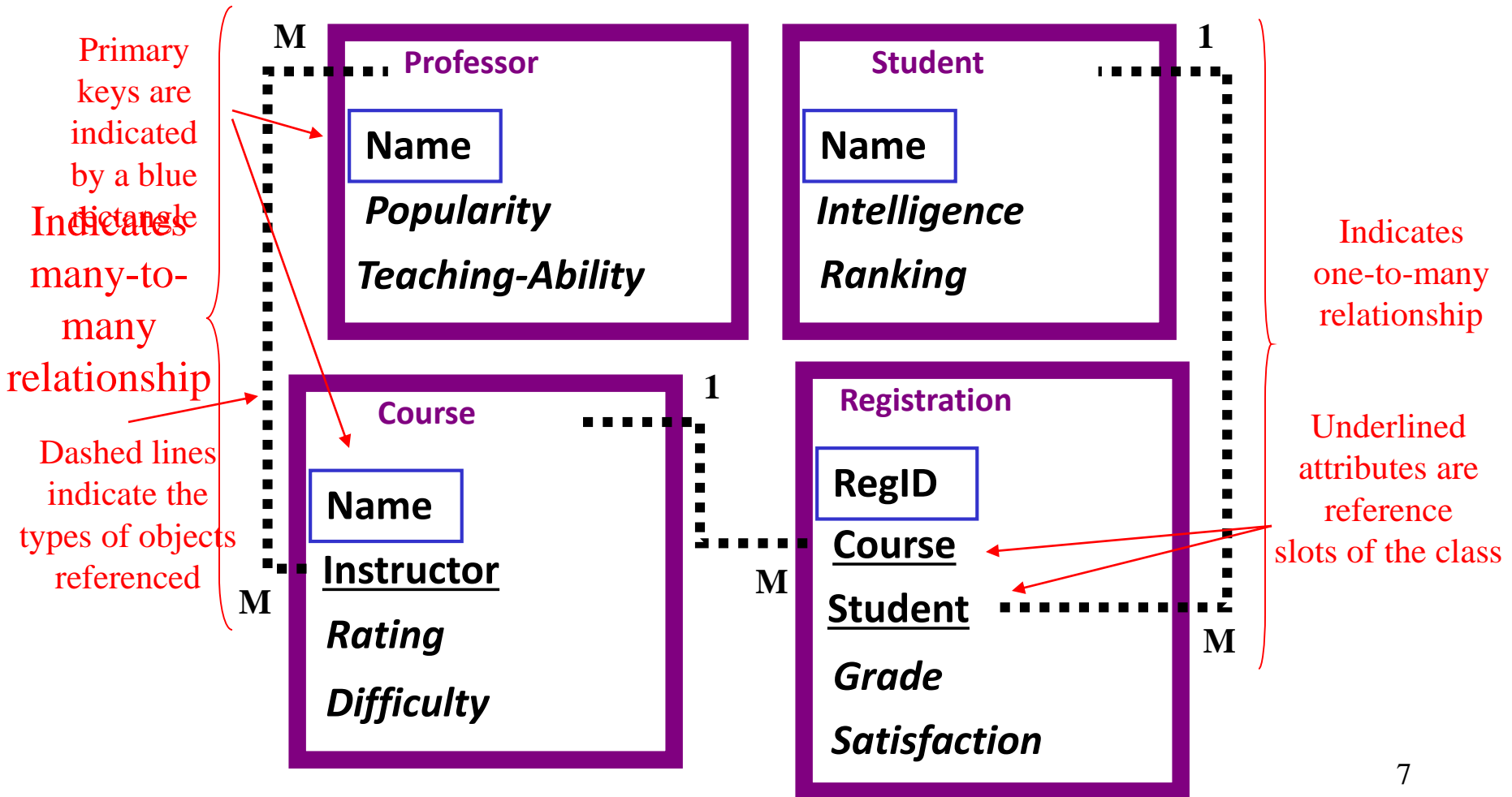
Professor

Teaching-Ability

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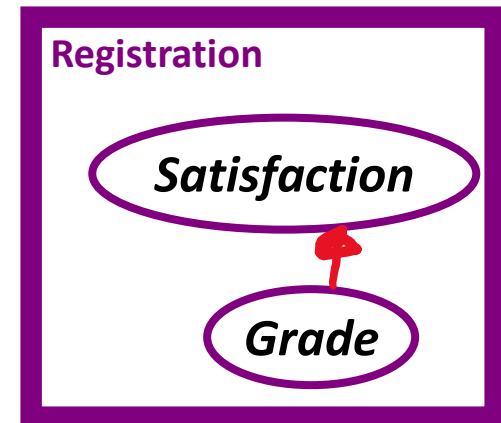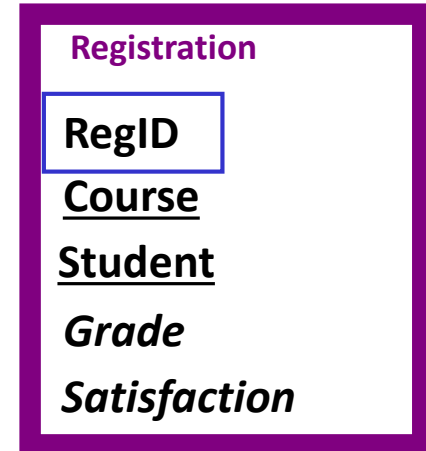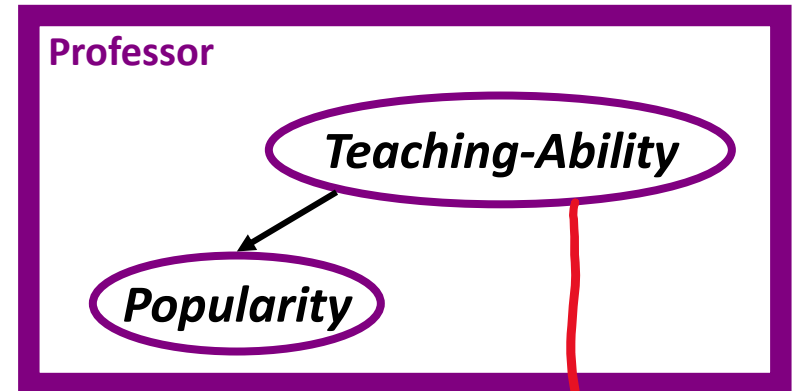
2. **PRMs also allow** properties of an entity to depend probabilistically on properties of other related entities
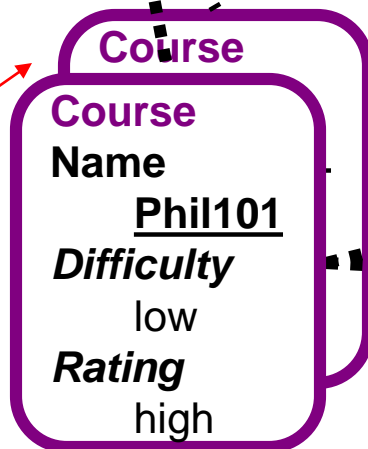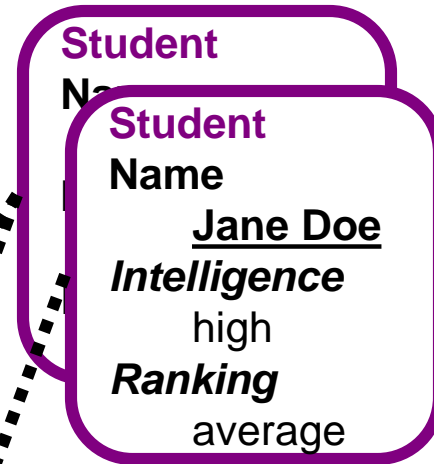
Registration

Satisfaction

Grade

5

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_ $X_1,...,X_n$ and a set of _relations_ $R_1,...,R_m$, where each relation R_i is typed**

University Domain Example – Full Relational Schema

M

Professor

Name

Popularity

Teaching-Ability

Student

Name

Intelligence

Ranking

1

Course

Name

Instructor

Rating

Difficulty

1

M

Registration

RegID

Course

Student

Grade

Satisfaction

M

M

Primary keys are indicated by a blue rectangle

Indicates many-to-many relationship

Dashed lines indicate the types of objects referenced

Indicates one-to-many relationship

Underlined attributes are reference slots of the class

7

University Domain Example – An Instance of the Schema

Professor
Name
> **Prof. Gump**
Popularity
> high
Teaching-Ability
> medium

Student
Na

Student
Name
> **Jane Doe**
Intelligence
> high
Ranking
> average

One professor is the instructor for both courses

Course

Course
Name
> **Phil101**
Difficulty
> low
Rating
> high

Registration

Registration

Registration
RegID
> **#5639**
Grade
> A
Satisfaction
> 3

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

8

University Domain Example – Another Instance of the Schema

Professor
Name
  Prof. Vincent
Popularity
  high
Teaching-Ability
  high
...ty
  medium

Course
Name
  Phil201
Difficulty
  low
Rating
  high
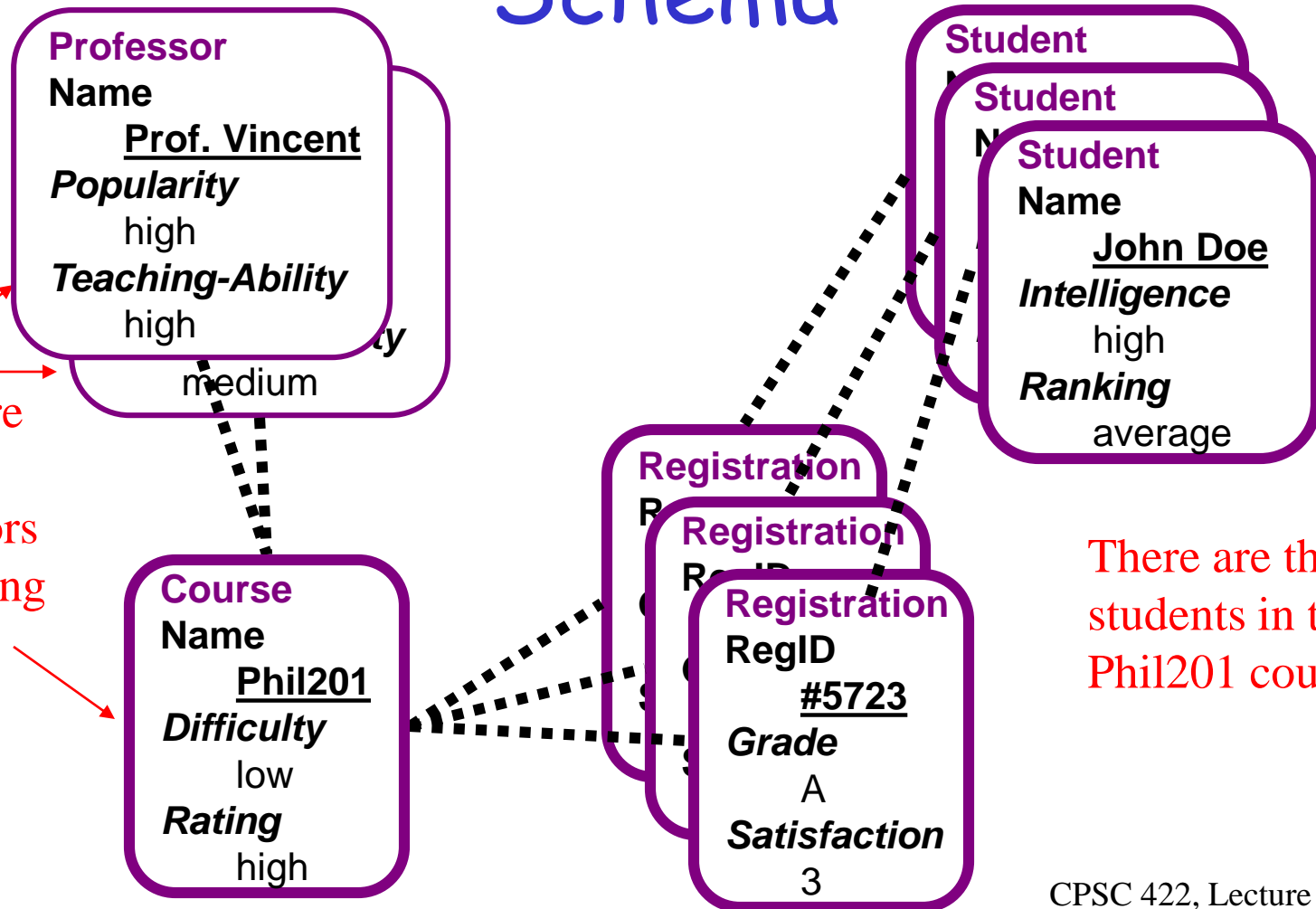
There are two professors instructing a course

Student
Student
Student
Name
  John Doe
Intelligence
  high
Ranking
  average

Registration
Registration
Registration
RegID
  #5723
Grade
  A
Satisfaction
  3

There are three students in the Phil201 course

University Domain Example – fixed vs. probabilistic attributes

M

Professor

Name

Popularity

Teaching-Ability

Student

Name

Intelligence

Ranking

1

Course

Name

Instructor

Rating

Difficulty

1

M

Registration

RegID

Course

Student

Grade

Satisfaction

M

Probabilistic attributes are also shown in italic

Fixed attributes are shown in regular font

Probabilistic attributes are shown in italic
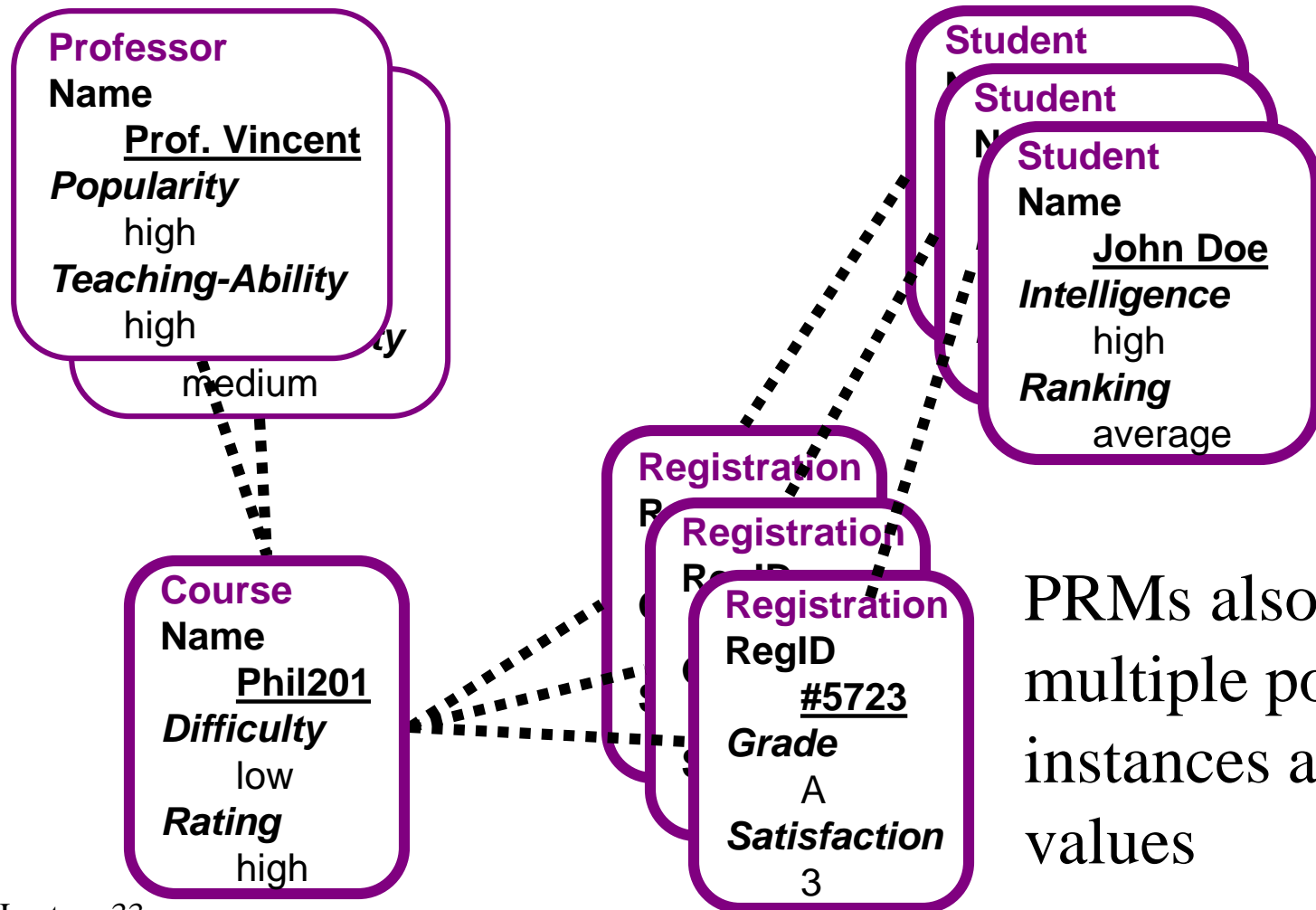
Fixed attributes are shown in regular font

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

Professor
Name
 Prof. Vincent
Popularity
 ???
Teaching-Ability
 ???
 ty
 ???

Student
Student
Student
Name
 John Doe
Intelligence
 ???
Ranking
 ???

Course
Name
 Phil201
Difficulty
 ???
Rating
 ???

Registration
Registration
Registration
RegID
 #5723
Grade
 ???
Satisfaction
 ???

12
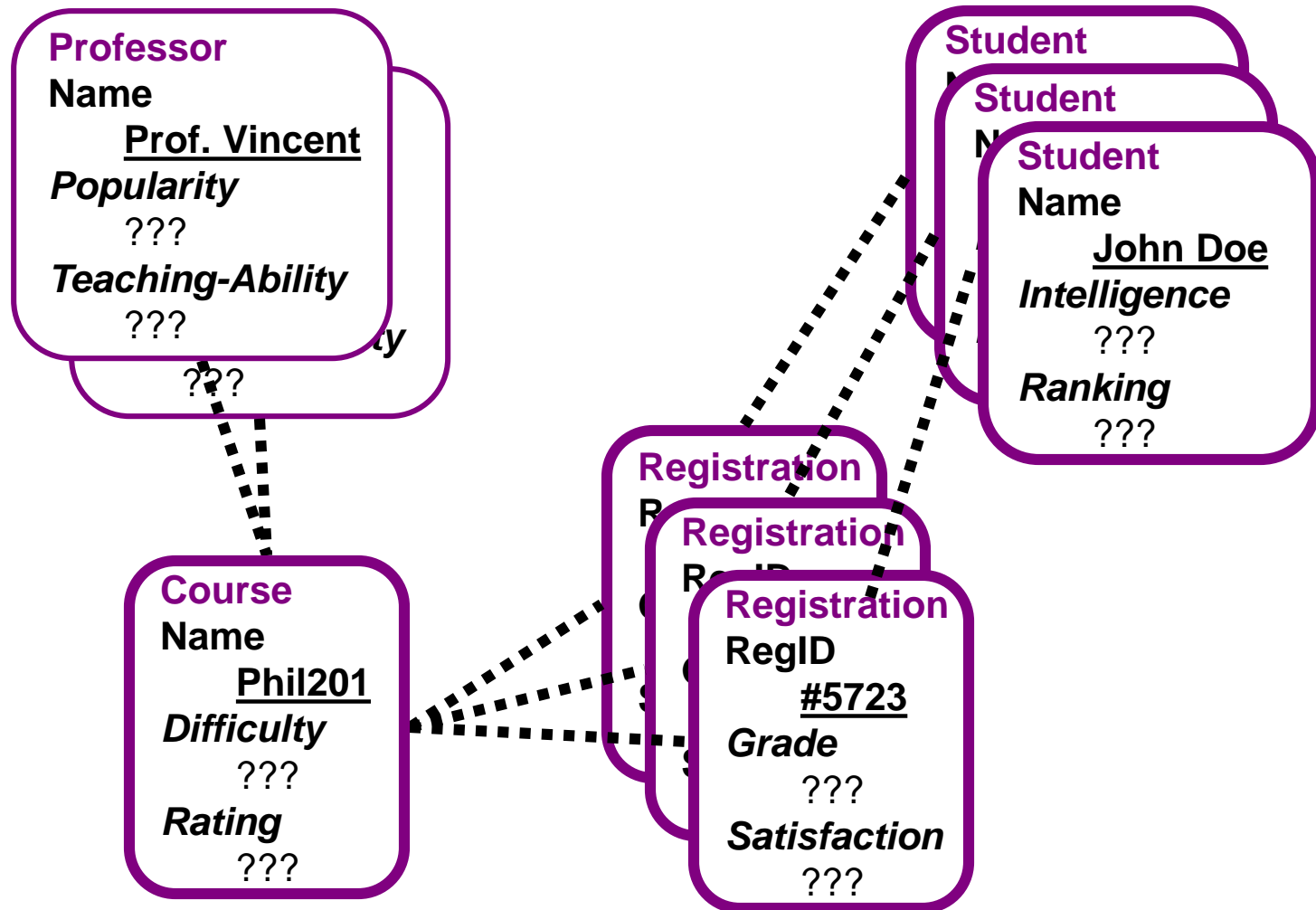
University Domain Example – The Completion Instance *I*

Professor
Name
 Prof. Vincent
Popularity
 high
Teaching-Ability
 high

 medium

Course
Name
 Phil201
Difficulty
 low
Rating
 high

Student

Student

Student
Name
 John Doe
Intelligence
 high
Ranking
 average

Registration

Registration

Registration
RegID
 #5723
Grade
 A
Satisfaction
 3

PRMs also allow multiple possible instances and values

13

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, $\mathcal{S}$**
  - **the parameters associated with it, $\theta_\mathcal{S}$**

- **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**
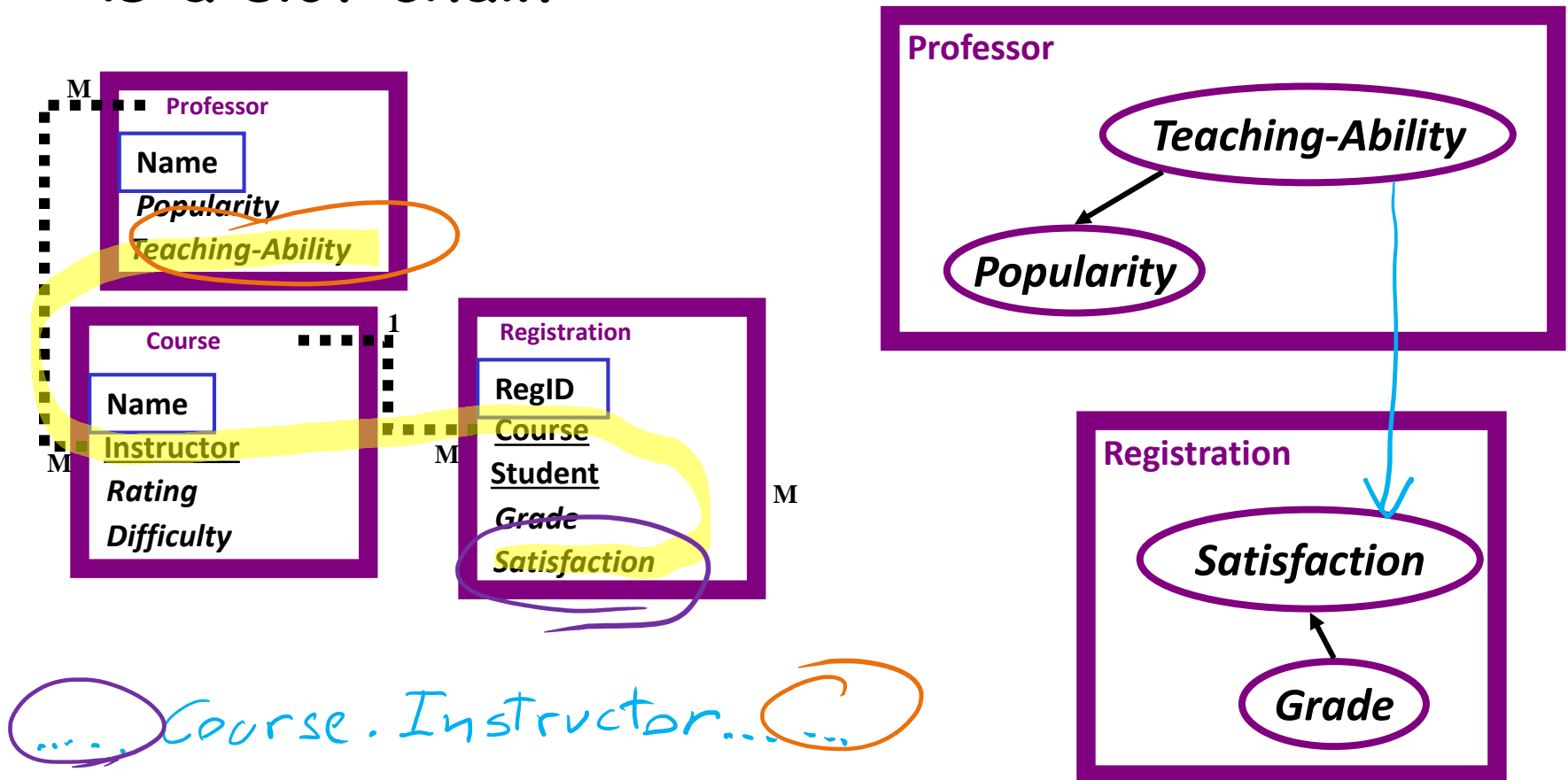
Dependencies within a class

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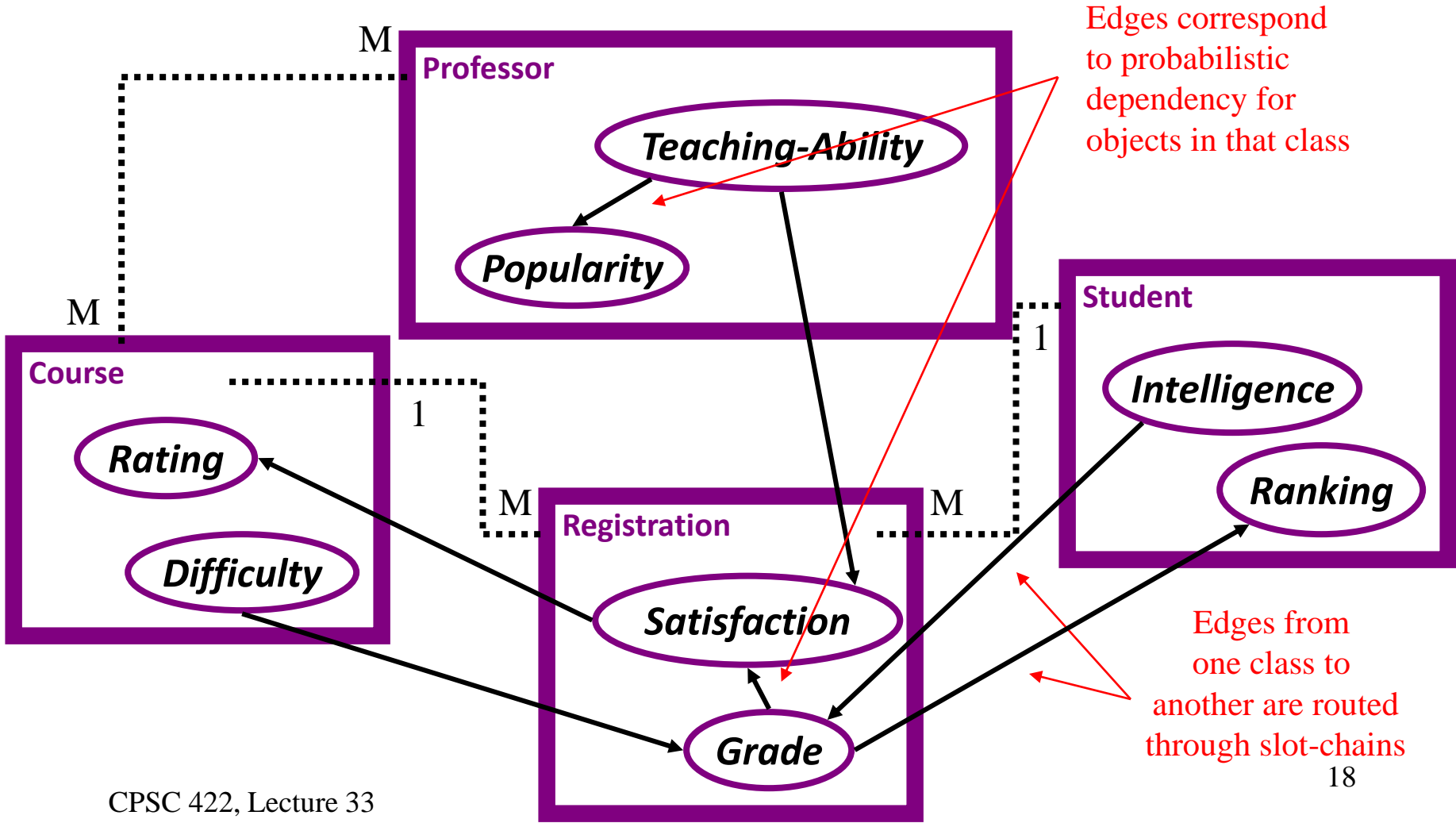
Registration

> **RegID**
> **Course**
> **Student**
> *Grade*
> *Satisfaction*

Registration
RegID
  #5723
Grade
  ....
Satisfaction
  .....

Registration

Satisfaction ← *Grade*

Registration(#5723). Satisfaction ← Registration(#5723). Grade
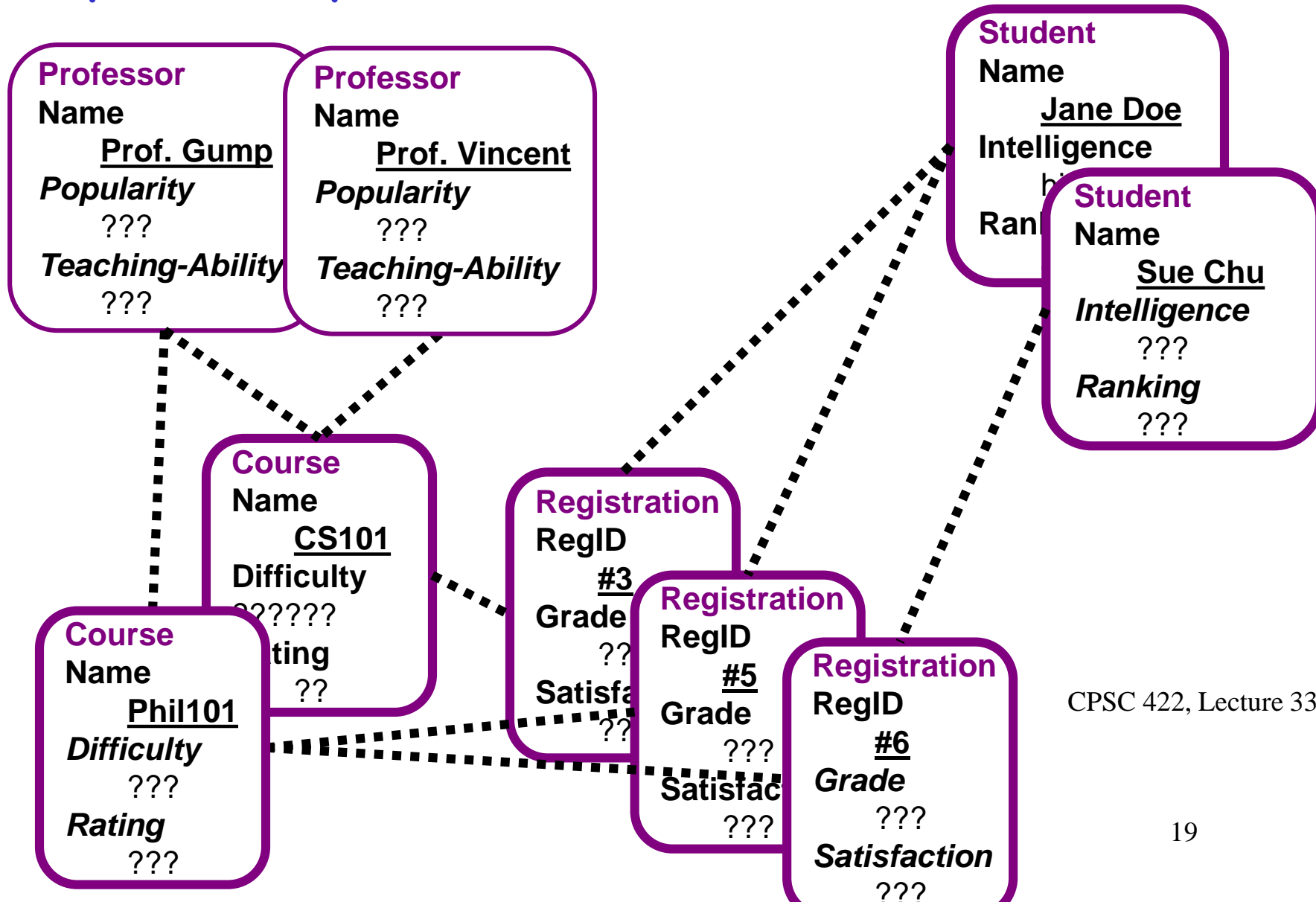
Dependencies across classes

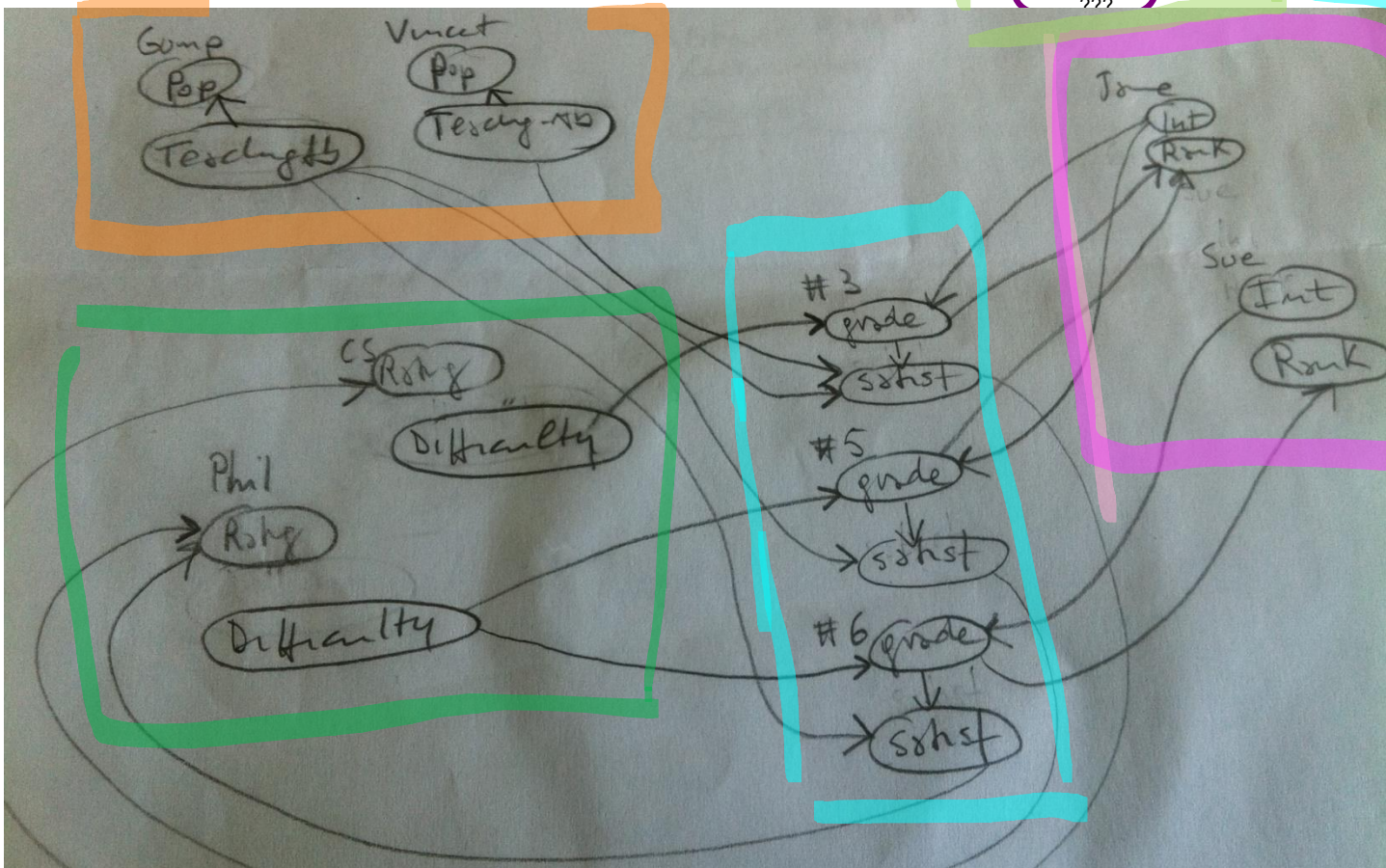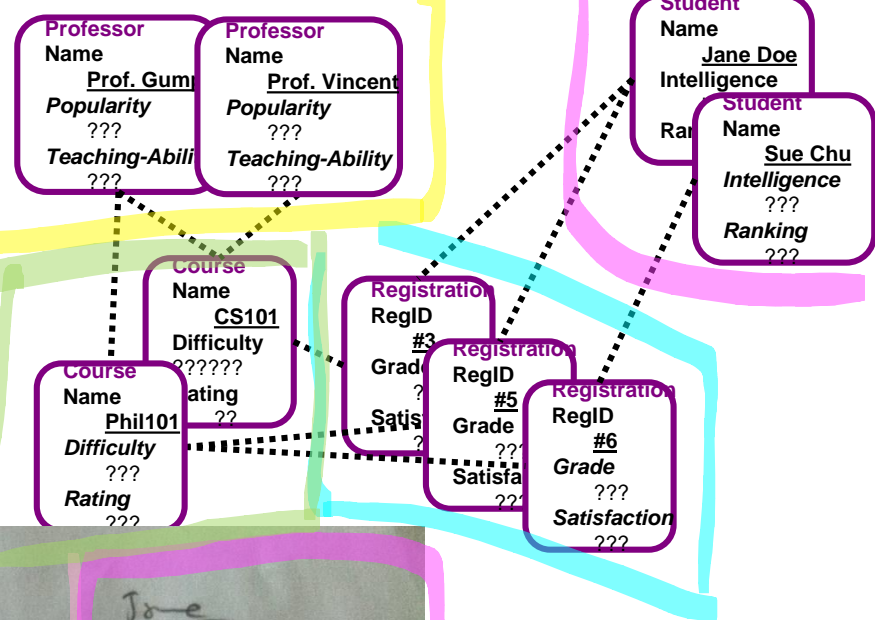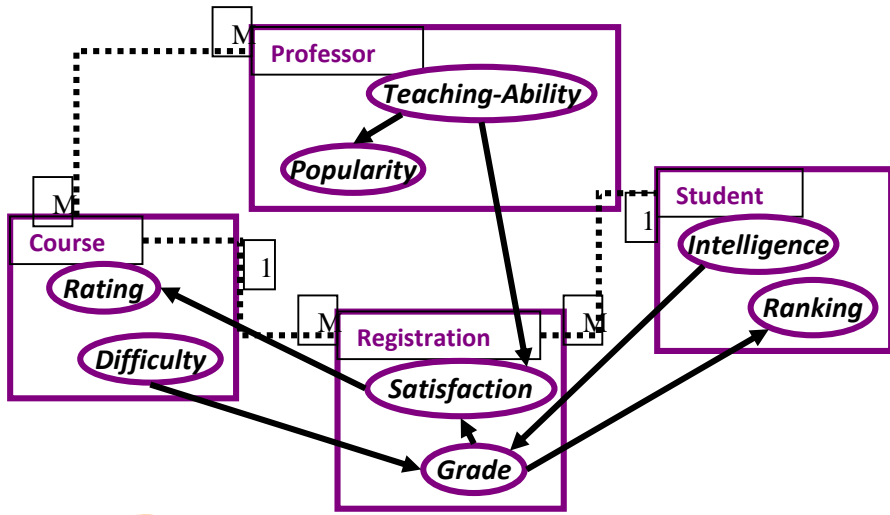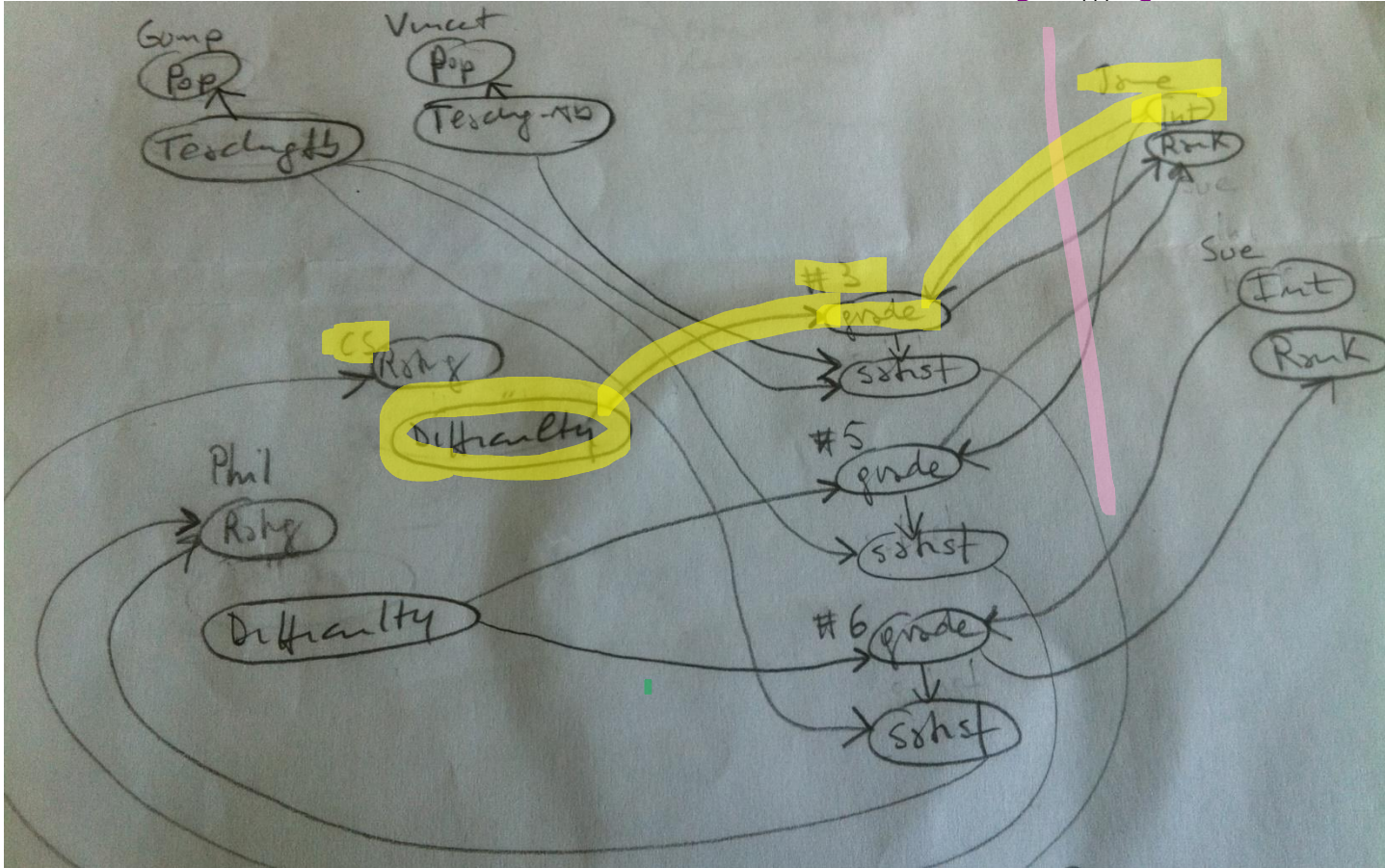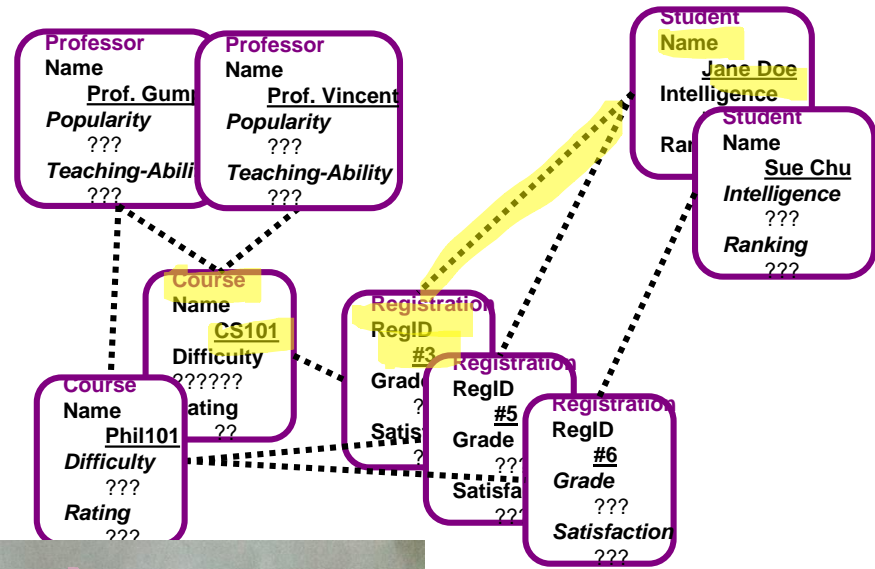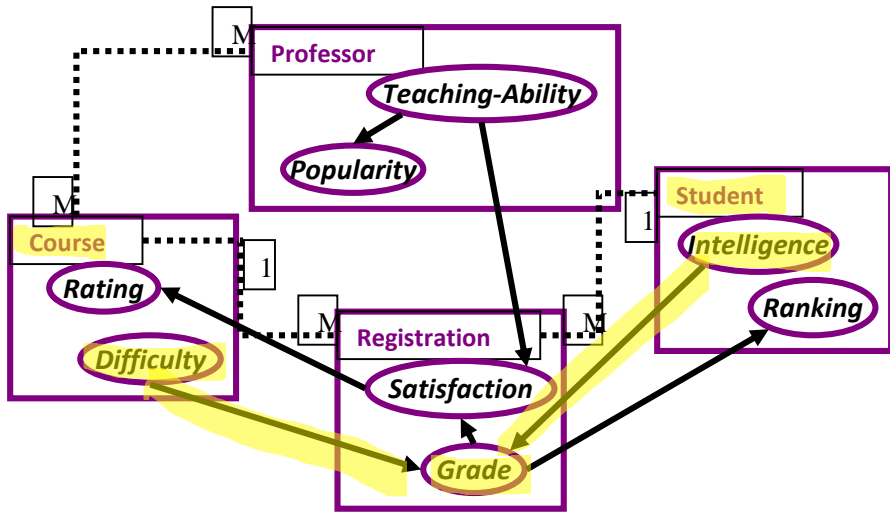- **The attribute *X.A* can also depend on attributes of related objects *X.τ.B*, where τ is a slot chain**

Possible PRM Dependency Structure for the University Domain

Edges correspond to probabilistic dependency for objects in that class

Edges from one class to another are routed through slot-chains

18

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

Professor
Name
   Prof. Gump
Popularity
   ???
Teaching-Ability
   ???

Professor
Name
   Prof. Vincent
Popularity
   ???
Teaching-Ability
   ???

Student
Name
   Jane Doe
Intelligence
   hi
Rank

Student
Name
   Sue Chu
Intelligence
   ???
Ranking
   ???

Course
Name
   CS101
Difficulty
   ??????
ting
   ??

Course
Name
   Phil101
Difficulty
   ???
Rating
   ???

Registration
RegID
   #3
Grade
   ??
Satisfa

Registration
RegID
   #5
Grade
   ???
Satisfac

Registration
RegID
   #6
Grade
   ???
Satisfaction
   ???

CPSC 422, Lecture 33

19

Professor M

Teaching-Ability

Popularity

Student

Course M

Intelligence

Rating

Ranking

Difficulty

Registration

Satisfaction

Grade

1

Professor
Name
Prof. Gump
Popularity
???
Teaching-Abili
???

Professor
Name
Prof. Vincent
Popularity
???
Teaching-Ability
???

Student
Name
Jane Doe
Intelligence
Ran

Student
Name
Sue Chu
Intelligence
???
Ranking
???

Course
Name
CS101
Difficulty
?????
ating
??

Course
Name
Phil101
Difficulty
???
Rating
???

Registration
RegID
#3
Grade
?
Satis

Registration
RegID
#5
Grade
???
Satisfa
???

Registration
RegID
#6
Grade
???
Satisfaction
???

Gump
Pop
Teachy-ab

Vincet
Pop
Teachy-Ab

Jane
Int
Rnk
Sue

Sue
Int
Rnk

CS
Rtng
Difficulty

Phil
Rtng
Difficulty

#3
grade
sshst

#5
grade
sshst

#6 grade
sshst

Professor
- *Teaching-Ability*
- *Popularity*

M

Course
- *Rating*
- *Difficulty*

M 1

Student
- *Intelligence*
- *Ranking*

Registration
- *Satisfaction*
- *Grade*

M M 1

Professor
Name
Prof. Gump
Popularity
???
Teaching-Ability
???

Professor
Name
Prof. Vincent
Popularity
???
Teaching-Ability
???

Student
Name
Jane Doe
Intelligence
Ran

Student
Name
Sue Chu
Intelligence
???
Ranking
???

Course
Name
CS101
Difficulty
?????
ating
??

Course
Name
Phil101
Difficulty
???
Rating
???

Registration
RegID
#3
Grade
?
Satis
?

Registration
RegID
#5
Grade
???
Satisfaction
???

Registration
RegID
#6
Grade
???
Satisfaction
???

Gump
Pop
Teachg.Ab

Vincet
Pop
Teachg-Ab

Jane
Int
Rank
Sue

Sue
Int
Rank

#3
grade
satst

CS
Rtng
Difficulty

#5
grade
sstst

Phil
Rotg
Difficulty

#6 grade
sstst

Professor
Teaching-Ability
Popularity

Student
Intelligence
Ranking

Course
Rating
Difficulty

Registration
Satisfaction
Grade

Professor	Professor
Name	Name
Prof. Gump	Prof. Vincent
Popularity	Popularity
???	???
Teaching-Abili	Teaching-Ability
???	???

Student
Name
Jane Doe
Intelligence
Ran

Student
Name
Sue Chu
Intelligence
???
Ranking
???

Course
Name
CS101
Difficulty
?????
ating
??

Course
Name
Phil101
Difficulty
???
Rating
???

Registration
RegID
#3
Grad
?
Satis
?

Registration
RegID
#5
Grade
???
Satisfa
?

Registration
RegID
#6
Grade
???
Satisfaction
???

Gump
Pop
Teaching

Vincet
Pop
Teachy-Ab

CS Rtg
Difficulty

Phil
Rtg
Difficulty

#3 grade
satsst

#5 grade
satsst

#6 grade
satst

Jane
Int
Rnk

Sue
Int
Rnk

Now, what are the parameters θ$_S$

Professor M

Teaching-Ability

Popularity

M

Course

Rating

Difficulty

M

1

1

Student

Intelligence

Ranking

M

Registration

Satisfaction

Grade

| 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

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 e.g.,**

P(Registration.Grade | Course.Difficulty, Student.Intelligence)

- **More precisely, let U be the set of parents of $X.A$. For each tuple of values $u \in V(U)$, the CPD specifies a distribution $P(X.A|u)$ over $V(X.A)$.**

Course.Difficulty = {high, low}
Student.Intelligence = {high, low}
Registration.Grade = {A, B, C}

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

The parameters in all of these CPDs comprise θ_S
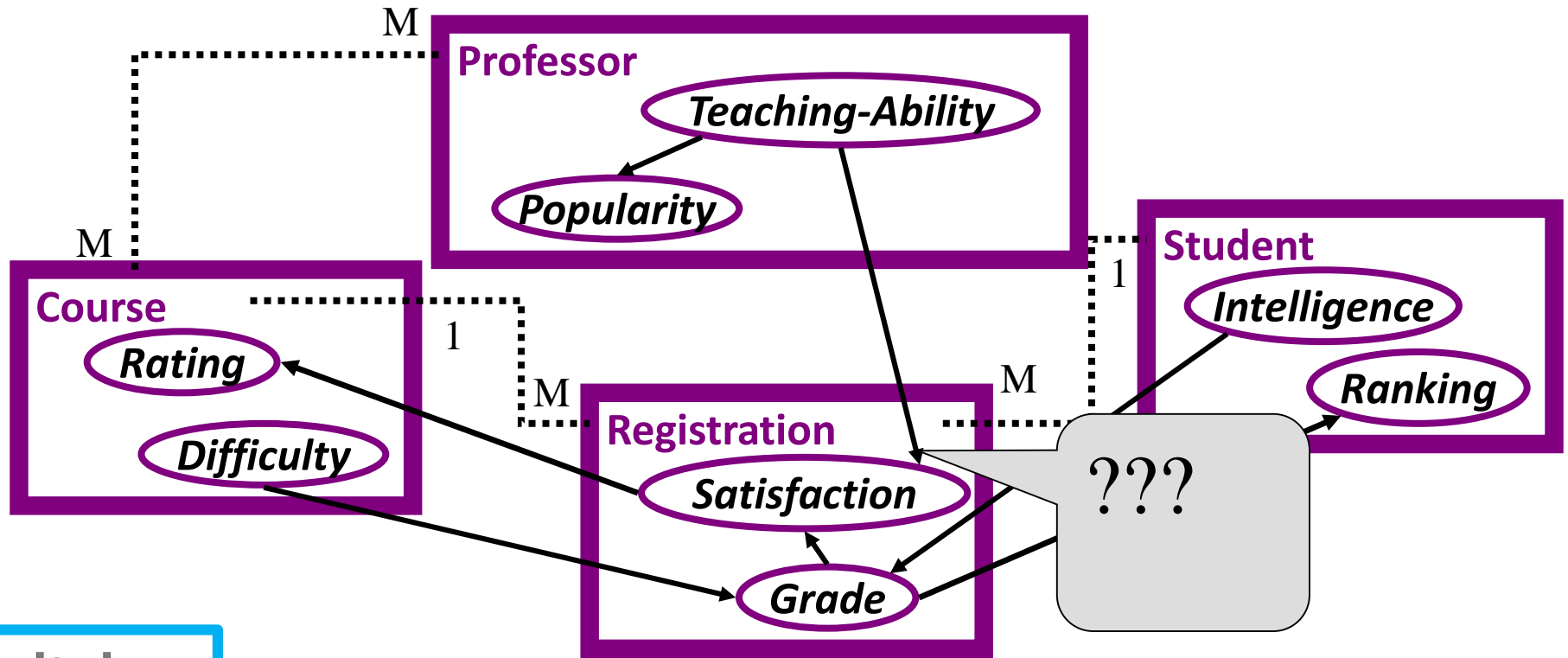
Problem with some parameters θ$_S$

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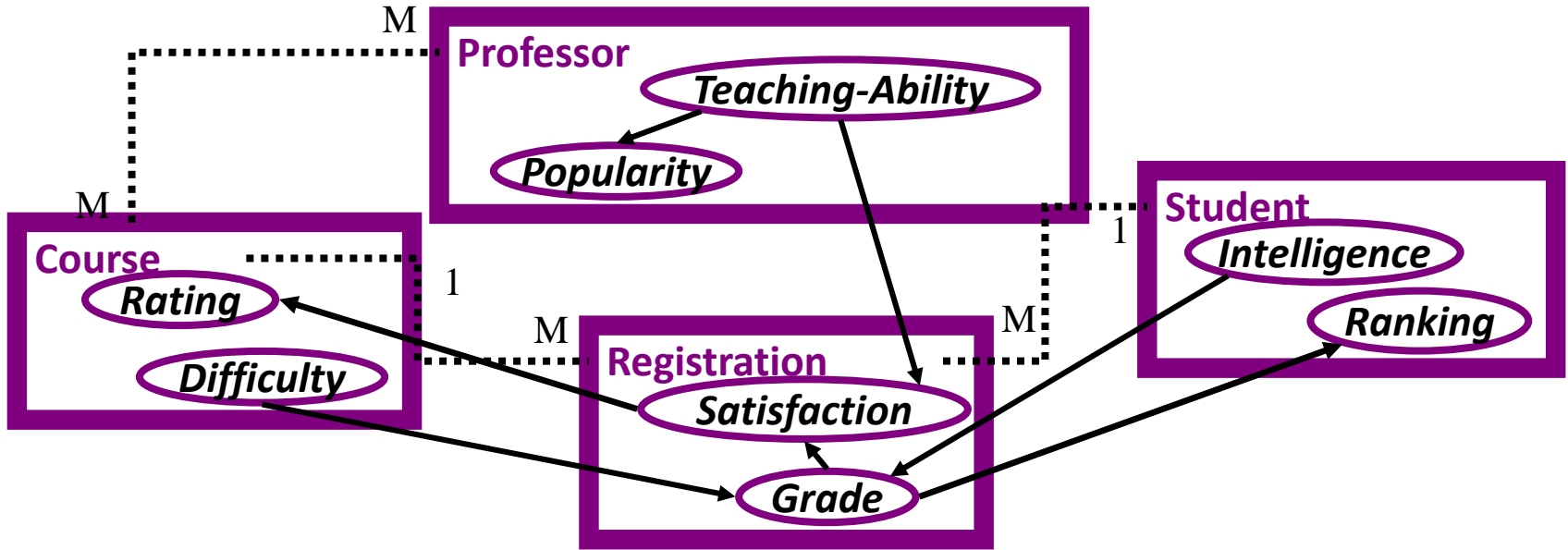
When the slot chain τ (e.g. Course. Instructor)
is not guaranteed to be single-valued, we must specify the probabilistic dependence of

- **x.A** — Registration. Satisfaction
- **on the set** {y.B: y ∈ x.τ}

who are instructors of the Course

The Teaching-Ability of all the profs

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

M

M

Professor

Teaching-Ability

Popularity

Student

1

Intelligence

Ranking

Course

Rating

1

Difficulty

AGG

M

AGG

Registration

Satisfaction

AGG

M

Grade

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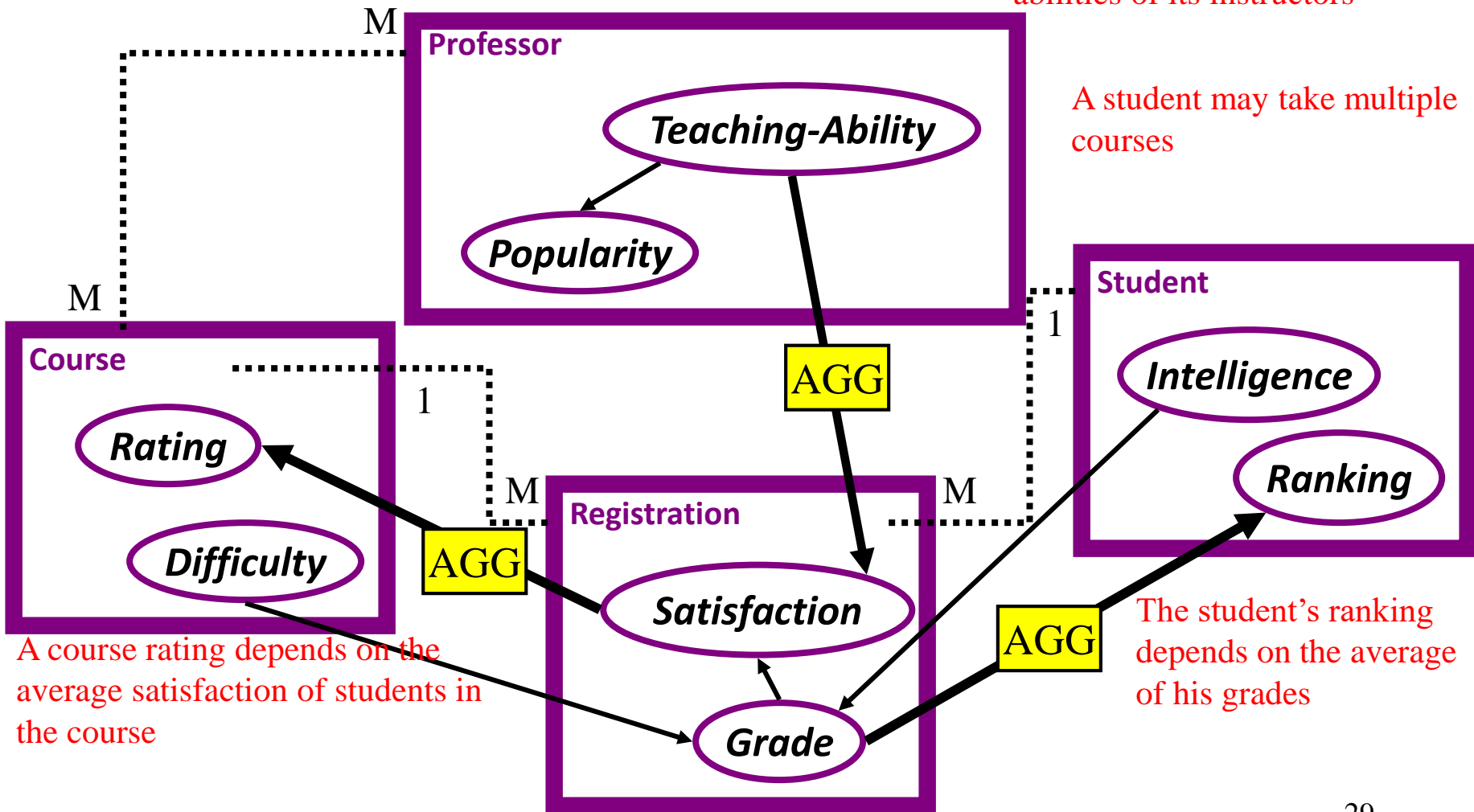
A course rating depends on the average satisfaction of students in the course

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

CPDs in PRMs

Professor

Teaching-Ability

Popularity

M

M

Course

Rating

Difficulty

1

Student

Intelligence

Ranking

1

M

M

AGG

Registration

Satisfaction

Grade

AGG

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
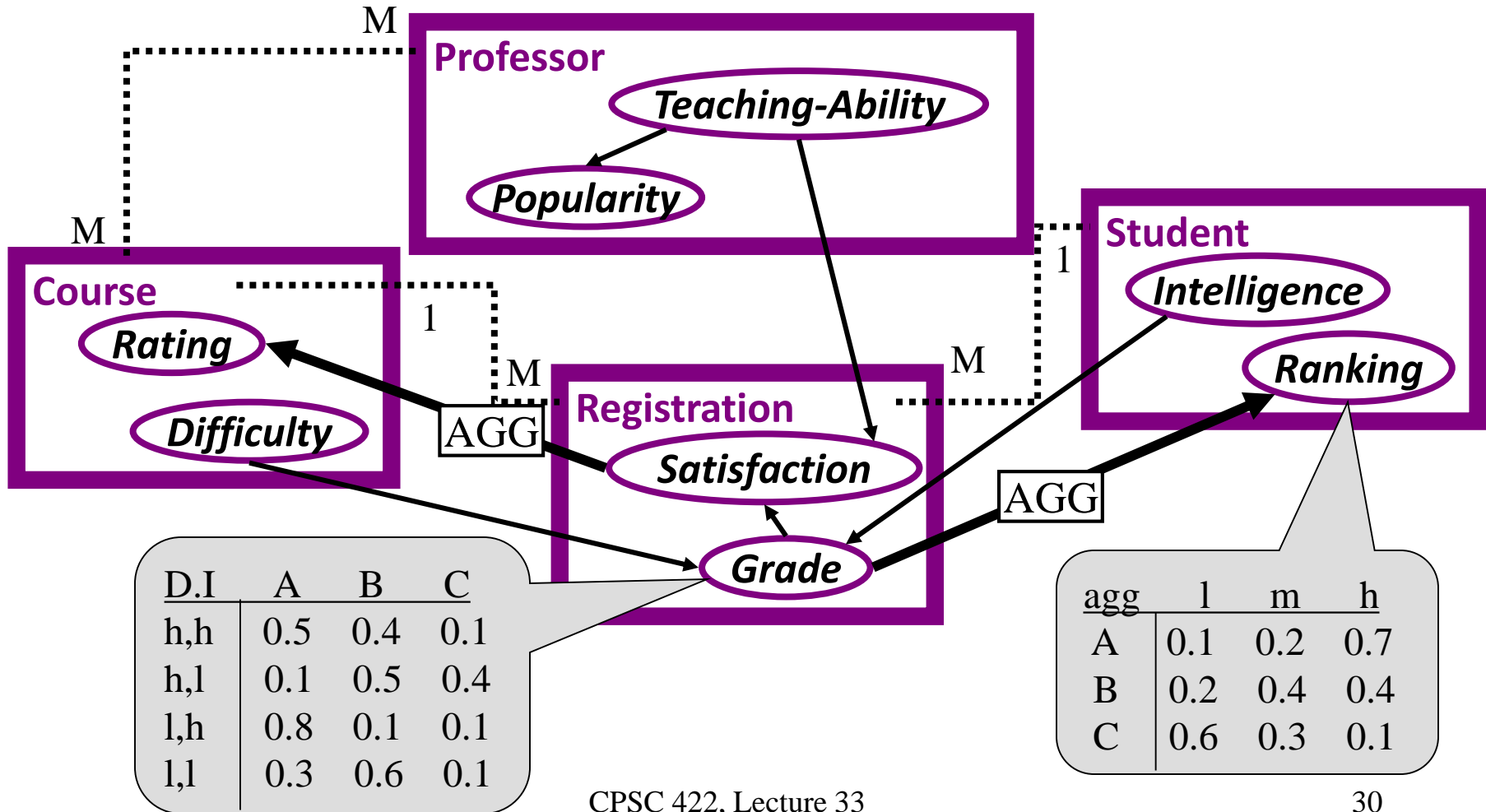
JPD in PRMs

- 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**

Professor
Teaching-Ability
Popularity

Course
Rating
Difficulty

Student
Intelligence
Ranking

Registration
Satisfaction
Grade

M 1 M 1 M M

Professor
Name
Prof. Gump
Popularity
???
Teaching-Abili
???

Professor
Name
Prof. Vincent
Popularity
???
Teaching-Ability
???

Student
Name
Jane Doe
Intelligence
Ran

Student
Name
Sue Chu
Intelligence
???
Ranking
???

Course
Name
CS101
Difficulty
??????
ating
??

Course
Name
Phil101
Difficulty
???
Rating
???

Registration
RegID
#3
Grad
?
Satis

Registration
RegID
#5
Grade
???
Satisfa
???

Registration
RegID
#6
Grade
???
Satisfaction
???

Gump
Pop
Teaching-Ab

Vincet
Pop
Teschg-Ab

Jane
Int
Rank
Sue

Sue
Int
Rank

CS
Rate
Difficulty

Phil
Rate
Difficulty

#3 grade
satisf

#5 grade
satisf

#6 grade
satisf

same CPD

Parameter Sharing / CPTs reuse, where else?

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

Final Issue….

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

      A. Polynomial
      B.   Acyclic
      C. Cyclic
      D. Recursive

i·clicker.

Class Dependency Graph for the University Domain
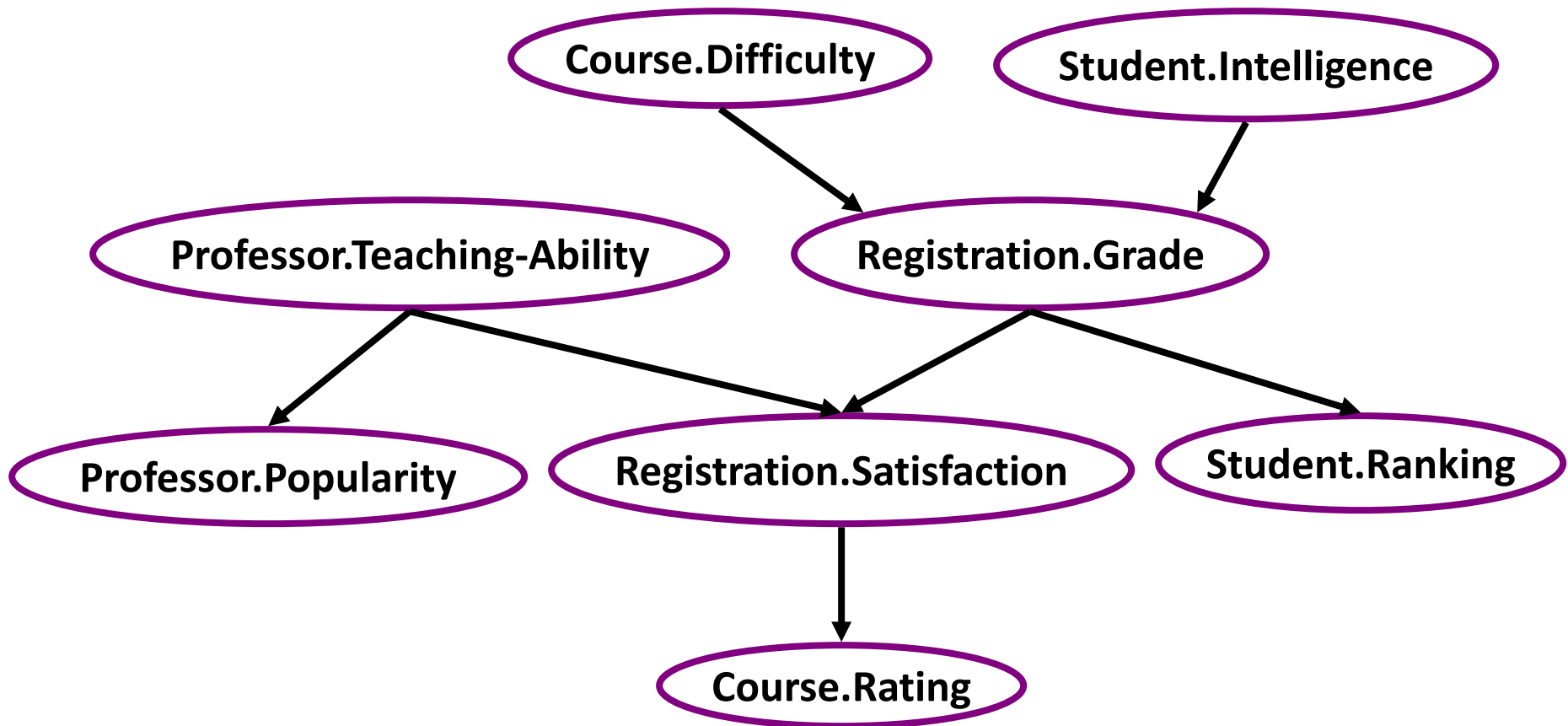
Ensuring Acyclic Dependencies

- 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

Person

> **Name**

Blood-Type
P-Chromosome
M-Chromosome

1

M

Blood Test

> **BT-ID**

Patient

Results

Contaminated
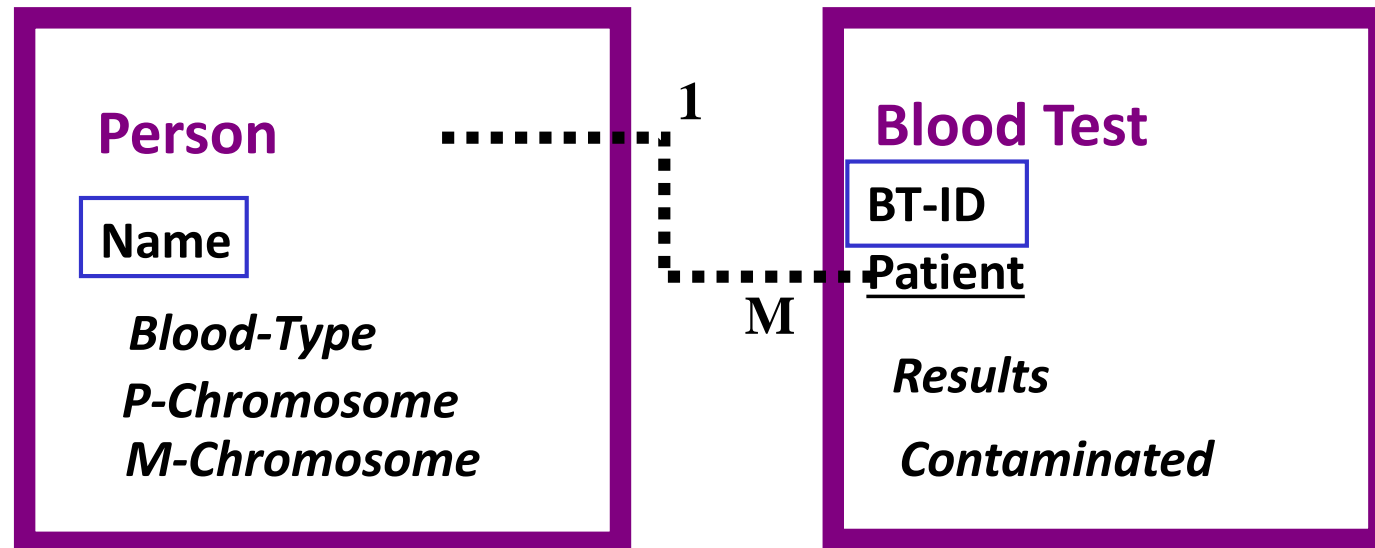
M = Maternal
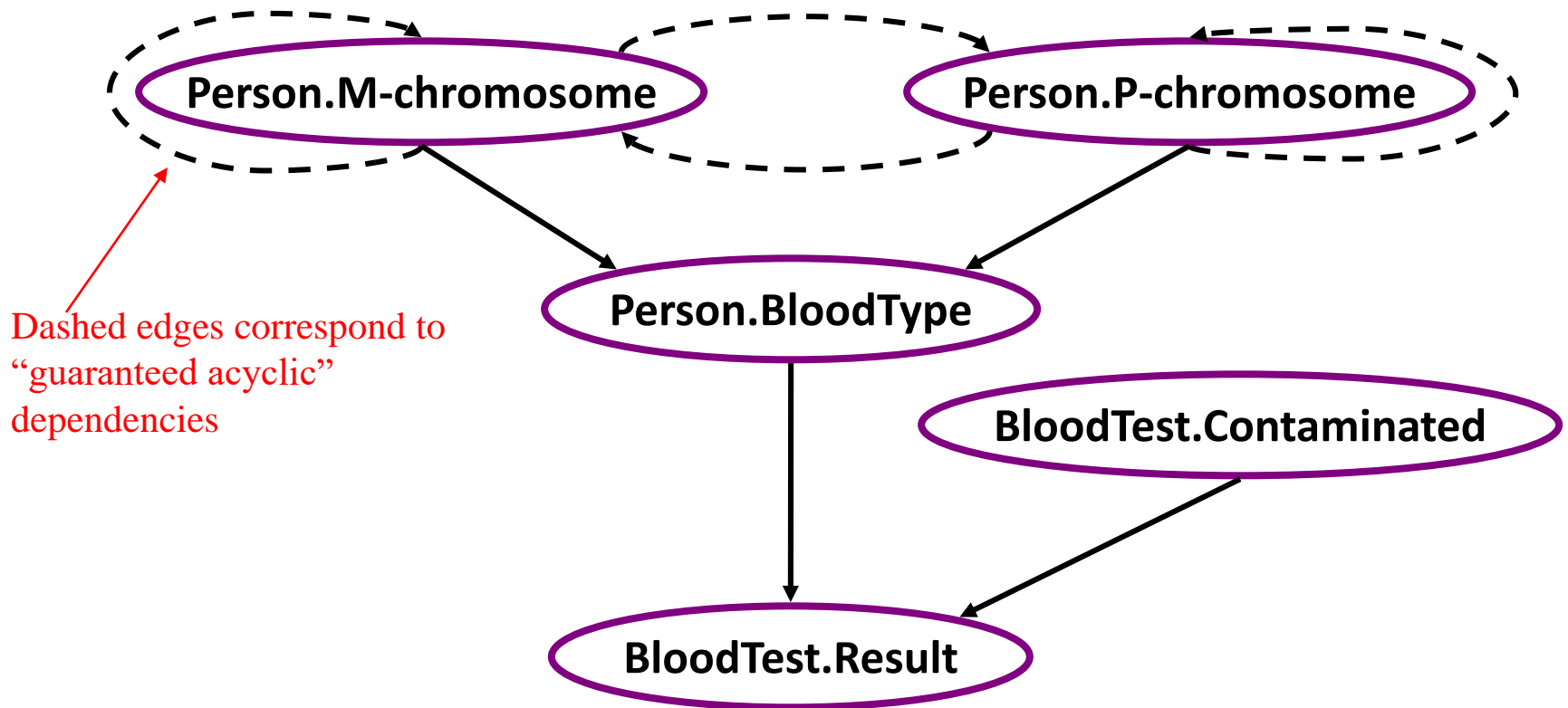P = Paternal

Relational Schema for the Genetics Domain
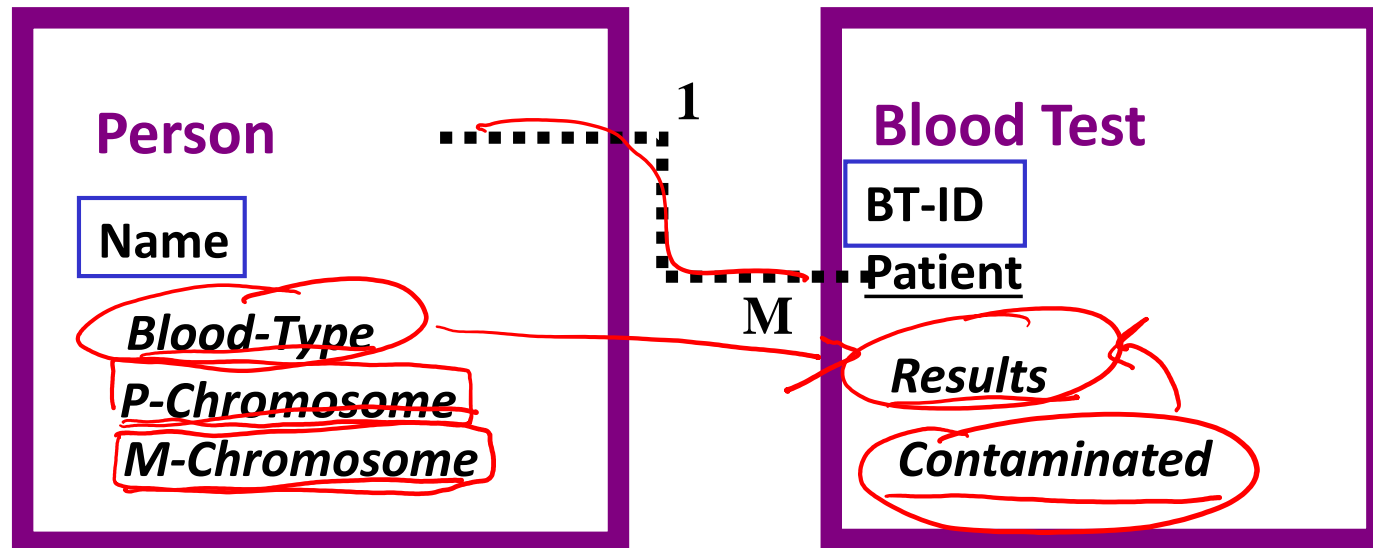
Dependency Graph: Genetics Domain

Person

Name

Blood-Type
P-Chromosome
M-Chromosome

1

M

Blood Test

BT-ID
Patient

Results
Contaminated

Person.M-chromosome

Person.P-chromosome

Person.BloodType

BloodTest.Contaminated

BloodTest.Result
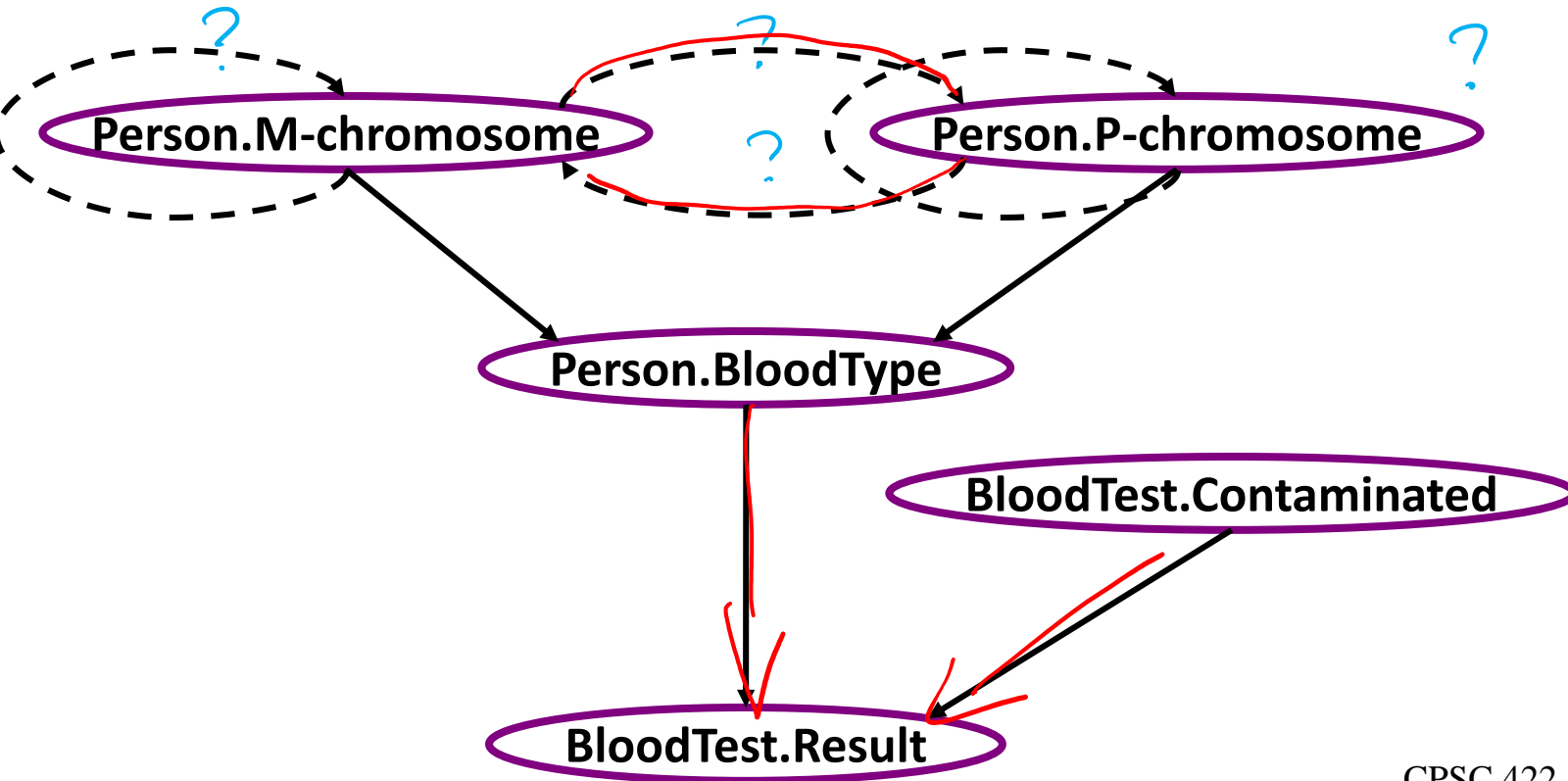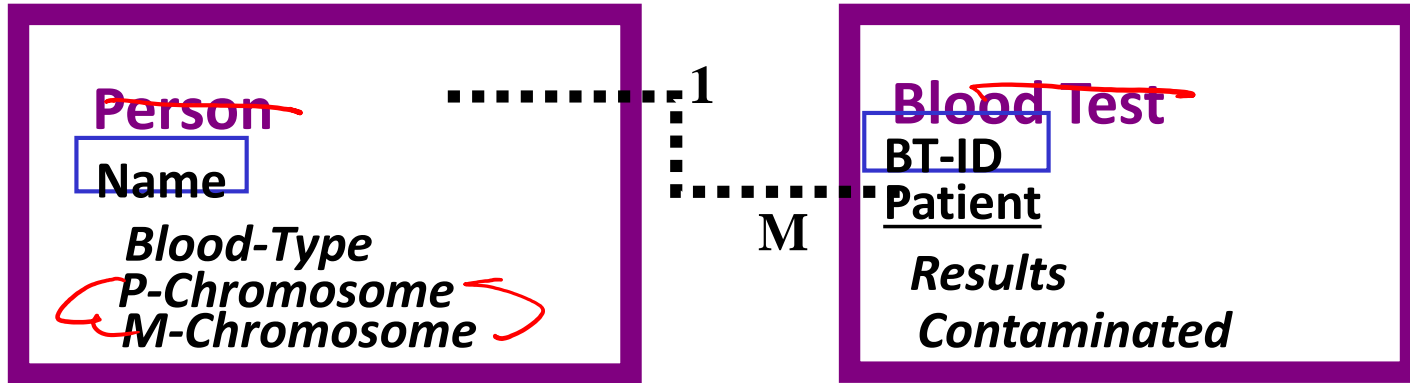
PRM for the Genetics Domain
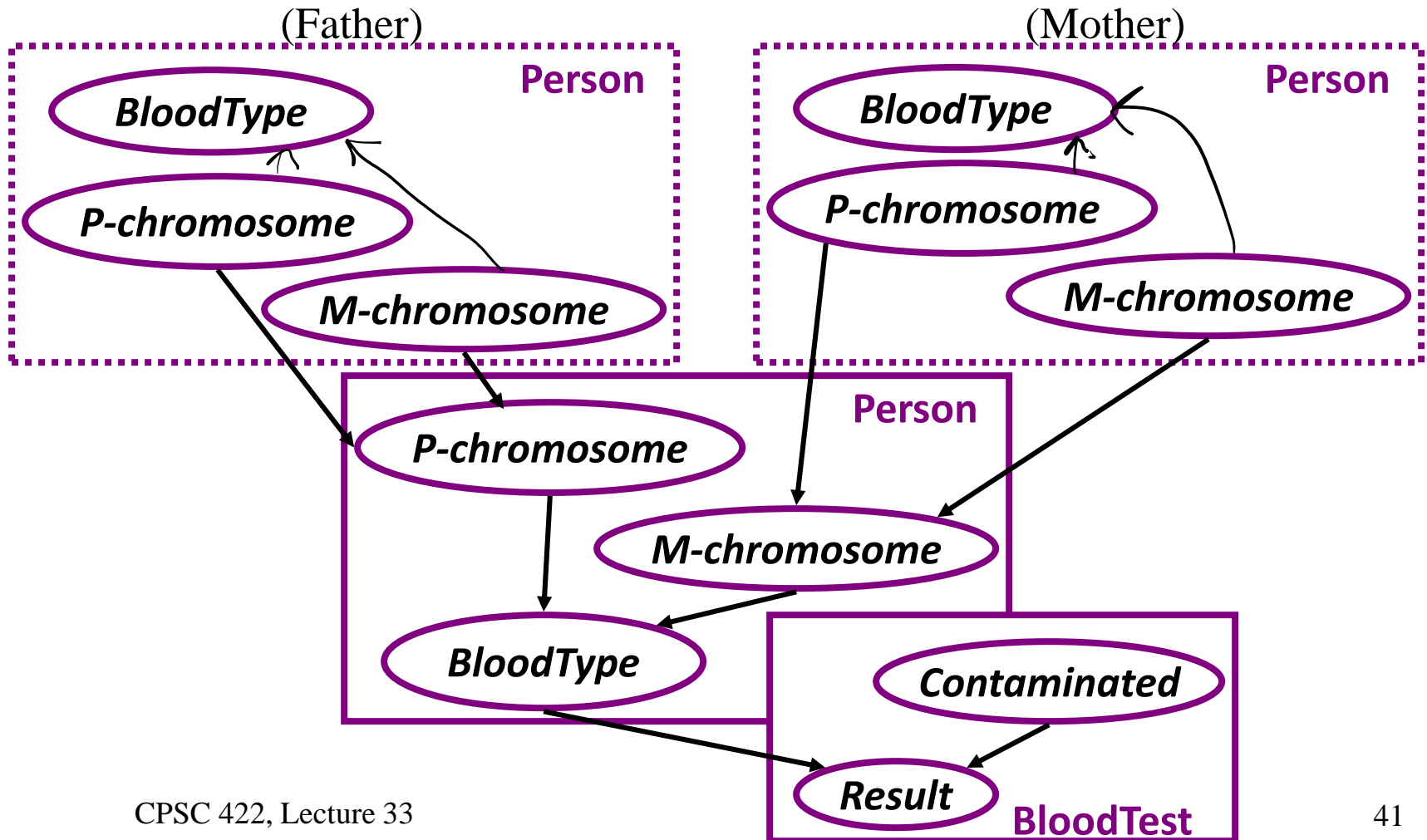
Dependency Graph for Genetics Domain

Person.M-chromosome

Person.P-chromosome

Dashed edges correspond to "guaranteed acyclic" dependencies

Person.BloodType

BloodTest.Contaminated

BloodTest.Result

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**

MONDAY: Third and final research paper reading

- **EMNLP 2020 paper** MEGA RST Discourse Treebanks with Structure and Nuclearity from Scalable Distant Sentiment Supervision *Patrick Huber, Giuseppe Carenini*

- (guest speaker: first author of the paper PhD student Patrick Huber !

- **Material to review** before reading:
  – CKY,
  – Exploration/Exploitation trade-off in RL,
  – Beam Search (from 322),
  – Recurrent Neural Networks (if you have seen them in 340 or other courses)

44

Relatively Recent Book + VideoTutorial on Star-AI

Book*: Logic, Probability, and Computation* 2016
Luc De Raedt, KU Leuven, Belgium,
Kristian Kersting, Technical University of Dortmund, Germany,
Sriraam Natarajan, Indiana University,
David Poole, University of British Columbia

An intelligent agent interacting with the real world will encounter individual people, courses, test results, drugs prescriptions, chairs, boxes, etc., and needs to reason about properties of these individuals and relations among them as well as cope with uncertainty.

Uncertainty has been studied in probability theory and graphical models, and relations have been studied in logic, in particular in the predicate calculus and its extensions. This book examines the **foundations of combining logic and probability** into what are called relational probabilistic models. It introduces representations, inference, and learning techniques for probability, logic, and their combinations.

The book focuses on two representations in detail: **Markov logic networks**, a relational extension of undirected graphical models and weighted first-order predicate calculus formula, and **Problog**, a probabilistic extension of logic programs that can also be viewed as a Turing-complete relational extension of Bayesian networks.

Statistical Relational AI tutorial at NIPS (now NEURIPS) 2017.
https://www.facebook.com/watch/live/?v=1552222671535633&ref=watch_permalink
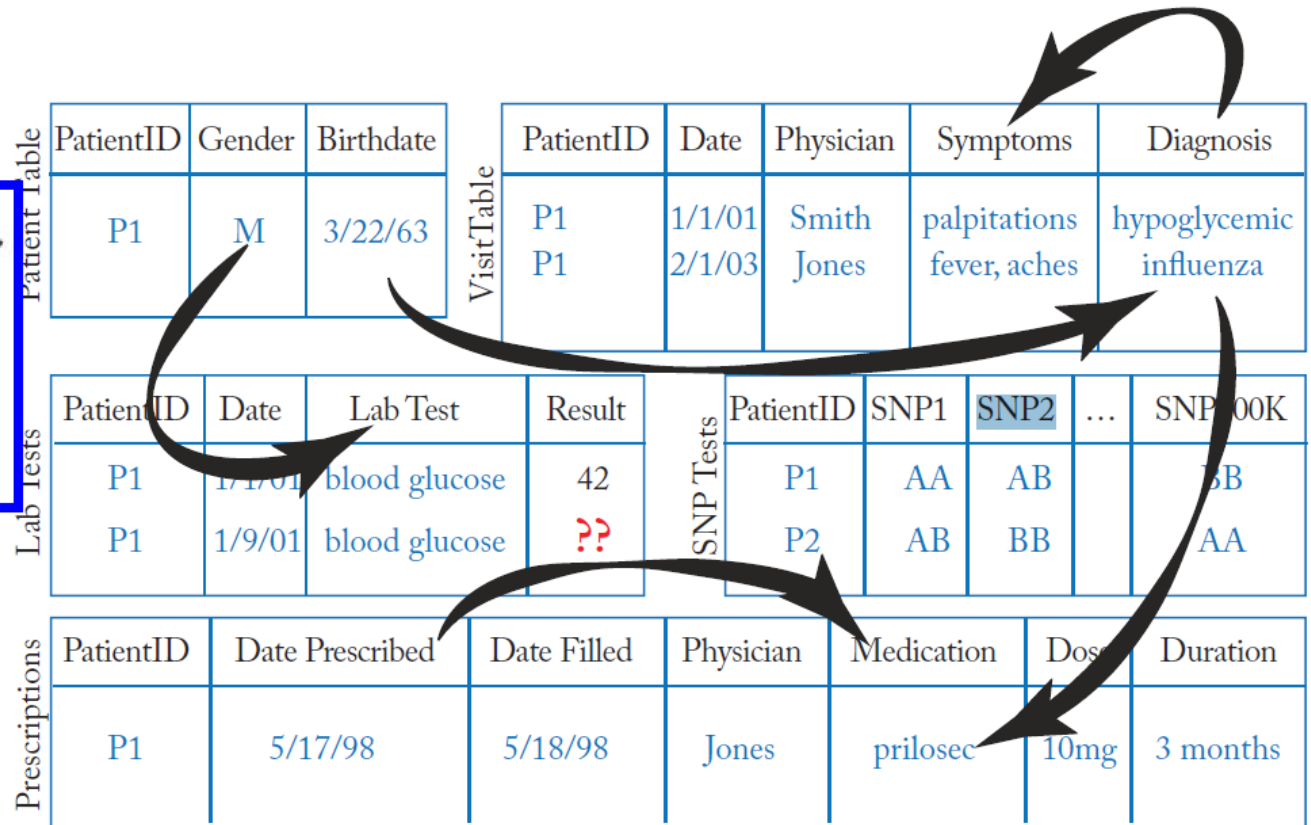
Figure 1.3: Electronic Health Records (EHRs) are relational databases capturing noisy and missing information with probabilistic dependencies (the black arrows) within and across tables.
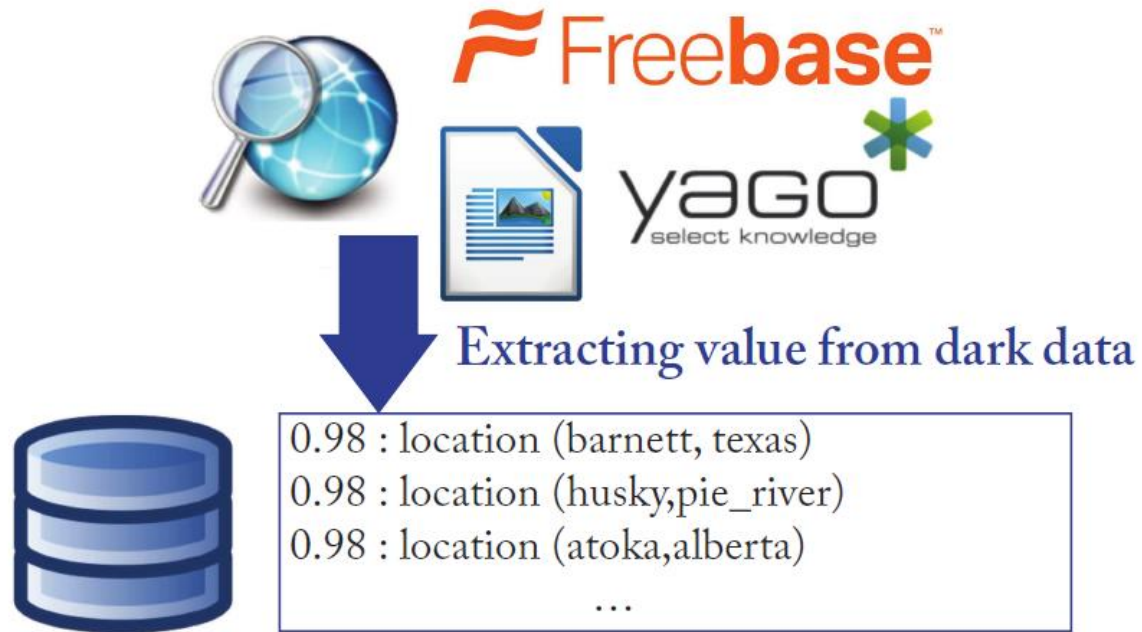
Figure 1.4: Populating a knowledge base with probabilistic facts (or assertions) extracted from dark data (e.g., text, audio, video, tables, diagrams, etc.) and background knowledge.
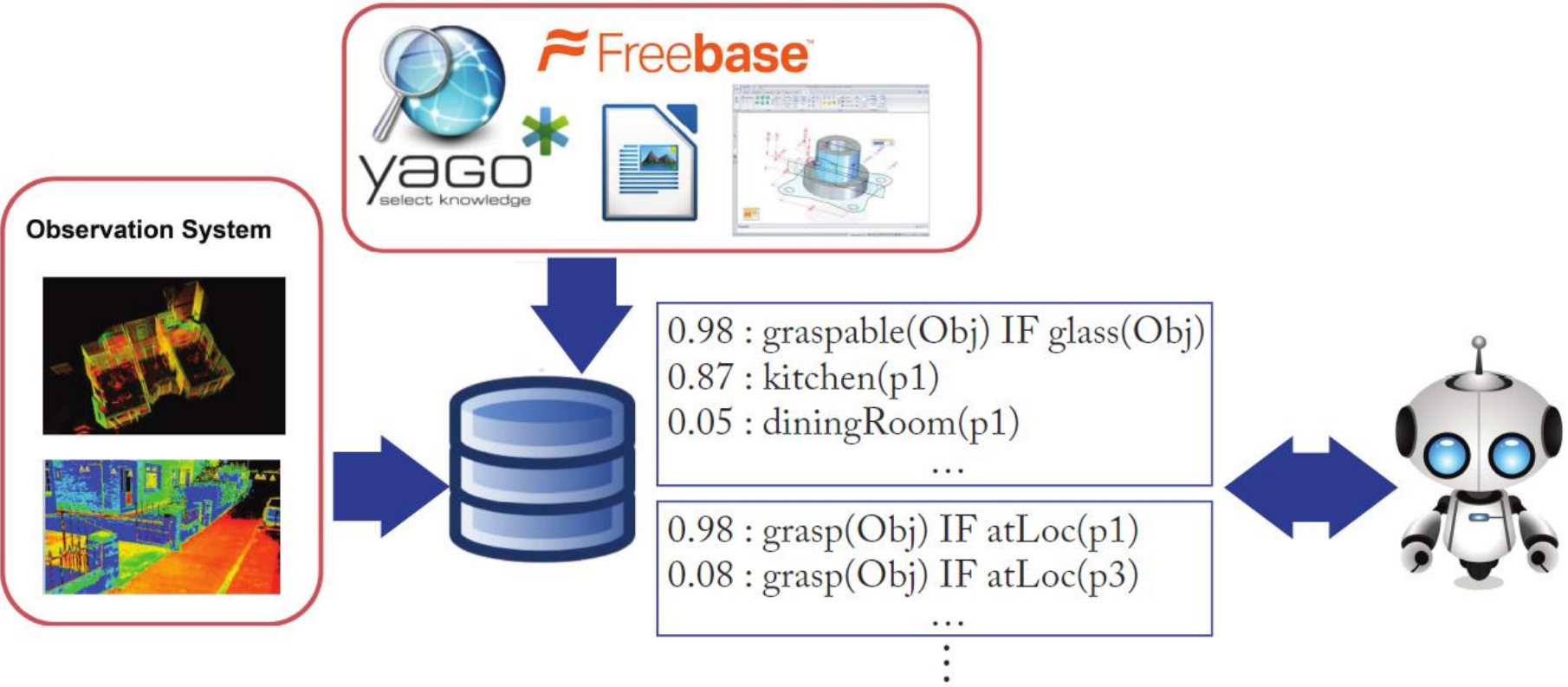
Figure 1.7: The robot's relational probabilistic model is populated with the data produced by its perception and experience (robot log data, human motion tracking, environment information, etc.) as well as with facts (or assertions) extracted from other dark data.

Last class on Wed

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

Assignment-4 Due !

Fill out on-line Teaching Evaluation