Learning Probabilistic Relational Models

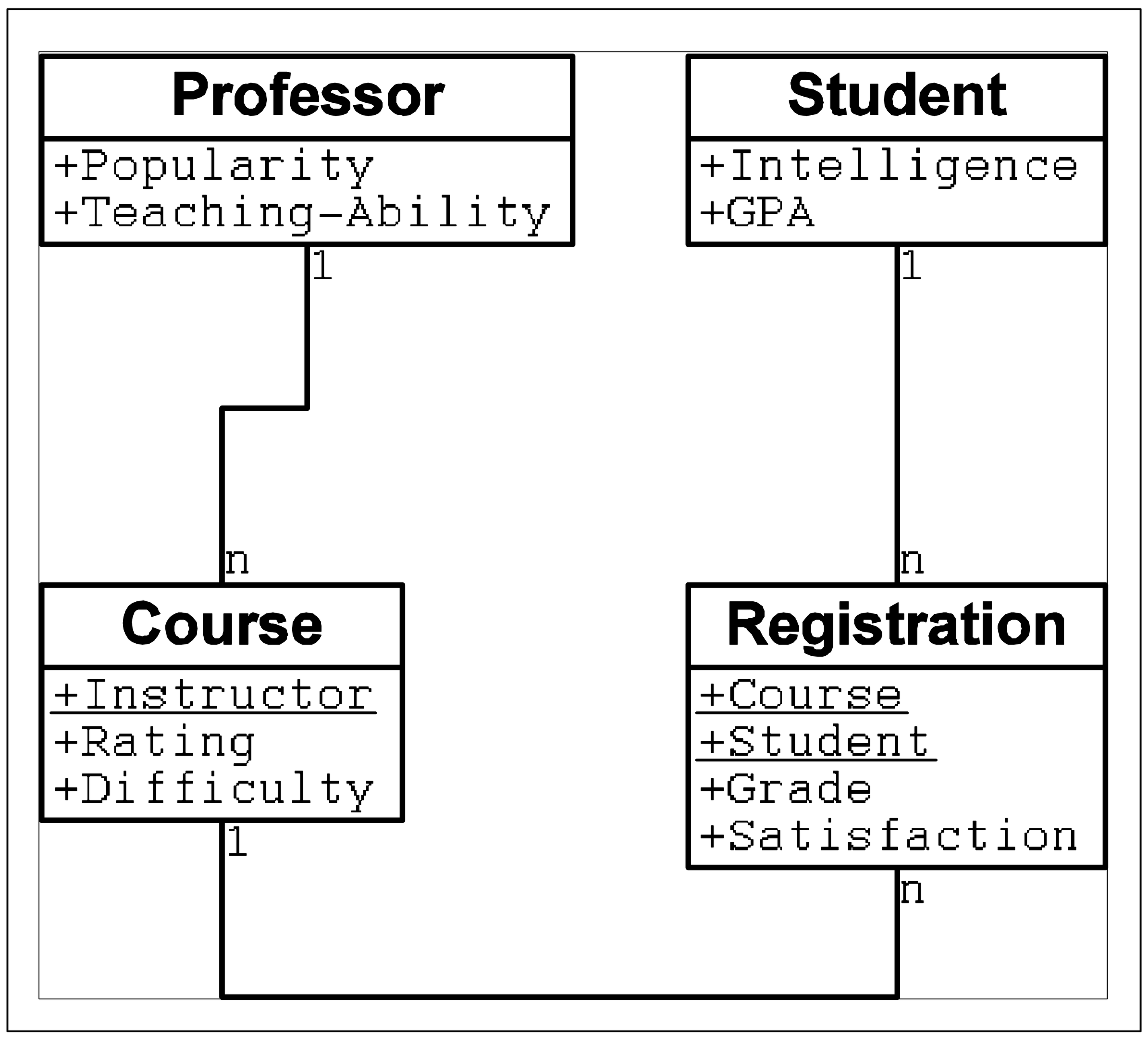
Getoor, Friedman, Koller, Pfeffer

Probabilistic Relational Models

Course.Instructor is foreign key for Professor relation

Registration.Course is foreign key for Course

Registration.Student is foreign key for Student



Corresponding Database

Professor	Popularity	Teaching-Ability
Gump	high	medium

Student	Intelligence	GPA
Gomer Pyle	low	2.0
Jane Doe	high	4.0

Course	Professor	Difficulty	Rating
Phil101	Gump	low	high
Com301	Gump	high	medium

Registration	Course	Student	Grade	Satisfaction
Reg123	Com301	Gomer Pyle	C	1
Reg333	Phil101	Jane Doe	A	3
Reg135	Com301	Jane Doe	A	2

Relational Schema

- Set of classes $X = \{X_1,\ldots,X_n\}$ (equivalent to relational tables)
- Each class has
  - descriptive attributes $A(X_i)$
    - A(Student) = {Intelligence, GPA}
    - JaneDoe.Intelligence
  - reference slots (foreign keys that point to other relations): $R(X_i)$
    - R(Registration) = {Student, Course}
    - Reg333.Student = JaneDoe
    - Reg333.Course = Phil101
  - inverse reference slots:
    - JaneDoe.RegisteredIn = {Reg333, Reg334}
      - Constructed automatically

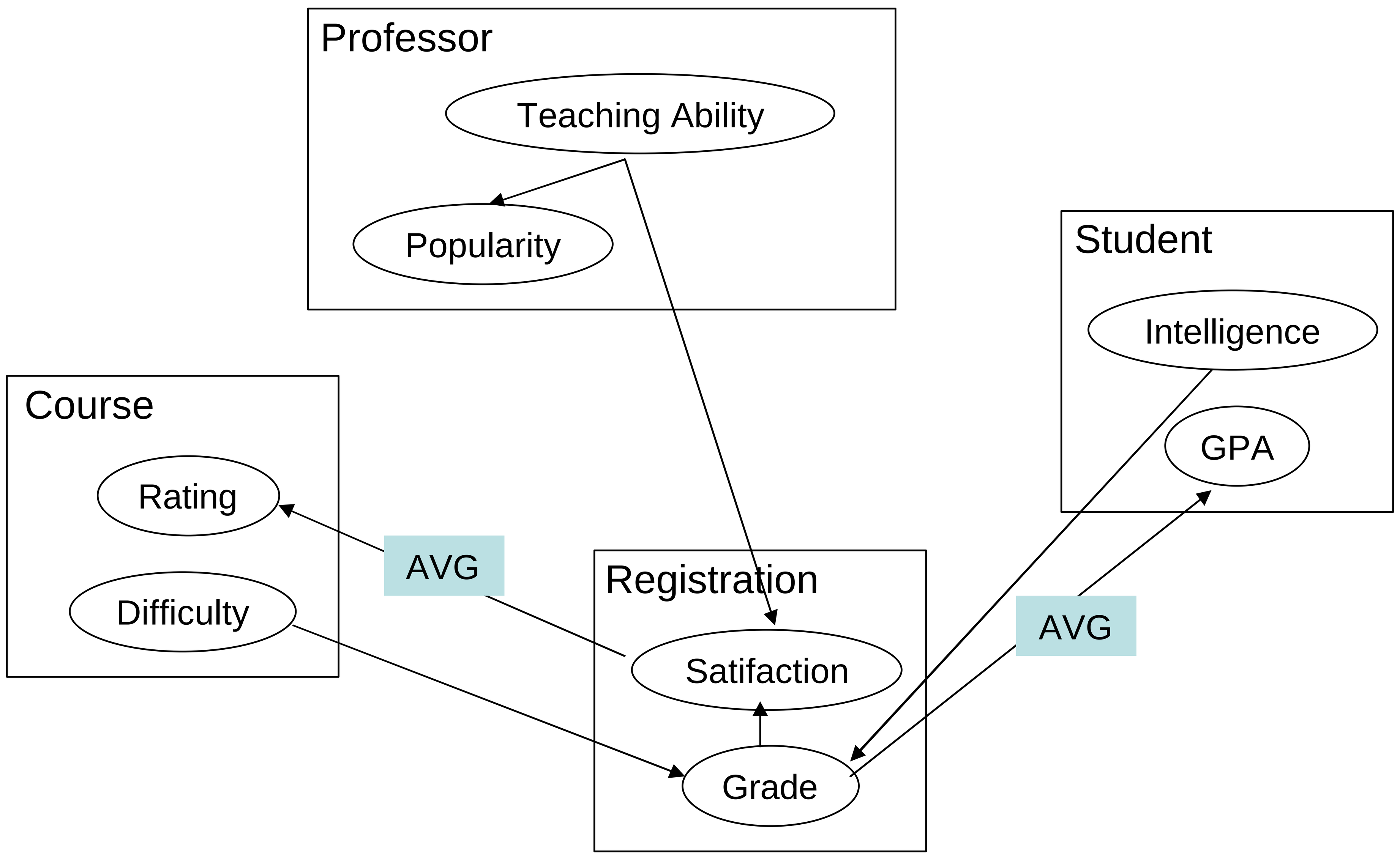
Slot Chains
(path expressions)

- Student.registered-in.Course.Instructor
  = bag of instructors of the courses that the student is registered in
  – bag is a set with multiple occurrences allowed
  – JaneDoe.registered-in.Course.Instructor = {Gump,Gump}
- Aggregations: Mean, Average, Mode
  – AVG(Student.registered-in.Grade)
    - Average grade of student
  – MODE(Student.registered-in.Course.Instructor)
    - Professor from whom student has taken the most courses

PRM Schema = Relational Schema + Probabilistic Parents

- Each attribute has a set of path expressions describing the parents of that attribute
  – parents(Student.gpa) = {AVG(Student.registered-in.Grade)}
  – parents(Registration.satisfaction) = {Registration.Course.Professor.TeachingAbility, Registration.Grade}
  – parents(Registration.grade) = {Registration.Student.Intelligence, Registration.Course.Difficulty}
  – parents(Professor.Popularity) = {Professor.TeachingAbility}
  – parents(Course.rating) = {AVG(Course.Registrations.Satisfaction)}

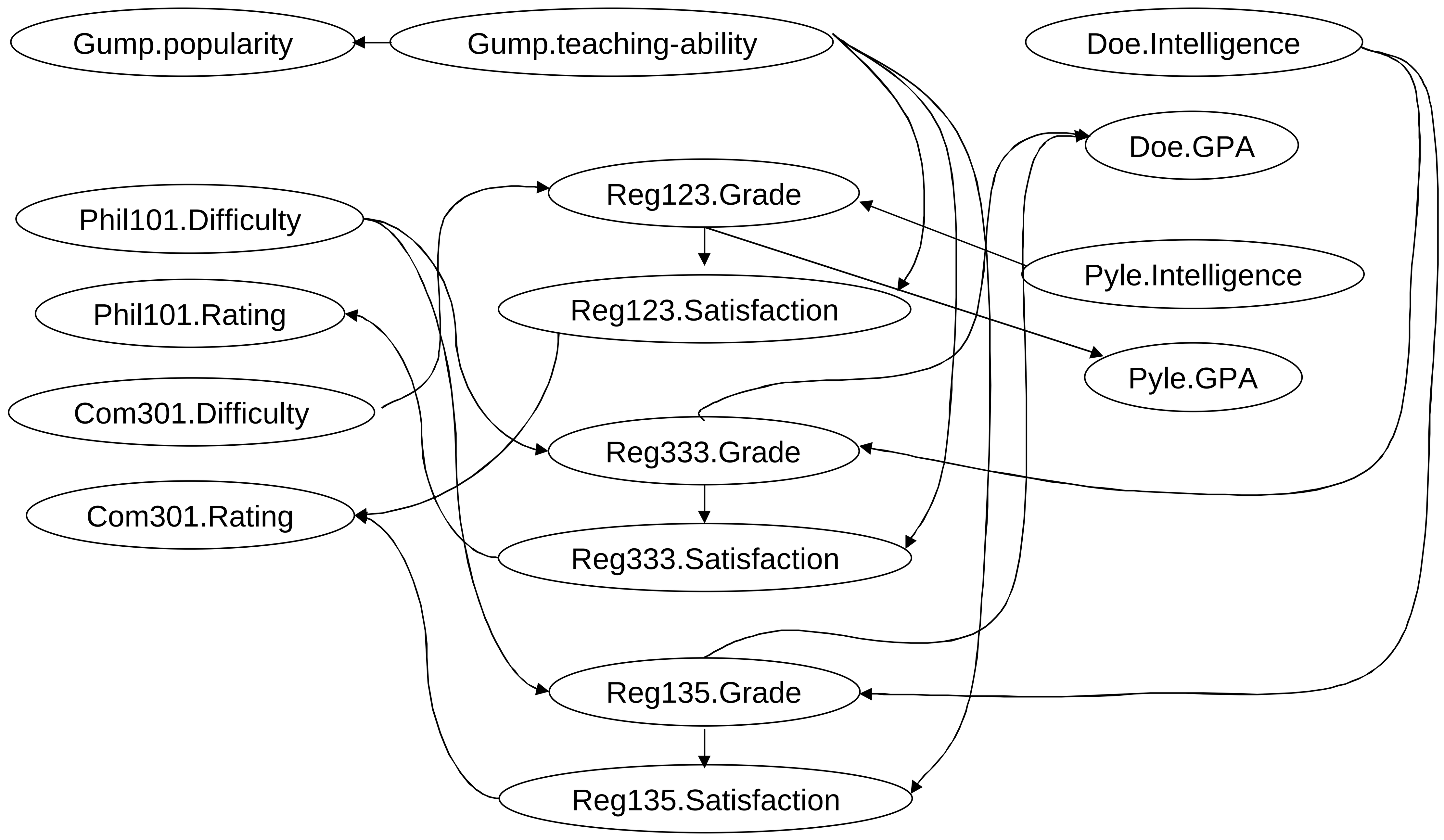
Visualizing the PRM Schema



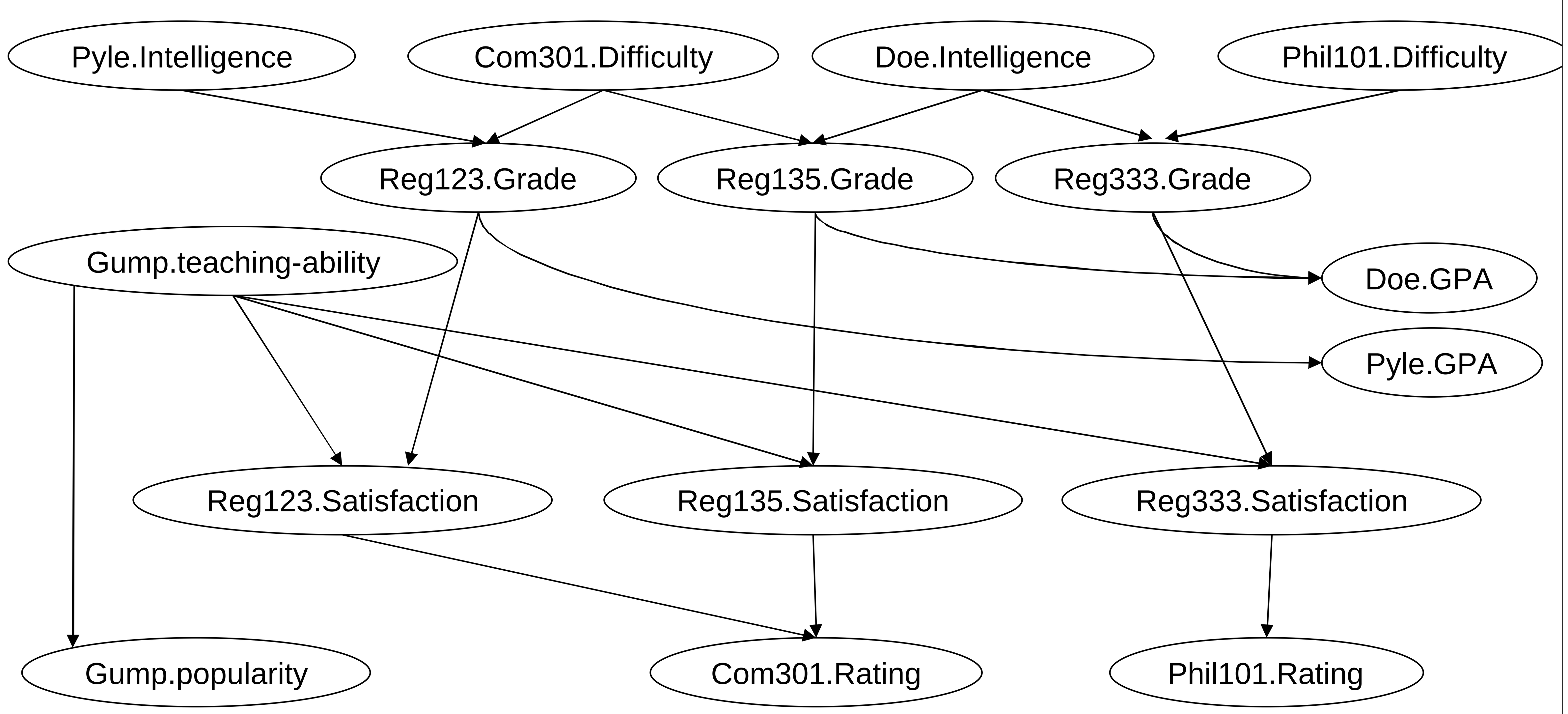
Probabilistic Relational Model

1. Relational Schema
2. Specification of the parents of each descriptive attribute (in terms of path expressions)
3. Conditional Probability Distribution for each attribute in each class
   - Conditional probability table:
     P(attribute | parents(attribute))
   - Parametric model:
     P(attribute | parents(attribute)) = F(attribute, parents(attribute); θ) for some parameters θ.

Instantiating the PRM on a database

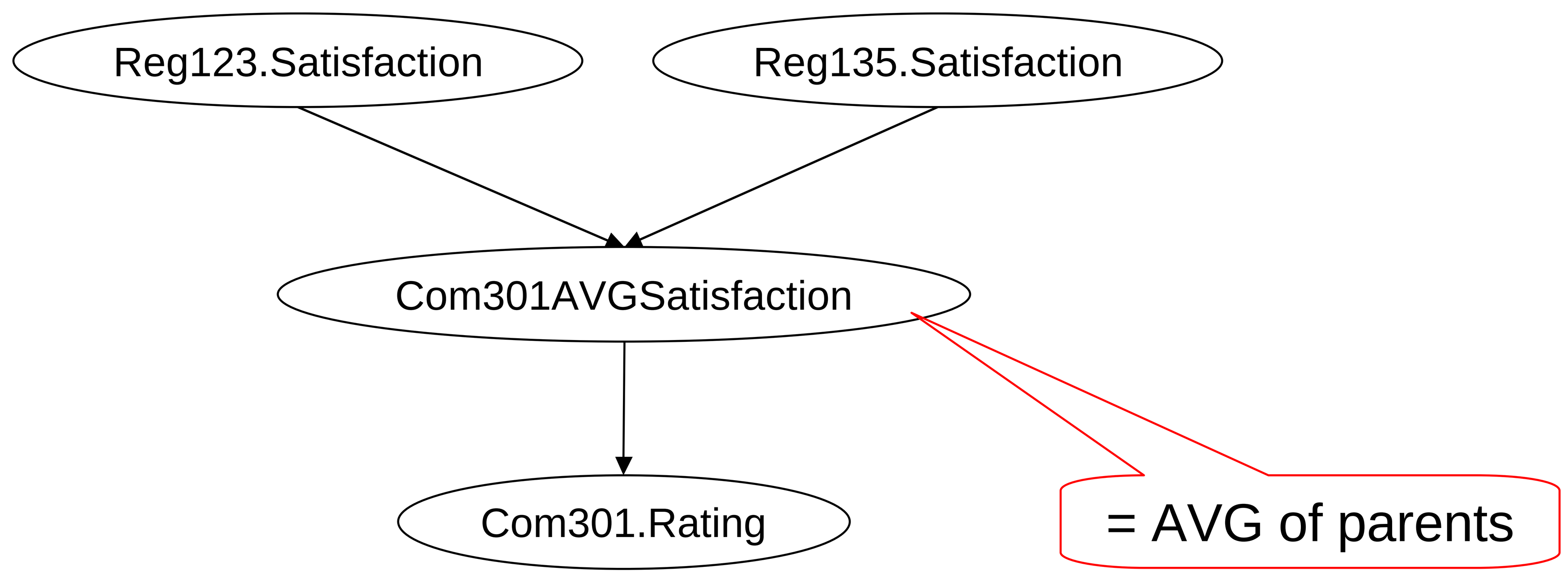


Redrawn to show DAG



Aggregations

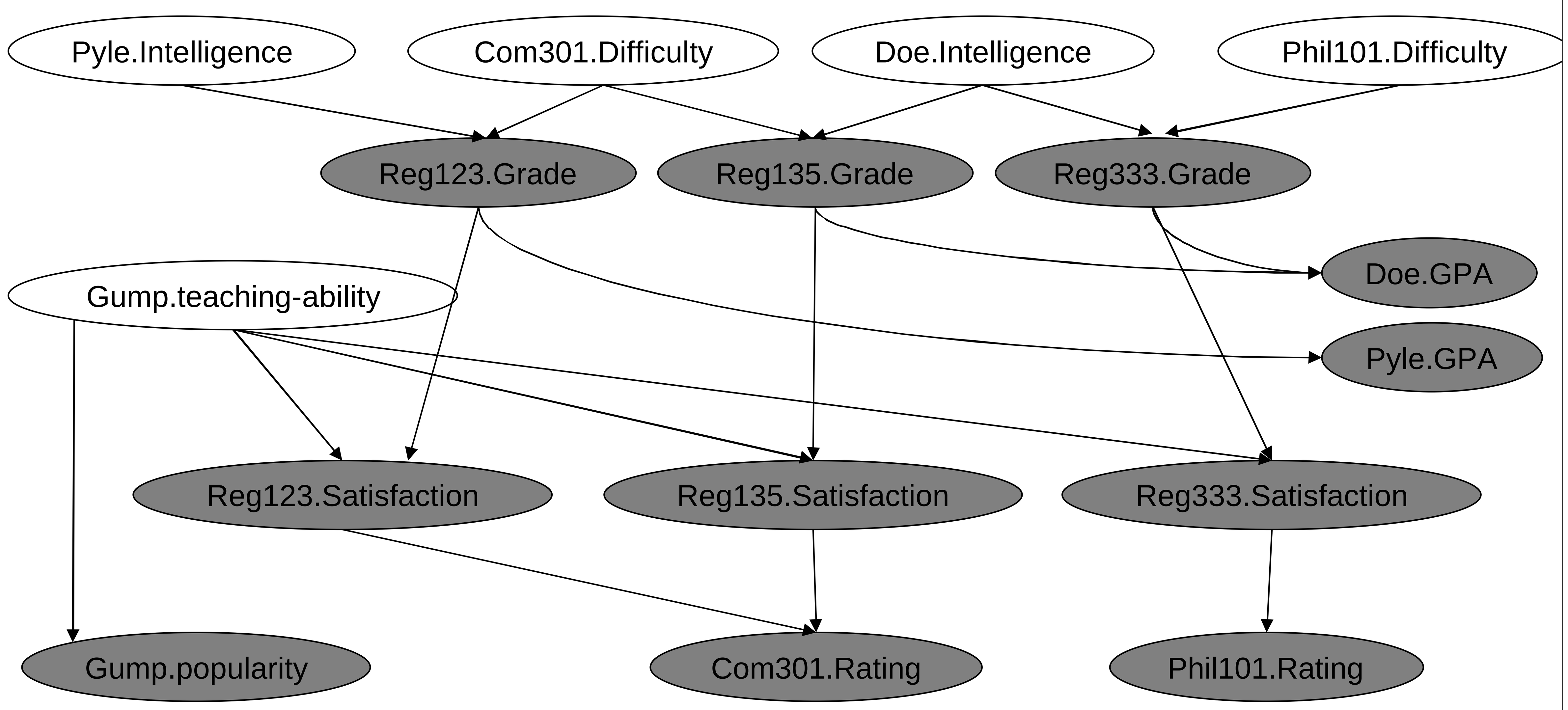
- We must introduce deterministic intermediate nodes to represent the aggregated value



Example Inferences
useful for tenure and letters of reference

- Observe Registration.Grade (and Student.GPA), Registration.Satisfaction (and Course.Rating), and Professor.Popularity
- Infer Student.Intelligence and Professor.TeachingAbility
- P(Gump.TeachingAbility, Pyle.Intelligence, Doe.Intelligence | …)

Example Inference (2)



Example Inference (3)

- Example: We might observe that Pyle has a GPA of 4.0. This could be explained either by Pyle.Intelligence or by Course.Difficulty for all of the courses that he took.
- The grades of other students in the same classes that Pyle took can tell us Course.Difficulty, which in turn can help us explain away the 4.0 GPA (e.g., because Pyle took only easy courses).
- This is a form of relational inference! We could not figure it out only from looking at Pyle's courses and grades.

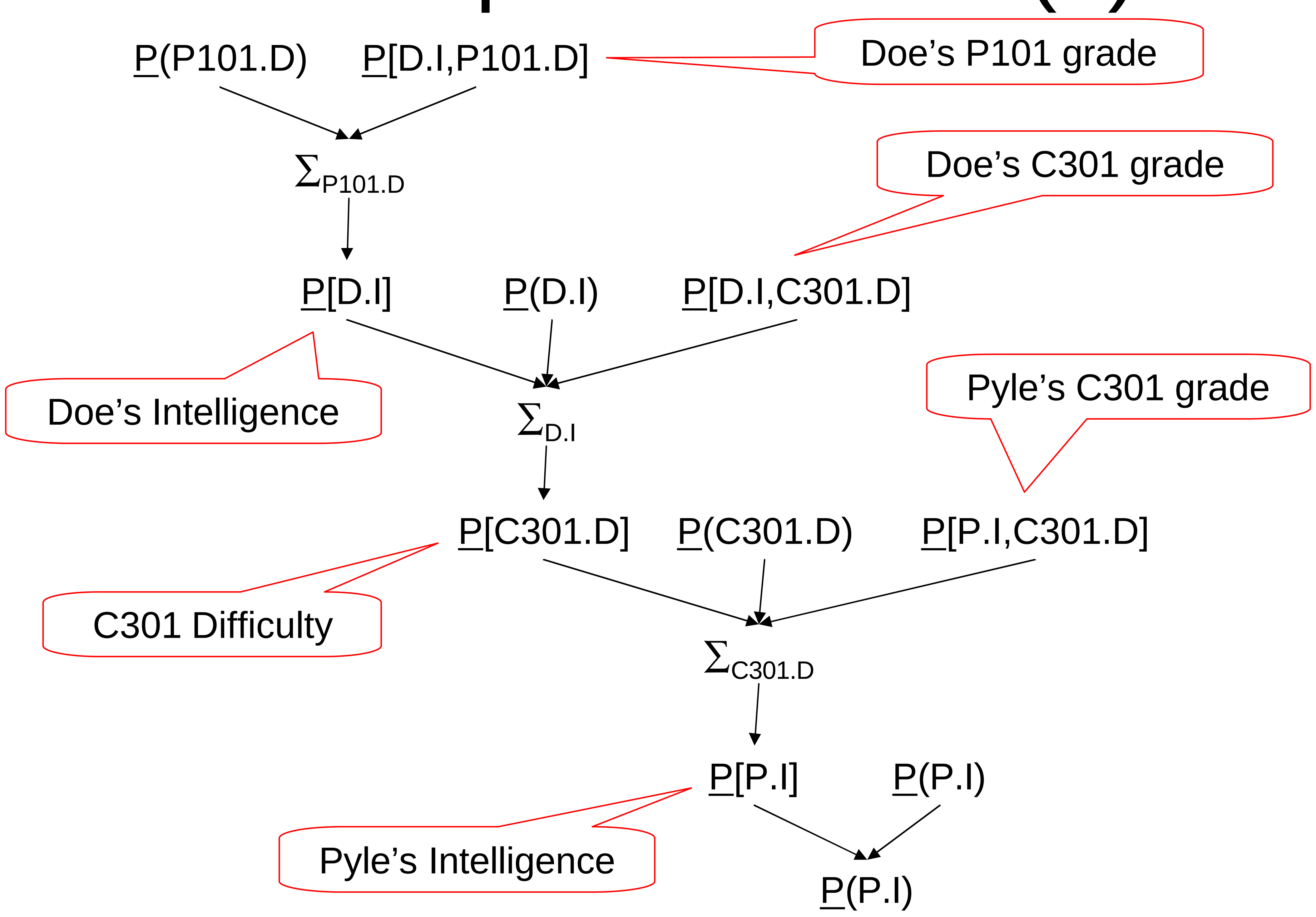
Example Inference (4)

$\underline{P}$(P.I | …) = $\Sigma_{C301.D}\ \Sigma_{D.I}\ \Sigma_{P101.D}$ $\underline{P}$(R123.G | P.I, C301.D) ¢ $\underline{P}$(R135.G | D.I, C301.D) ¢ $\underline{P}$(R333.G | D.I, P101.D) ¢ $\underline{P}$(P.I) ¢ $\underline{P}$(C301.D) ¢ $\underline{P}$(D.I) ¢ $\underline{P}$(P101.D)

$\underline{P}$(P.I | …) = $\Sigma_{C301.D}\ \Sigma_{D.I}\ \Sigma_{P101.D}$ $\underline{P}$[P.I, C301.D] ¢ $\underline{P}$[D.I, C301.D] ¢ $\underline{P}$[D.I, P101.D] ¢ $\underline{P}$(P.I) ¢ $\underline{P}$(C301.D) ¢ $\underline{P}$(D.I) ¢ $\underline{P}$(P101.D)

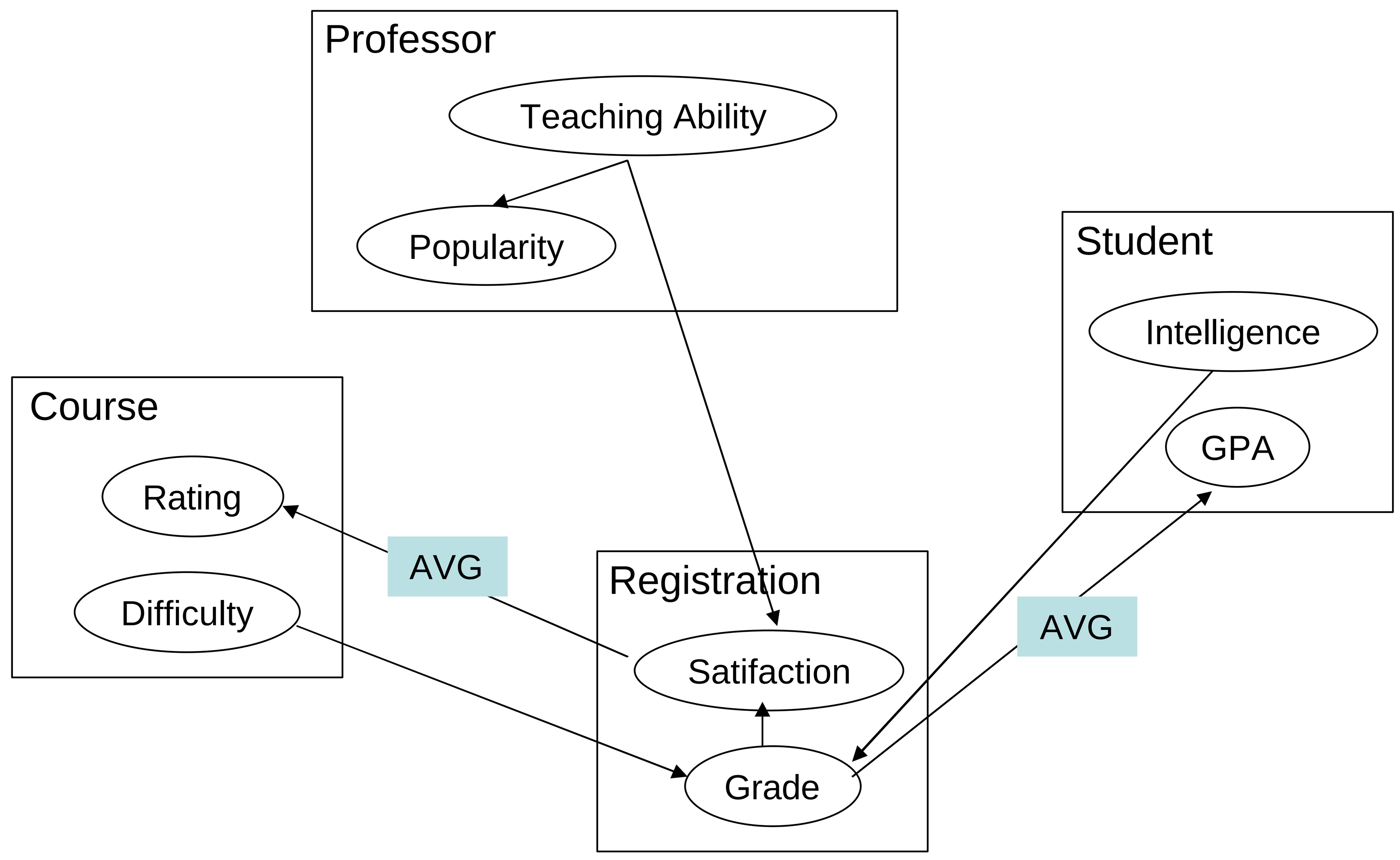
$\underline{P}$(P.I | …) = $\underline{P}$(P.I) ¢ $\Sigma_{C301.D}$ $\underline{P}$[P.I, C301.D] ¢ $\underline{P}$(C301.D) ¢ $\Sigma_{D.I}$ $\underline{P}$[D.I, C301.D] ¢ $\underline{P}$(D.I) ¢ $\Sigma_{P101.D}$ $\underline{P}$[D.I, P101.D] ¢ $\underline{P}$(P101.D)

Example Inference (5)

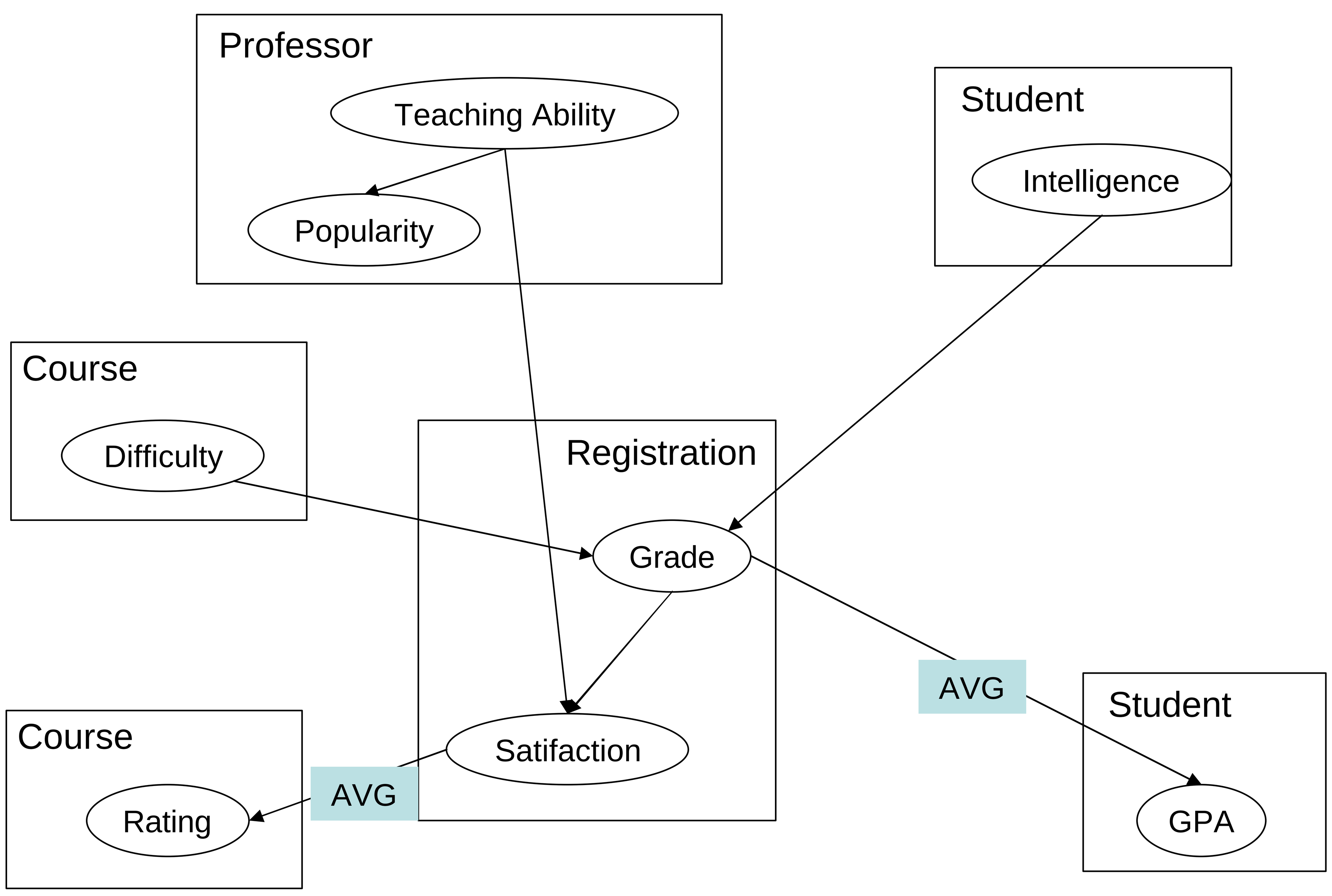


Can we be sure that the instantiated PRM gives a DAG?

- Case 1: Check at the skeleton level

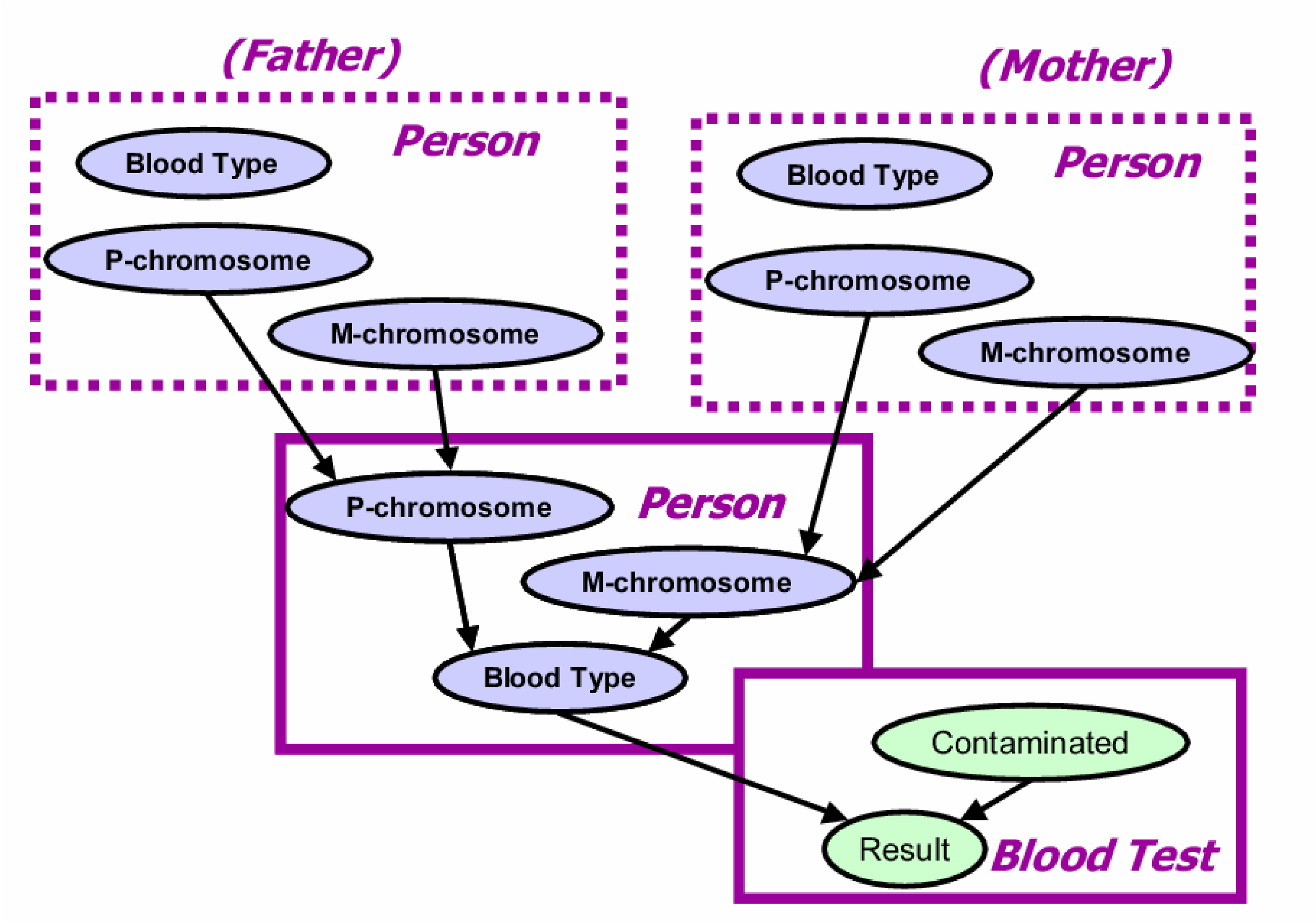


The graph is a DAG at the skeleton level



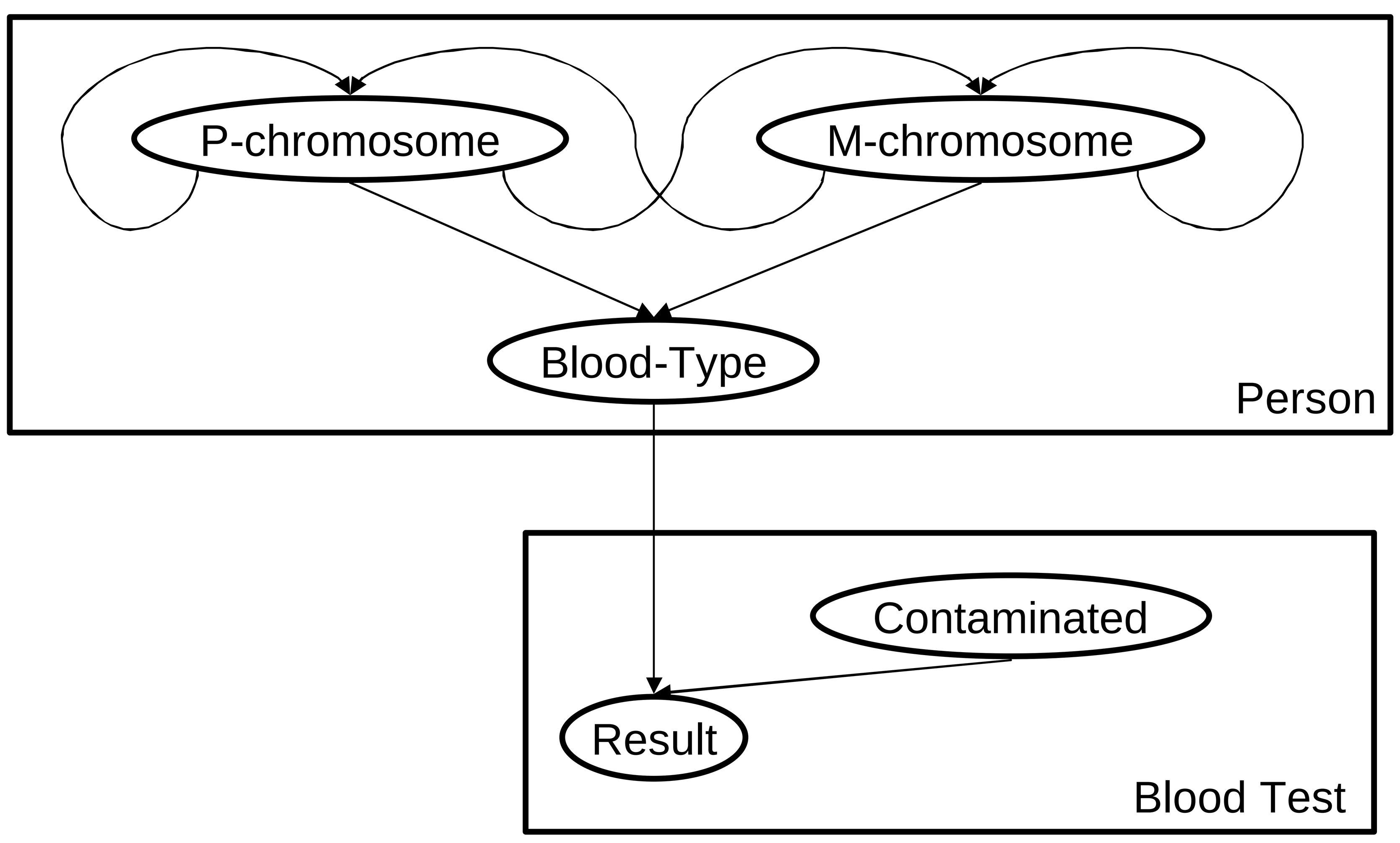
Case 2: Skeleton graph contains cycles, but instantiated graph does not

Blood type depends on chromosomes inherited from parents



parents(Person.M-chromosome)=
{Person.Mother.M-chromosome, Person.Mother.P-chromosome}

Case 2: Skeleton graph contains cycles, but instantiated graph does not



parents(Person.M-chromosome)=
{Person.Mother.M-chromosome, Person.Mother.P-chromosome}

PRM Semantics: PRM Skeleton

- Take database: keep Reference attributes, but replace all Descriptive attributes by random variables
- PRM defines the joint distribution of these random variables

PRM Skeleton:
??? denotes random variable

Professor	Popularity	Teaching-Ability
Gump	???	???

Student	Intelligence	GPA
Gomer Pyle	???	???
Jane Doe	???	???

Course	Professor	Difficulty	Rating
Phil101	Gump	???	???
Com301	Gump	???	???

Registration	Course	Student	Grade	Satisfaction
Reg123	Com301	Gomer Pyle	???	???
Reg333	Phil101	Jane Doe	???	???
Reg135	Com301	Jane Doe	???	???

PRM Semantics (2)

- The PRM does not provide a probabilistic model over the reference attributes (i.e., over the "link structure") of the database
- The PRM does not provide a model of all possible databases involving these relations. It does not model, for example, the number and nature of the courses that a student takes or the number of classes that a professor teaches.

Learning

- Known Skeleton, Fully Observed
  - Constrain corresponding CPT's to have the same parameters

$$P(\text{Reg.Grade} = A | \text{Course.Diff} = high, \text{Student.Int} = low) = \frac{N(\text{Reg.Grade} = A, \text{Course.Diff} = High, \text{Student.Int} = low)}{N(\text{Course.Diff} = high, \text{Student.Int} = low)}$$

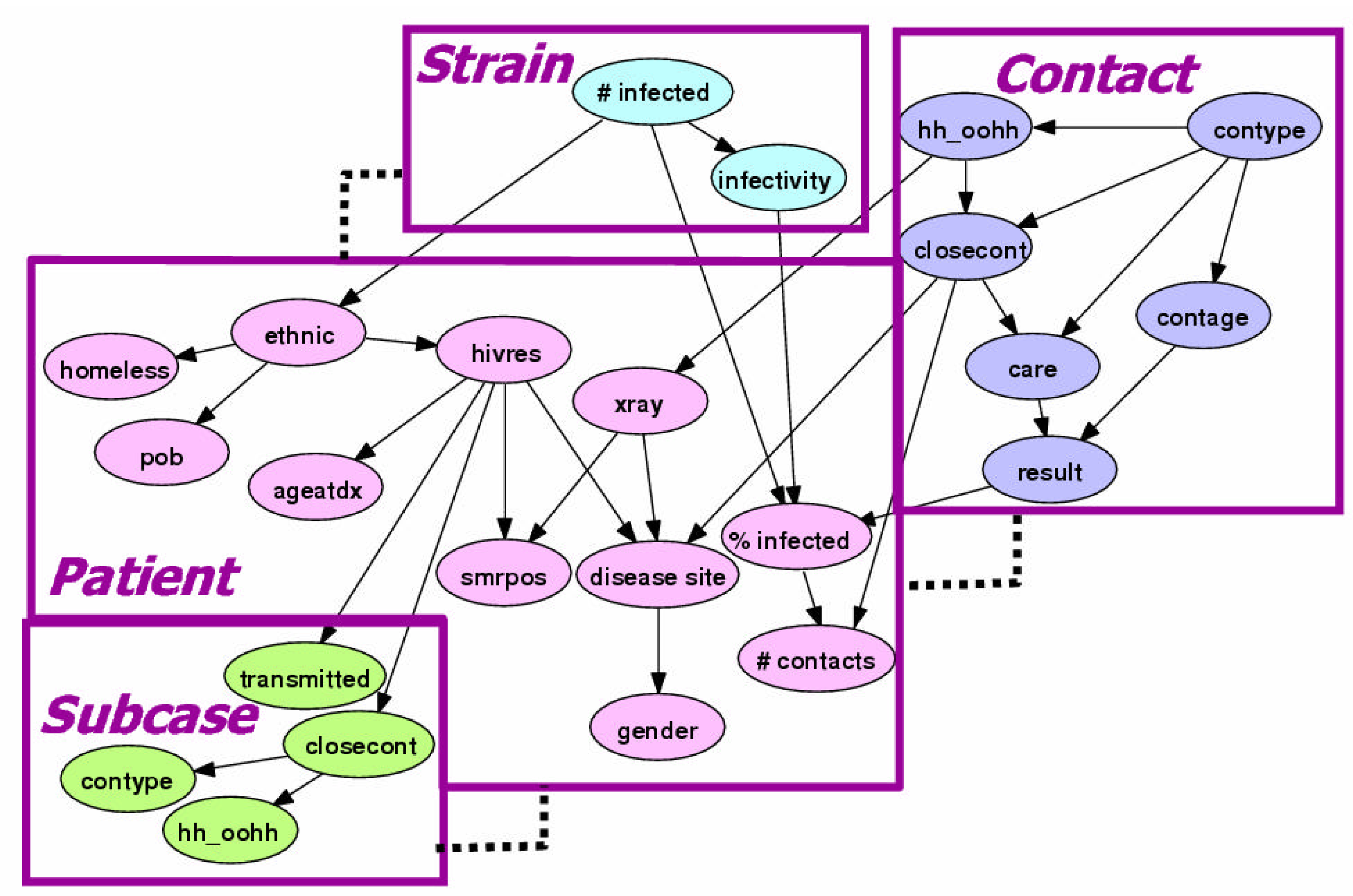
Learning the Structure

- Case 1: We know how individual objects are connected and we just need to learn the parents of each attribute
- Case 2: We need to learn how objects are connected as well as learning the parents of each attribute. This is the subject of our next paper.

Case 1: Learning the parents of each attribute

- Search in the space of path expressions and aggregators
  - infinite space!
  - impose some complexity limits?

Application: Tuberculosis



Application: Banking