Graphical Models for Inference and Decision Making

Unit 3: Representing Knowledge in an Uncertain World

Instructor: Kathryn Blackmond Laskey

Spring 2019

Learning Objectives

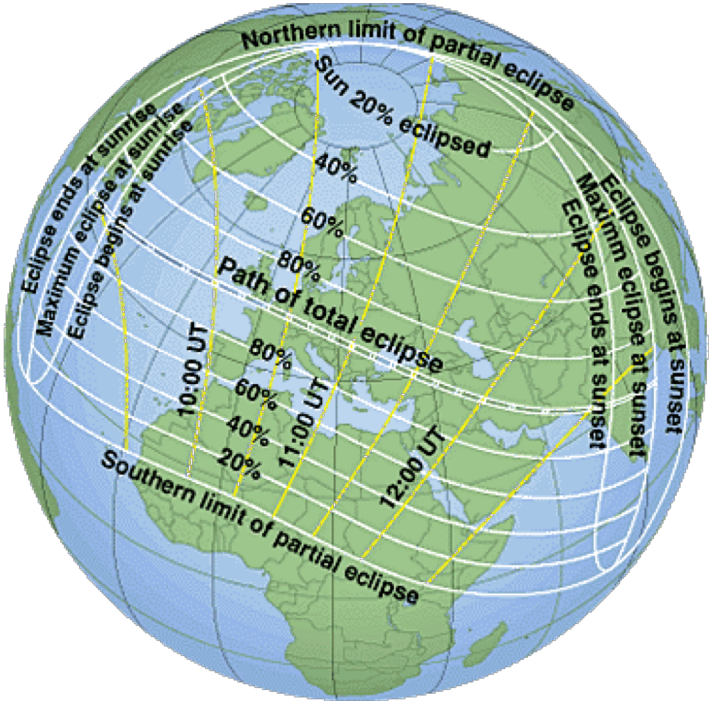
- Describe
  - The elements of a knowledge representation
  - The difference between propositional and first-order logics
  - Why first-order logics are important for intelligent reasoning
- Name and describe some major expressive languages for probabilistic reasoning
  - OOBNs; PRMs; MEBN; RBN, MLN
- Define an ontology and a probabilistic ontology
- Define a Multi-Entity Bayesian Network
  - Reusable model components
  - Basis for knowledge-based model construction
- Define a situation-specific Bayesian network.
- Given a simple MEBN model and a query, construct a situation-specific BN to respond to the query

Unit 3 Outline

- Knowledge Representation
- Logic and Ontologies
- Expressive Probabilistic Languages
- Probabilistic Ontology

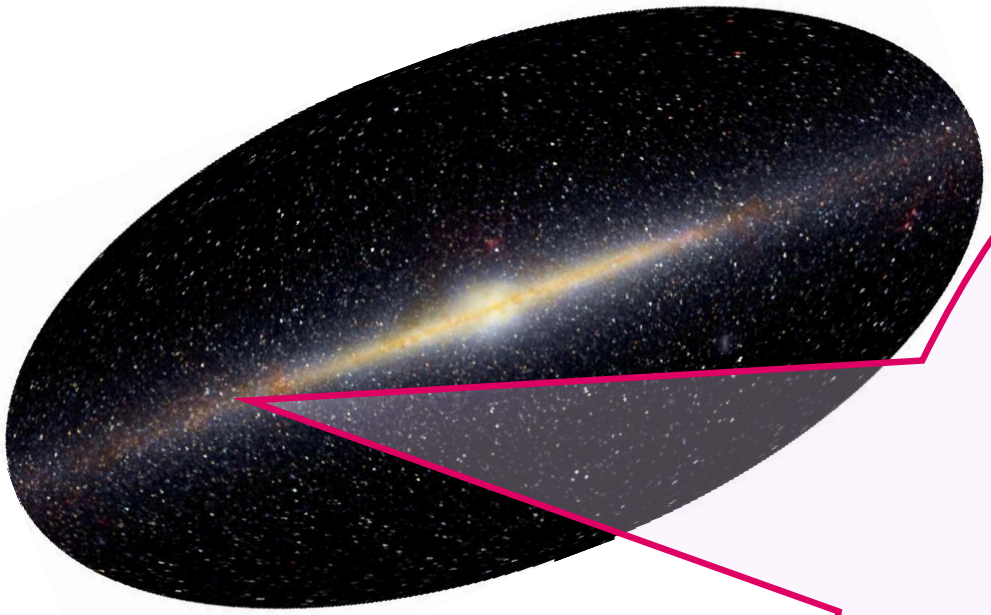
Observations

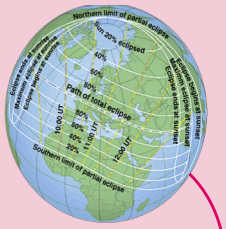
Real World



Representation

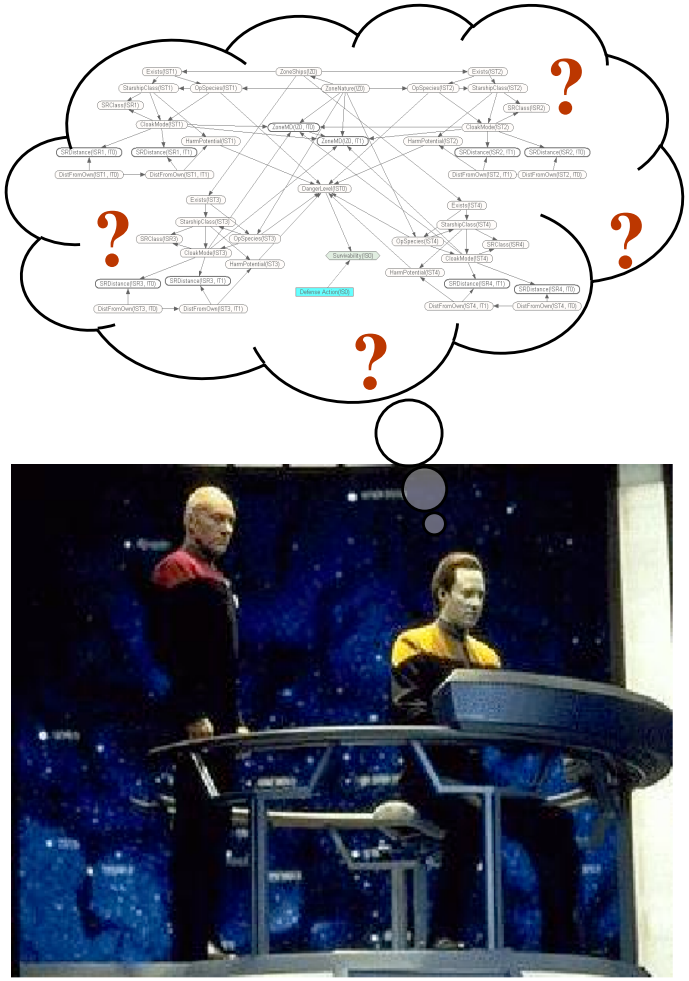
Actions




Northern limit of partial eclipse
20% eclipsed
40%
60%
80%
Path of total eclipse
80%
60%
40%
20%
10:00 UT
11:00 UT
12:00 UT
Southern limit of partial eclipse
Eclipse ends at sunset
Maximum eclipse at sunset
Eclipse begins at sunrise
Eclipse ends at sunset
Maximum eclipse at sunrise
Eclipse begins at sunrise

Representing The World

- Representation requires:
  - A representing system
  - A represented system
  - A correspondence between the representing system and the represented system
- Intelligent systems use representations to respond intelligently to their environments
  - Sense
  - Recognize
  - Plan and act
- A good representation should:
  - Contain features corresponding to important properties of the represented system
  - Capture how structure & behavior of represented system gives rise to observable evidence
  - Improve with experience
  - Rest on a mathematically sound and scientifically principled logical foundation



http://www.oz-q.com/brent/pic/

Knowledge Representation
(Davis, 1993)
http://groups.csail.mit.edu/medg/ftp/psz/k-rep.html

- A knowledge representation is a surrogate
  - Cannot store physical objects and processes in a computer
  - Symbols and links form model of an external system
    - » Variables serve as surrogates for entities they designate
    - » Variables are transformed to simulate behavior of system
- A knowledge representation is a set of ontological commitments
  - Ontology determines categories of things that can exist in the model
- A knowledge representation is a fragmentary theory of intelligent reasoning
  - Describes things, behavior, interactions
  - Declarative: stated as explicit axioms & allowable transformations
  - Procedural: compiled into executable programs
- A knowledge representation is a medium for efficient computation
  - Must encode knowledge in form that can be processed efficiently
- A knowledge representation is a medium of human expression
  - Vehicle of communication between knowledge engineers and domain experts

Components of Computational Representation

- Vocabulary
  - Elementary components of expressions
  - Variables, constants, connectives, functions
- Syntax
  - Rules for composing legal expressions
  - Organization into higher level structures or patterns
    - » Frames
    - » Objects
    - » Scripts
- Reasoning methods
  - Rules for deriving expressions from other expressions
  - Corresponds to *operational semantics* of computer language
- Semantics - characterizes meaning of expressions
  - Ontology or theory of reference (*denotational semantics*)
  - Theory of truth (*axiomatic semantics*)

Advantages of a Good Knowledge Representation

- Knowledge is different from data or information
  - *Data*: symbols or signals denoting facts or observations about the world; must be processed to be meaningful
  - *Information*: stimulus that has meaning in some context to its receiver
  - *Knowledge*: Expertise and skills acquired through experience or education; the theoretical or practical understanding of a subject.
- Problems that are difficult in one representation become easy in another
  - Find MCMXLIV – (CDXLIV + MD)
  - Find 1944 – (444 + 1500)
- Formal knowledge representation represents *semantics* of a domain
  - Knowledge structures reflect structure of the domain
  - Facilitates maintenance and reuse
  - Supports sharing and semantic interoperability
  - Supports efficient reasoning

Examples of Knowledge Representation Formalisms

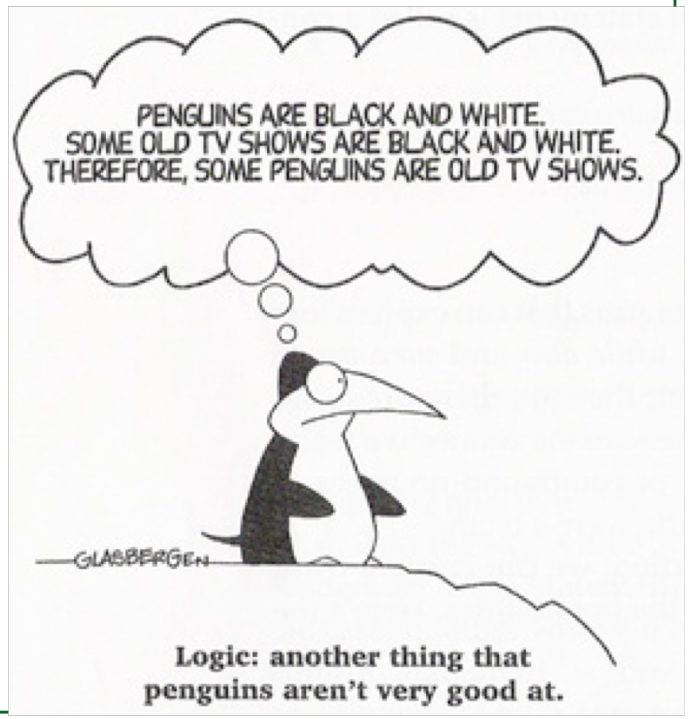
- Logical representations
  - Well-defined vocabulary, syntax with clear, mathematically precise semantics, e.g.
    - » Predicate logic: propositional, first-order, higher-order
    - » Fuzzy logic, modal logic, multiple-valued logic
  - Representation languages have been formalized after the fact as logics
- Semantic networks
  - Nodes represent concepts, linked by relationships, inference traces links
- Production rules
  - Knowledge is organized as a set of if-then rules
  - Inference by forward chaining, backward chaining, or mixed
- Frame systems
  - Precursor to object-oriented computer languages
  - Frame represents a kind of entity; slots represent attributes; can make instances of frames
  - Slots can have associated production rules

Unit 3 Outline

- Knowledge Representation
- Logic and Ontologies
- Expressive Probabilistic Languages
- Probabilistic Ontology

Logic

- Logic is the study of precise patterns of reasoning
  - Aristotle developed classical syllogisms
  - Boole developed Boolean logic (19th century)
  - Leibnitz formalized Aristotle's syllogisms (17th century)
  - Frege and Pierce developed first-order logic (late 19th century)
  - Undecidability results, higher order logics, modal logics, (20th century)
- Russell and Norvig (2009) define a logic as:
  - A formal language for expressing knowledge
    - » Precisely defined syntax and semantics
  - A means of carrying out reasoning
    - » Precisely defined reasoning processes that correspond to semantics
- Logic concerns *structure* of sentences and proofs, not *content*
  - A valid syllogism:
    - » All querxsms are frjplantes
    - » morxengro is a querxsms
    - » Therefore, morxengro is a frjplantes
  - Logic formalizes reasoning so it can be performed mechanistically



Logic: another thing that penguins aren't very good at.

Propositional Logic

- Studies logical relationships among *propositions*
  - A proposition is a declarative statement
  - Complex propositions are built up from elementary propositions using logical connectives
  - Reasoning derives truth-value of conclusions from truth-values of premises
- Not expressive enough for most real-world problems
  - Cannot express generalizations
  - Elementary propositions are indivisible units with no inner structure
- But useful as a starting point

p	q	$p \& q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

Example: Vehicle Identification

Elementary Propositions:

- » K (Tracked vehicle)
- » R (On road)
- » F (Traveling fast)

Axioms:

- » $\neg K \rightarrow R$ (Wheeled vehicle cannot go off-road)
- » $K \rightarrow \neg F$ (Tracked vehicle cannot be traveling fast)

To reason about more than one vehicle, we need to replicate the propositions and axioms "by hand":

$\neg K_i \rightarrow R_i$ *and* $K_i \rightarrow \neg F_i$ *for* i=1, …, N

Possible Worlds

- Axioms define a set of "possible worlds" consistent with axioms
  - Worlds with tracked vehicle traveling fast and wheeled vehicle off-road are impossible
- A truth table uses truth-values of the elementary propositions to determine which worlds are possible

K	R	F	¬K→R	K→¬F
~~T~~	~~T~~	~~T~~	~~T~~	~~F~~
T	T	F	T	T
~~T~~	~~F~~	~~T~~	~~T~~	~~F~~
T	F	F	T	T
F	T	T	T	T
F	T	F	T	T
~~F~~	~~F~~	~~T~~	~~F~~	~~T~~
~~F~~	~~F~~	~~F~~	~~F~~	~~T~~

Expressive Power

- *Propositional* logic can express particular facts but not generalizations
  - Language has statements but no variables
  - We can say "If V23 is wheeled then V23 cannot go off-road"
  - We cannot say "Wheeled vehicles cannot go off-road"
- *First-order* logic can express generalizations about objects in the domain of application
  - Variables can refer to objects
  - Quantifiers can express generalizations about objects in the domain and state the existence of objects having given properties
    - » For all numbers *n* and *m*, *n*+*m* is equal to *m*+*n*
    - » There is a fire station in every town
- *Higher-order* logic can express generalizations about sets of objects in the domain, functions defined on the domain, or properties of objects
  - Some things can be said in higher-order logic that cannot be said in first-order logic (such as the full principle of mathematical induction)
- *Modal* logics can reason not just about truth and falsehood, but also about necessity, possibility, desirability, permissibility, and other non truth-functional attributes of propositions
  - Modal logics are also strictly more expressive than first-order logic

First-Order Logic

- Extends expressive power of propositional logic
  - Propositions have inner structure
  - Can express generalizations
    - » There is an air defense site next to every airport
    - » No wheeled vehicle can travel off-road
- First-order logic is to propositional logic as algebra is to arithmetic
- Most ontology languages are based on some fragment of first-order logic
  - Need to be able to express generalizations
  - Full first-order logic is undecidable
    - » There are statements that can be neither proven nor disproven

Classical First-Order Logic

- Vocabulary:
  - Constants (stand for particular named objects)
  - Variables (stand for generic unnamed objects)
  - Functions (represent attributes of objects or sets of objects)
    - » *Location*(*x*); *MotherOf*(*y*); *TeacherOf*(*c*,*s*)
  - Predicates (represent hypotheses that can be true or false; also called relations)
    - » *Guilty*(*s*)
    - » *Near*(*John*,*GroceryStore32*)
  - Connectives
    - » Quantifiers, conjunction, disjunction, implication, negation, equality
- Syntax:
  - Atomic sentences
  - Composition rules for forming compound sentences from atomic sentences
- Semantics
  - Possible worlds are abstract structures that specify truth-values of sentences
  - A sentence is *valid* if it is true in all possible worlds
  - A sentence follows logically from a set of axioms if it is true in every possible world in which the axioms are true
- Proof rules
  - Natural deduction
  - Resolution with refutation

Vehicles Revisited: FOL Version

- Propositions have inner structure

  V(x) : x is a vehicle K(x) : x is tracked

  L(x) : location of x F(x) : x is traveling fast

  R(x) : x is a road

- Can represent:
  - Different types of entity, e.g., vehicles and roads
  - Properties of entities, e.g., being tracked; traveling fast
  - Relationships among entities
  - Functional relationships, e.g., location of object
  - Rules that apply to all entities of a given type, e.g.:
    - » $\forall x\ V(x) \rightarrow K(x) \rightarrow \neg F(x)$
    - » $\forall x\ V(x) \rightarrow \neg K(x) \rightarrow R(L(x))$
  - Particular individual entities, e.g., O_3, O_7
  - Equality, e.g., $O_3 = L(O_7)$

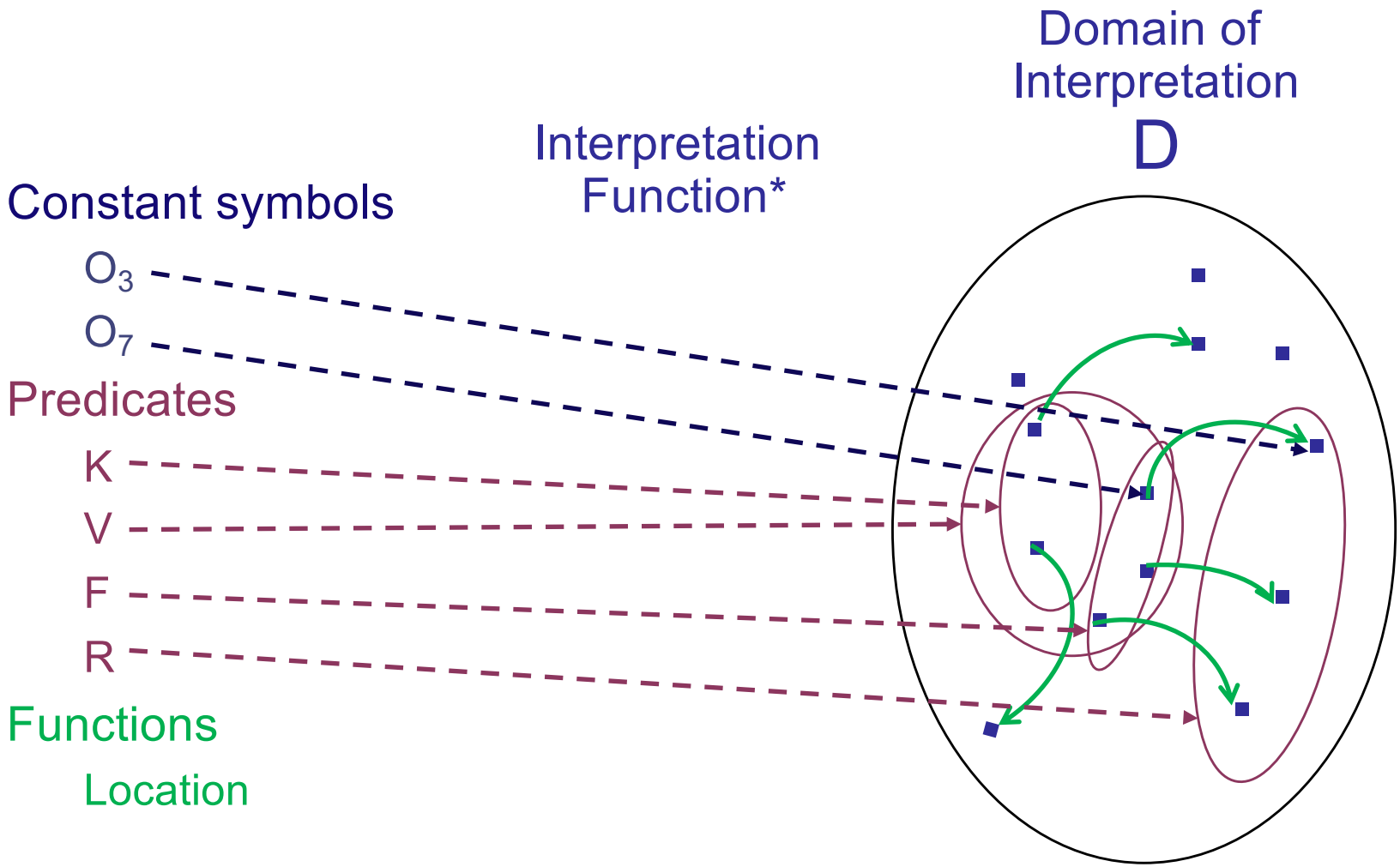
Privileged Status of FOL

- Has been proposed as unifying language for
  - Defining extended logics
  - Interchanging knowledge
- Many common KR systems have theoretical basis in FOL
- Common Logic issued as ISO standard October 2007
  - Family of first-order logic syntaxes sharing common semantics
  - Designed for knowledge interchange
- Issues:
  - Cannot express generalizations about sets, predicates, functions
  - Full first-order logic is undecidable
    - » Some statements can be neither proven nor disproven
    - » Whether arbitrary statement is provable is undecidable
- No built-in structures for
  - Categories
  - Time and space
  - Causality
  - Action
  - Events
  - Value / utility
  - *Plausibility / probability*

First-Order Possible Worlds: Model Theoretic Semantics

- An *interpretation* of a first-order vocabulary consists of:
  - A set D called the *domain*
    - » *e.g.. vehicles, roads, and possibly other things*
  - An element of D for each constant symbol
    - » *e.g., a road for O_3 and a vehicle for O_7*
  - A relation on D for each predicate (relation) symbol
    - » *e.g., the set of objects which are roads for R(x)*
  - A function taking arguments in D and having value in D for each function symbol
    - » *e.g., a function mapping each object to another object for L(x)*
- A *model* for a set of axioms is an interpretation in which all the axioms are true
  - Interpretations are worlds; models are possible worlds
  - (The word "model" has a different meaning in logic than in engineering)

Model Theory: Illustration



*Note: Interpretation of L(x) is shown only for objects that are vehicles

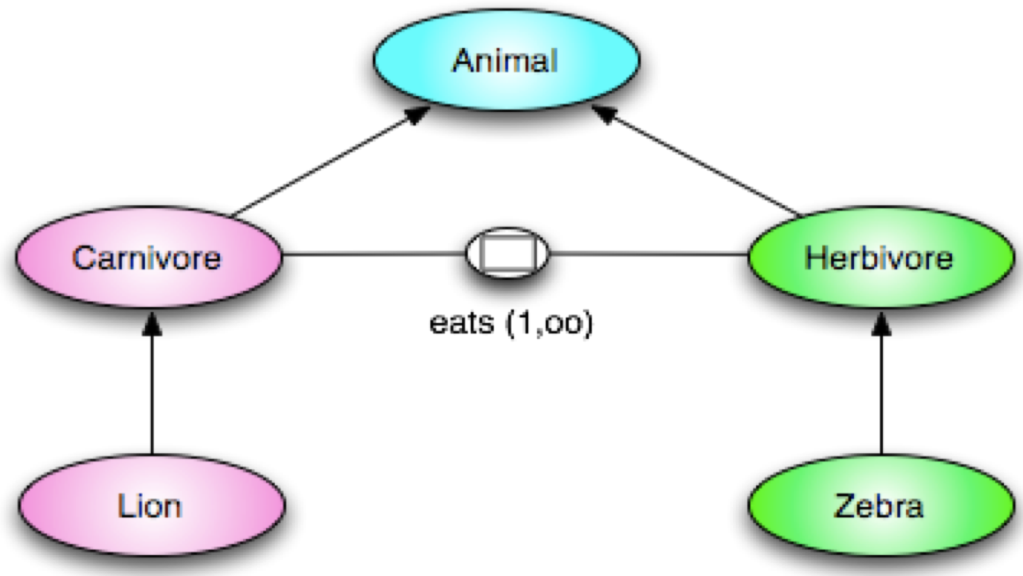
Ontology

- Ontology connects formal language to things in the world
  - Categories things that can exist in a domain
  - Organized hierarchically into types / subtypes
  - Objects of a given type have:
    - » Similar structure (attributes)
    - » Similar relationships
    - » Similar behavior (processes)
- Ontology for a first-order language represents:
  - Allowable predicates and functions
  - Types of entities variable symbols can refer to
  - Entities denoted by constant symbols
- Specifying an ontology:
  - Formal - defined by logical rules
  - Informal - specified via prototypical instances
- Explicit computational specification of ontology facilitates interoperability and reuse

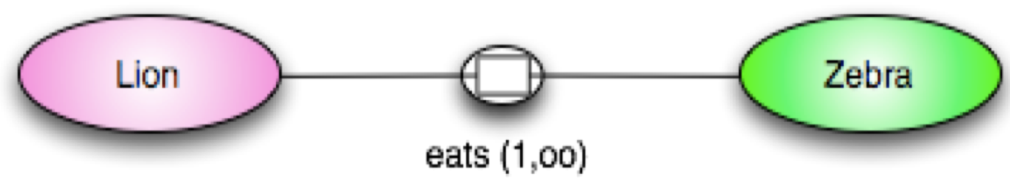
Ontology Supports Logical Reasoning

Concepts
animal, carnivore, herbivore

Relationships
carnivore **is-a** animal
herbivore **is-a** animal
carnivore **eats** herbivore
lion **is-a** carnivore
zebra **is-a** herbivore



Reasoning
lion eats zebra



Logical Reasoning

may be inadequate

- Our knowledge base:
  - Birds lay eggs
  - Aquatic birds can swim
  - Aquatic birds can hold their breath
  - Aquatic birds have duck-like bills
  - Aquatic birds have webbed feet
- The problem-specific data:
  - Pamela lays eggs
  - Pamela can swim
  - Pamela can hold her breath
  - Pamela has a duck-like bill
  - Pamela has webbed feet
- Pamela is a…

Mammal !

Duck-billed platypus, an egg-laying mammal

Ontology versus DB Schema

- DB schema and ontology both represent:
  - Types of entities that can exist
  - Relationships that can hold among entities
- Purpose and usage is different
  - DB schema represents structure of data in a particular data store and is typically not designed for interoperability
  - Ontology is intended to represent knowledge about a domain in a structured, shareable way

Unit 3 Outline

- Knowledge Representation
- Logic and Ontologies
- Expressive Probabilistic Languages
- Probabilistic Ontology

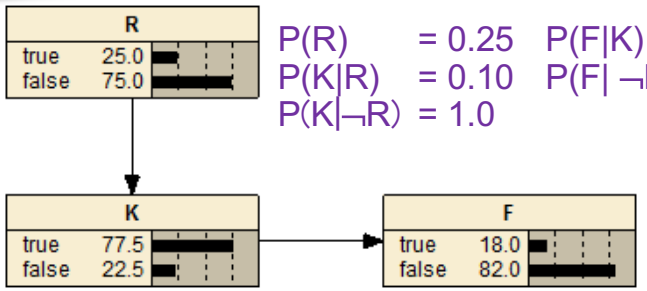
Possible and Probable Worlds

- A classical logic KB consists of sentences called *axioms*
- The axioms implicitly define a set of *possible worlds*
- To reason about a given problem:
  - We supply some additional sentences to represent problem-specific facts
  - We pose a *query* to infer the truth-value of a sentence of interest
- In classical logic the possible results are:
  - We may find a proof that the query sentence is true;
  - We may find a proof that the query sentence is false;
  - No proof may exist (either truth-value may be consistent with the axioms)
- Classical logic cannot assign plausibility to statements if they can be neither proven true nor proven false
- A probabilistic logic assigns probabilities to possible worlds
  - Can respond to a query with a probability even if truth-value for sentence is not determined by the axioms
  - Provable sentences have probability 1; sentence that contradict the axioms have probability 0

Possible and Probable Worlds

- Propositional logic can be extended to incorporate uncertainty by assigning a probability to each possible world

K	R	F	¬K→R	K→¬F	F∨R	Prob
~~T~~	~~T~~	~~T~~	~~T~~	~~F~~	T	0
T	T	F	T	T	T	2.5%
~~T~~	~~F~~	~~T~~	~~T~~	~~F~~	T	0
T	F	F	T	T	F	75.0%
F	T	T	T	T	T	18.0%
F	T	F	T	T	T	4.5%
~~F~~	~~F~~	~~T~~	~~F~~	~~T~~	T	0
~~F~~	~~F~~	~~F~~	~~F~~	~~T~~	F	0



P(R) = 0.25 P(F|K) = 0.0
P(K|R) = 0.10 P(F| ¬K) = 0.8
P(K|¬R) = 1.0

Total Probability = 100%

Pr(F∨R) = 2.5% + 18% + 4.5% = 25%

Probable Worlds: Incorporating Evidence

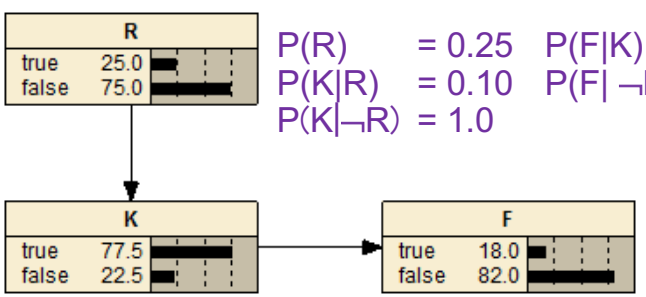
K	R	F	¬K→R	K→¬F	Prob
T	T	T	T	F	0
T	T	F	T	T	3.05%
T	F	T	T	F	0
T	F	F	T	T	91.46%
F	T	T	T	T	0
F	T	F	T	T	5.49%
F	F	T	F	T	0
F	F	F	F	T	0

Pr(K)=77.5%

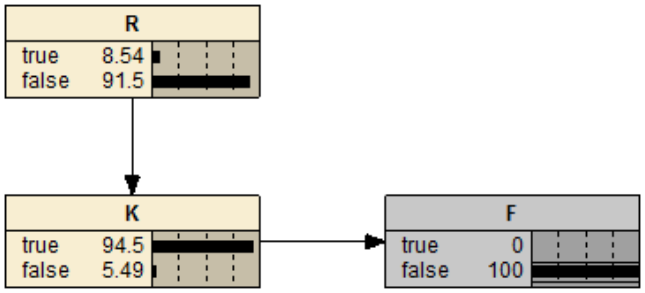
Pr(K|¬F) = 94.5%

We learn vehicle is not traveling fast

BN expresses uncertainty over possible worlds for a propositional logic

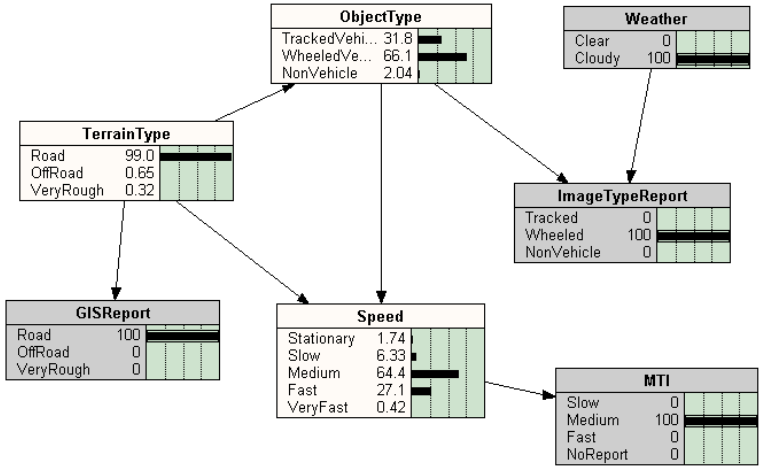


P(R) = 0.25 P(F|K) = 0.0
P(K|R) = 0.10 P(F| ¬K) = 0.8
P(K|¬R) = 1.0

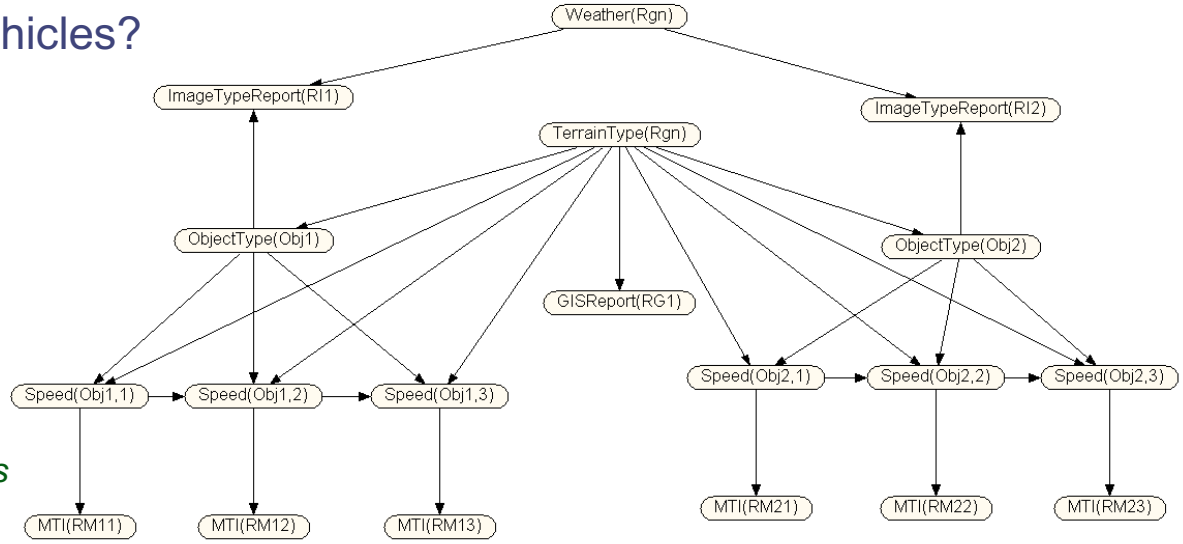


The Trouble with BNs

- Standard Bayesian networks:
  - Designed for a single instance
    - » e.g., single vehicle at a single time
- Many problems require greater representation power:
  - Multiple entities of interest
    - » Many vehicles
    - » Many reports coming in at different times
  - Entities are related
    - » Which vehicles go with which reports?
    - » Are there unreported vehicles? Spurious reports?
  - Situation evolves in time
    - » Vehicles move
    - » New reports arrive
- Greater expressive power is needed



Single object, single time step



Two objects, three time steps

Extending Expressiveness of Graphical Models

- Standard graphical probability model includes random variables and dependence relationships expressed as conditional probability statements
  - Pr(A | B, C, D) = belief_table
  - Standard probabilistic graphical models assign probabilities to statements in a propositional logic
- First-order Bayesian logic assigns probabilities to statements in a more expressive logic:
  - Variables and constants represent entities in domain
  - Predicates and functions represent attributes and relationships
  - Specify probability distribution over values of predicates and functions, e.g.
    - » (∀x) [Pr(A(x) | B(x), C(x), D(x)) = belief_table]
    - » B(R17) = b2
  - A probabilistic KB implicitly represents probabilities for possible worlds
  - Rules of inference
    - » Probability calculus
    - » Derive probability statements from other probability statements

History of Probabilistic KR

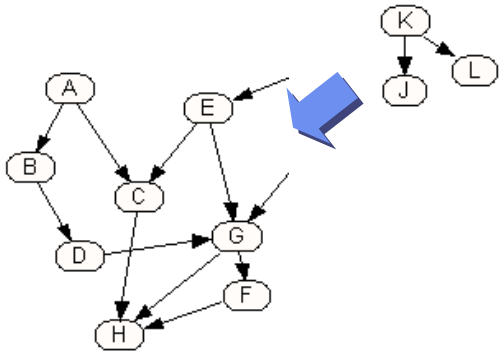
- Mid to late 80's: Heuristic knowledge-based model construction (KBMC)
  - Levitt, Binford, Ettinger applied KBMC to computer vision and automated target recognition
  - Breese PhD dissertation on rule-based KBMC
  - Goldman PhD dissertation on KBMC for natural language understanding
- 1990's: Probabilistic logics
  - Charniak & Goldman (1993) was the first true first-order probabilistic logic based on BN fragments
  - Halpern, Bacchus & collaborators - theoretical work in first-order probabilistic representations
- 2000 and beyond
  - Convergence between BN, database, and logic programming communities
  - Intensive development of probabilistic languages with first-order expressive power
  - Probabilistic programming: unifying general-purpose programming with probabilistic modeling (http://probabilistic-programming.org)
  - Software support is growing

Expressive Probabilistic Languages

- Using different metaphors
  - Object-oriented Bayesian networks
  - Probabilistic relational models
  - Multi-entity Bayesian networks
  - Markov Logic Networks
  - Bayesian Logic Programs
  - Plates
  - … and many more
- They appeal to people from different communities
- All attempt to combine expressive language with ability to represent and manipulate uncertainty

Knowledge-Based Model Construction

- Represent probabilistic knowledge in an expressive base representation
  - Represent knowledge as fragments of a Bayesian network
  - Fragments capture stable patterns of probabilistic interrelationships
  - Fragments can be replicated
- At problem solving time
  - Bring fragments together to construct a problem-specific Bayesian network
  - Use constructed model to process queries
- A KBMC system must contain at least the following elements:
  - A base representation that represents domain dependencies, constraints, etc.
  - A model construction procedure that maps a context and/or query into a target model
- Advantages of more expressive representation
  - Understandability
  - Maintainability
  - Knowledge reuse (agile modeling)
  - Exploit repeated structure



(Breese, Goldman and Wellman, 1992)

Object-Oriented Knowledge Representation

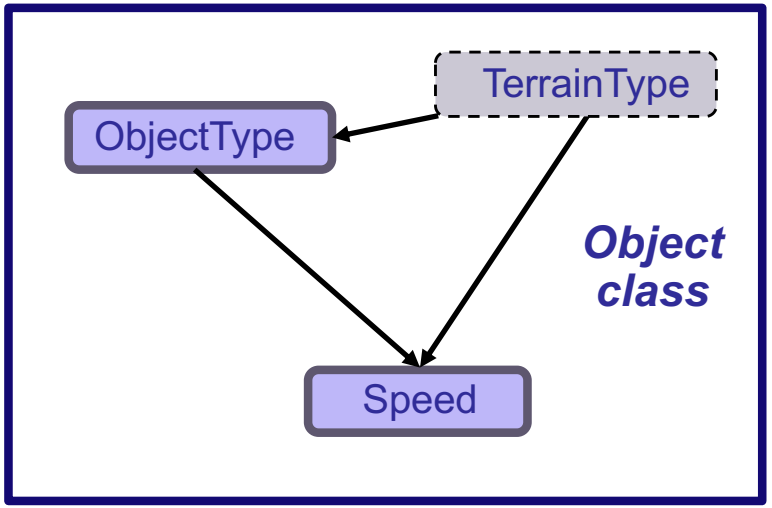
- An object-oriented representation
  - Represents abstract data types
    - » *Classes* represent types of object
    - » *Instances* represent individual objects in a class
    - » Objects of a given type share structure and behavior
    - » Subclasses inherit structure and behavior from superclasses
  - Encapsulation
    - » Public structure and methods are "seen" externally
    - » Private structure and methods are available only to the object itself
  - Facilitates knowledge base development, maintenance, reuse
    - » Modularity and defined interfaces

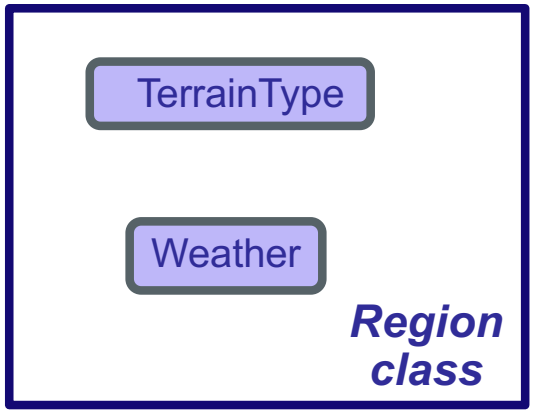
Object-Oriented Bayesian Networks

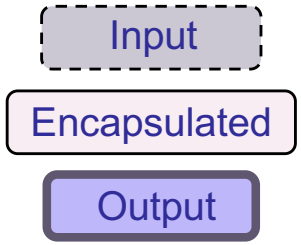
- Classes represent types of object
  - Attributes for a class are represented as OOBN nodes
  - *Input* nodes refer to instances of another class
  - *Output* nodes can be referred to by other classes
  - *Encapsulated* nodes are private
    - » Conditionally independent of other objects given input and output nodes
- Classes may have subclasses
  - Subclass inherits attributes from superclass
  - Subclass may have additional attributes not in superclass
- Classes may be *instantiated*
  - Instances represent particular members of the class

Hugin® has support for OOBNs

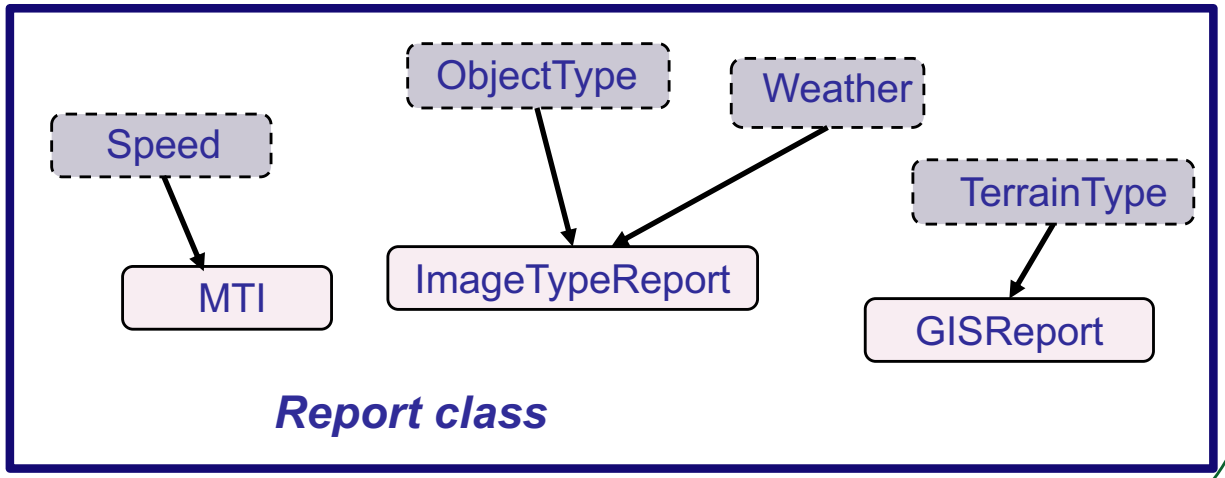
Vehicle ID Example: OOBN Representation







- *Can make multiple instances of classes*
- *Connect instances of output nodes to input nodes of other classes*
  - *Cannot express uncertainty about which output instance goes with which input instance*
- *Encapsulated nodes are not visible outside the object*



Example: OOBNs for Forensic Genetics

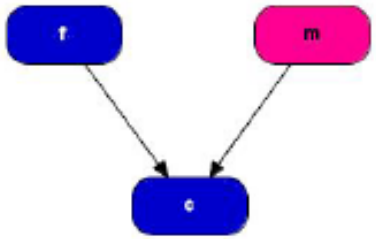


Figure 1: Pedigree for nuclear family

- f & m are instances of the class "founder" (no parents represented)
- c is an instance of class "child" (has parents represented)

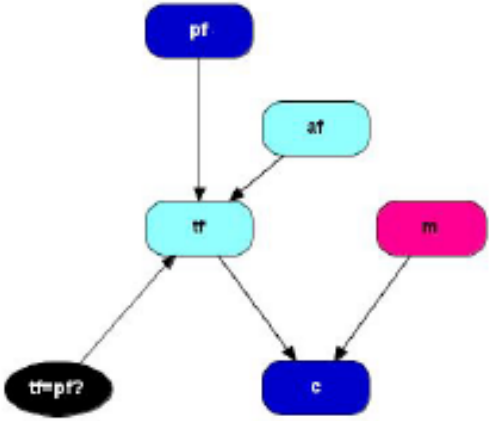


Figure 2: Pedigree for simple disputed paternity

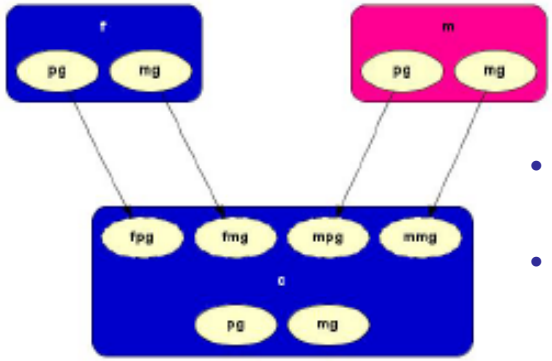


Figure 7: Expanded pedigree for nuclear family

- Input and output nodes are shown
- "founder" and "child" classes have additional hidden attributes shown at right

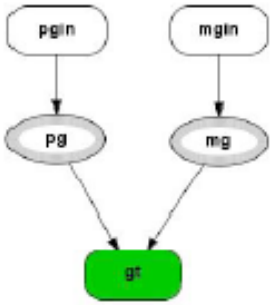


Figure 9: Network founder

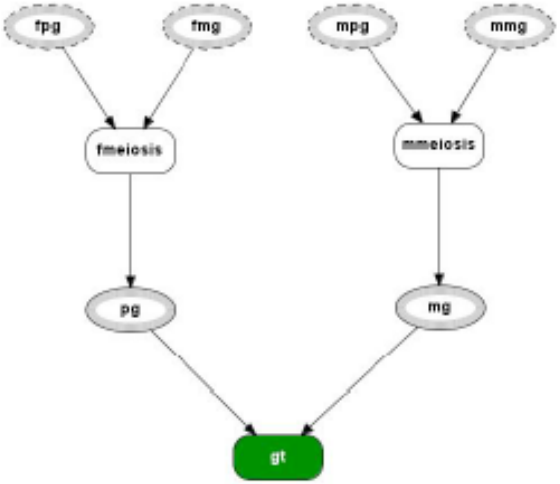


Figure 11: Network child

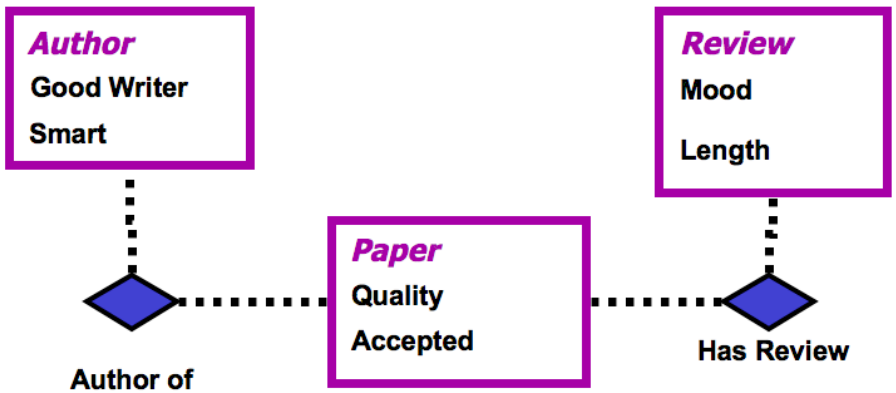
Reference: Dawid, et al. (2005)

Probabilistic Relational Models

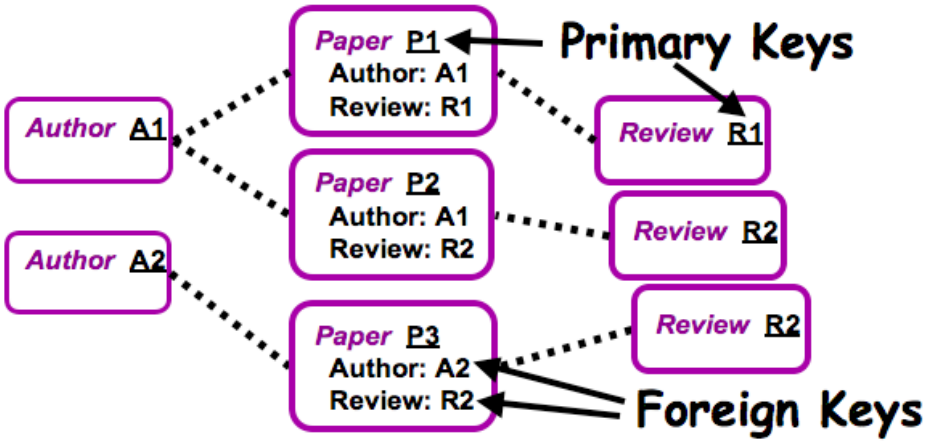
- Elements of PRM
  - Relational schema - Represents classes, attributes and relationships (corresponds to table in relational DB schema)
  - PRM structure - Represents probabilistic dependencies & numerical probabilities
  - Skeleton - Unique identifier and template for each instance
  - Data - Fills in values for instances
- An instance of a relational schema consists of a set of objects
  - Each object belongs to one of the classes
  - A value is specified for each descriptive attribute
  - An object of the appropriate type is specified for each reference attribute
- PRM structure represents probabilistic information
  - Allows representation of repeated structure
  - Can be viewed as a set of BN fragments
  - Can be learned from data

PRM Example: Publishing Papers

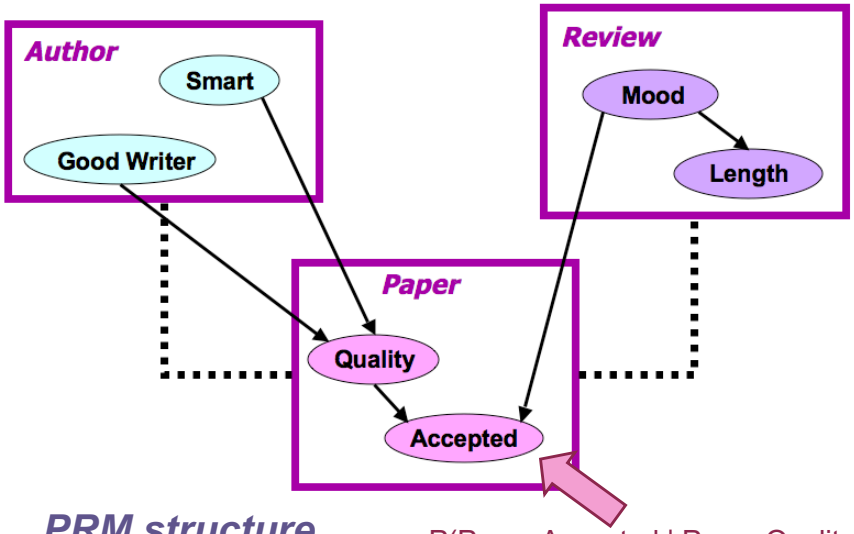
(Getoor and Pfeffer, 2005)



Relational schema



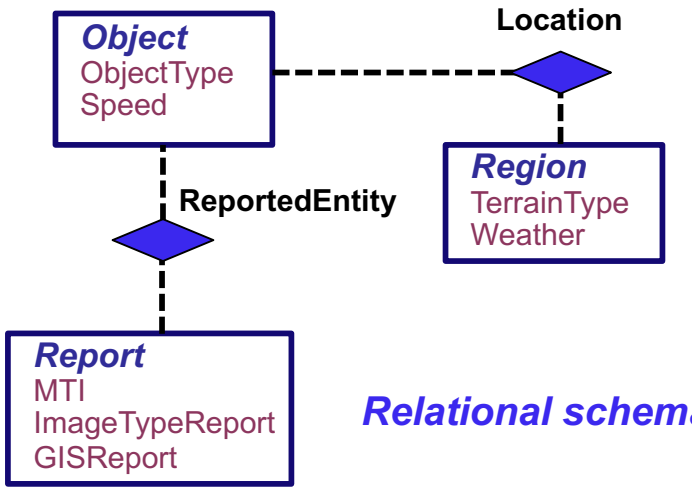
Relational skeleton



PRM structure

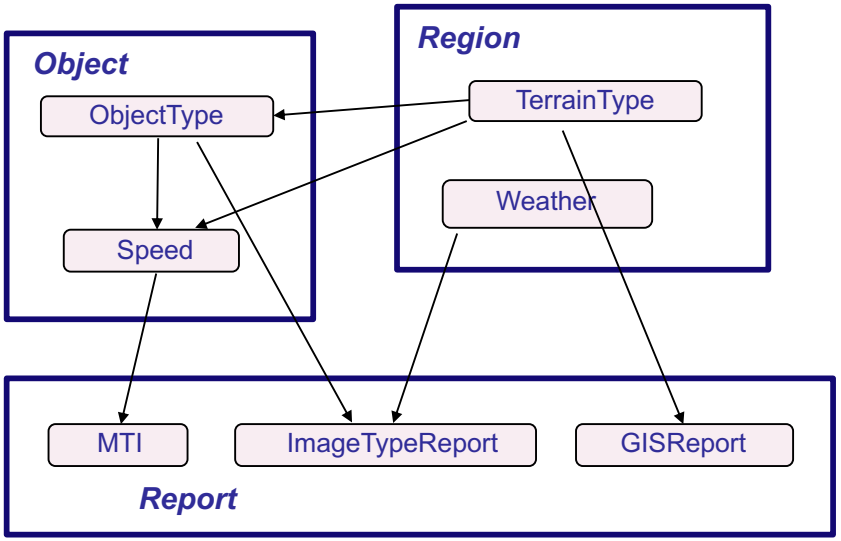
P(Paper.Accepted | Paper.Quality, Paper.Review.Mood)

Vehicle ID Example: PRM Representation

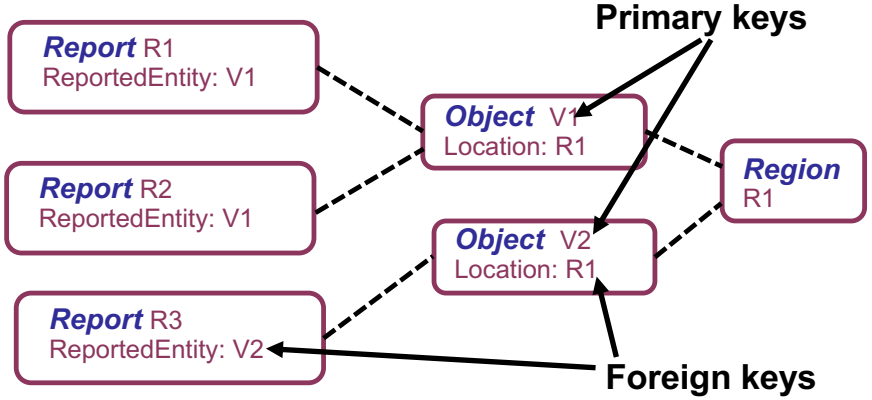


Relational schema

- Relational schema represents
  - Entities: objects, regions, reports
  - Relationships:
    - » ReportedEntity relationship between reports and objects
    - » Location relationship between objects and regions
- PRM structure represents uncertain relationships among attributes of objects, regions, reports
- Relational skeleton represents individual instances of entities



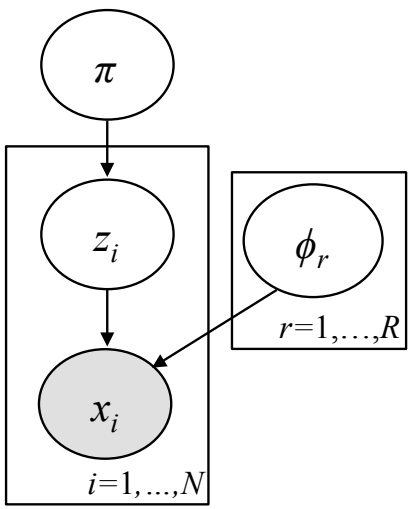
PRM structure



Relational skeleton

Plate Notation

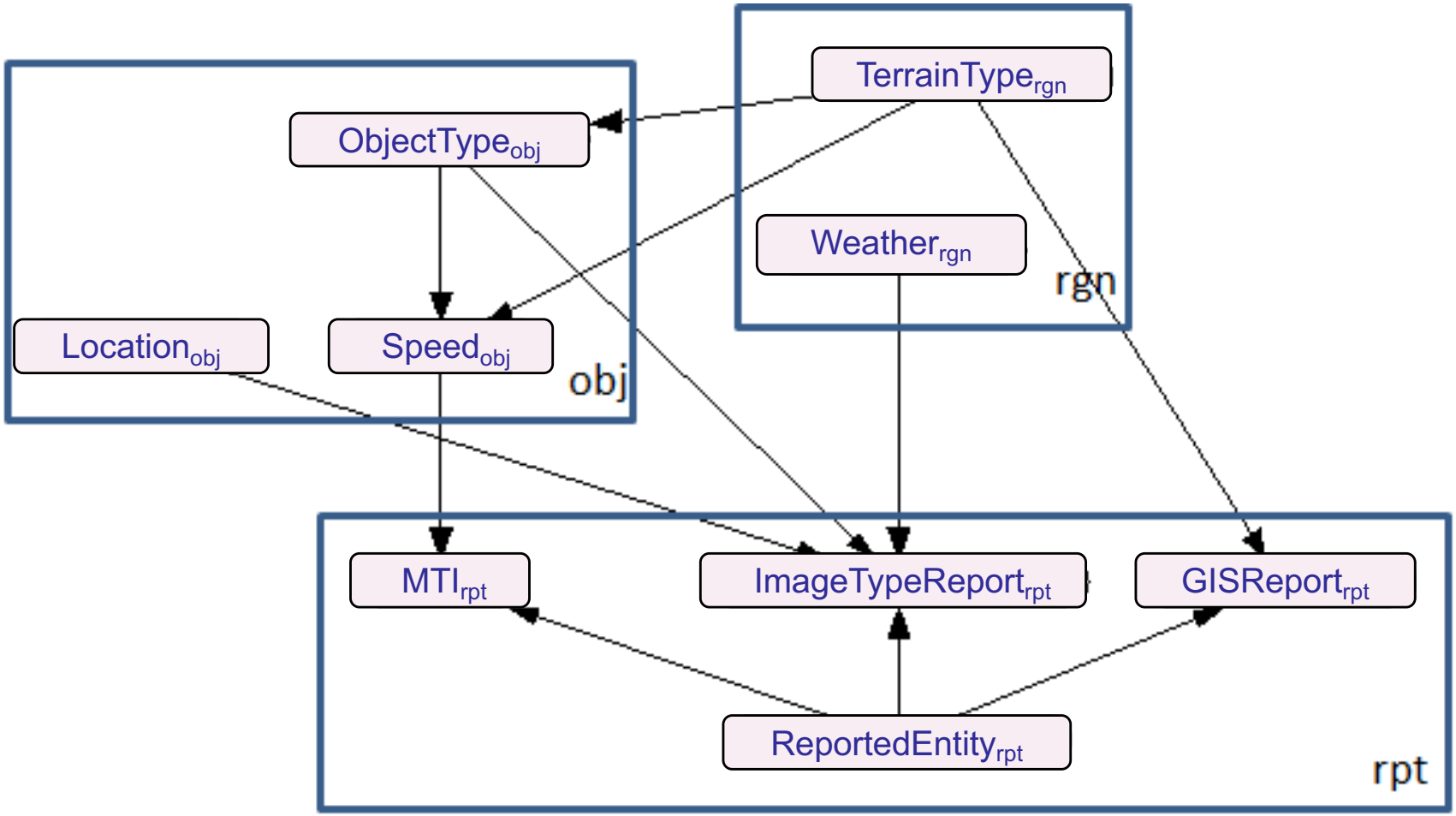
- Plates are a representation for repeated structure that arose in the statistical community and has become popular in machine learning
- Plates are a notational device but have not been formalized as a language with precise syntax and semantics
- A plate encloses a part of a graphical model that is repeated
  - Random variables in a plate have a plate-specific index
  - Index range is indicated in lower right corner of plate



$$\pi \sim Dirichlet(\alpha_1,\ldots\alpha_R)$$
$$\phi_r \sim h(\phi)$$
$$z_i \mid \pi \sim Multinomial(1,\pi)$$
$$x_i \mid z_i,\phi_{z_i} \sim f(x \mid \phi_{z_i})$$

Plate notation for a simple clustering model
- R clusters
- N observations $X_1, \ldots X_N$
- Z_i denotes cluster membership of X_i
- ϕ_r is a parameter for the rth cluster
- π is a vector of prior cluster probabilities

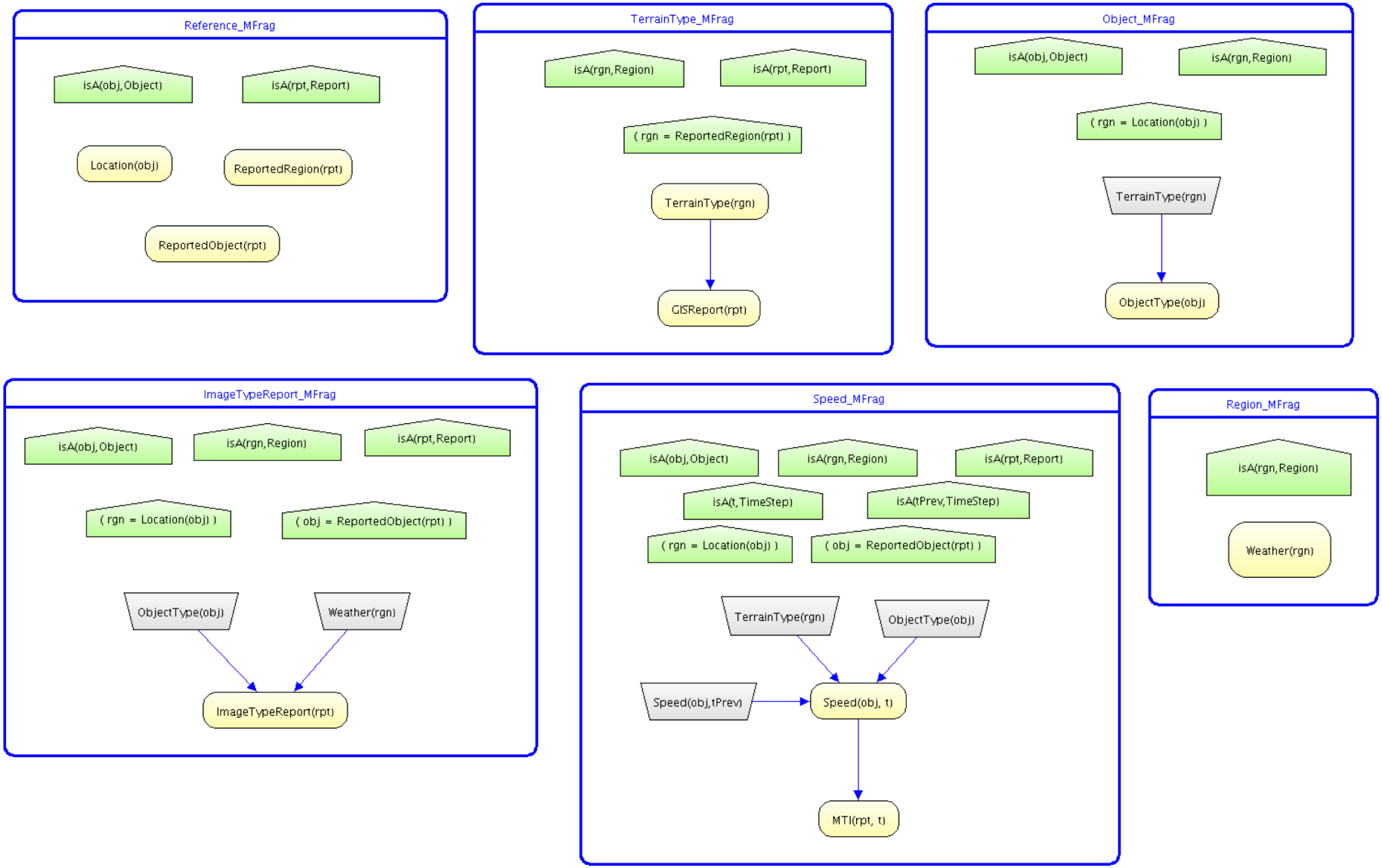
Vehicle ID Example: Plate Notation



Multi-Entity Bayesian Networks

- Syntax similar to first-order predicate calculus
- *MEBN fragments* represent probabilistic dependencies among related random variables
  - Random variable syntax is similar to first-order logic notation
    - » RVName(variable1, variable2, …, variablen)
    - » (Ordinary) variables are placeholders for entity instances
  - Inserting instance names for the ordinary variables creates an instance of the MFrag
  - There are built-in MEBN fragments for standard logical operators (and, or..) and quantifiers
  - *Influence combination* rules specify how influences from arbitrarily many parents are combined
- *MEBN theory* is a collection of MEBN fragments that satisfies global consistency conditions
- Inference: Situation-specific Bayesian network
  - Constructed from MEBN knowledge base
  - Contains instances of MEBN fragments
- Unlike OOBNs and PRMS, MEBN has native representation for n-ary relations

Vehicle ID Example: MEBN Representation



Implemented in UnBBayes-MEBN open-source PGM framework

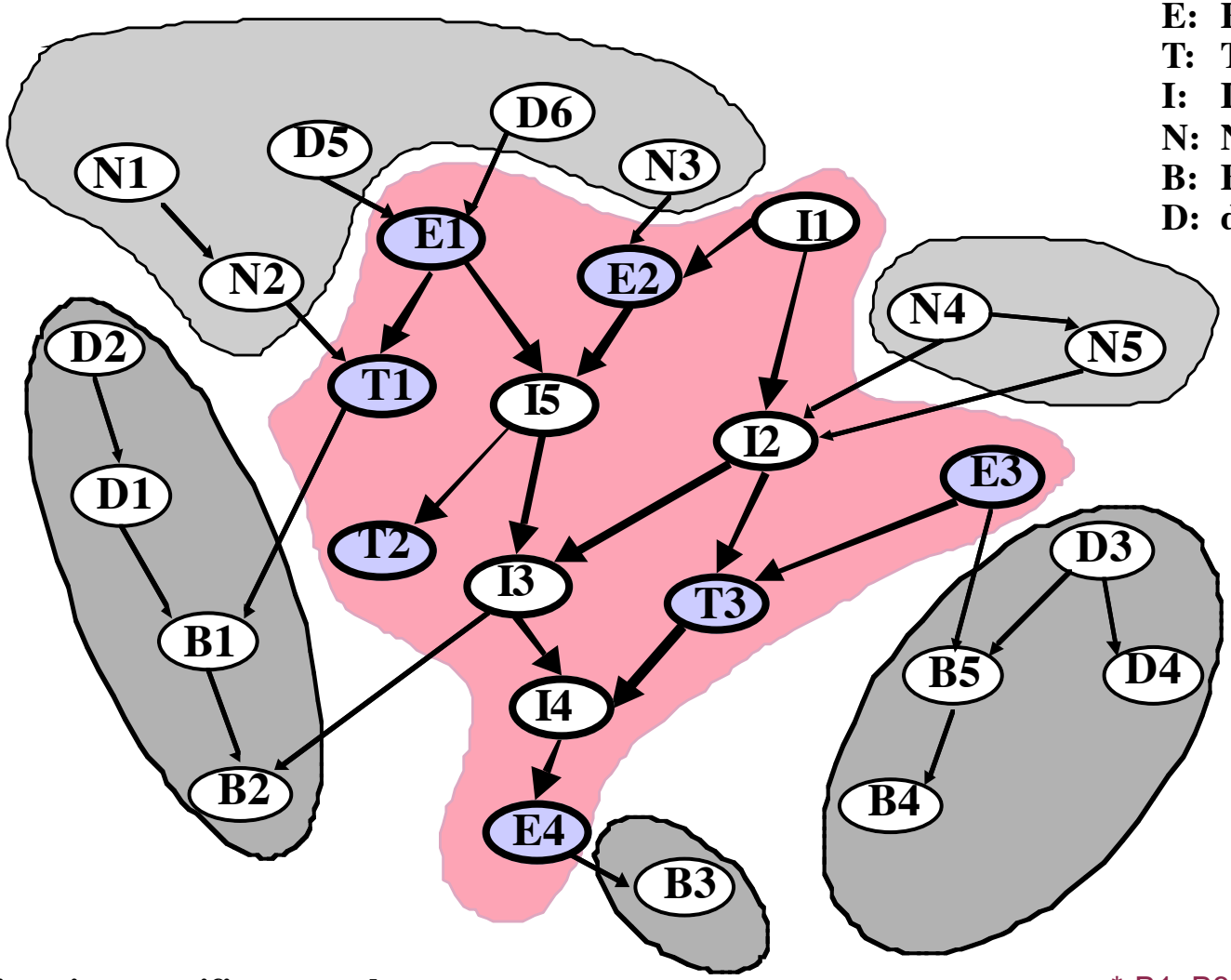
MEBN Specifics

- MFrag
  - Contains random variables and a fragment graph
  - Random variables may be *resident*, *input*, or *context*
  - Input random variables are roots
  - Context random variables are shown as isolated nodes
- Random variables
  - Every random variable is resident in exactly one MFrag,called its *home MFrag*
  - Random variables may be *input* or *context* random variables in any number of MFrags
  - A random variable's distribution is defined in its home MFrag
    - » Local distribution specifies how to construct a belief table for an instance of a resident random variable
    - » Context random variables specify a context in which the influences defined in the fragment graph hold
- MFrag instances
  - Substitute entity identifiers for variables in the MFrag for which it is possible for all context random variables to have value True
  - Context random variables known to be true or false do not have to be represented explicitly
  - Context random variables that are uncertain are parents to all resident random variables
  - Each MFrag instance specifies a parents -> child influence
  - MFrag specifies a rule for combining influences when a random variable is resident in more than one MFrag instance

Inference: Situation-Specific BN (aka "grounding" the first-order theory)

- A situation specific network is a Bayesian network constructed specifically for a problem instance
  - Represents only relevant parts of a larger model
  - "Top down" construction - prune explicilty represented larger network to obtain smaller situation-specific network
  - "Bottom up" construction builds a situation-specific network from a knowledge base of component knowledge elements
- Key ideas for situation-specific network construction
  - Construction is query and evidence driven
    - » Compute P(Target Nodes | Evidence Nodes)
  - Network must contain query and evidence random variables
  - Random variables not computationally relevant to query
    - » d-separated from target variables given evidence variables
    - » "barren" nodes – have no descendents that are either evidence or target random variables
  - Nuisance nodes
    - » May be computationally relevant to query
    - » Can be marginalized out prior to evidence propagation without affecting posterior distribution of target random variables

Situation-Specific Bayesian Network



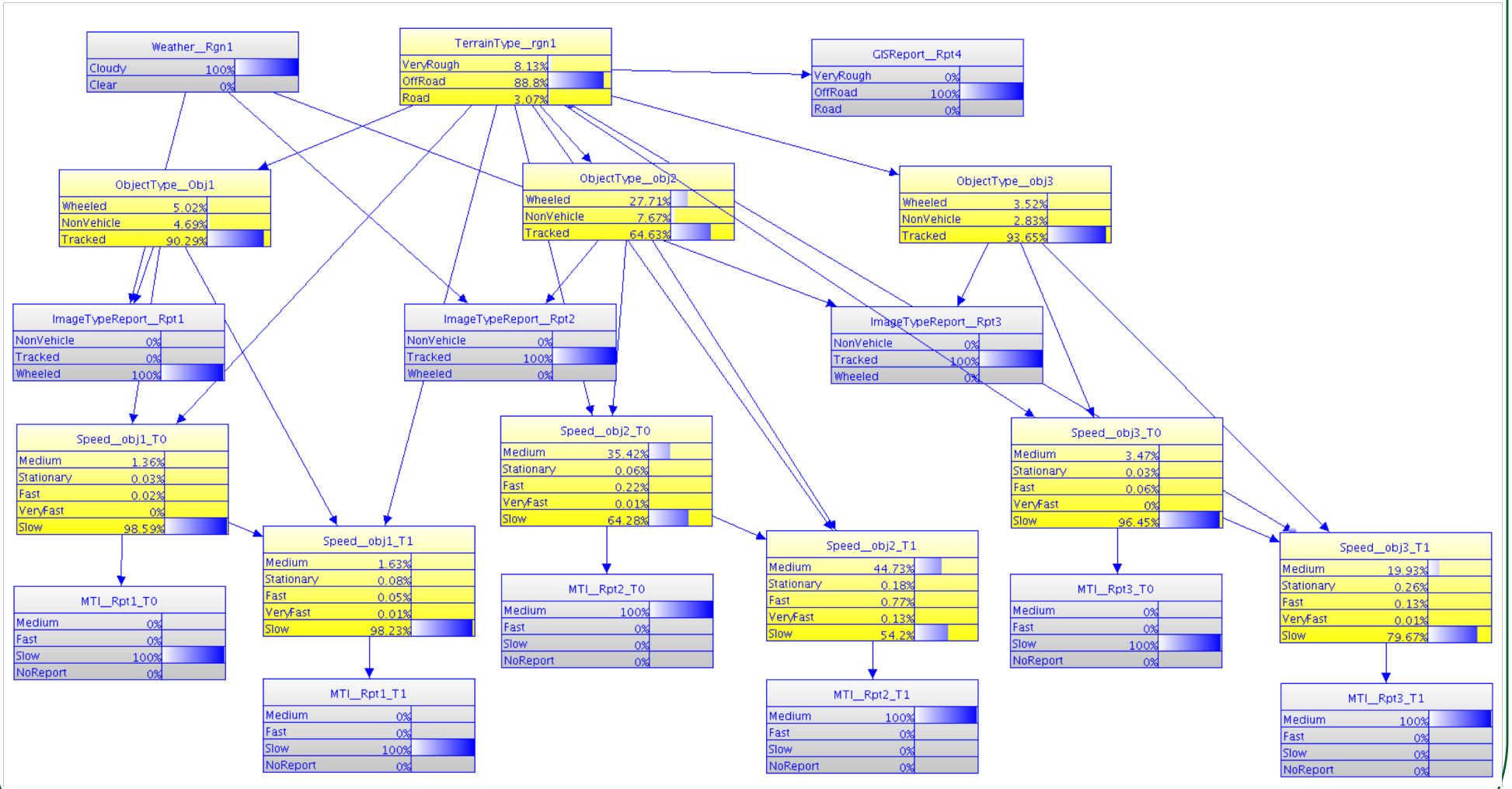
E: Evidence node
T: Target node
I: Internal node
N: Nuisance node
B: Barren node*
D: d-separated node

In situation-specific network

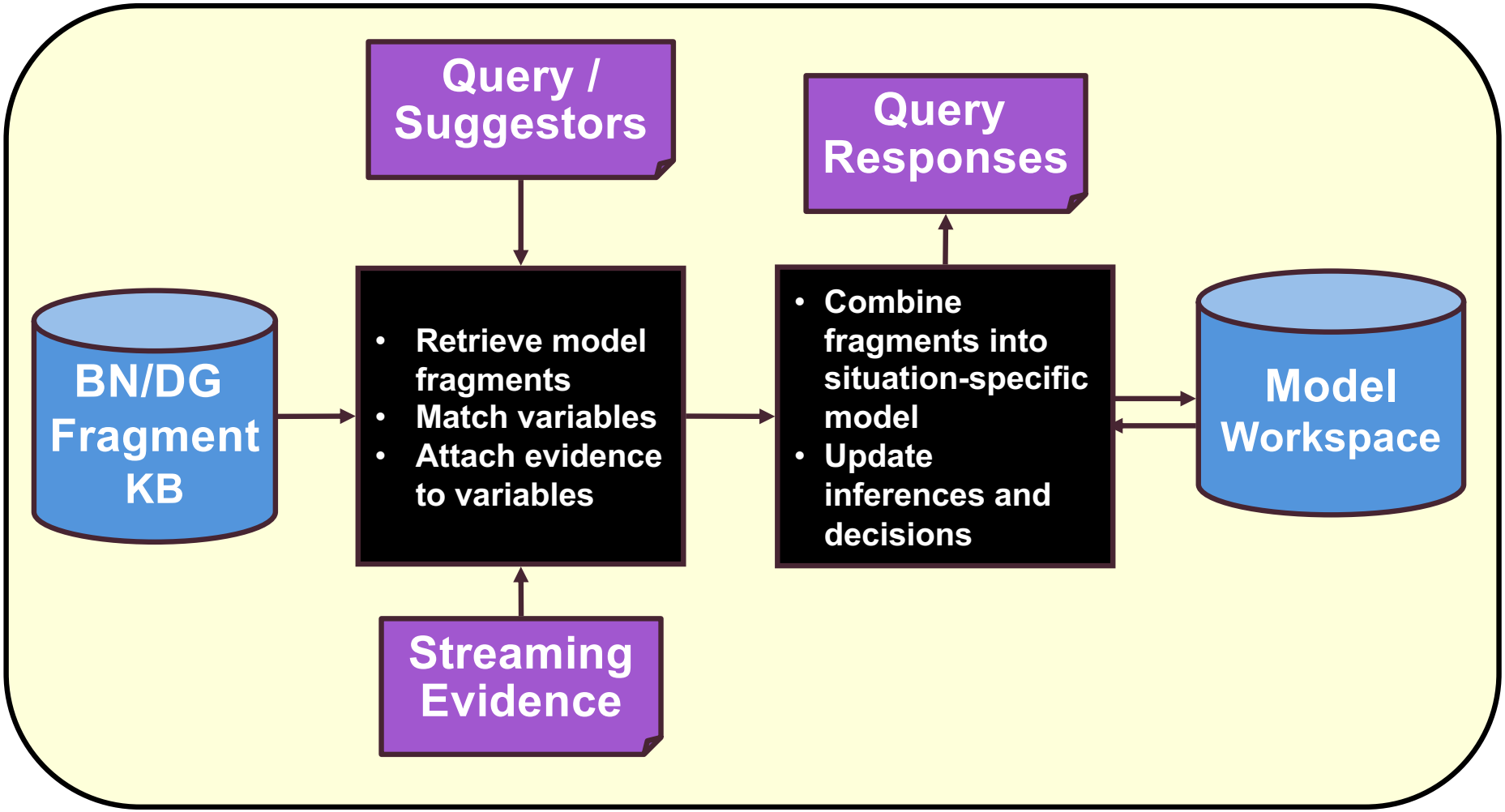
Not in situation-specific network

* B1, B3, B4, B5 are barren and also are d-separated from target nodes given evidence nodes

Example SSBN (aka "ground BN") for 3 vehicles, 2 time steps

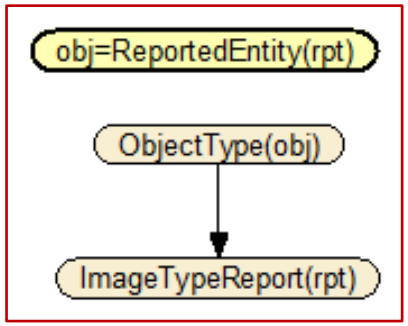


SSBN Construction

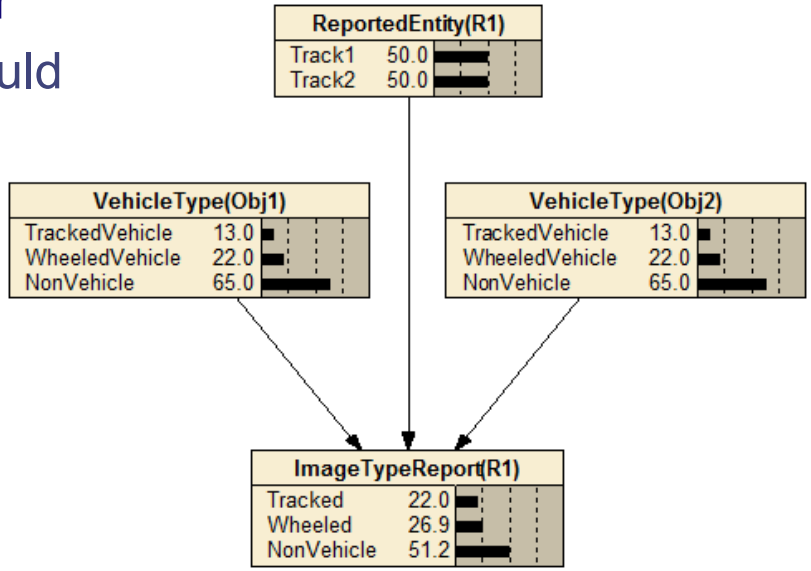


Reference Uncertainty

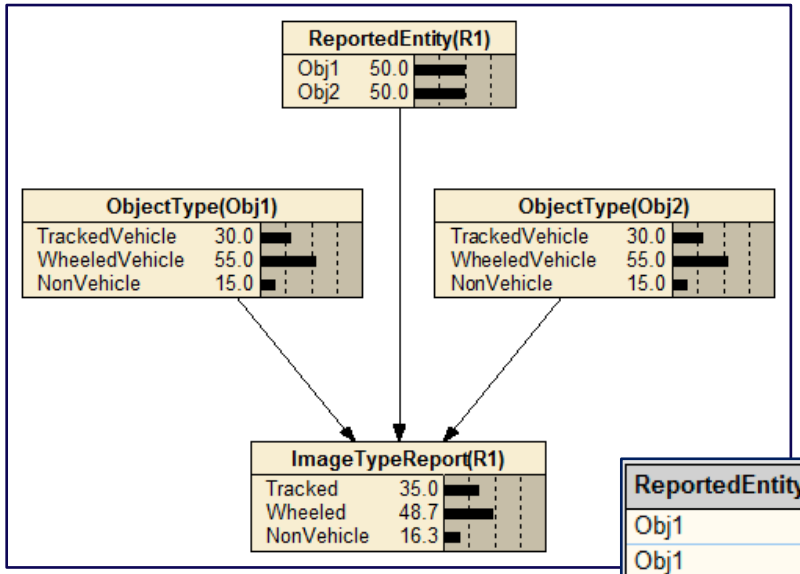
- The Image Type Report MFrag describes the relationship between the type of an object and the type reported by an image sensor
  - ObjectType(*obj*) is the type of the object *obj*
  - ImageTypeReport(*rpt*) is the type reported by sensor report *rpt*
  - The link between these RVs is valid when *obj* is the object in view of the imaging sensor



- Reference uncertainty occurs when we are uncertain about which object is in view of the sensor
  - SSBN has a parent for each object that could have produced the report, and a *reference parent* indicating which parent it was
  - Local distribution in SSBN is specified using the *multiplexor* combination rule



Multiplexor Combination Rule



SSBN with 2 objects

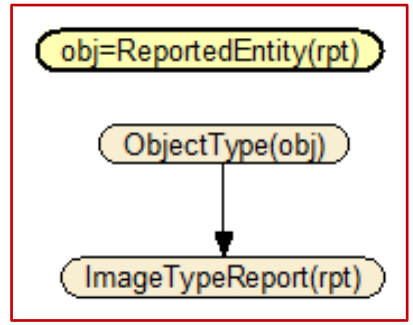


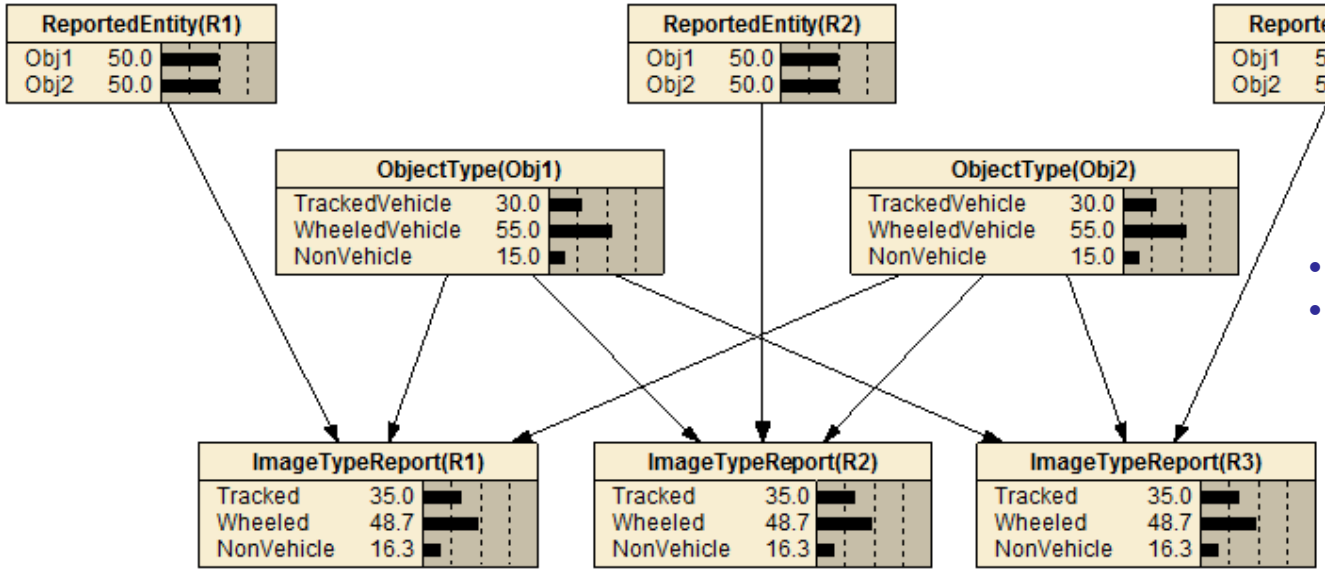
Image Type Report MFrag

ObjectType(obj)	Tracked	Wheeled	NonVehicle
TrackedVehicle	75	20	5
WheeledVehicle	20	75	5
NonVehicle	10	10	80

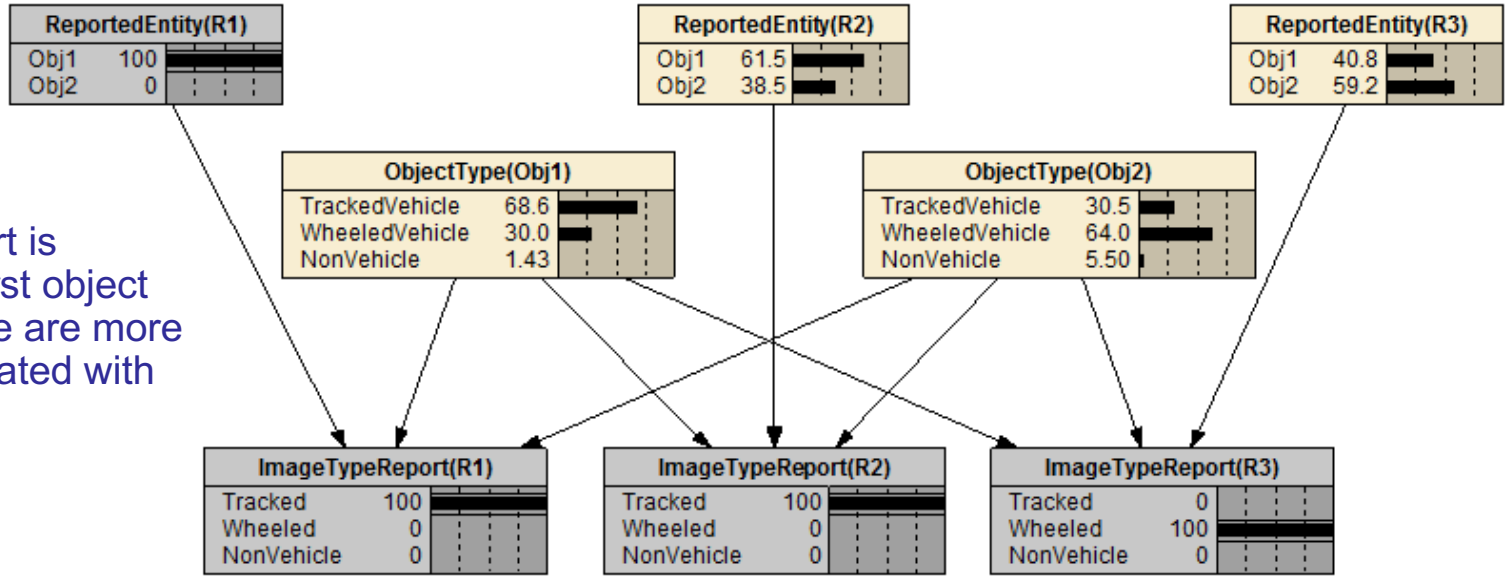
ReportedEntity(R1)	ObjectType(Obj2)	ObjectType(Obj1)	Tracked	Wheeled	NonVehicle
Obj1	TrackedVehicle	TrackedVehicle	75	20	5
Obj1	TrackedVehicle	WheeledVehicle	20	75	5
Obj1	TrackedVehicle	NonVehicle	10	10	80
Obj1	WheeledVehicle	TrackedVehicle	75	20	5
Obj1	WheeledVehicle	WheeledVehicle	20	75	5
Obj1	WheeledVehicle	NonVehicle	10	10	80
Obj1	NonVehicle	TrackedVehicle	75	20	5
Obj1	NonVehicle	WheeledVehicle	20	75	5
Obj1	NonVehicle	NonVehicle	10	10	80
Obj2	TrackedVehicle	TrackedVehicle	75	20	5
Obj2	TrackedVehicle	WheeledVehicle	75	20	5
Obj2	TrackedVehicle	NonVehicle	75	20	5
Obj2	WheeledVehicle	TrackedVehicle	20	75	5
Obj2	WheeledVehicle	WheeledVehicle	20	75	5
Obj2	WheeledVehicle	NonVehicle	20	75	5
Obj2	NonVehicle	TrackedVehicle	10	10	80
Obj2	NonVehicle	WheeledVehicle	10	10	80
Obj2	NonVehicle	NonVehicle	10	10	80

- Distribution for ImageTypeReport (*rpt*) is based on the actual type of the object in view of the sensor
- ImageTypeReport (*rpt*) is independent of ObjectType(*obj*) for objects not in view of sensor

Reference Uncertainty Example

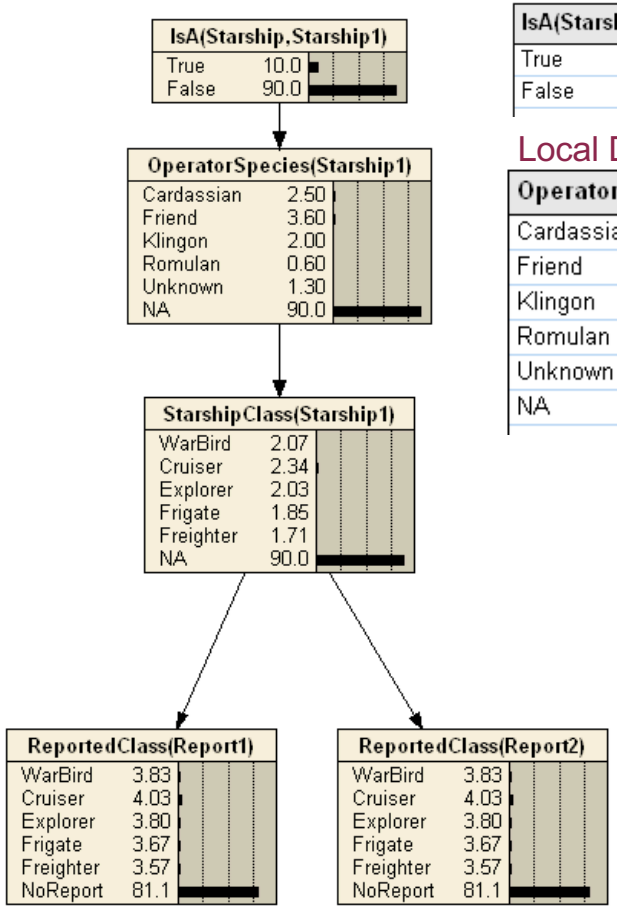


- Two objects and three reports
- Association of reports to objects is unknown *a priori*



- Assume first report is associated with first object
- Reports that agree are more likely to be associated with same object

Existence Uncertainty



Local Distribution for OperatorSpecies(Starship1)

IsA(Starship,Starship1)	Cardassian	Friend	Klingon	Romulan	Unknown	NA
True	25.000	36.000	20.000	6.000	13.000	0.000
False	0.000	0.000	0.000	0.000	0.000	100.00

Local Distribution for StarshipClass(Starship1)

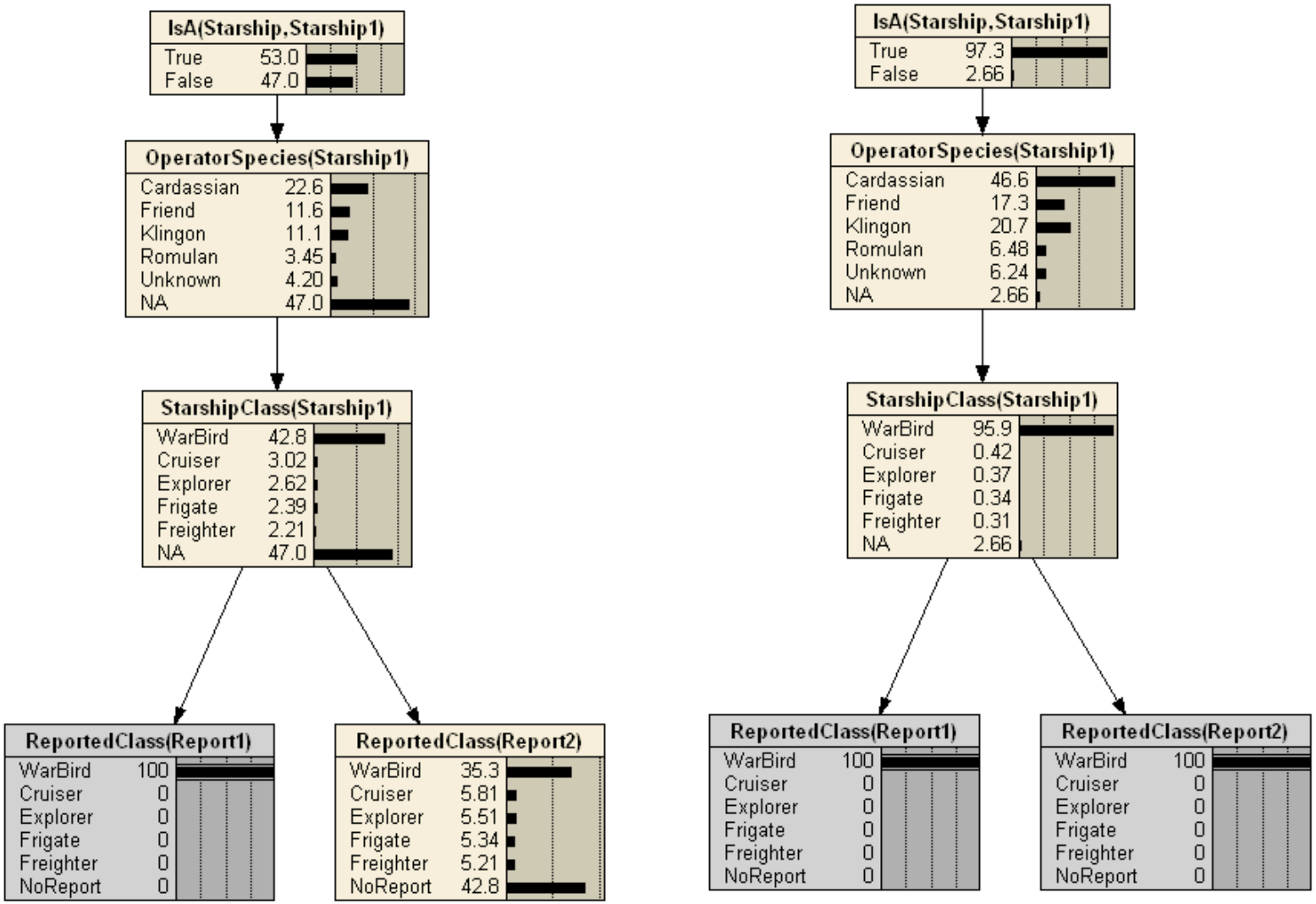
OperatorSpecies(Starship1)	WarBird	Cruiser	Explorer	Frigate	Freighter	NA
Cardassian	40.000	10.000	0.000	40.000	10.000	0.000
Friend	10.000	30.000	25.000	20.000	15.000	0.000
Klingon	22.000	44.000	20.000	0.000	14.000	0.000
Romulan	23.000	0.000	46.000	0.000	31.000	0.000
Unknown	10.000	10.000	35.000	10.000	35.000	0.000
NA	0.000	0.000	0.000	0.000	0.000	100.00

Local Distribution for ReportedClass(Report1) and ReportedClass(Report2)

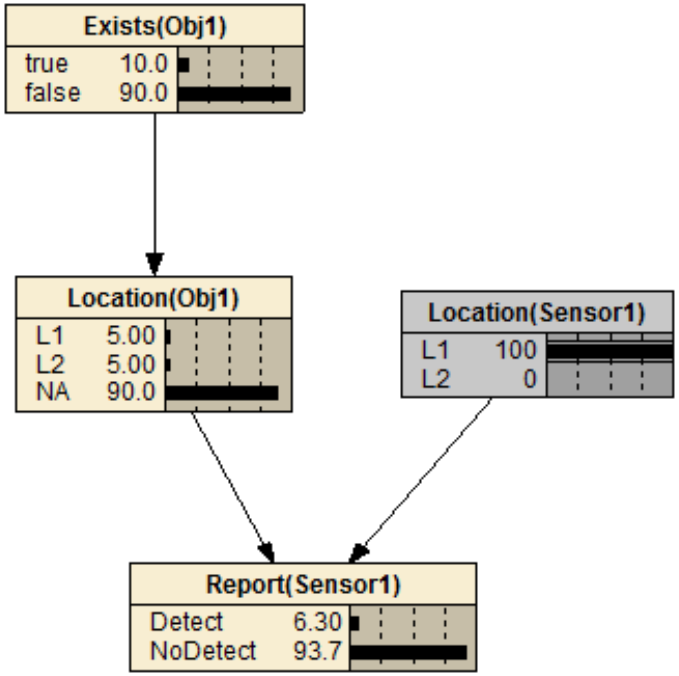
StarshipClass(Starship1)	WarBird	Cruiser	Explorer	Frigate	Freighter	NoReport
WarBird	79.208	4.950	4.950	4.950	4.950	0.990
Cruiser	4.950	79.208	4.950	4.950	4.950	0.990
Explorer	4.950	4.950	79.208	4.950	4.950	0.990
Frigate	4.950	4.950	4.950	79.208	4.950	0.990
Freighter	4.950	4.950	4.950	4.950	79.208	0.990
NA	2.000	2.000	2.000	2.000	2.000	90.000

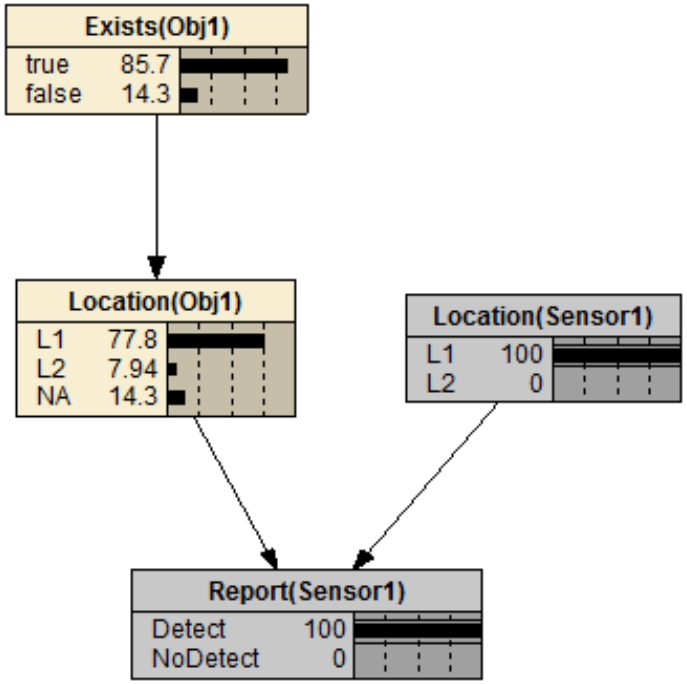
- Reports are fallible
  - Sensor may produce spurious reports
  - Starship may fail to be reported
- Report instance generates hypothetical starship
  - Attributes of nonexistent starship have value “NA”

Multiple Confirming Reports Resolve Existence Uncertainty



Existence Uncertainty and Miss/False Alarm Probabilities





- Non-existent object has NA as location
- Probability of detect given non-existent object is called false alarm probability
- Probability of NoDetect given not-NA is called miss probability
  - Depends on location of object with respect to sensor

Location(Obj1)	Location(Sensor1)	Detect	NoDetect
L1	L1	98	2
L1	L2	10	90
L2	L1	10	90
L2	L2	98	2
NA	L1	1	99
NA	L2	1	99

Types of Uncertainty

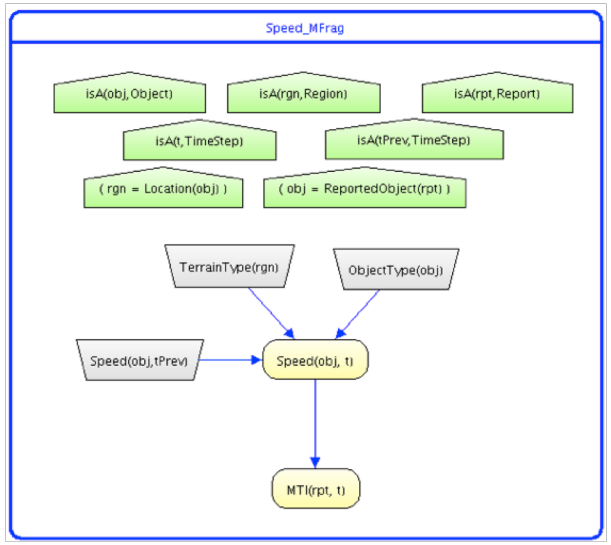
- First-order uncertainty
  - Attribute value uncertainty
    - » TempLight(M) = ?
  - Type uncertainty
    - » Type(M) = ?
  - Existence uncertainty
    - » Exists(M) = ?
  - Reference uncertainty
    - » MachineLocation(M1) = ?
- Higher-order uncertainty
  - Parameter uncertainty
    - » Pr(RoomTemp(r)=HIGH | ACStatus(r)=BROKEN) = ?
  - Structural uncertainty
    - » Is there an arc from RoomTemp(r) to ACStatus(r)?
  - Entity-Relationship uncertainty
    - » Does a relationship exist between two types of entity?

Subtasks in SSBN Construction with Existence and Reference Uncertainty

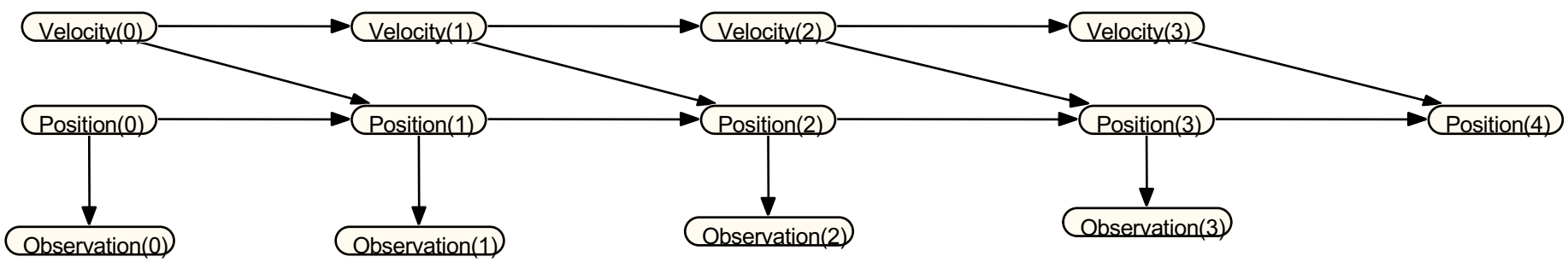
- Data association
  - Given: a new report and a set of hypothesized entities
  - Task: identify which entity gave rise to the report
  - Approaches:
    - » "Hard" assignment to best-fit entity
    - » "Soft" assignment to multiple entities
    - » Multiple-hypothesis assignment
- Hypothesis management
  - Hypothesize new entities when reports don't match existing entities
  - Prune hypotheses that have too little support to be maintained
  - Combine similar hypotheses
- MEBN fragment retrieval and model construction
- Inference and projection
  - Declare reports as evidence for the hypotheses they support
  - Infer properties of entities and relationships among entities
  - Project forward to time of next report

Reasoning About Time

- *Dynamic Bayesian Networks* (DBNs) are commonly used to reason in domains where temporal effects are present
- Random variables are repeated in time
  - Special case: Kalman filter
  - Special case: Hidden Markov model
- More complex temporal effects can also be modeled
- Active area of research
- We will do a unit on dynamic models



MFrag with Dynamic Node



DBN Example

Markov Logic Networks

- "Just add weights" to first-order logic sentences
- Yields first-order undirected graphical model

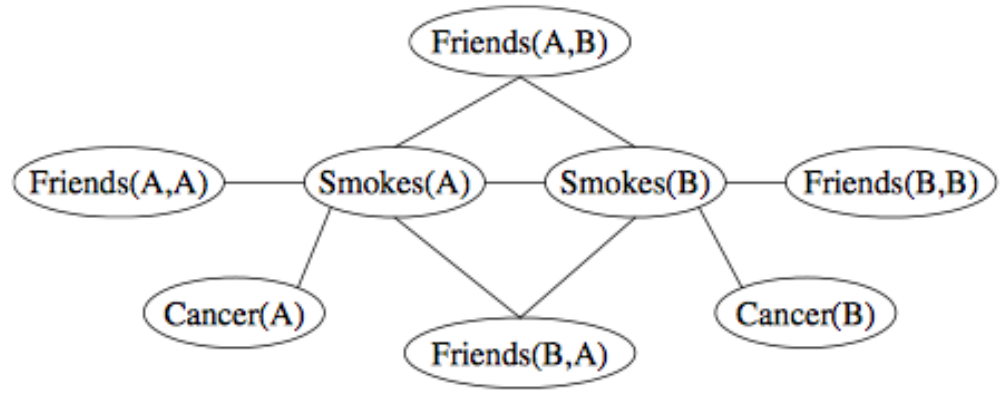


Figure 1. Ground Markov network obtained by applying the last two formulas in Table I to the constants `Anna(A)` and `Bob(B)`.

Table I. Example of a first-order knowledge base and MLN. `Fr()` is short for `Friends()`, `Sm()` for `Smokes()`, and `Ca()` for `Cancer()`.

English	First-Order Logic	Clausal Form	Weight
Friends of friends are friends.	$\forall x \forall y \forall z\ \mathtt{Fr}(x,y) \wedge \mathtt{Fr}(y,z) \Rightarrow \mathtt{Fr}(x,z)$	$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Fr}(y,z) \vee \mathtt{Fr}(x,z)$	0.7
Friendless people smoke.	$\forall x\ (\neg(\exists y\ \mathtt{Fr}(x,y)) \Rightarrow \mathtt{Sm}(x))$	$\mathtt{Fr}(x,g(x)) \vee \mathtt{Sm}(x)$	2.3
Smoking causes cancer.	$\forall x\ \mathtt{Sm}(x) \Rightarrow \mathtt{Ca}(x)$	$\neg\mathtt{Sm}(x) \vee \mathtt{Ca}(x)$	1.5
If two people are friends, either	$\forall x \forall y\ \mathtt{Fr}(x,y) \Rightarrow (\mathtt{Sm}(x) \Leftrightarrow \mathtt{Sm}(y))$	$\neg\mathtt{Fr}(x,y) \vee \mathtt{Sm}(x) \vee \neg\mathtt{Sm}(y)$,	1.1
both smoke or neither does.		$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Sm}(x) \vee \mathtt{Sm}(y)$	1.1

Software for MLNs

- Alchemy
  - http://alchemy.cs.washington.edu/
- Tuffy
  - http://research.cs.wisc.edu/hazy/tuffy/

Unit 3 Outline

- Knowledge Representation
- Logic and Ontologies
- Expressive Probabilistic Languages
- Probabilistic Ontology

Ontology

- An ontology is an explicit, formal representation of knowledge about a domain of application. This includes:
  - Types of entities that exist in the domain; `Fighter, APV, Missile, ...`
  - Properties of those entities; `hasMaxSpeed, hasNetWeight, hasMaxGRate, ...`
  - Relationships among entities; `isCommissiontedAt, supports, hasLaunchBase, ...`
  - Processes and events that happen with those entities; `participate in mission ...`

  where the term entity refers to any concept (real or fictitious, concrete or abstract) that can be described and reasoned about within the domain of application. ■

Costa, P.C.G., "Bayesian Semantics for the Semantic Web", Doctoral Dissertation, George Mason University, 2005.

Probabilistic Ontology

- A probabilistic ontology is an explicit, formal representation of knowledge about a domain of application. This includes:
  - Types of entities that exist in the domain; `Fighter, APV, Missile, ...`
  - Properties of those entities; `hasMaxSpeed, hasNetWeight, hasMaxGRate, ...`
  - Relationships among entities; `isCommissiontedAt, supports, hasLaunchBase, ...`
  - Processes and events that happen with those entities; `participate in mission ...`
  - Statistical regularities that characterize the domain;
  - Inconclusive, ambigious, incomplete, unreliable and dissonant knowledge related to entities of the domain;
  - uncertainty about all the above forms of knowledge;

```
P(isEnemyIncursor|
hasIFF = False,
groundSpeed = >420Kt,
isFormationMember = True)
= 90%, ...
```

where the term entity refers to any concept (real or fictitious, concrete or abstract) that can be described and reasoned about within the domain of application. ■

Costa, P.C.G., "Bayesian Semantics for the Semantic Web", Doctoral Dissertation, George Mason University, 2005.

PR-OWL Probabilistic Ontology Language

- Expresses MEBN theories in World Wide Web Consortium (W3C) recommended OWL ontology language.
  - Some aspects of MEBN are not supported by OWL
  - PR-OWL reasoners are expected to respect these conditions although they cannot be handled by OWL reasoners
- Open-source, freely available solution for representing knowledge and associated uncertainty.
- UnBBayes-MEBN (developed in collaboration between GMU and University of Brasilia)
  - Represents and reasons with MEBN theories
  - Stores MFrags in PR-OWL format

Summary and Synthesis

- First-order logic
  - Basis for standard symbolic AI
  - Expressive
  - Cannot represent ambiguity or uncertainty
- Probability (traditional)
  - Moving rapidly into mainstream AI
  - Propositional representational power (no mechanisms for reasoning about classes of individuals)
  - Represents uncertainty
  - Non-modular
- Statistics (traditional)
  - Probability theory applied to classes of individuals
  - Limited expressive power
- First-order probabilistic logics synthesize logic, probability & statistics
  - First-order expressive power
  - Represent uncertainty
  - Modular elements with global consistency constraint
  - Learning theory based on Bayesian statistics

Some References for Unit 3

- Costa, P. (2005) *Bayesian Semantics for the Semantic Web*, PhD dissertation, George Mason University.
- Costa, P., Ladeira, M., Carvalho, R.N., Laskey, K.B., Lima, Laécio, Matsumoto, S. (2008) A First-Order Bayesian Tool for Probabilistic Ontologies. *Proceedings of FLAIRS 2008.*
- Charniak, E. and Goldman, R. (1993) A Bayesian Model of Plan Recognition. *Artificial Intelligence,* 64: 53-79.
- Davis, R., Schrobe, H. and Szlovits, P. (1993) What is a Knowledge Representation? *AI Magazine* 14:1, 17-33.
- Dawid, A.P., Mortera, J. and Vicard, P. (2005) *Object-Oriented Bayesian Networks for Complex Forensic DNA Profiling Problems.* http://www.ucl.ac.uk/Stats/research/Resrprts/abs05.html#256
- Getoor, L., Friedman, N., Koller, D. and Pfeffer, A. (2001) Learning Probabilistic Relational Models. In Saso Dzeroski and Nada Lavrac, editors. *Relational Data Mining*, Springer-Verlag, New York, New York.
- Getoor, L. and Pfeffer, A. (2005) Representation, Inference and Learning in Relational Probabilistic Languages. Tutorial presented at IJCAI05, available from http://www.eecs.harvard.edu/~avi/.
- Haddawy, P. (1994) Generating Bayesian networks from Probability Logic Knowledge Bases. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA pp. 262-269.
- Koller, D. and A. Pfeffer (1997) Object-Oriented Bayesian Networks In Geiger, D. and Shenoy, P. (eds) *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, San Francisco, CA: Morgan Kaufmann.
- Laskey, K.B. (2007) MEBN: A Language for First-Order Bayesian Knowledge Bases. *Artificial Intelligence*, 172(2-3).
- Pfeffer, A. (2000). *Probabilistic Reasoning for Complex Systems*. PhD dissertation, Stanford University.
- Sowa, J. (2000) *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole.
- Wellman, M.P., J.S. Breese, and R.P. Goldman (1992) From knowledge bases to decision models. *The Knowledge Engineering Review*, 7(1):35-53.