Description Logics

What is Description Logic?

- A family of logic based KR formalisms
  - Descendants of semantic networks and KL-ONE
  - Describe domain in terms of concepts (classes), roles (relationships) and individuals
- Distinguished by:
  - Formal semantics (typically model theoretic) based on a decidable fragments of FOL
  - Provision of inference services
    - Sound and complete decision procedures for key problems
    - Implemented systems (highly optimized)
- Formal basis for OWL (DL profile)

Informally, What is Description Logic?

- We define a concept using a simple noun phrase in a human language like English
  - A red car
  - A tall person who works for IBM
  - A tall person who works for a Bay-area Technology company
- We don’t do this using a set of rules
- Natural languages have multiple ways of attaching modifiers to a simple concept
  - E.g., adjectives, propositional phrases, clausal modifiers, connectives (and, or, not)
- Description logics, like OWL-DL, designed to define concepts in a similar way

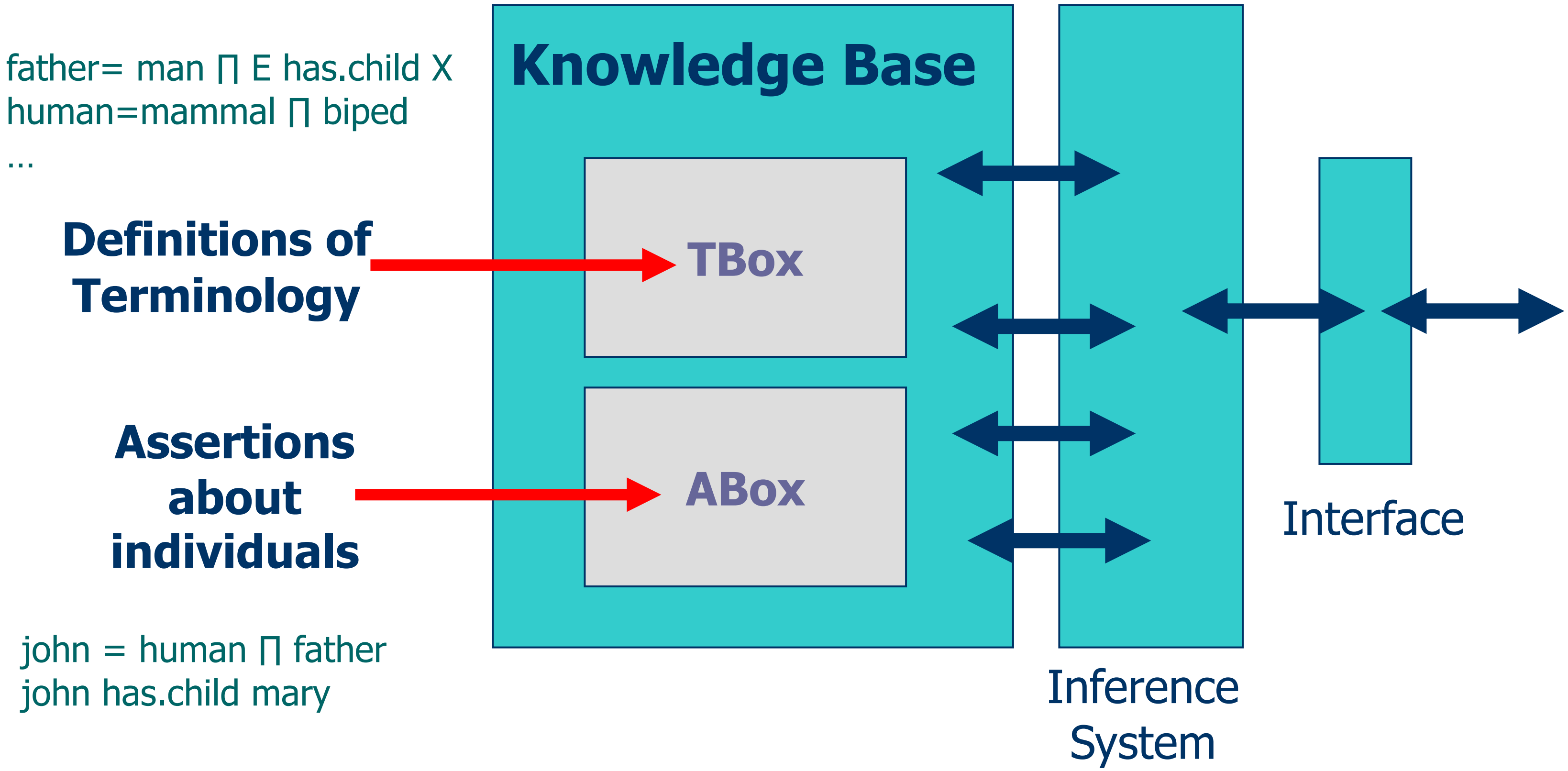
Description Logic History

- Major focus of KR research in the 1980's
  - Grew out of early network-based KR systems like semantic networks and frames
  - Inspired by 1975 paper by Bill Woods, What's in a Link
- Major systems and languages
  - 80s: KL-ONE, NIKL, KANDOR, BACK, CLASSIC, LOOM
  - 90s: FACT, RACER, …
  - 00s: DAML+OIL, OWL, Pellet, Jena, FACT++, …
  - 10s: HermiT, ELK, …
- Basis for semantic web language OWL

DL Paradigm

- **Description Logic** characterized by a set of constructors that allow one to build complex *descriptions* or *terms* out of **concepts** and **roles** from atomic ones
  - **Concepts**: classes interpreted as sets of objects,
  - **Roles**: relations interpreted as binary relations on objects
- Set of axioms for asserting **facts** about concepts, roles and **individuals**

Typical KB Architecture



Division into **TBox** and **ABox** has no logical significance, but is made for conceptual & implementation convenience

- Tbox ≈ Ontology and Abox ≈ Data

DL defines a family of languages

- The expressiveness of a description logic is determined by the **operators** that it uses
  - Adding or removing operators (e.g., $\neg$, $\cup$) increases or decreases the kinds of statements expressible
  - Higher expressiveness usually means higher reasoning complexity
- *AL* or *Attributive Language* is the base and includes just a few operators
- Other DLs are described by the additional operators they include

AL: Attributive Language

Constructor	**Syntax**	**Example**
atomic **concept**	C	Human
atomic negation	~ C	~ Human
atomic **role**	R	hasChild
conjunction	C ∧ D	Human ∧ Male
value **restriction**	R.C	Human ∃ hasChild.Blond
existential rest. (lim)	∃ R	Human **∃** hasChild
Top (universal concept)	T	T
bottom (null concept)	⊥	⊥

for concepts C and D and role R

ALC Adds Complements

ALC is the smallest DL that is *propositionally closed* (i.e., includes full negation and disjunction) and include booleans (and, or, not) and restrictions on role values

constructor	**Syntax**	**Example**
atomic concept	C	Human
negation	**~ C**	**~ (Human V Ape)**
atomic role	R	hasChild
conjunction	C ^ D	Human ^ Male
disjunction	**C V D**	**Nice V Rich**
value restrict.	∃ R.C	Human ∃ hasChild.Blond
existential restrict.	**∃ R.C**	**Human ∃ hasChild.Male**
Top (univ. conconcept)	T	T
bottom (null concept)	⊥	⊥

Examples of ALC concepts

- **Person ⋀ ∀hasChild.Male** *(everybody whose children are all male)*
- **Person ⋀ ∀hasChild.Male ⋀∃hasChild.T** *(everybody who has a child and whose children are all male)*
- **Living_being ⋀ ¬Human_*being*** *(all living beings that are not human beings)*
- **Student ⋀ ¬∃interestedIn.Mathematics** *(all students not interested in mathematics)*
- **Student ⋀ ∀drinks.tea** *(all students who only drink tea)*
- **∃hasChild.Male V ∀hasChild.⊥** *(everybody who has a son or no child)*

Other Constructors

The general DL model has additional constructors…

Constructor	Syntax	Example
Number restriction	>= n R	>= 7 hasChild
	<= n R	<= 1 hasmother
Inverse role	R-	haschild-
Transitive role	R*	hasChild*
Role composition	R ◦ R	hasParent ◦ hasBrother
Qualified # restrict.	>= n R.C	>= 2 hasChild.Female
Singleton concepts	{<name>}	{Italy}

Special names and combinations

See http://en.wikipedia.org/wiki/Description_logic

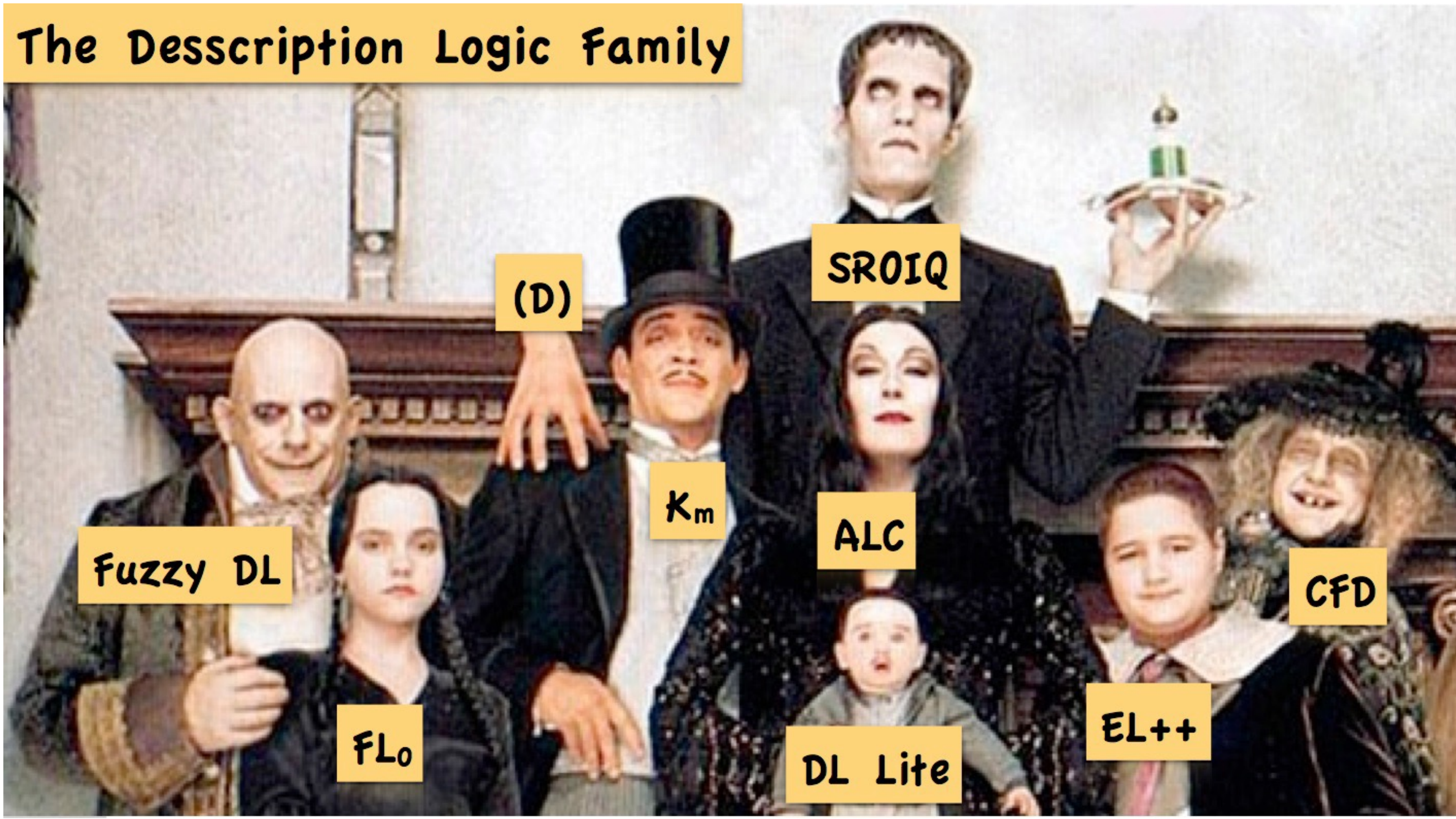
- S = ALC + transitive properties
- H = role hierarchy, e.g., rdfs:subPropertyOf
- O = nominals, e.g., values constrained by enumerated classes (e.g., days of week) , as in owl:oneOf and owl:hasValue
- I = inverse properties
- N = cardinality restrictions (owl:cardinality, maxCardonality)
- $^{(D)}$ = use of datatypes properties
- R = complex role axioms (e.g. (ir)reflexivity, disjointedness)
- Q = Qualified cardinality (e.g., at least two female children)

➔ **OWL-DL is SHOIN$^{(D)}$**

➔ **OWL 2 is SROIQ$^{(D)}$**

Note: R->H and Q->N

DL defines a family of languages



OWL as a DL

- OWL-DL is $SHOIN^{(D)}$
- We can think of OWL as having three kinds of statements
  - Ways to specify classes
    - the intersection of humans and males
  - Ways to state axioms about those classes
    - Humans are a subclass of apes
  - Ways to talk about individuals
    - John is a human, a male, and has a child Mary

Subsumption: $D \subseteq C$?

- Concept C **subsumes** D iff for every interpretation *I*

  $I(D) \subseteq I(C)$
  - This means the same as ∀(x)(D(x) → C(x)) for complex statements D & C
- Determining whether one concept *logically* contains another is called the *subsumption problem*
- Subsumption is **undecidable** for reasonably expressive languages
  - e.g.; for FOL, subsumption means "does one FOL sentence imply another"
- and non-polynomial for fairly restricted ones

Other reasoning problems

These problems can be reduced to subsumption (for languages with negation) and to the satisfiability problem

- **Concept satisfiability** is C (necessarily) empty?
- **Instance Checking** Father(john)?
- **Equivalence** CreatureWithHeart ≡ CreatureWithKidney
- **Disjointness** C ⊓ D
- **Retrieval** Father(X)? X = {john, robert}
- **Realization** X(john)? X = {Father}

Definitions

- A **definition** is a description of a concept or a relationship
- It is used to assign a meaning to a term
- In description logics, definitions use a specialized logical language
- Description logics are able to do limited reasoning about concepts defined in their logic
- One important inference is **classification**, i.e., the computation of subsumption relations

Necessary vs. Sufficient Properties

- *Necessary* properties of an object are common to all objects of that type
  - Being a *man* is a **necessary** condition for being a *father*
- *Sufficient* properties allow one to identify an object as belonging to a type and need not be common to all members of the type
  - *Speeding* is a **sufficient** reason for being stopped by the police (but there are others!)
- **Definitions** typically specify **both** *necessary and sufficient* properties

Subsumption (1)

- Meaning of subsumption in knowledge representation

  A more general concept/description **subsumes** a more specific one. Members of a subsumed concept are necessarily members of a subsuming concept

- It's a familiar concept in programming languages, especially object-oriented ones
- Example: *Animal* subsumes *Person*
- Notations differ: IS-A, rdfs:subClassOf, P279

Subsumption (2)

Two ways to formalize meaning of subsumption

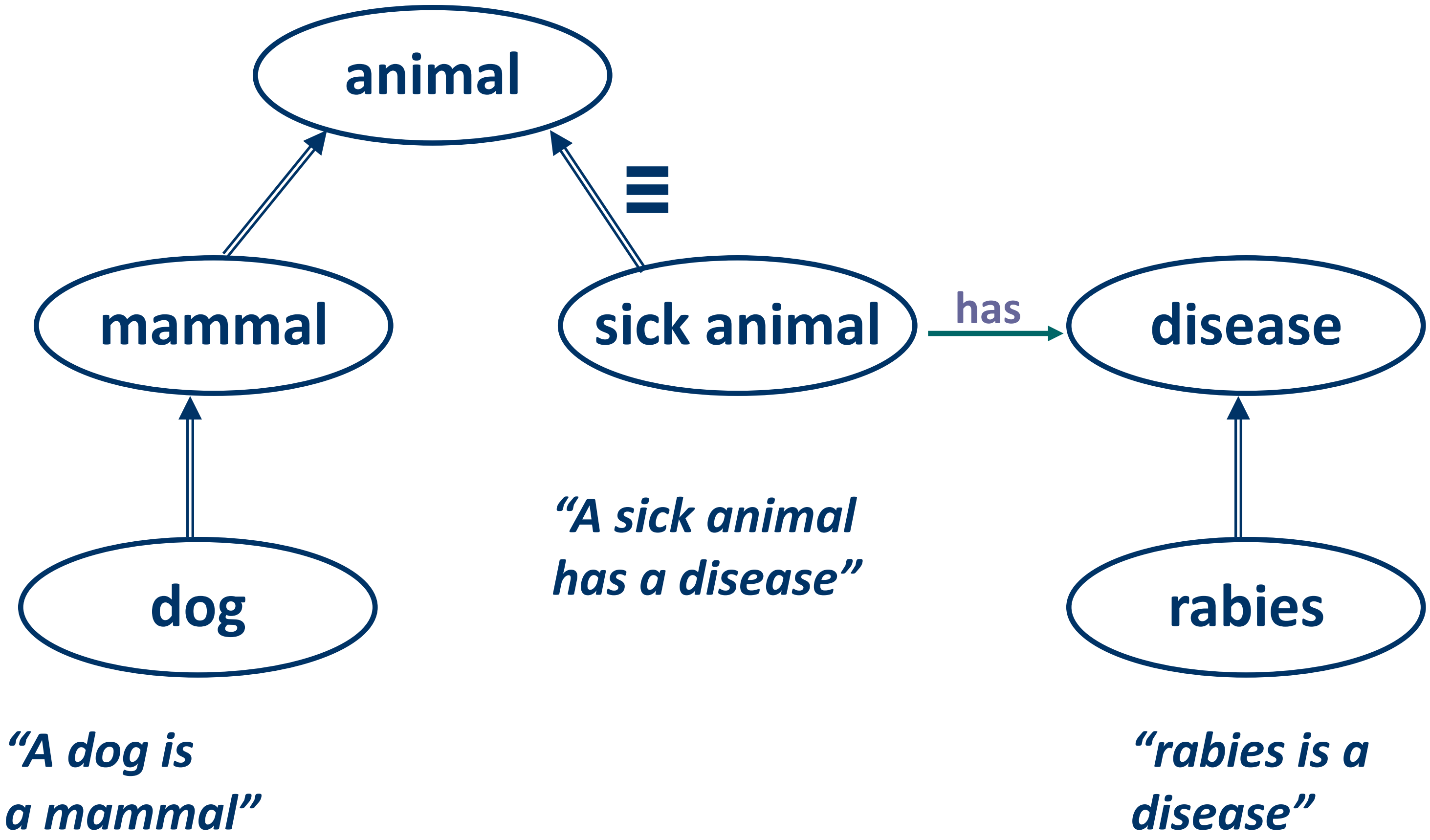
- Using logic: satisfying a subsumed concept implies that the subsuming concept is also satisfied

  E.g., if john is a person, he is also an animal

- Using set theory: instances of subsumed concept are necessarily a subset of subsuming concept's instances

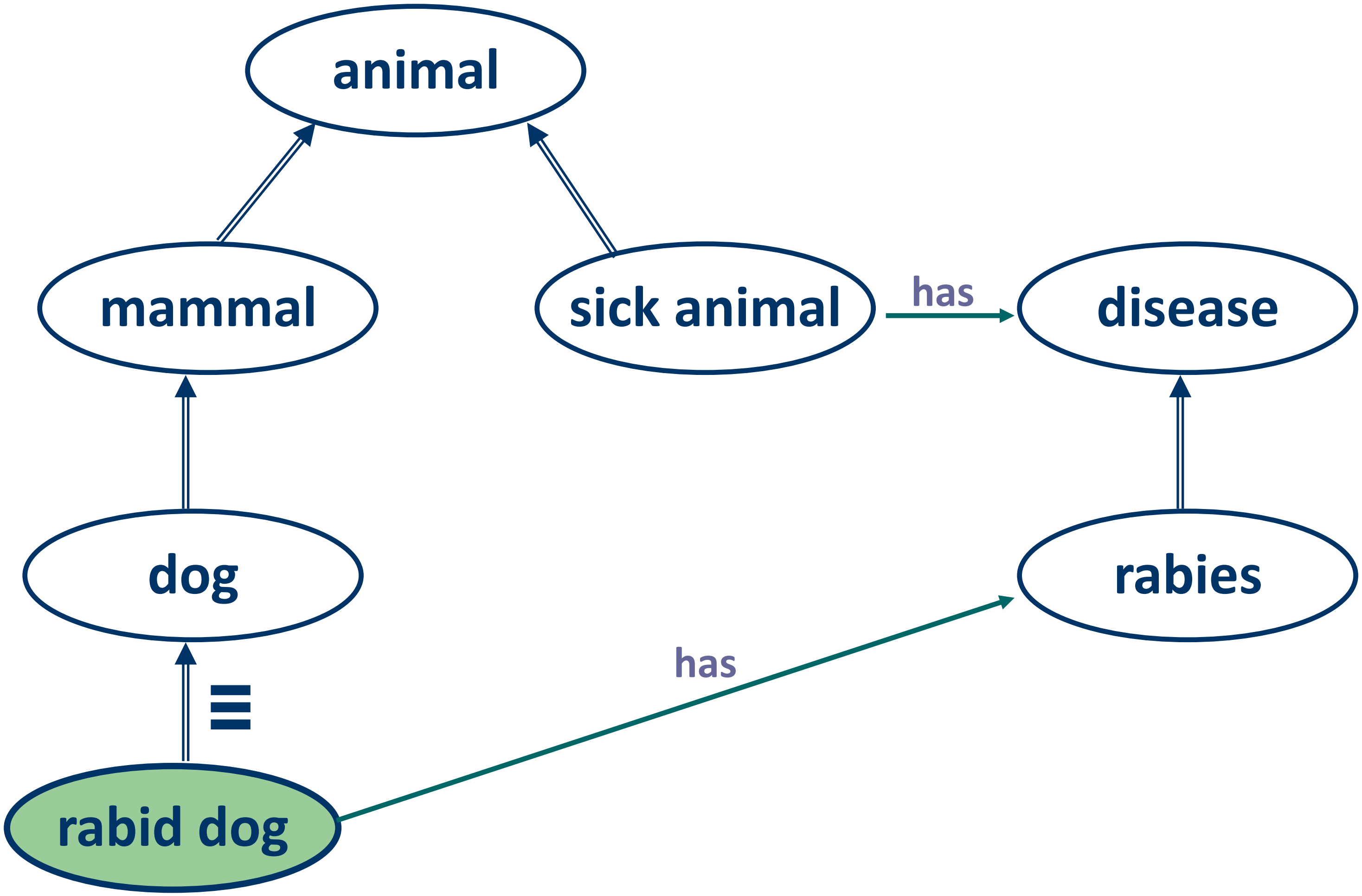
  E.g., the set of all persons is a subset of all animals

How Does Classification Work?



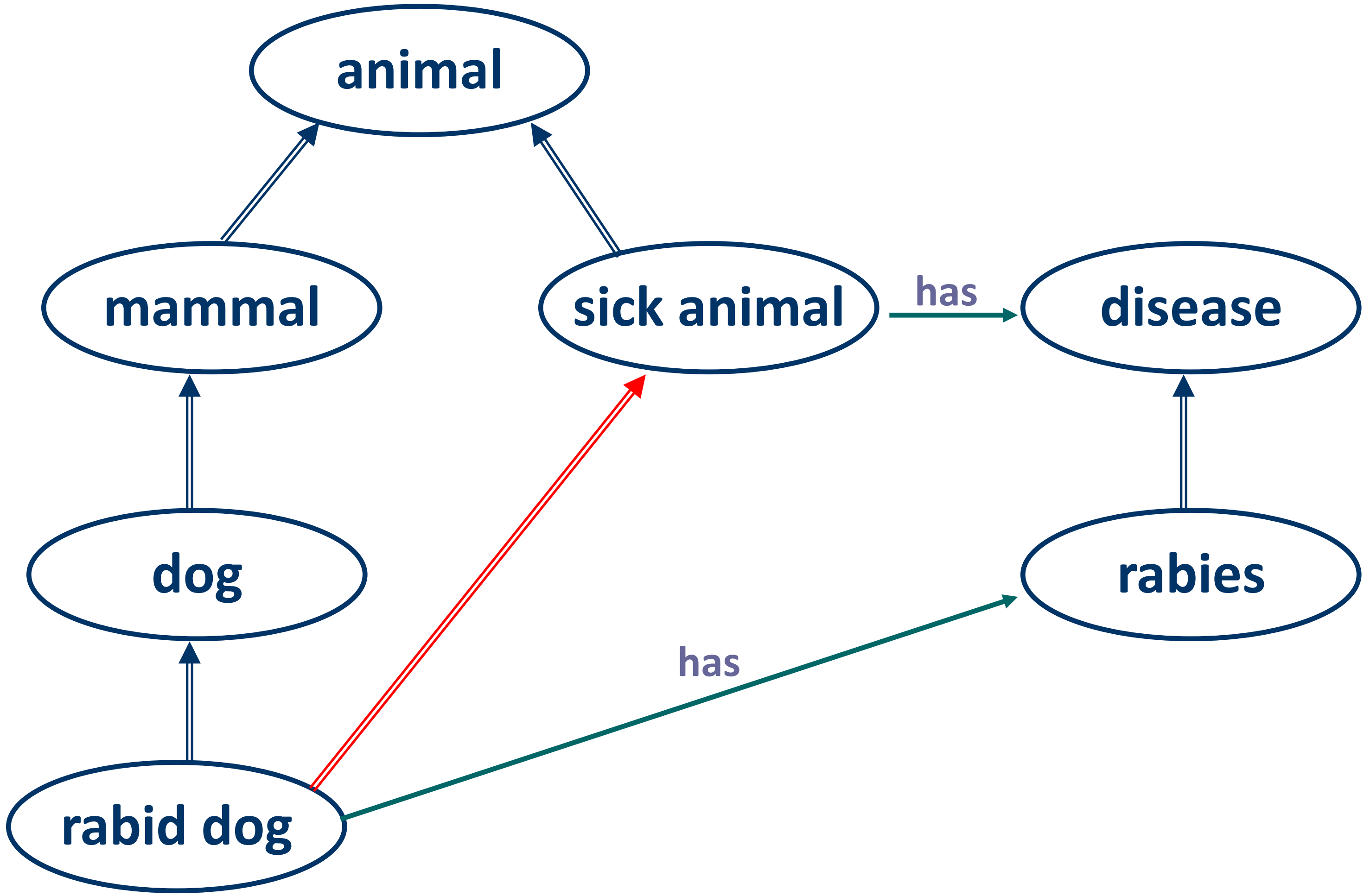
A sick animal **defined** as something that's both an animal and has at least one thing that is a kind of a disease

Defining a “rabid dog”



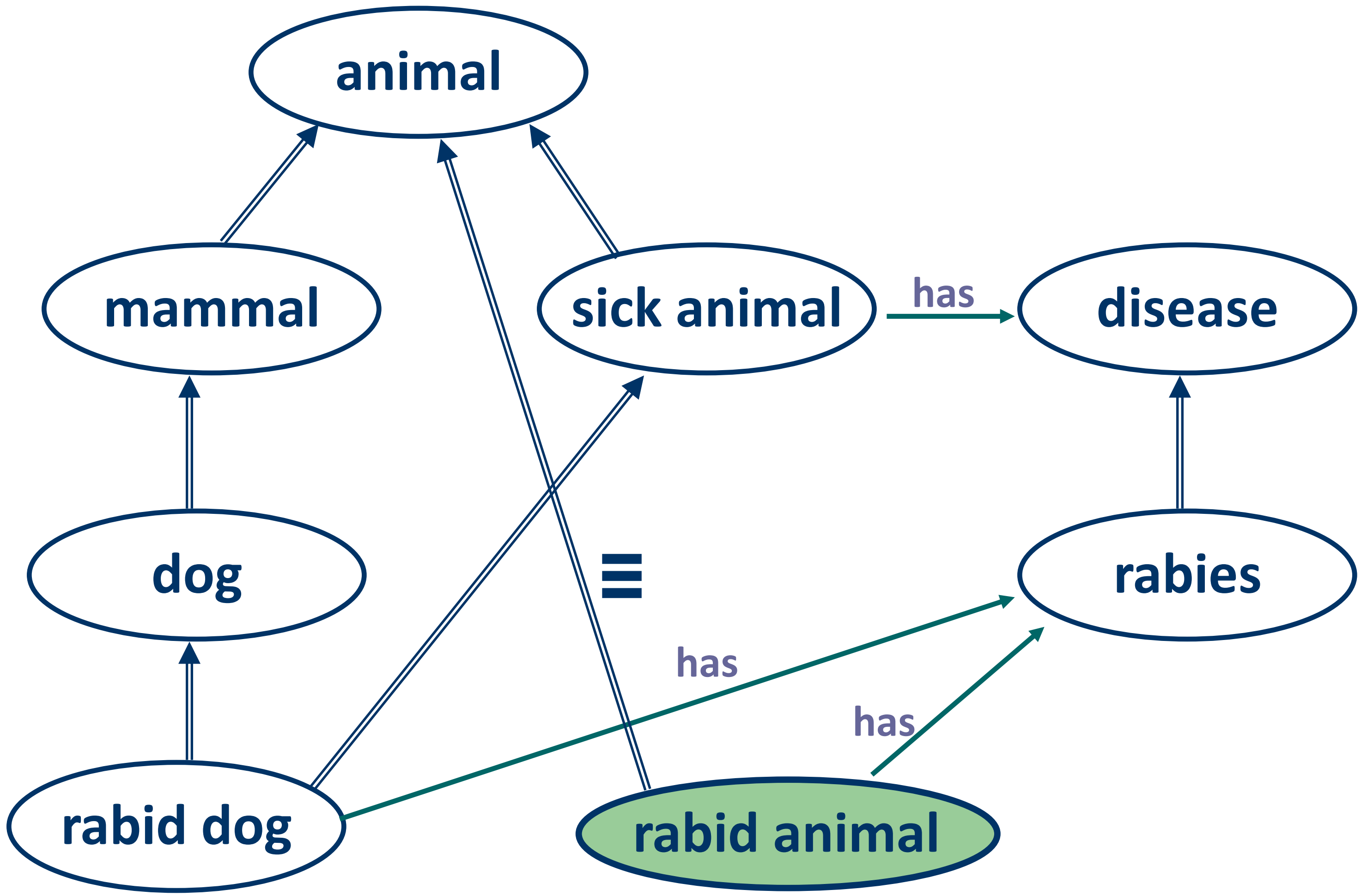
The **rabid dog** concept is **defined** as something that is both a dog and has rabies

Classification as a "sick animal"



We can easily prove that a rabid dog is a kind of sick animal

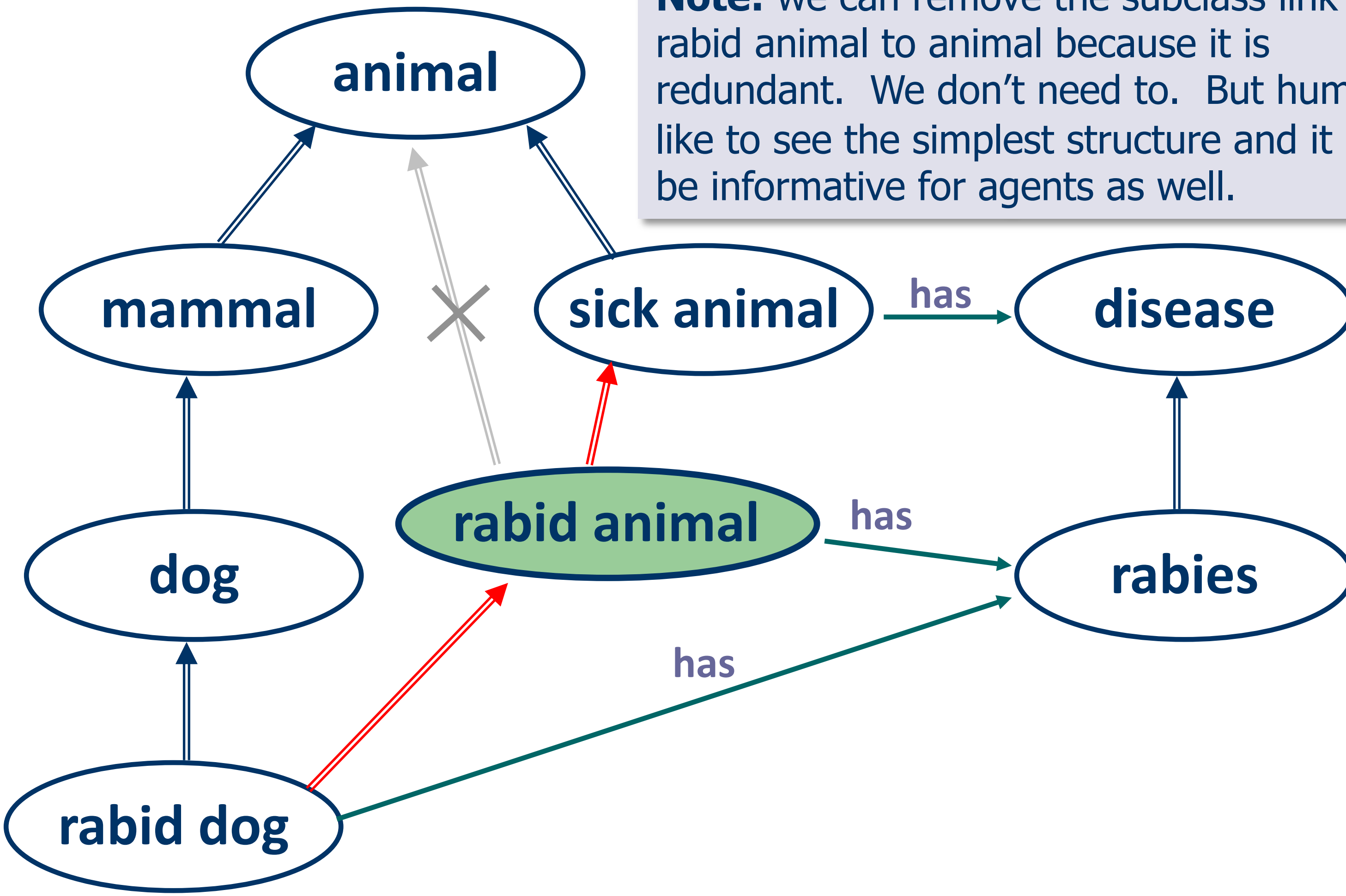
Defining "rabid animal"



The **rabid animal** concept is **defined** as something that is both an animal and has rabies

DL reasoners places concepts in hierarchy

Note: we can remove the subclass link from rabid animal to animal because it is redundant. We don't need to. But humans like to see the simplest structure and it may be informative for agents as well.



The **rabid animal** concept is **defined** as something that is both an animal and has rabies

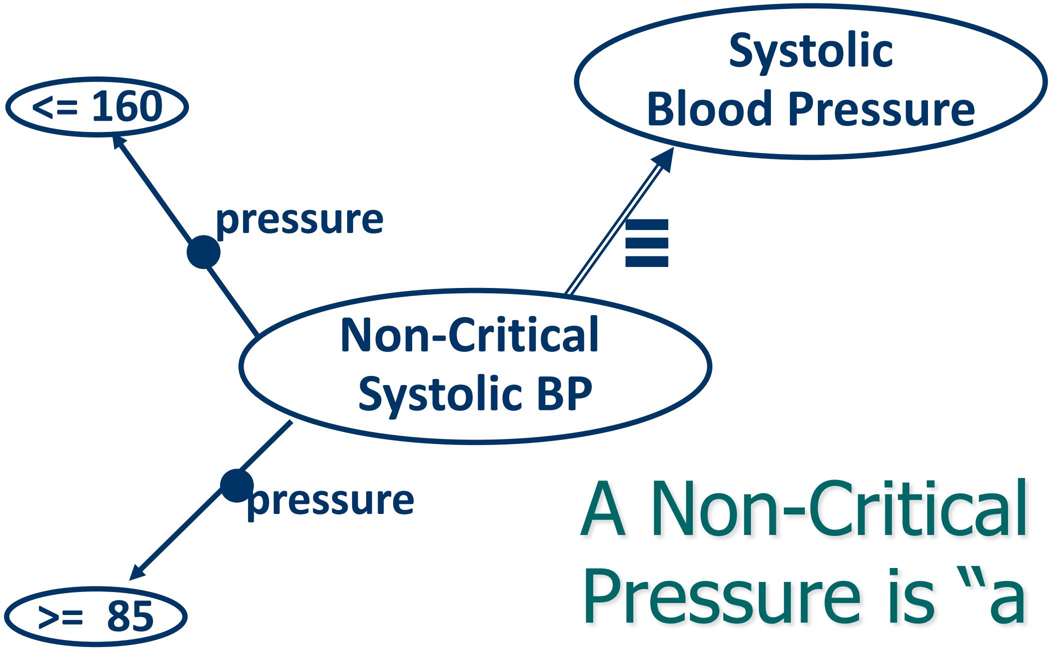
Primitive versus Structured (Defined)

- Description logics reason with definitions
  - They prefer to have *complete* descriptions
  - A complete definition includes both **necessary** conditions and **sufficient** conditions
- Often impractical or impossible, especially with natural kinds
- A “primitive” definition is an incomplete one
  - Limits amount of classification that can be done automatically
- Example:
  - Primitive: a Person
  - Defined: Parent = Person with at least one child

Classification is very useful

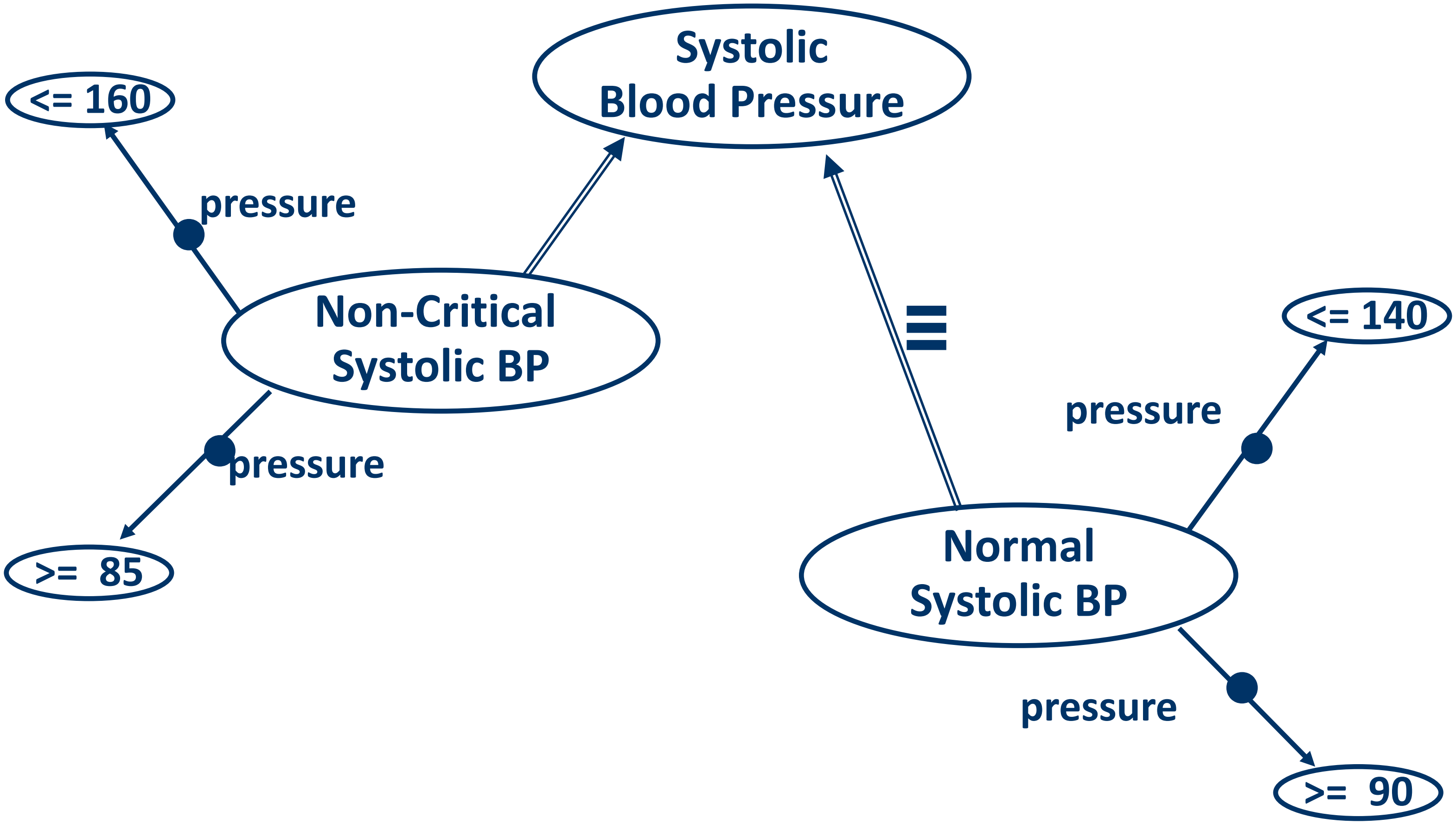
- Classification is a powerful kind of reasoning that is very useful
- Many AI systems can be usefully thought of as doing “heuristic classification”
- Logical classification over structured descriptions and individuals is also quite useful
- But... can classification ever deduce something about an individual other than what classes it belongs to?
- And what does **that** tell us?

Example: Blood Pressure



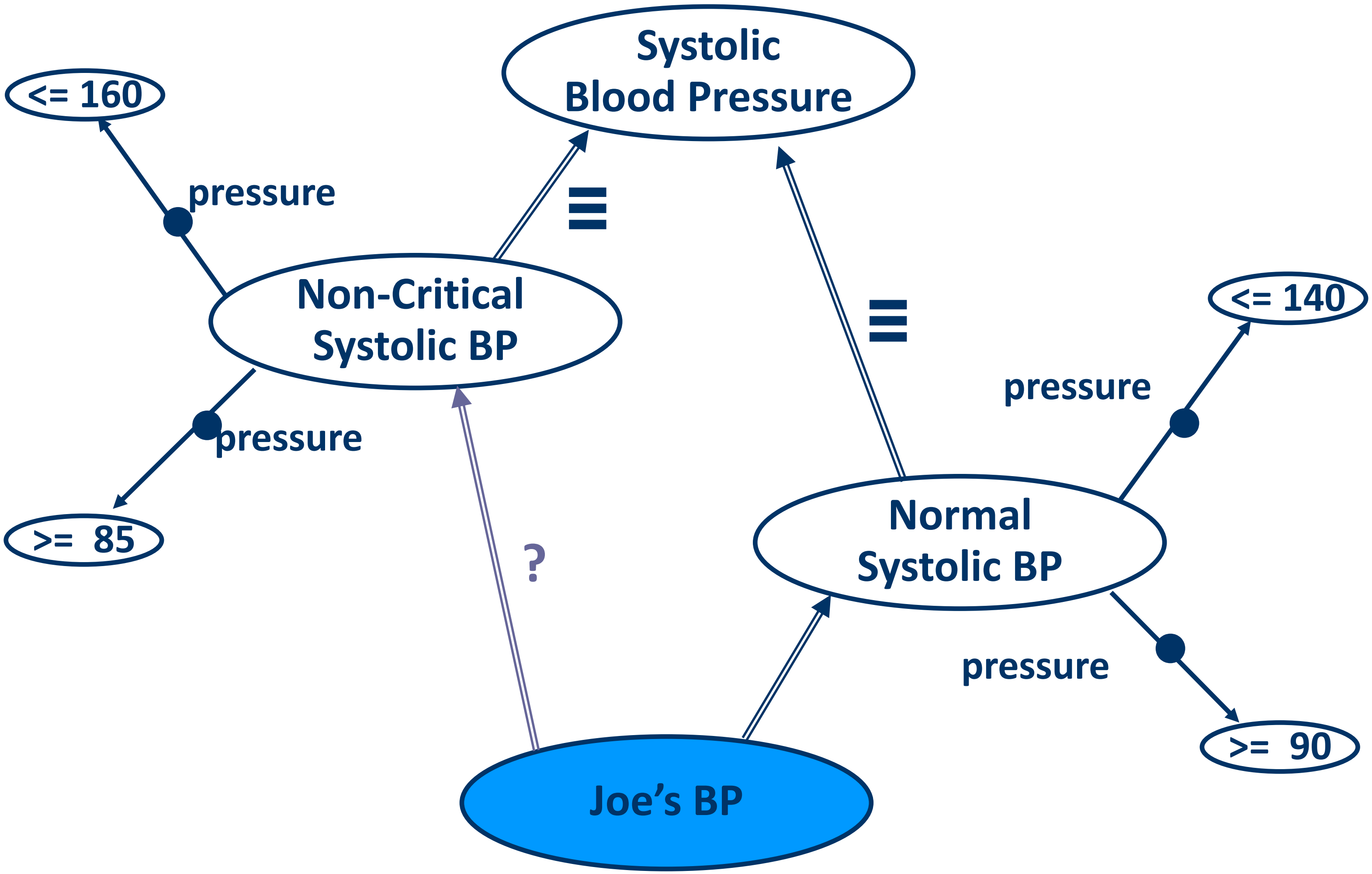
A Non-Critical Blood Pressure is "a Systolic B.P. between 85 and 160."

Example: Blood Pressure

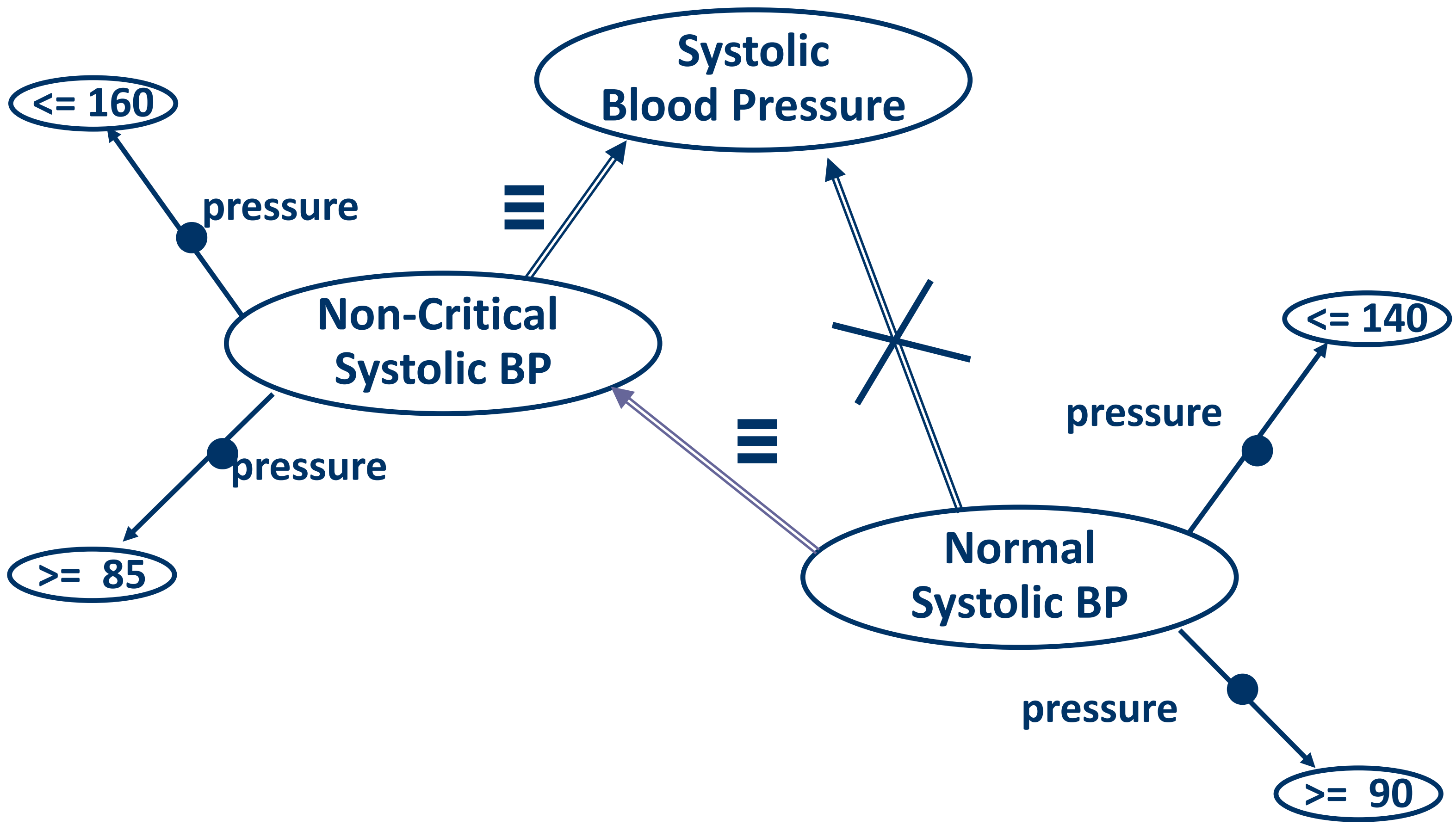


Normal Systolic B.P. is a Systolic B.P. between 90 and 140

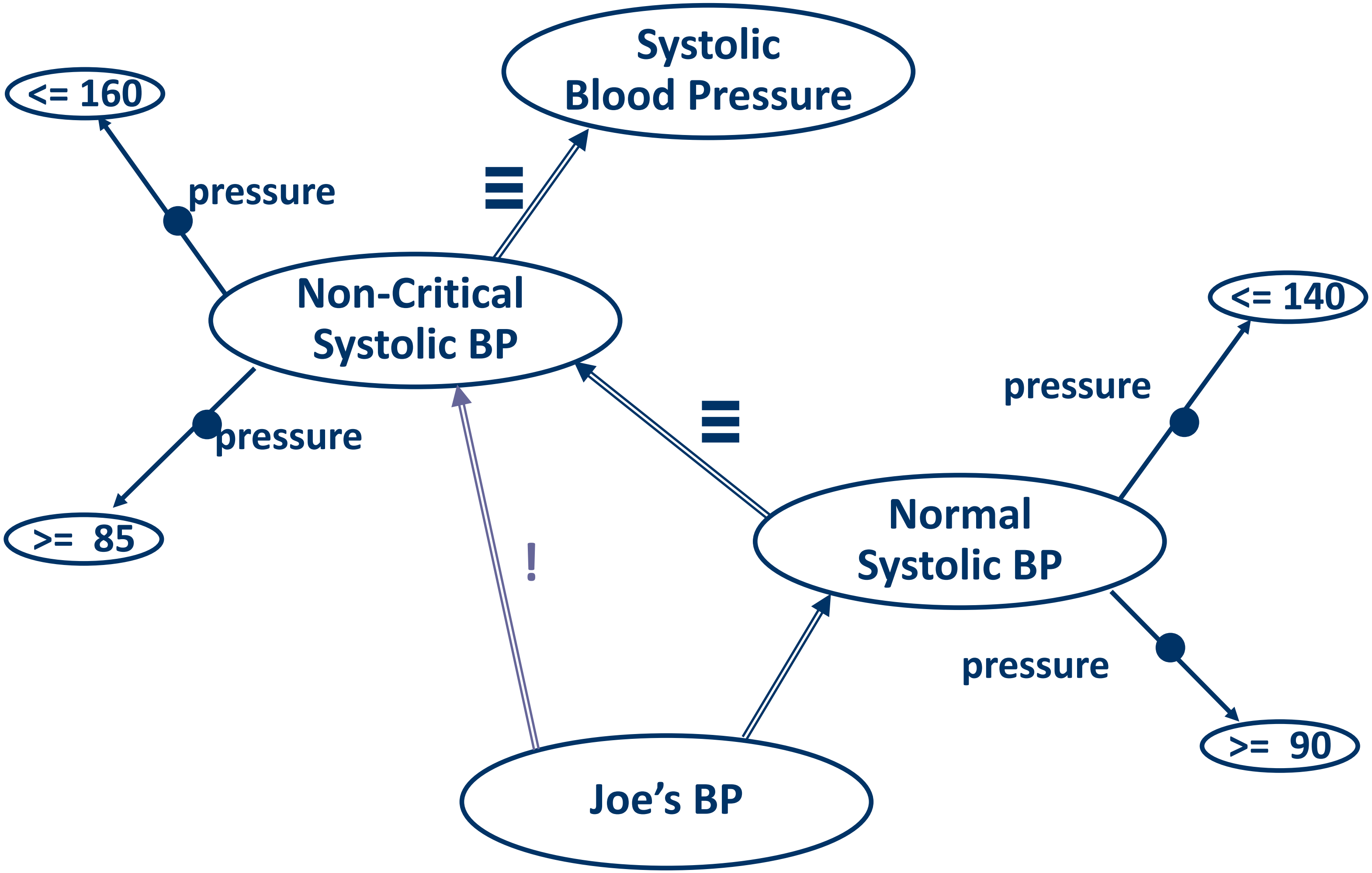
If Joe's BP is Normal is it also Non-Critical?



Classification Infers Normal BP is Subsumed by Non-Critical BP



Answer is Easy to Compute with Classified Concepts



Incidental properties

- We consider properties that are not part of any definition to be **incidental**
- Classification based on non-incidental properties allow inference of incidental properties
- Examples:
  - E.g., **red cars** have been observed to have a high accident rate by insurance companies
  - **Birds weighing more than 25kg** can not fly
  - People with **non-critical blood pressure** require no medication

DL Conclusion

- Description logic was the model for OWL reasoning
- More expressive than rule-based systems without being undecidable or intractable
- It can reason over general statements (e.g., a dog with rabies is a sick animal), unlike most rule-based systems
- It still has limitations, of course
- More powerful logics might be needed in some cases