COMP3411/9814: Artificial Intelligence

Knowledge Representation

Lecture Overview

- The Knowledge Level
- Knowledge Representation
- Ontologies, Taxonomy, Categories and Objects
- Semantic Networks
- Rule based representation
- Inference Networks
- Deduction, Abduction, and Induction

The Knowledge Level

- Knowledge Level Hypothesis. There exists a distinct computer systems level, lying immediately above the symbol level, which is characterized by knowledge as the medium and the principle of rationality as the law of behaviour.
- Principle of Rationality. If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action.
- Knowledge. Whatever can be ascribed to an agent, such that its behaviour can be computed according to the principle of rationality.

“The Knowledge Level” (Newell, 1982)

Knowledge Representation

- Any agent can be described on different level
  - Knowledge level (knowledge ascribed to agent)
  - Logical level (algorithms for manipulating knowledge)
  - Implementation level (how algorithms are implemented)
- Knowledge Representation is concerned with expressing knowledge explicitly in a computer-tractable way (for use by an agent in reasoning) – not the same as Newell's view
- Reasoning attempts to take this knowledge and draw inferences (e.g. answer queries, determine facts that follow from the knowledge, decide what to do, etc.) – as part of the agent architecture

Knowledge Representation and Reasoning

- A knowledge-based agent has at its core a knowledge base
- A knowledge base is an explicit set of sentences about some domain expressed in a suitable formal representation language
- Sentences express facts (true) or non-facts (false)
- Fundamental Questions
  - How do we write down knowledge about a domain/problem?
  - How do we automate reasoning to deduce new facts or ensure consistency of a knowledge base?

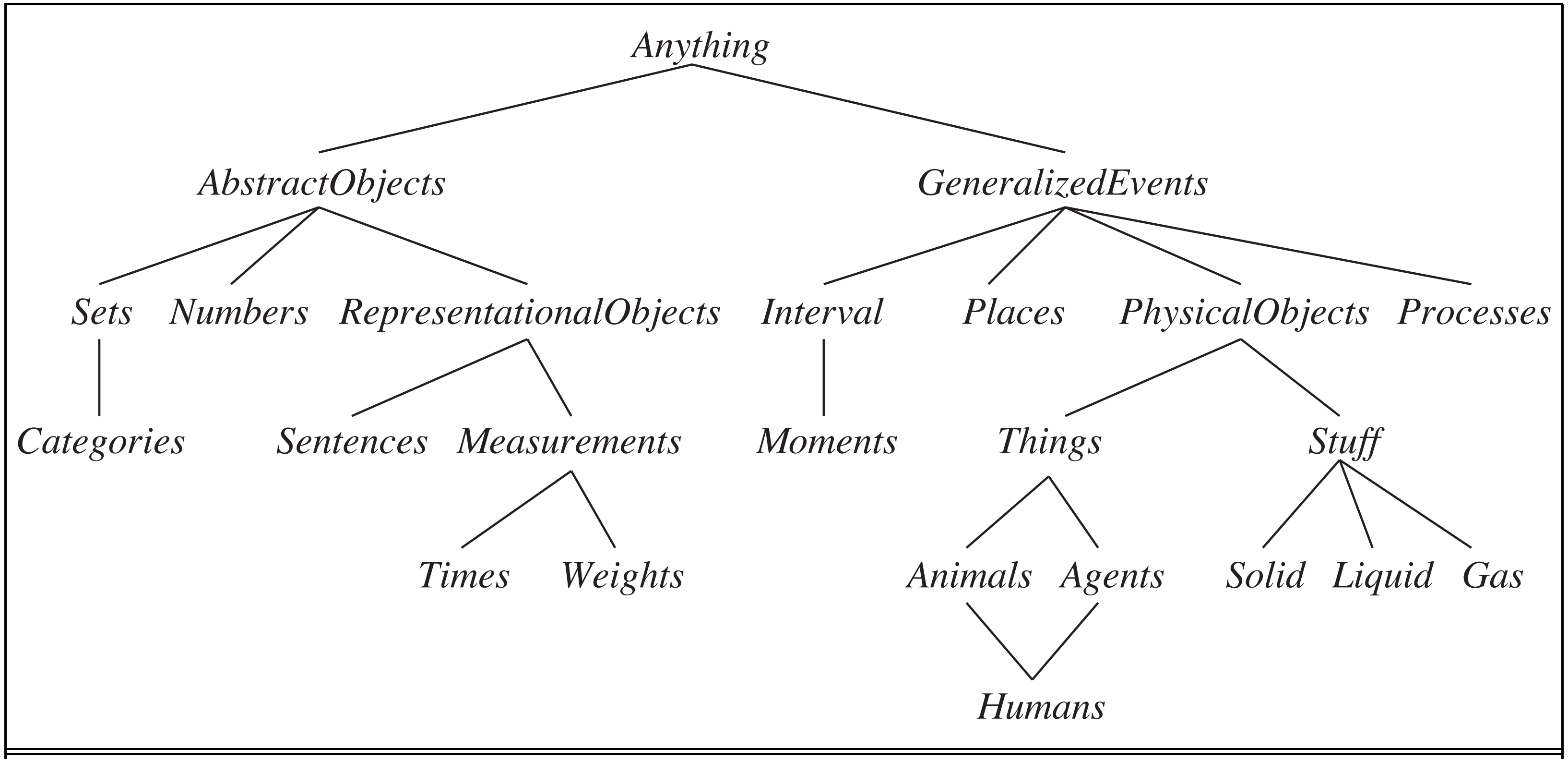
Knowledge representation

- We are looking at the technology for knowledge-based agents: the syntax, semantics, and proof theory of propositional and first-order logic, and the implementation of agents that use these logics.
- We also need to address the question: What *content* to put into such an agent's knowledge base?
  - how to represent facts about the world.

Ontologies and ontological engineering

- A general ontology organizes everything in the world into a hierarchy of categories.
- The prospect of representing *everything* in the world is daunting.
  - We won't actually write a complete description of everything—that would be far too much - but we will leave placeholders where new knowledge for any domain can fit in.
  - We will define what it means to be a physical object, and the details of different types of objects—robots, televisions, books, or whatever—can be filled in later.
- Similar to OO programming framework

Ontologies - example



An ontology of the world,
Each link indicates that the lower concept is a specialization of the upper one. Specializations are not necessarily disjoint;
- a human is both an animal and an agent

Example – Ontologies

- AfPak Ontology
  - Ashraf Ghani is President Ghani – equality
  - Ashraf Ghani is the President of Afghanistan – role
  - Ashraf Ghani is in the government – part of
  - Nangarhar is a province – a kind of
  - Nangarhar is in Afghanistan – part of

  - Bombing implies Attacking – linguistic meaning/semantics

Ontology = Set of such facts

Categories and Objects

- The organization of objects into categories is a vital part of knowledge representation.
- Although interaction with the world takes place at the level of individual objects, *much reasoning takes place at the level of categories.*
- Categories also serve to make predictions about objects once they are classified.
- There are two choices for representing categories in first-order logic
  - predicates and objects.

Categories and Objects

- One infers the presence of certain objects from perceptual input, infers category membership from the perceived properties of the objects, and then uses category information to make predictions about the objects.
  - For example, from its green and yellow mottled skin, one-foot diameter, ovoid shape, red flesh, black seeds, and presence in the fruit aisle, one can infer that an object is a watermelon;
  - from this, one infers that it would be useful for fruit salad.

Categories and Objects

- Categories serve to organize and simplify the knowledge base through inheritance.
- If we say that all instances of the category Food are edible, and if we assert that Fruit is a subclass of Food and Apples is a subclass of Fruit, then we can infer that every apple is edible.
- We say that the individual apples inherit the property of edibility, in this case from their membership in the Food category.

Taxonomic hierarchy

- Subclass relations organize categories into a **taxonomy**, or taxonomic hierarchy.
  - Taxonomies have been used explicitly for centuries in technical fields.
- Examples:
  - The largest such taxonomy organizes about 10 million living and extinct species, many of them beetles into a single hierarchy
  - library science has developed a taxonomy of all fields of knowledge
- Taxonomies are also an important aspect of general commonsense knowledge.

Categories and Objects and FOL

- First-order logic makes it easy to state facts about categories, either by relating objects to categories or by quantifying over their members.

An object is a member of a category.
$BB_9 \in Basketballs$

A category is a subclass of another category.
$Basketballs \subset Balls$

All members of a category have some properties.
$(x \in Basketballs) \Rightarrow Spherical(x)$

Members of a category can be recognized by some properties.
$Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5'' \wedge x \in Balls \Rightarrow x \in Basketballs$

A category as a whole has some properties.
$Dogs \in DomesticatedSpecies$

Reasoning system for categories

- Categories are the primary building blocks of large-scale knowledge representation schemes.
- There are two closely related families of systems:
  - semantic networks provide graphical aids for visualizing a knowledge base and efficient algorithms for inferring properties of an object on the basis of its category membership; and
  - **description logics** provide a formal language for constructing and combining category definitions and efficient algorithms for deciding subset and superset relationships between categories.

Semantic Networks

- Fact , Objects, Attributes and Relationships
  - Relationships exist among instances of objects and classes of objects.
- Attributes and relationships can be represented as a network, known as an associative network or semantic network
- We can build a model of the subject area of interest

Semantic networks

In 1909, Charles S. Peirce proposed a graphical notation of nodes and edges called **existential graphs** that he called "the logic of the future."

A long-running debate between advocates of "logic" and advocates of "semantic networks."

- the semantics networks—at least those with well-defined semantics—*are* a form of logic.
- The notation that semantic networks provide for certain kinds of sentences is often more convenient,
- the underlying concepts
  - objects, relations, quantification, and so on…
  - are the same as in logic.

Knowledge and Semantic Networks

Facts

- Facts may be static, in which case they can be written into the knowledge base.
    - Static facts need not be permanent, but they change sufficiently infrequently that changes can be accommodated by updating the knowledge base when necessary.
- Facts may be transient - apply at a specific instance only or for a single run of the system

Knowledge and Semantic Networks

- One of the most important aspects of semantic networks is their ability to represent default values for categories.
  - The knowledge base may contain *defaults* that can be used as facts in the absence of transient facts.

Example – A simple set of statements

- My car is a car
- A car is a vehicle
- A car has four wheels
- A car's speed is 0 mph
- My car is red
- My car is in my garage
- My garage is a garage
- A garage is a building
- My garage is made from brick
- My car is in the High Street
- The High Street is a street
- A street is a road

Example – facts, objects and relations

- My car is a **car**
- A car is a **vehicle**
- A car has four wheels
- A car's speed is 0 mph
- My car is red
- My car is in my garage
- My garage is a **garage**
- A garage is a **building**
- My garage is made from brick
- My car is in the High Street
- The High Street is a **street**
- A street is a **road**

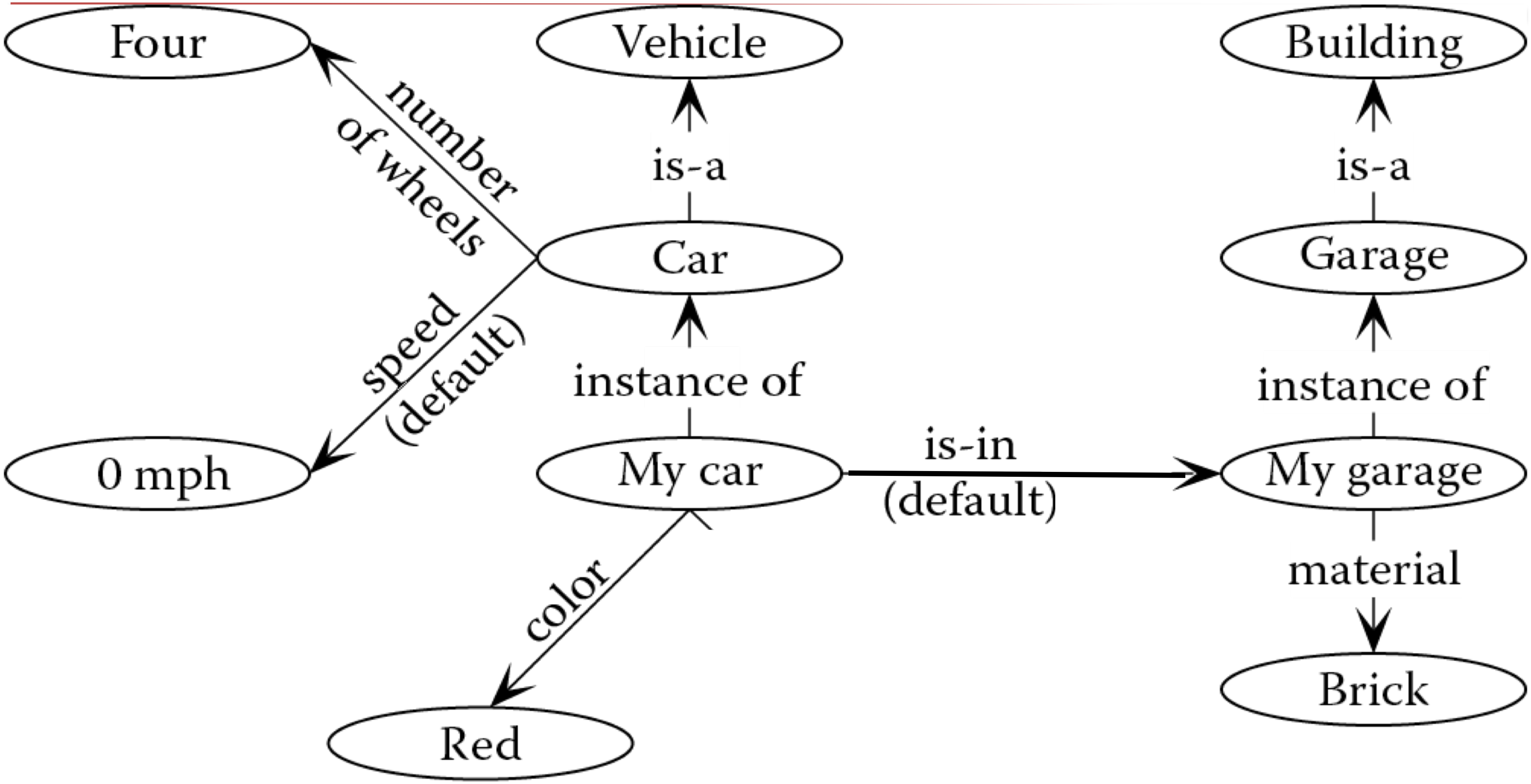
Example – facts, objects and relations

- My car is a **car** (static relationship)
- A car is a **vehicle** (static relationship)
- A car has four wheels (static attribute)
- A car’s speed is 0 mph (default attribute)
- My car is red (static attribute)
- My car is in my garage (default relationship)
- My garage is a **garage** (static relationship)
- A garage is a **building** (static relationship)
- My garage is made from brick (static attribute)
- My car is in the High Street (transient relationship)
- The High Street is a **street** (static relationship)
- A street is a **road** (static relationship)

Semantic Network - representation

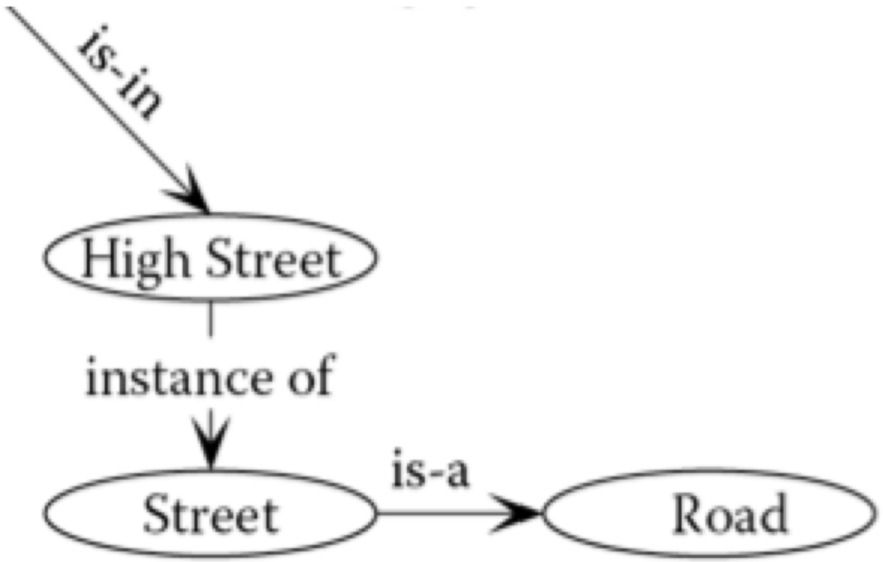
- Semantic Network consists of
  - Facts, Objects, Attributes and Relationships
  - Relationships exist among instances of objects and classes of objects.
- Attributes and relationships can be represented as a network, known as an associative network or **semantic network**
- We can build a model – a semantic network representation of our example

A semantic network (with a default)

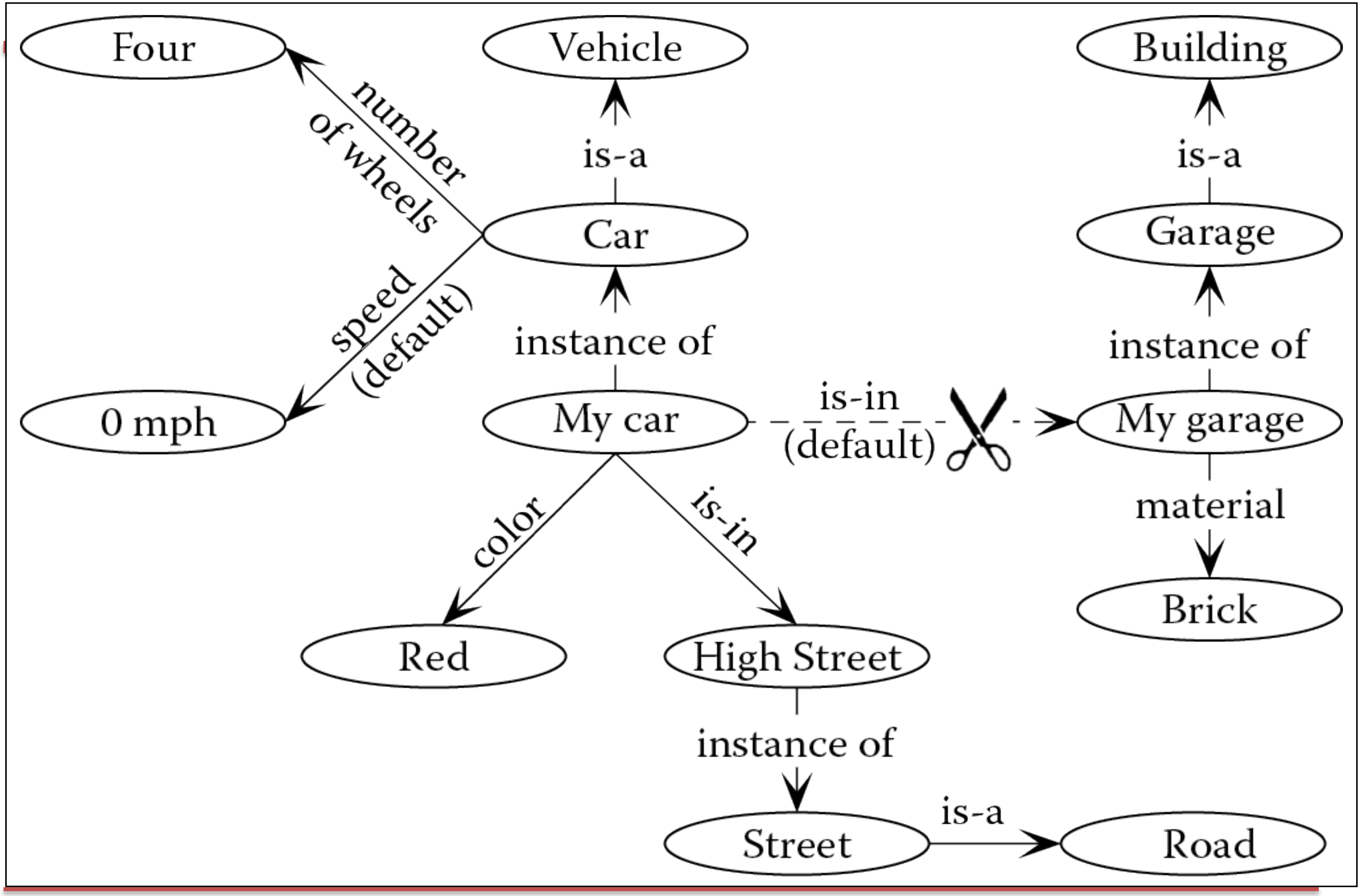


New fact

My car is in the High Street
The High Street is a street
A street is a road



A semantic network with an overridden default



Object instances and classes of objects

It is important to understand the distinction between object instances and classes of objects:

- Car and vehicle are both classes of object. They are linked by the relation "is a", which could be expanded to "is a subclass of."
  - The direction of the arrow is important, indicating that "car is a vehicle" and not "vehicle is a car."
- My car, on the other hand, is a unique entity. There may be an identical car somewhere, but they are distinct instances. So the relationship between my car and car takes the form "is an instance of."

Semantic Networks - reasoning

- The central reasoning mechanism in a semantic network is inheritance: a category and its instances inherit the properties of the categories that contain them.
- The advantages of the semantic network architecture are that it is easy and natural to use, its meaning can be defined precisely, and the inheritance mechanism is easy to implement and efficient.

Main Components Of Semantic Networks

- Lexical component: nodes denoting physical objects or links are relationships between objects; labels denote the specific objects and relationships
- Structural component: the links or nodes from a diagram which is directed.
- Semantic component: Here the definitions are related only to the links and label of nodes, whereas facts depend on the approval areas.
- Procedural part: constructors permit the creation of the new links and nodes. The removal of links and nodes are permitted by destructors.

Semantic Networks

- Semantic Networks are usually used for:
  - Representing data
  - Revealing structure (relations, proximity, relative importance)
  - Supporting conceptual edition
  - Supporting navigation

Semantic network

- At this stage, the semantic network is an unstructured collection of object instances, classes, and attributes.
  - In this representation, attributes are treated in the same way as relationships.

- Alternative representation is with *agents, objects, and frames*:
  - the relationships and attributes are represented explicitly in a formalized manner.
  - This more structured approach turns these ideas into a powerful programming technique.

The Knowledge Base – rule based

- The simplest type of rule is called a production rule and takes the form

```
if <condition> then <conclusion>
```

- Production rule for dealing with the payroll of ACME, Inc., might be

```
rule r1_1

if the employer of Person is acme

then the salary of Person becomes large.
```

The Knowledge Base – rule based

```
rule r1_1

if the employer of Person is acme

then the salary of Person becomes large.
```

- Part of the attraction of using production rules is that they can often be written in a form that closely resembles natural language, as opposed to a computer language.

```
/* fact f1_1 */

the employer of joe_bloggs is acme.
```

- The use of capitalization indicates that “Person” is a local variable that can be replaced by a constant value, such as “joe_bloggs” or “mary_smith”, through the process of pattern matching.

The Knowledge Base – rule based

```
rule r1_1
if the employer of Person is acme
then the salary of Person becomes large.
```

```
rule r1_2
if the salary of Person is large
or the job_satisfaction of Person is true
then the professional_contentment of Person becomes true.
```

This in turn may lead to the generation of a new derived fact.

Rules r1_1 and r1_2 are interdependent since the conclusion of one can satisfy the condition of the other.

Uncertainty in rules

- Rules such as rule r1_1 are a useful way of expressing many types of knowledge!
- This is not always the case, In the case of rule r1_1, *uncertainty may arise* from three distinct sources:
  - Uncertain evidence. (Perhaps we are not certain that Joe Bloggs works for ACME.)
  - Uncertain link between evidence and conclusion. (We cannot be certain that an ACME employee earns a large salary, we just know that it is likely.)
  - Vague rule. (What is a “large” salary anyway?)

Uncertainty in rules

- In the case of rule r1_1, uncertainty may arise from three distinct sources:
  - Uncertain evidence. (Perhaps we are not certain that Joe Bloggs works for ACME.)
  - Uncertain link between evidence and conclusion. (We cannot be certain that an ACME employee earns a large salary, we just know that it is likely.)
  - Vague rule. (What is a “large” salary anyway?)

Bayesian inference

Uncertainty in rules

- In the case of rule r1_1, uncertainty may arise from three distinct sources:
  - Uncertain evidence. (Perhaps we are not certain that Joe Bloggs works for ACME.)
  - Uncertain link between evidence and conclusion. (We cannot be certain that an ACME employee earns a large salary, we just know that it is likely.)
  - Vague rule. (What is a "large" salary anyway?)

Bayesian inference

fuzzy logic

The Knowledge Base – rule based

```
rule r1_1

if the employer of Person is acme

then the salary of Person becomes large.
```

```
/* fact f1_1 */

the employer of joe_bloggs is acme.
```

```
rule r1_2

if the salary of Person is large

or the job_satisfaction of Person is true

then the professional_contentment of Person becomes true.
```

```
/* derived fact f1_2 */

the salary of joe_bloggs is large.
```

Inference Networks

- The interdependencies among rules, such as r1_1 and r1_2 given earlier, define a network
- The inference network illustrates the way in which facts can be logically combined to form new facts or conclusions
- The facts can be combined logically using "and" and "or".
  - Thus, professional contentment is true if either job satisfaction or large salary is true (or both are true).

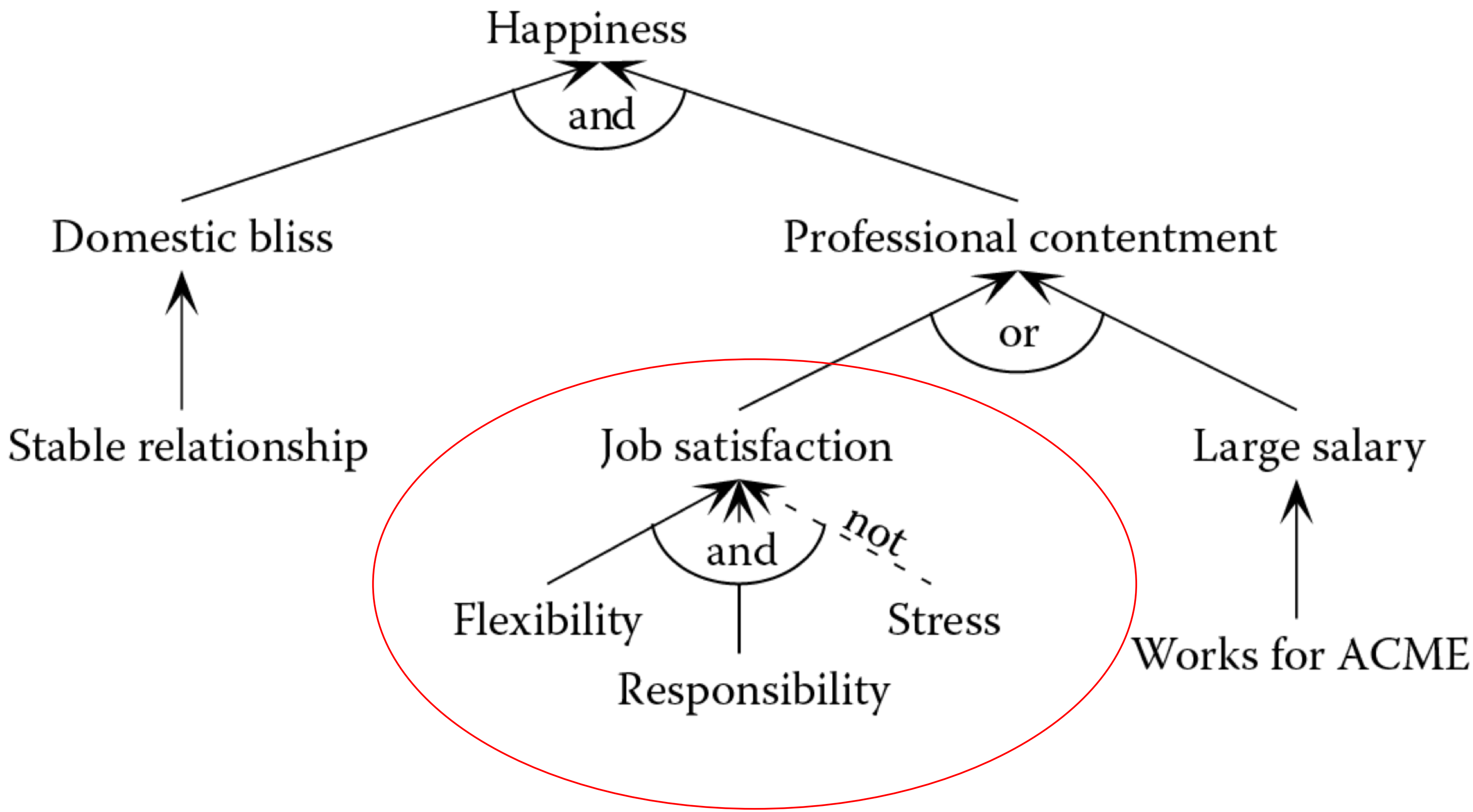
Inference Networks

Another type of connection is “not”.

- Job satisfaction is achieved through *flexibility*, *responsibility*, and *the absence of stress*.

- The negative relationship with stress is shown by a dashed line labeled “not”.

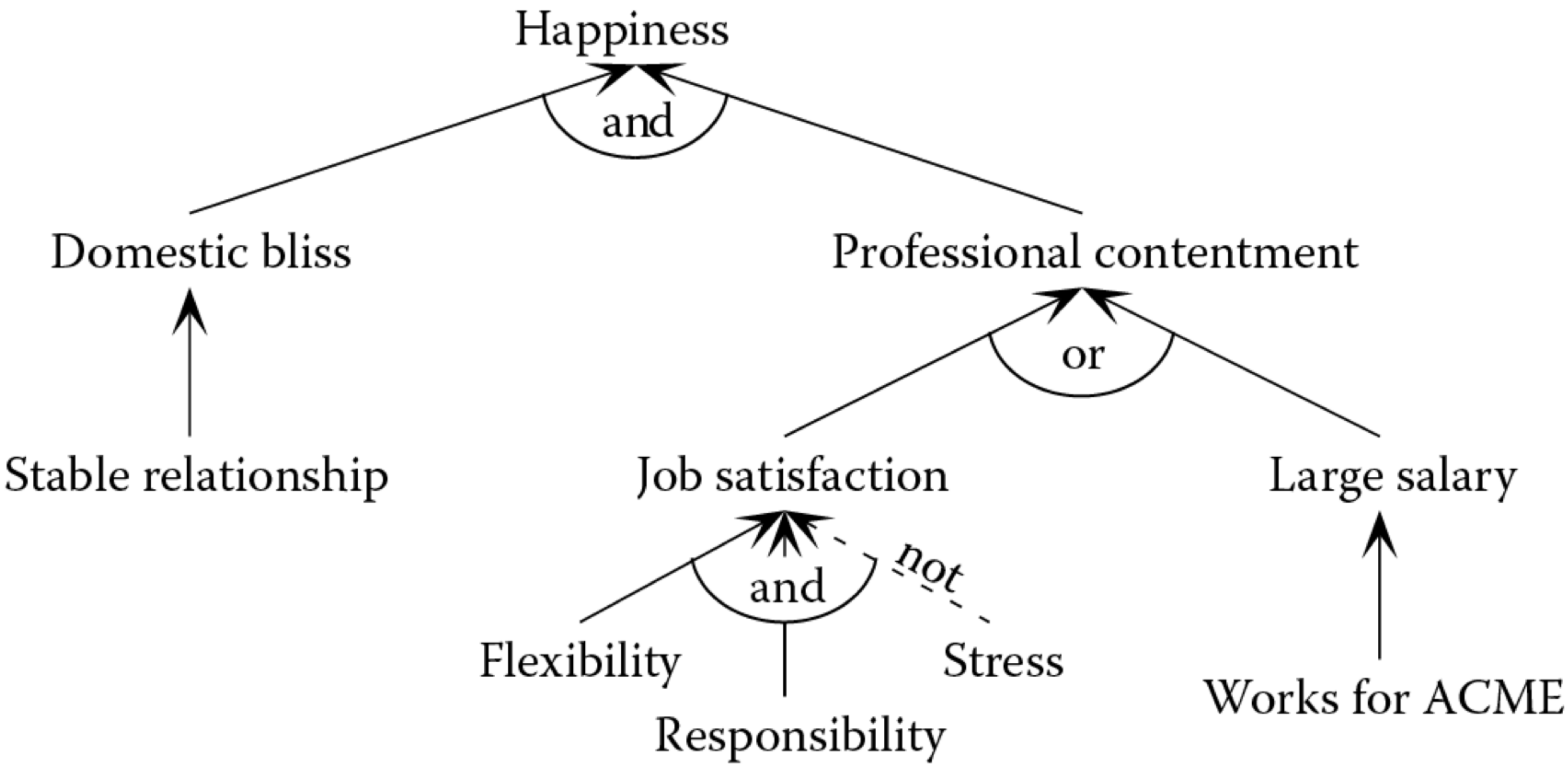
An inference network



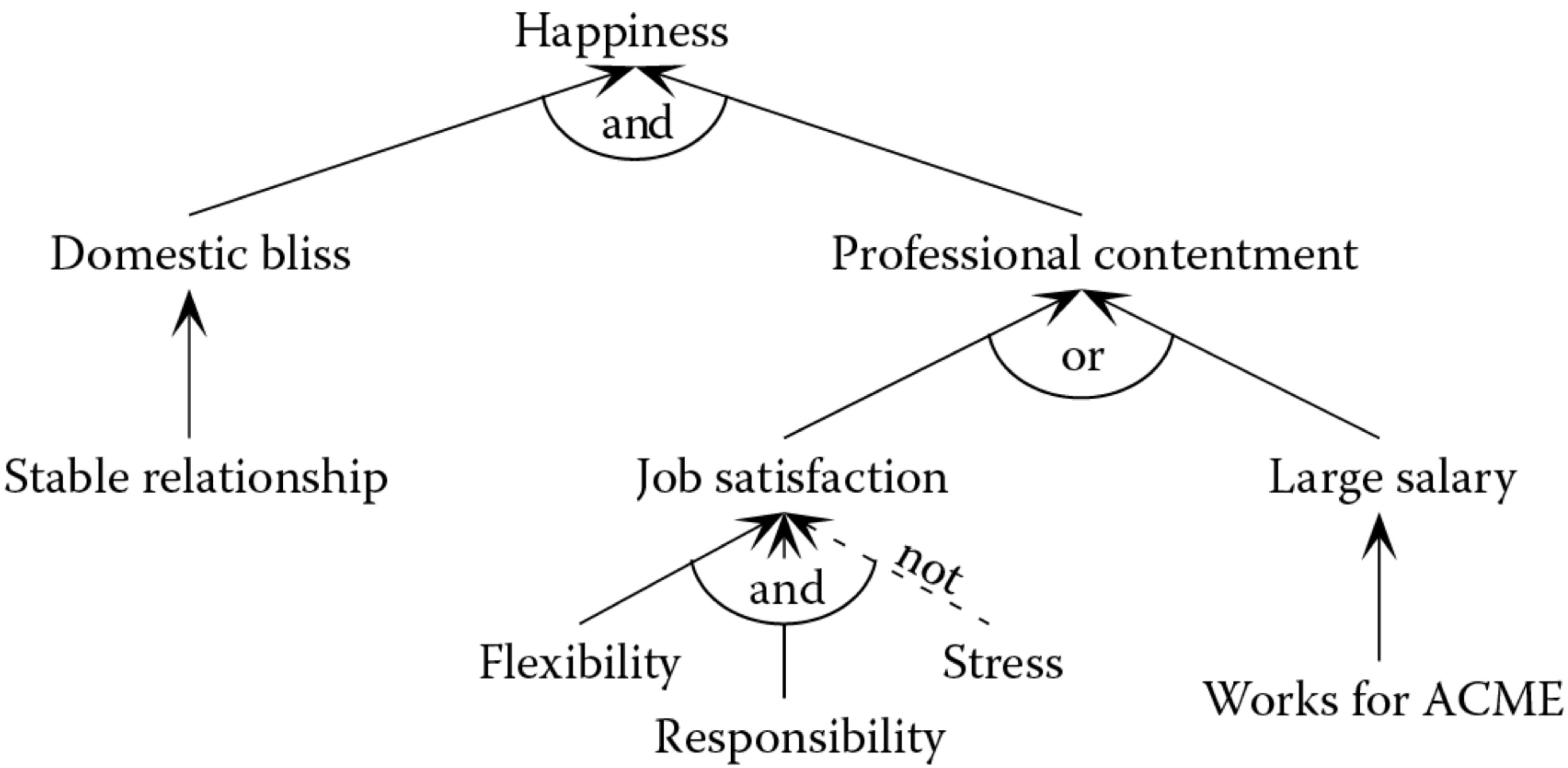
An inference network

- An inference network can be constructed by
  - taking *facts* and working out what conditions have to be met for those facts to be true.
  - After these conditions are found, they can be added to the diagram and linked by a *logical expression* (such as and, or, not).
  - This usually involves breaking down a complex logical expression into smaller parts.

An inference network

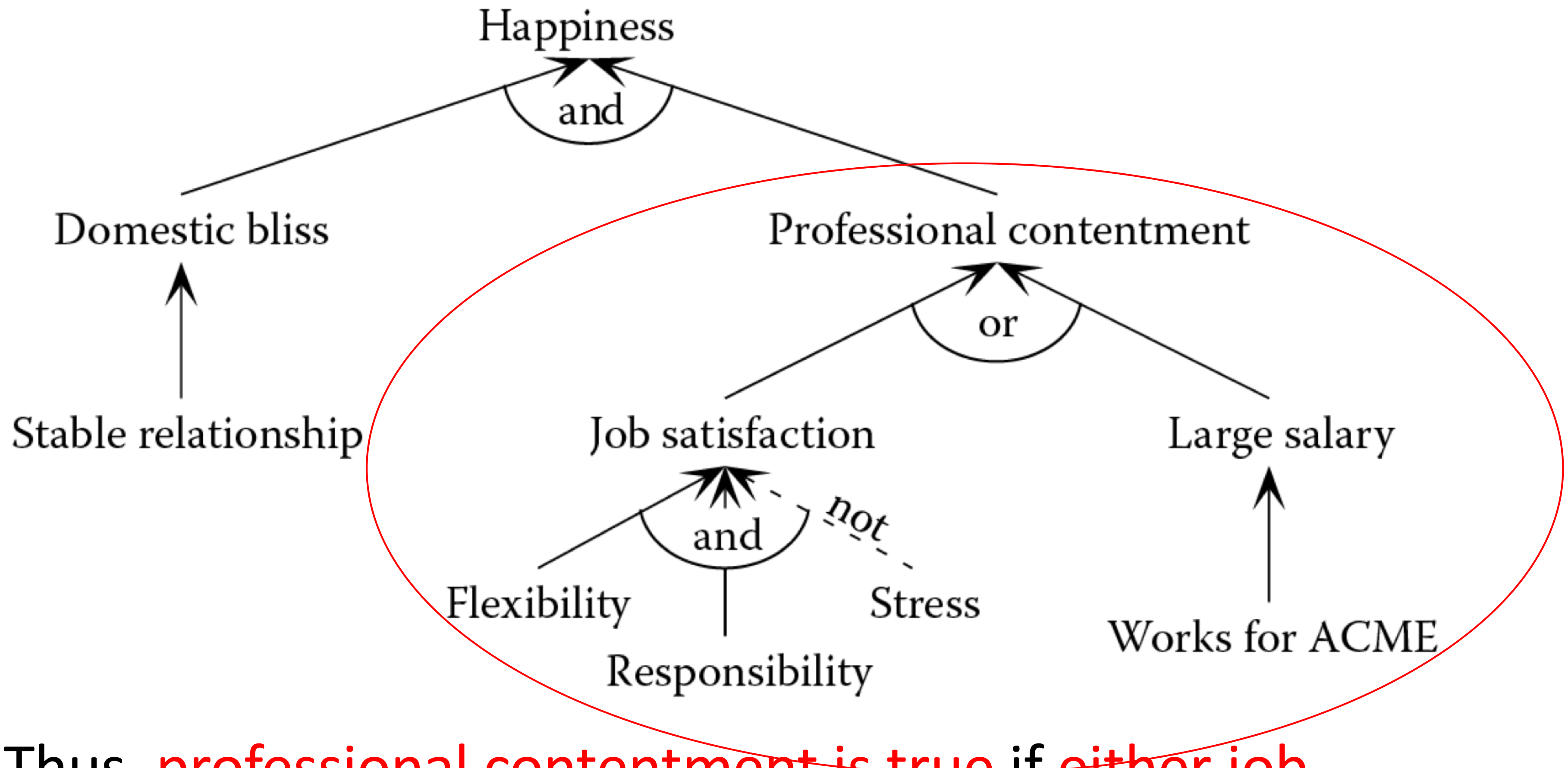


Inference Networks



- Thus, professional contentment is true if either job satisfaction or large salary is true (or both are true).

Inference Networks



- Thus, professional contentment is true if either job satisfaction or large salary is true (or both are true).

Deduction, Abduction and Induction

- The rules that make up the inference network about "happiness" and the network taken as a whole, are used to link cause and effect:

  if <cause> then <effect>

We can infer that:

if

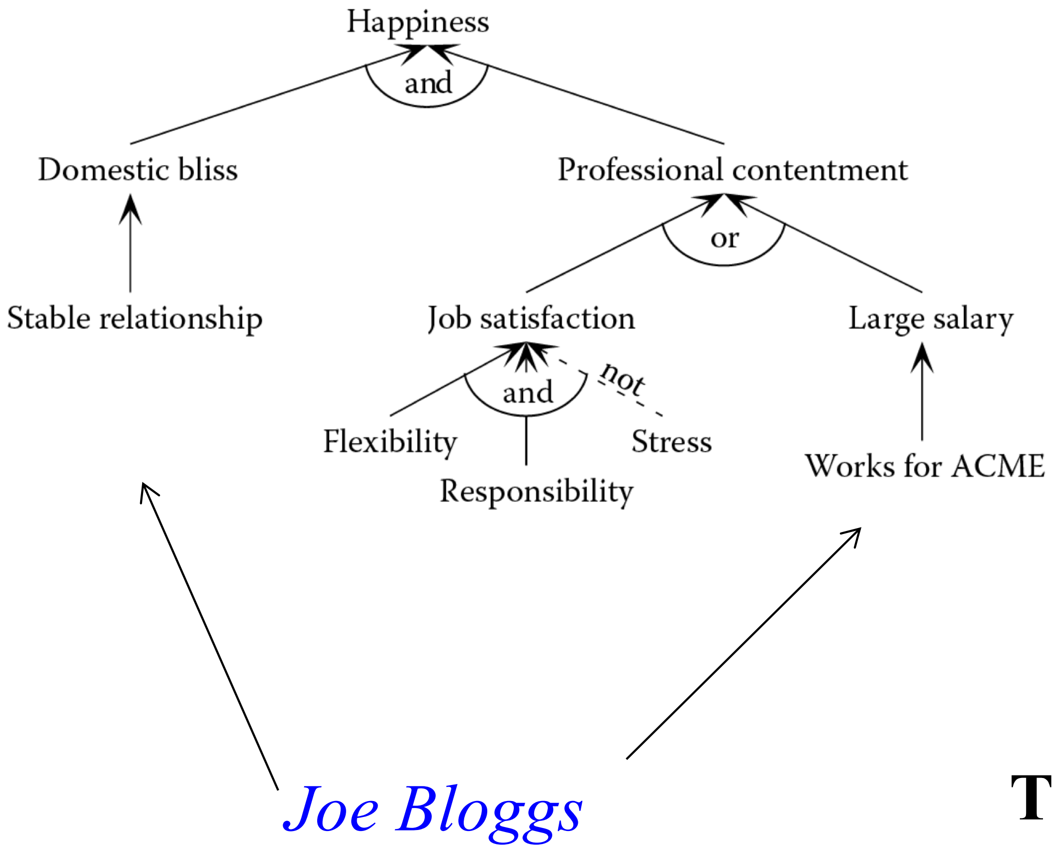
Joe Bloggs works for ACME and is in a stable relationship (the causes),

then

he is happy (the effect).

Deduction, Abduction, and Induction

if <cause> then <effect>

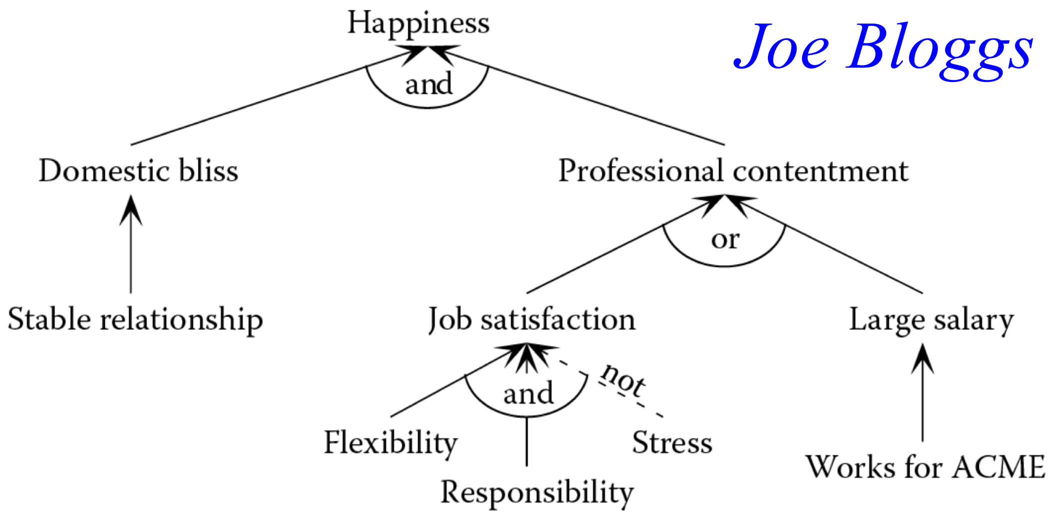


We can infer that:
if *Joe Bloggs* works for ACME and is in a stable relationship (the causes),
then
he is happy (the effect).

This is the process of deduction.

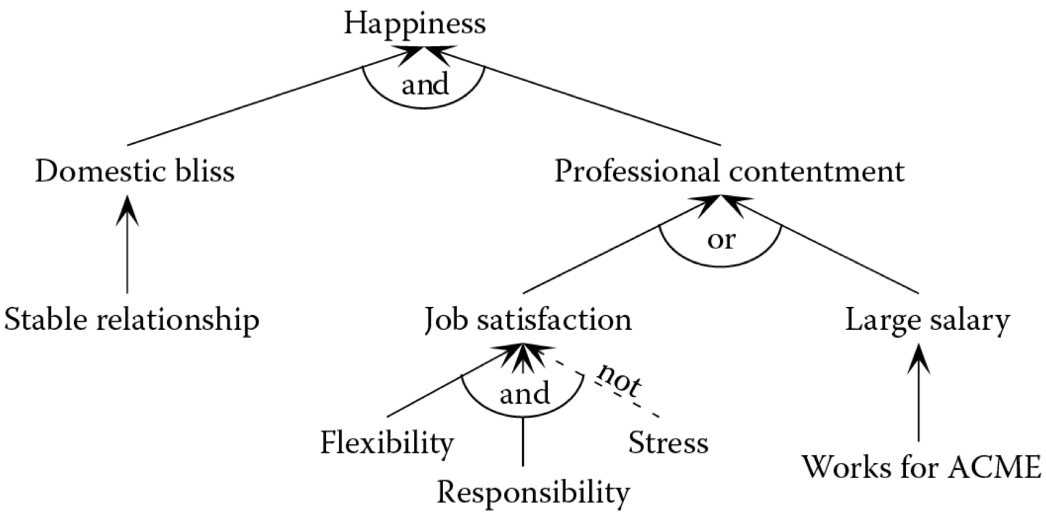
Deduction, Abduction, and Induction

- Abduction - Many problems, such as diagnosis, involve reasoning in the reverse direction, that is, we wish to ascertain a cause, given an effect.
- Given the observation that Joe Bloggs is happy, we can infer by abduction that Joe Bloggs enjoys domestic bliss and professional contentment.



Deduction, Abduction, and Induction

if <cause> then <effect>



We can infer that:
Joe Bloggs he is happy (the effect)
then
he
works for ACME and is in a stable relationship (the causes),

This is the process of abduction.

Deduction, Abduction, and Induction

- If we have many examples of cause and effect, we can infer the rule (or inference network) that links them.
- For instance, if every employee of ACME that we have met earns a large salary, then we might infer

rule r1_1

if the employer of Person is acme

then the salary of Person becomes large.

Inferring a rule from a set of example cases of cause and effect is termed induction.

Deduction, Abduction, and Induction

We can summarize deduction, abduction, and induction as follows:

- deduction: cause + rule $\Rightarrow$ effect
- abduction: effect + rule $\Rightarrow$ cause
- induction: cause + effect $\Rightarrow$ rule

Closed-world assumption

- The inference network represents a closed world where nothing is known beyond its boundaries.
- As each node represents a possible state of some aspect of the world, a model of the current overall state of the world can be maintained.
- Such a model is dependent on the extent of the relationships between the nodes in the inference network.
- This is only a valid conclusion if the inference network shows all of the ways in which a person can find happiness.

Artificial Intelligence

- A simple definition (Hopgood 2005):
  - Artificial intelligence is the science of mimicking human mental faculties in a computer.

- The ultimate achievement in this field would be to construct a machine that can mimic or exceed human mental capabilities, including reasoning, understanding, imagination, recognition, creativity, and emotions.

A spectrum of intelligent behavior

- expertise
- planning
- adaptability
- interaction
- language
- common sense
- vision / perception
- coordination
- regulation
- reaction

Level of understanding

References

- Poole &Mackworth, Artificial Intelligence: Foundations of Computational Agents, Chapter 5
- Russell & Norvig, *Artificial Intelligence: a Modern Approach*, Chapter 12.
- Adrian A. Hopgood. *Intelligent Systems for Engineers and Scientists (3rd Edition)*. CRC Press, 2011 – Chapter 1.