Outline for today

- Learning a theory and new concepts in first-order logic.
- The debate about structure in people's mental representations of concepts
  - Hierarchies or hidden units?
  - Logical relations or hidden units?
  - Definitions or prototypes?

First-order logic

- A language for talking about things, their properties and their relations.
  - Specific facts about particular objects
  - General laws about kinds of objects and their relations.
- In other words, a language for expressing theories of the world.

First-order logic in cognition

- Intuitive theories and theory-based concepts.
  - Rational agent

$$\begin{aligned}&rational(x) \Leftrightarrow \forall y, a\\ &\qquad (desires(x, y) \wedge believes(x, causes(a, y)) \Rightarrow todo(x, a))\end{aligned}$$

  - Magnetism

$$\forall x, y\ magnet(x) \wedge magnetic(y) \wedge near(x, y) \Rightarrow\ forceBetween(x, y)$$

$$\forall x, y\ \ forceBetween(x, y) \Leftrightarrow\ forceBetween(y, x)$$

$$\forall x\ magnet(x) \Rightarrow magnetic(x)$$

First-order logic

- More powerful than propositional logic
  - Contains predicates and quantifiers.
  - Can express general laws, e.g., a syllogism

All men are mortal
Socrates is a man

Socrates is a mortal

$$\begin{aligned}&\left(\forall x\ man(x) \Rightarrow mortal(x)\right) \land man(Socrates)\\ &\quad \Rightarrow mortal(Socrates)\end{aligned}$$

First-order logic

- Less powerful than second-order logic
  - Cannot apply predicates or quantifiers to predicates
  - Cannot express generalizations about all concepts, or all concepts of a certain kind

$$\begin{array}{l} \text{All F are G} \\ \text{X is an F} \\ \hline \text{X is a G} \end{array}$$

$$\forall f, g, x \left[\left(\forall y \ f(y) \Rightarrow g(y) \right) \wedge f(x) \Rightarrow g(x) \right]$$

Or, grue vs. green….

Representational power

- Like grammars, FOL representations are abstract, recursive, and generates structures of greatly varying (potentially infinite) complexity.
- E.g., kinship domain

Kinship domain

- Define arbitrarily complex relations

$$\forall x, y\ greatgrandmother(x, y) \Leftrightarrow \exists w, z$$
$$(mother(x, w) \wedge parent(w, z) \wedge parent(z, y))$$

- Define new relations in terms of other new relations.

$$\forall x, y\ cousin(x, y) \Leftrightarrow \exists w, z$$
$$(sibling(w, z) \wedge parent(w, x) \wedge parent(z, y))$$

$$\forall x, y\ sibling(x, y) \Leftrightarrow \exists w\ parent(w, x) \wedge parent(w, y) \wedge \neg(x = y)$$

- Generate arbitrarily complex structures.

$John = father(Tony)$ $\qquad$ $Ann = mother(Susan)$

$brother(Tony, Ann)$ $\qquad$ $Susan = greatgrandmother(Joe)$

- Generate infinite structures.

$$\forall x\ (\exists y\ (y = mother(x)) \qquad \forall x\ \neg(x = mother(x))$$

A kinship dataset

father(Christopher, Arthur)
father(Christopher, Victoria)
father(Andrew, James)
father(Andrew, Jennifer)
father(James, Colin)
father(James, Charlotte)

mother(Penelope, Arthur)
mother(Penelope, Victoria)
mother(Christine, James)
mother(Christine, Jennifer)
mother(Victoria, Colin)
mother(Victoria, Charlotte)

husband(Christopher, Penelope)
husband(Andrew, Christine)
husband(Arthur, Margaret)
husband(James, Victoria)
husband(Charles, Jennifer)

wife(Penelope, Christopher)
wife(Christine, Andrew)
wife(Margaret, Arthur)
wife(Victoria, James)
wife(Jennifer, Charles)

son(Arthur, Christopher)
son(Arthur, Penelope)
son(James, Andrew)
son(James, Christine)
son(Colin, Victoria)
son(Colin, James)

daughter(Victoria, Christopher)
daughter(Victoria, Penelope)
daughter(Jennifer, Andrew)
daughter(Jennifer, Christine)
daughter(Charlotte, Victoria)
daughter(Charlotte, James)

brother(Arthur, Victoria)
brother(James, Jennifer)
brother(Colin, Charlotte)

sister(Victoria, Arthur)
sister(Jennifer, James)
sister(Charlotte, Colin)

uncle(Arthur, Colin)
uncle(Charles, Colin)
uncle(Arthur, Charlotte)
uncle(Charles, Charlotte)

aunt(Jennifer, Colin)
aunt(Margaret, Colin)
aunt(Jennifer, Charlotte)
aunt(Margaret, Charlotte)

nephew(Colin, Arthur)
nephew(Colin, Jennifer)
nephew(Colin, Margaret)
nephew(Colin, Charles)

niece(Charlotte, Arthur)
niece(Charlotte, Jennifer)
niece(Charlotte, Margaret)
niece(Charlotte, Charles)

A more intuitive representation

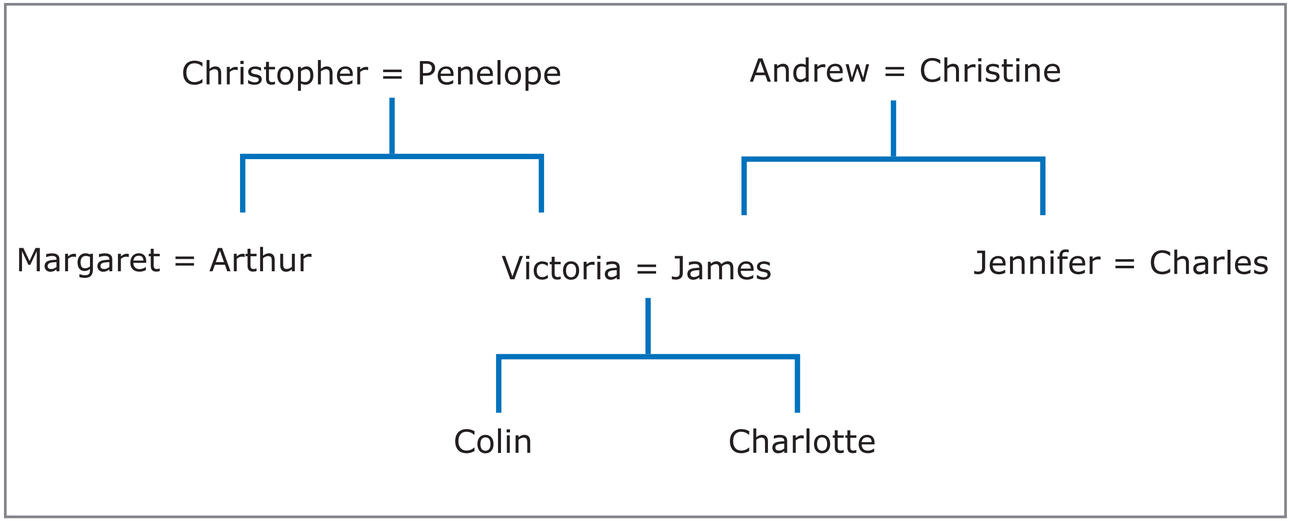


Figure by MIT OCW.

- Relations: spouse (“=“), parent (solid line)
- Attribute: male or female (type of name)
- Define other relations in terms of basic relations spouse, parent, and the attributes male, female.

A kinship dataset

spouse(Christopher,Penelope)
spouse(Andrew,Christine)
spouse(Arthur,Margaret)
spouse(James,Victoria)
spouse(Charles,Jennifer)

female(Penelope)
female(Christine)
female(Margaret)
female(Victoria)
female(Jennifer)
female(Charlotte)
NOT female(Colin)

parent(Penelope, Arthur)
parent(Penelope, Victoria)
parent(Christine, James)
parent(Christine, Jennifer)
parent(Victoria, Colin)
parent(Victoria, Charlotte)

spouse(x,y) <=> spouse(y,x)
NOT female(x) <= spouse(x,y) AND female(y)
parent(x,y) <= spouse(x,z) AND parent(z,y)

father(x,y) <=> parent(x,y) AND NOT female(x)
mother(x,y) <=> parent(x,y) AND female(x)
husband(x,y) <=> spouse(x,y) AND NOT female(x)
wife(x,y) <=> spouse(x,y) AND female(x)
son(x,y) <=> parent(y,x) AND NOT female(x)
daughter(x,y) <=> parent(y,x) AND female(x)
sibling(x,y) <=> parent(z,x) AND parent(z,y) AND ~(x=y)
brother(x,y) <=> sibling(x,y) AND NOT female(x)
sister(x,y) <=>sibling(x,y) AND female(x)
uncle(x,y) <=> (parent(z,y) AND brother(x,z))
OR (aunt(z,y) AND spouse(x,z))
aunt(x,y) <=> (parent(z,y) AND sister(x,z)) OR
OR (uncle(z,y) AND spouse(x,z))
nephew(x,y) <=> (parent(z,x) AND sibling(y,z) AND NOT female(x))
OR (nephew(x,z) AND spouse(y,z))
niece(x,y) <=> (parent(z,x) AND sibling(y,z) AND female(x))
OR (niece(x,z) AND spouse(y,z))

A kinship dataset

spouse(Christopher,Penelope)
spouse(Andrew,Christine)
spouse(Arthur,Margaret)
spouse(James,Victoria)
spouse(Charles,Jennifer)

female(Penelope)
female(Christine)
female(Margaret)
female(Victoria)
female(Jennifer)
female(Charlotte)
NOT female(Colin)

parent(Penelope, Arthur)
parent(Penelope, Victoria)
parent(Christine, James)
parent(Christine, Jennifer)
parent(Victoria, Colin)
parent(Victoria, Charlotte)

```
spouse(x,y) <=> spouse(y,x)
NOT female(x) <= spouse(x,y) AND female(y)
parent(x,y) <= spouse(x,z) AND parent(z,y)

father(x,y) <=> parent(x,y) AND NOT female(x)
mother(x,y) <=> parent(x,y) AND female(x)
husband(x,y) <=> spouse(x,y) AND NOT female(x)
wife(x,y) <=> spouse(x,y) AND female(x)
son(x,y) <=> parent(y,x) AND NOT female(x)
daughter(x,y) <=> parent(y,x) AND female(x)
sibling(x,y) <=> parent(z,x) AND parent(z,y) AND ~(x=y)
brother(x,y) <=> sibling(x,y) AND NOT female(x)
sister(x,y) <=>sibling(x,y) AND female(x)
uncle(x,y) <=> (parent(z,y) AND brother(x,z))
                    OR (aunt(z,y) AND spouse(x,z))
aunt(x,y) <=> (parent(z,y) AND sister(x,z)) OR
                    OR (uncle(z,y) AND spouse(x,z))
nephew(x,y) <=> (parent(z,x) AND sibling(y,z) AND NOT female(x))
                    OR (nephew(x,z) AND spouse(y,z))
niece(x,y) <=> (parent(z,x) AND sibling(y,z) AND female(x))
                    OR (niece(x,z) AND spouse(y,z))
```

Properties of this representation

- Useful for compression (memory)
- Useful for predicting unknown relations
  - Margaret is Arthur’s wife. What else do we know about her?

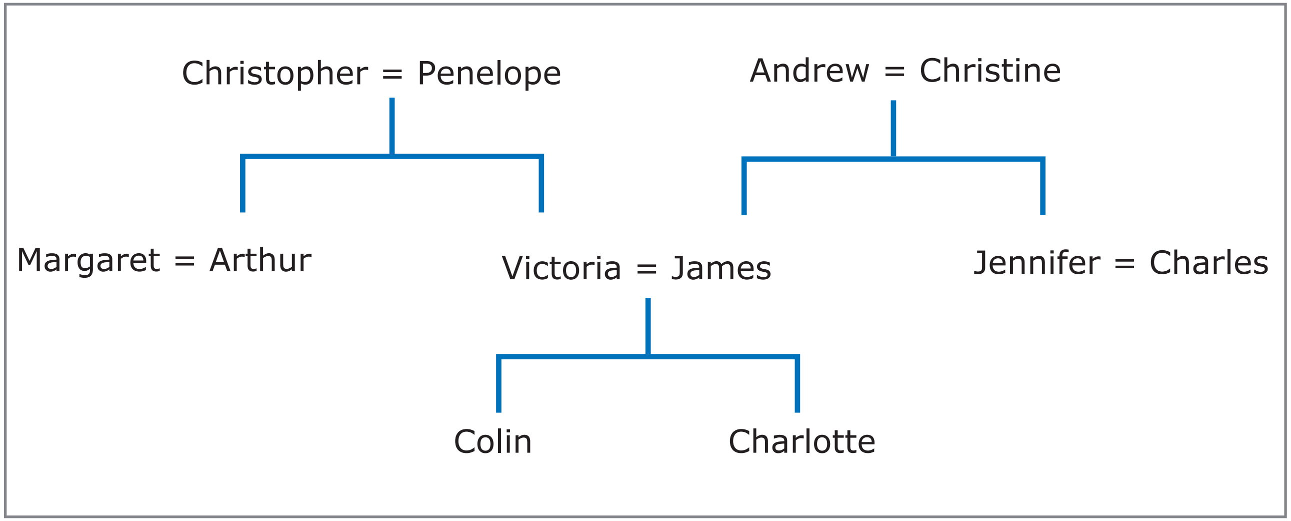


Figure by MIT OCW.

Reasoning about kinship

- Consider a new person, Boris.
  - Is the mother of Boris's father his grandmother?
  - Is the mother of Boris's sister his mother?
  - Is Boris's uncle his grandfather?
  - Is the son of Boris's sister his son?

- Is first-order logic sufficient (or necessary) to explain these inferences?

Properties of this representation

- Useful for compression (memory)
- Useful for predicting unknown relations
- Could it be discovered?
  - Hard problem: Given a good basis set, learn to represent all relations most compactly.
  - Harder problem: Discover a good basis set, which consists of all novel concepts

    parent, *spouse*, *female*

    (or, *child*, *spouse*, *male*)

How can we learn this theory?

- Computational level (Marr level 1)
  - Theory is learnable if it compresses the data.

- Algorithmic level (Marr level 2)
  - Inductive Logic Programming (ILP) algorithms.
  - Require procedures for inventing new predicates in the course of searching for the most compact theory.

How to discover new basis concepts

- Predicate invention via inverse resolution:

Intra-construction: $$\frac{p \,\square\!\leftarrow A, B \square\square \quad \square \quad \square \quad \square \quad p \square\!\leftarrow A, C}{q \,\square\!\leftarrow B \qquad p \square\!\leftarrow A, q \square \quad \square \quad q \square\!\leftarrow C}$$

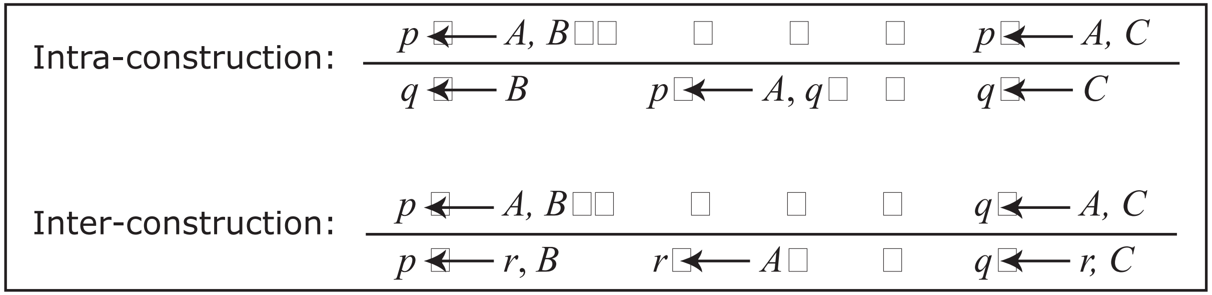
Inter-construction: $$\frac{p \,\square\!\leftarrow A, B \square\square \quad \square \quad \square \quad \square \quad q \square\!\leftarrow A, C}{p \,\square\!\leftarrow r, B \qquad r \square\!\leftarrow A \square \quad \square \quad q \square\!\leftarrow r, C}$$

Figure by MIT OCW.

Mill’s Canons of Induction (1840’s)

How to discover new basis concepts

- Predicate invention via inverse resolution:



Intra-construction:

$$\frac{p \leftarrow A, B \qquad p \leftarrow A, C}{q \leftarrow B \qquad p \leftarrow A, q \qquad q \leftarrow C}$$

Inter-construction:

$$\frac{p \leftarrow A, B \qquad q \leftarrow A, C}{p \leftarrow r, B \qquad r \leftarrow A \qquad q \leftarrow r, C}$$

Figure by MIT OCW.

son(x,y) <= brother(x,z) AND father(y,z)
son(x,y) <= brother(x,z) AND mother(y,z)

---

parent(x,y) <= father(x,y)
parent(x,y) <= mother(x,y)
son(x,y) <= brother(x,z) AND parent(y,z)

```
son(x,y) <= brother(x,z) AND father(y,z)
son(x,y) <= brother(x,z) AND mother(y,z)
son(x,y) <= husband(x,z) AND father(y,x)
son(x,y) <= husband(x,z) AND mother(y,x)
son(x,y) <= uncle(x,z) AND father(y,x)
son(x,y) <= uncle(x,z) AND mother(y,x)
daughter(x,y) <= sister(x,z) AND father(y,z)
daughter(x,y) <= sister(x,z) AND mother(y,z)

…
```

```
parent(x,y) <= father(x,y)
parent(x,y) <= mother(x,y)
   son(x,y) <= brother(x,z) AND parent(y,z)
   son(x,y) <= husband(x,z) AND parent(y,x)
   son(x,y) <= uncle(x,z) AND parent(y,x)
daughter(x,y) <= sister(x,z) AND parent(y,z)

…
```

How to discover new basis concepts

- Predicate invention via inverse resolution:

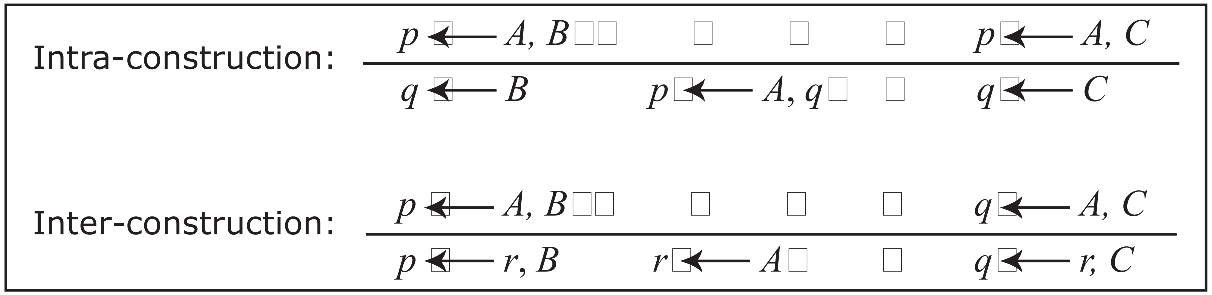


Figure by MIT OCW.

mother(x,y) <= son(y,z) AND wife(x,z)
mother(x,y) <= son(y,z) AND husband(z,x)

---

spouse(x,y) <= wife(x,y)
spouse(x,y) <= husband(y,x)
mother(x,y) <= son(y,z) AND spouse(x,z)

How to discover new basis concepts

- Predicate invention via inverse resolution:

Intra-construction:
$$\frac{p \leftarrow A, B \qquad p \leftarrow A, C}{q \leftarrow B \qquad p \leftarrow A, q \qquad q \leftarrow C}$$

Inter-construction:
$$\frac{p \leftarrow A, B \qquad q \leftarrow A, C}{p \leftarrow r, B \qquad r \leftarrow A \qquad q \leftarrow r, C}$$

Figure by MIT OCW.

son(x,y) <= parent(y,x) AND brother(x,z)
son(x,y) <= parent(y,x) AND father(x,z)

---

male-relative-of(x,y) <= brother(x,y)
male-relative-of(x,y) <= father(x,y)
son(x,y) <= parent(y,x) AND male-relative-of(x,z)

- Clustering on relational roles: e.g., males

father(Christopher, Arthur)
father(Christopher, Victoria)
father(Andrew, James)
father(Andrew, Jennifer)
father(James, Colin)
father(James, Charlotte)

mother(Penelope, Arthur)
mother(Penelope, Victoria)
mother(Christine, James)
mother(Christine, Jennifer)
mother(Victoria, Colin)
mother(Victoria, Charlotte)

husband(Christopher, Penelope)
husband(Andrew, Christine)
husband(Arthur, Margaret)
husband(James, Victoria)
husband(Charles, Jennifer)

wife(Penelope, Christopher)
wife(Christine, Andrew)
wife(Margaret, Arthur)
wife(Victoria, James)
wife(Jennifer, Charles)

son(Arthur, Christopher)
son(Arthur, Penelope)
son(James, Andrew)
son(James, Christine)
son(Colin, Victoria)
son(Colin, James)

daughter(Victoria, Christopher)
daughter(Victoria, Penelope)
daughter(Jennifer, Andrew)
daughter(Jennifer, Christine)
daughter(Charlotte, Victoria)
daughter(Charlotte, James)

brother(Arthur, Victoria)
brother(James, Jennifer)
brother(Colin, Charlotte)

sister(Victoria, Arthur)
sister(Jennifer, James)
sister(Charlotte, Colin)

uncle(Arthur, Colin)
uncle(Charles, Colin)
uncle(Arthur, Charlotte)
uncle(Charles, Charlotte)

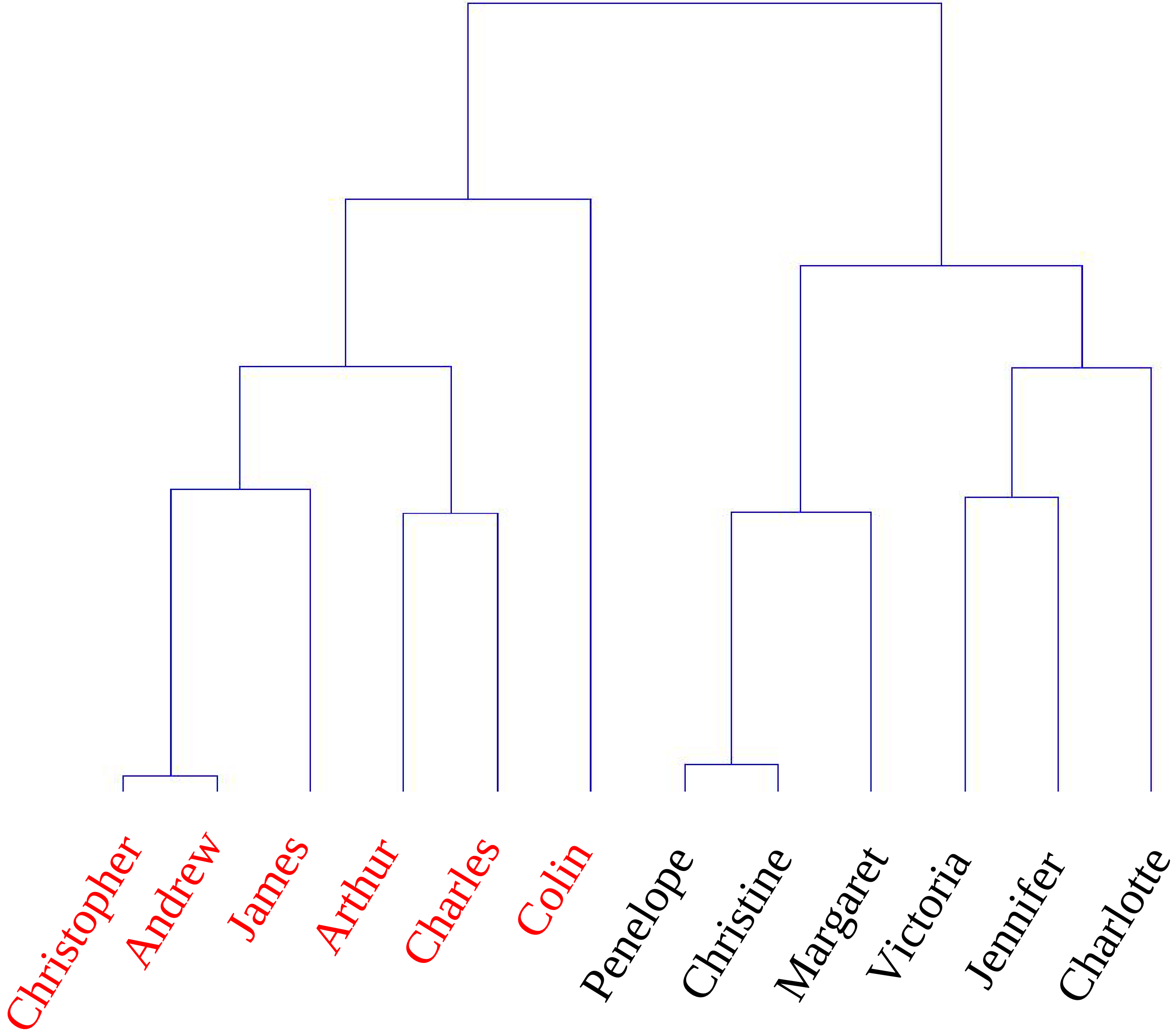
aunt(Jennifer, Colin)
aunt(Margaret, Colin)
aunt(Jennifer, Charlotte)
aunt(Margaret, Charlotte)

nephew(Colin, Arthur)
nephew(Colin, Jennifer)
nephew(Colin, Margaret)
nephew(Colin, Charles)

niece(Charlotte, Arthur)
niece(Charlotte, Jennifer)
niece(Charlotte, Margaret)
niece(Charlotte, Charles)

- Clustering on relational roles: e.g., males

	Is a father	Is a mother	Is a husband	Is a wife	Is a son	Is a daughter	Is a brother	Is a sister	Is an uncle	Is an aunt	Is a nephew	Is a niece
Christopher	x		x									
Andrew	x		x									
James	x		x		x		x					
Arthur			x		x		x		x			
Charles			x						x			
Colin					x		x				x	
Penelope		x		x								
Christine		x		x								
Margaret				x						x		
Victoria		x		x		x		x				
Jennifer				x		x		x		x		
Charlotte						x		x				x


Christopher
Andrew
James
Arthur
Charles
Colin
Penelope
Christine
Margaret
Victoria
Jennifer
Charlotte

How general is this problem?

- Parent, spouse, male are in fact hidden concepts to the child.
- But in real life:
  - Have only sparse data on relations, which makes the problem hard (DEMO: family_tree.m)
  - Have attribute data (hair length, clothes, size, vocal register, etc.), which makes it easier.
- Can we develop more realistic algorithms for this kind of concept learning?
- Could clustering on pairs of people allows us to learn new relations, e.g. spouse(x,y)?

An answer to Fodor?

- Some of these new concepts can be simply defined in terms of old concepts:
  - E.g., spouse(x,y) <=> husband(x,y) OR wife(x,y)
- Others are atoms, defined only implicitly by the role they play in a system of relations to other concepts.
  - E.g., male(x)
  - Maybe spouse (or “married”) is primitive as well?
    - Which sounds more natural, “X is ‘married’ to Y if X is Y’s husband or Y’s wife”, or “X is Y’s ‘husband’ if X is married to Y and X is a man”?

An answer to Fodor?

- Some of these new concepts can be simply defined in terms of old concepts:
  - E.g., spouse(x,y) <=> husband(x,y) OR wife(x,y)
- Others are atoms, defined only implicitly by the role they play in a system of relations to other concepts.
  - E.g., male(x)
  - Maybe spouse (or “married”) is primitive as well?
- A new paradigm for concept learning:
  - Rather than learning new concepts that are simple to represent in terms of old concepts, learn new concepts in terms of which old concepts are simpler to represent.

Constructing new concepts via theory learning

One way to see what the approach comes to is to reflect on how one learned the concepts of elementary physics, or anyway, how I did. When I took my first physics course, I was confronted with quite a bit of new terminology all at once: ‘energy’, ‘momentum’, ‘acceleration’, ‘mass’, and the like. As should be no surprise to anyone who noted the failure of positivists to define theoretical terms in observation language, I never learned any definitions of these new terms in terms I already knew. Rather, what I learned was how to use the new terminology – I learned certain relations among the new terms themselves (e.g., the relation between force and mass, neither of which can be defined in old terms), some relations between the new terms and old terms, and, most importantly, how to generate the right numbers in answers to questions posed in the new terminology.

Block (1986)

Other cognitive applications for "theory learning"

- Grammar induction
- Physical objects
- Intentional agents
- Biological species
- Analogy to frameworks for learning simpler representations: MDS, PCA, hierarchical clustering

Simple object world

Demo: objects.m

Summary

- Structured representations are important
  - Abstract
  - Recursive
  - Generative
- New primitive concepts can be learned
  - Learning the most parsimonious theory
- How to combine structured representations and statistical inference?
  - Statistical parsing in language
  - Statistical grammar induction
  - Probabilistic inferences about kin relations.
  - Statistical learning of relational concepts and theories.