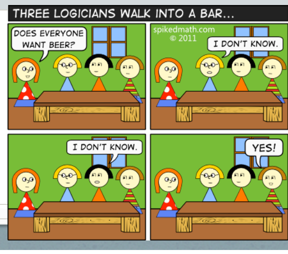
Propositional and First-Order Logic

Chapter 7.4–7.8, 8.1–8.3, 8.5



Material from Dr. Marie desJardin, and Dr. Tim Oates, Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer

Bookkeeping

- HW4 last night
- HW5 out today after class
  - Due evening of 11/20
- New Eleusis code today
  - Getting checked for dumb mistakes
- Reminder: no HW6!
- Project designs back tomorrow

Today's Class

- Last time we talked about **knowledge-based agents**
  - Agents have knowledge about the world, own state, etc.
- Knowledge is stored in a **Knowledge Base** (KB)
  - Formally represented statements
  - If it's something the agent knows, it's in the KB
  - Add: New discoveries, new sensor data, new conclusions
  - Delete: Old (discovered to be outdated) facts
- Agents can reason over knowledge in the KB
- But how is it represented and reasoned over?

Logic Roadmap

- Propositional logic
  - Problems with propositional logic
- First-order logic
  - Properties, relations, functions, quantifiers, …
  - Terms, sentences, wffs, axioms, theories, proofs, …
  - Extensions to first-order logic
- Logical agents
  - Reflex agents
  - Representing change: situation calculus, frame problem
  - Preferences on actions
  - Goal-based agents

Propositional Logic

Chapter 7.4-7.8

Big Ideas in Logic

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic:** simple foundation, fine for many AI problems
- **First order logic** (FOL): much more expressive KR language, more commonly used in AI
- **Many variations** on classical logics are used: Horn logic, higher order logic, three-valued logic, probabilistic logics, etc.

Material from Dr. Tim Oates

Propositional Logic Syntax

- **Logical constants**: true, false
- **Propositional symbols**: P, Q, S, ... (**atomic sentences**)
- **Parentheses**: (…)
- Sentences are built with **connectives**:

∧ ...and	[conjunction]
∨ ...or	[disjunction]
⇒...implies	[implication / conditional]
⇔..is equivalent	[biconditional]
¬ ...not	[negation]

- **Literal**: atomic sentence or negated atomic sentence

Propositional Logic (PL)

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols
  - E.g., P and Q
- User defines the **semantics** (meaning) of each propositional symbol:
  - P=“*It's hot*”
  - Q=“*It's humid*”

PL Sentences

- A **sentence** (or **well formed formula**) is:
  - Any symbol is a sentence
  - If **S** is a sentence, then **¬S** is a sentence
  - If **S** is a sentence, then **(S)** is a sentence
  - If **S** and **T** are sentences, then so are **(S ∨ T), (S ∧ T), (S → T),** and **(S ↔ T)**
- A sentence is created by any (finite) number of applications of these rules

Examples of PL Sentences

- (P ∧ Q) → R
  “If it is hot and humid, then it is raining”
- Q → P
  “If it is humid, then it is hot”
- Q
  “It is humid.”
- We're free to choose better symbols, e.g.:
  Ho = “It is hot”
  Hu = “It is humid”
  R = “It is raining”

Some Terms

- The meaning, or **semantics**, of a sentence determines its **interpretation**
- Given the truth values of all symbols in a sentence, it can be **evaluated** to determine its **truth value** (True or False)
- A **model** for a KB is a **possible world**—an assignment of truth values to propositional symbols that makes each sentence in KB True
  - E.g.: it is both hot and humid.

Model for a KB

- Let the KB be [P∧Q→R, Q → P]
- What are the possible models?
- Consider all possible assignments of {T|F} to P, Q and R and check truth tables

P: it's hot
Q: it's humid
R: it's raining

PQR	*{T/F}*
FFF	
FFT	
FTF	
FTT	
TFF	
TFT	
TTF	
TTT	

Model for a KB

- Let the KB be [P∧Q→R, Q → P, Q]
- What are the possible models?
- Consider all possible assignments of {T|F} to P, Q and R and check truth tables

P: it's hot
Q: it's humid
R: it's raining

- R is true in every model of the KB
- This KB entails that R is True

PQR	{T/F}
FFF	
FFT	
FTF	
FTT	
TFF	
FT	
TF	
TT	

More Terms

- **Valid sentence** or **tautology**: True under all interpretations, no matter the semantics or what the world is actually like.
  - "It's raining or it's not raining."
- **Inconsistent sentence** or **contradiction**: False under all interpretations. The world is never like what it describes.
  - "It's raining and it's not raining."
- **P entails Q** (P ⊨ Q): whenever P is True, so is Q. In other words, all models of P are also models of Q.

Truth Tables

- Truth tables are used to define logical connectives
- And to determine when a complex sentence is true given the values of the symbols in it

Truth tables for the five logical connectives

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

Example of a truth table used for a complex sentence

P	H	$P \lor H$	$(P \lor H) \land \neg H$	$((P \lor H) \land \neg H) \Rightarrow P$
False	False	False	False	True
False	True	True	False	True
True	False	True	True	True
True	True	True	False	True

On "implies": P → Q

- → is a **logical connective**
- So *P→ Q* is a logical sentence and has a truth value, i.e., is either true or false
- If we add this sentence to a KB, it can be used by an inference rule, *Modes Ponens*, to derive/infer/prove Q if P is also in the KB
- Given a KB where P=True and Q=True, we can also derive/infer/prove that P→Q is True

P → Q

- When is *P→Q* true? Check all that apply
  - ❑ P=Q=true
  - ❑ P=Q=false
  - ❑ P=true, Q=false
  - ❑ P=false, Q=true

P → Q

- When is *P→Q* true? Check all that apply
  - ☑ P=Q=true
  - ☑ P=Q=false
  - ❑ P=true, Q=false
  - ☑ P=false, Q=true
- We can get this from the truth table for →
- In FOL, it's hard to prove a conditional true
  - Consider proving prime(x) → odd(x)

Inference Rules

- **Logical inference** creates new sentences that logically follow from a set of sentences (the KB)
- An inference rule is **sound** if every sentence X produces when operating on a KB logically follows from the KB
  - I.e., inference rule does not create contradictions
- An inference rule is **complete** if it can produce every expression that logically follows from (is entailed by) the KB
  - Note the analogy to complete search algorithms

Sound Rules of Inference

- Here are some examples of sound rules of inference
  - *A rule is sound if its conclusion is true when the premise is true*
- Each can be shown to be sound using a truth table

RULE	PREMISE	CONCLUSION
Modus Ponens	A, A → B	B
And Introduction	A, B	A ∧ B
And Elimination	A ∧ B	A
Double Negation	¬¬A	A
Unit Resolution	A ∨ B, ¬B	A
Resolution	**A ∨ B, ¬B ∨ C**	**A ∨ C**

Resolution

- **Resolution** is an rule producing a new clause implied by two clauses containing complementary literals
  - Literal: atomic symbol or its negation, i.e., P, ~P
- Amazingly, this is the only interference rule needed to build a sound & complete theorem prover
  - Based on proof by contradiction and usually called resolution refutation

The resolution rule was discovered by Alan Robinson (CS, U. of Syracuse) in the mid 1960s

Resolution

- A KB is a set of sentences all of which are true, i.e., a conjunction of sentences
- To use resolution, put KB into *conjunctive normal form* (CNF) where each is a disjunction of literals (positive or negative atoms)
- Every KB can be put into CNF
  - Rewrite sentences using standard tautologies
  - P→Q ≡ ¬P∨Q

Proving Things

- **Proof:** a sequence of sentences, where each is a premise (i.e., a given) or is derived from earlier sentences in the proof by an inference rule
- Last sentence is the **theorem** (aka goal or query) that we want to prove

1	Hu	premise	It's humid
2	Hu→Ho	premise	If it's humid, it's hot
3	Ho	modus ponens (1,2)	It's hot
4	(Ho∧Hu)→R	premise	If it's hot and humid, it's raining
5	Ho∧Hu	and introduction	It is hot and humid
6	R	modus ponens (4,5)	It is raining

Horn sentences

- A **Horn sentence** or **Horn clause** has the form:
  P1 ∧ P2 ∧ P3 ... ∧ Pn → Qm *where n>=0, m in{0,1}*
- Note: a conjunction of 0 or more symbols to left of → and 0-1 symbols to right
- Special cases:
  - n=0, m=1: **P** *(assert P is true)*
  - n>0, m=0: **P∧Q** → *(constraint: both P and Q can't be true)*
  - n=0, m=0: *(well, there is nothing there!)*
- Put in CNF: each sentence is a disjunction of literals with at most one non-negative literal
  ¬P1 ∨ ¬P2 ∨ ¬P3 ... ∨ ¬Pn ∨ Q

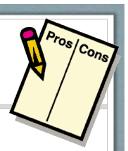
(P → Q) = (¬P ∨ Q)

Significance of Horn Logic

- We can also have horn sentences in FOL
- Reasoning with horn clauses is much simpler
  - Satisfiability of propositional KB (i.e., finding values for a symbols that will make it true) is NP complete
  - Restricting KB to horn sentences, satisfiability is in P
- FOL Horn sentences are the basis for many rule-based languages
- Horn logic can't handle **negation** and **disjunctions** (in general)

Problems with Propositional Logic

Propositional Logic



- **Advantages**
  - Simple KR language good for many problems
  - Lays foundation for higher logics (e.g., FOL)
  - Reasoning is decidable, though NP complete; efficient techniques exist for many problems
- **Disadvantages**
  - Not expressive enough for most problems
  - Even when it is, it can be very "un-concise"

Propositional Logic is a Weak Language

- Hard to identify ***individuals*** (e.g., Mary, 3)
- Can't directly talk about properties of individuals or relations between individuals (e.g., "Bill is tall")
- Generalizations, patterns, regularities can't easily be represented (e.g., "all triangles have 3 sides")
- First-Order Logic (abbreviated FOL) is expressive enough to concisely represent this kind of information
- FOL adds relations, variables, and quantifiers, e.g.,
  - *"Every elephant is gray":* ∀ x (elephant(x) → gray(x))
  - *"There is a white alligator":* ∃ x (alligator(X) ^ white(X))

Examples of PL Limits

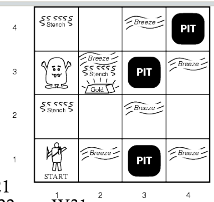
- Consider the problem of representing the following information:
  - Every person is mortal.
  - Confucius is a person.
  - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?

Example II

- In PL we have to create propositional symbols to stand for all or part of each sentence. For example, we might have:
  P = "person"; Q = "mortal"; R = "Confucius"
- so the above 3 sentences are represented as:
  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$
- Although the third sentence is entailed by the first two, we needed an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes "person" and "mortal"
- To represent other individuals we must introduce separate symbols for each one, with some way to represent the fact that all individuals who are "people" are also "mortal"

The "Hunt the Wumpus" Agent

- Some atomic propositions:
  - S12 = There is a stench in cell (1,2)
  - B34 = There is a breeze in cell (3,4)
  - W13 = The Wumpus is in cell (1,3)
  - V11 = We have visited cell (1,1)
  - OK11 = Cell (1,1) is safe
  - …
- Some rules:
  - (R1) ¬S11 → ¬W11 ∧ ¬ W12 ∧ ¬ W21
  - (R2) ¬ S21 → ¬W11 ∧ ¬ W21 ∧ ¬ W22 ∧ ¬ W31
  - (R3) ¬ S12 → ¬W11 ∧ ¬ W12 ∧ ¬ W22 ∧ ¬ W13
  - (R4) S12 → W13 ∨ W12 ∨ W22 ∨ W11
  - …
- Lack of variables forces similar rules for each cell



- **Prove Wumpus is in (1,3) and there is a pit in (3,1)!**
- If there is no stench in a cell, then there is no wumpus in any adjacent cell
- If there is a stench in a cell, then there is a wumpus in some adjacent cell
- If there is no breeze in a cell, then there is no pit in any adjacent cell
- If there is a breeze in a cell, then there is a pit in some adjacent cell
- If a cell has been visited, it has neither a wumpus nor a pit
- **FIRST write the propositional rules for the relevant cells**
- **NEXT write the proof steps and indicate what inference rules you used in each step**

INFERENCE RULES

Modus Ponens
A, A → B
ergo B

And Introduction
A, B
ergo A ∧ B

And Elimination
A ∧ B
ergo A

Double Negation
¬¬A
ergo A

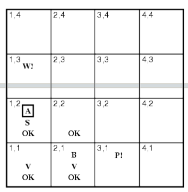
Unit Resolution
A ∨ B, ¬B
ergo A

Resolution
A ∨ B, ¬B ∨ C
ergo A ∨ C

A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

V12 S12 -B12	V22 -S22 -B22		
V11 -S11 -B11	V21 B21 -S21		

After 3[rd] move



A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

- We can prove that the Wumpus is in (1,3) using these rules:

(R1) ¬S11 → ¬W11 ∧ ¬W12 ∧ ¬W21

(R2) ¬S21 → ¬W11 ∧ ¬W21 ∧ ¬W22 ∧ ¬W31

(R3) ¬S12 → ¬W11 ∧ ¬W12 ∧ ¬W22 ∧ ¬W13

(R4) S12 → W13 ∨ W12 ∨ W22 ∨ W11

See R&N section 7.5

Proving W13

(R1) ¬S11 → ¬W11 ∧ ¬W12 ∧ ¬ W21
(R2) ¬ S21 → ¬W11 ∧ ¬W21 ∧ ¬ W22 ∧ ¬W31
(R3) ¬ S12 → ¬W11 ∧ ¬W12 ∧ ¬ W22 ∧ ¬W13
(R4) S12 → W13 ∨ W12 ∨ W22 ∨ W11

- Apply MP with ¬S11 and R1:
  ¬ W11 ∧ ¬ W12 ∧ ¬ W21
- Apply And-Elimination to this, yielding three sentences:
  ¬ W11, ¬ W12, ¬ W21
- Apply MP to ~S21 and R2, then apply And-Elimination:
  ¬ W22, ¬ W21, ¬ W31
- Apply MP to S12 and R4 to obtain:
  W13 ∨ W12 ∨ W22 ∨ W11
- Apply Unit Resolution on (W13 ∨ W12 ∨ W22 ∨ W11) and ¬W11:
  W13 ∨ W12 ∨ W22
- Apply Unit Resolution with (W13 ∨ W12 ∨ W22) and ¬W22:
  W13 ∨ W12
- Apply UR with (W13 ∨ W12) and ¬W12:
  W13
- QED

Propositional Wumpus Problems

- Lack of variables prevents stating more general rules
  - ∀ x, y V(x,y) → OK(x,y)
  - ∀ x, y S(x,y) → W(x-1,y) ∨ W(x+1,y) …
- Change of the KB over time is difficult to represent
  - In classical logic, a fact is true or false for all time
  - A standard technique is to index dynamic facts with the time when they're true
    - A(1, 1, t0)
  - So we have a separate KB for every time point ☹

Prop. Logic Summary

- **Inference:** the process of deriving new sentences from old
  - **Sound** inference derives true conclusions given true premises
  - **Complete** inference derives all true conclusions from a set of premises
- A **valid sentence** is true in all worlds under all interpretations
- If an implication sentence can be shown to be valid, then—given its premise—its consequent can be derived
- Different logics make different **commitments** about what the world is made of and what kind of beliefs we can have regarding the facts
- **Propositional logic** commits only to the existence of facts that may or may not be the case in the world being represented
  - Simple syntax and semantics suffice to illustrate the process of inference
  - Propositional logic quickly becomes impractical, even for very small worlds

First-Order Logic

Material from Dr. Marie desJardin, Some material adopted from notes by Andreas Geyer-Schulz and Chuck Dyer

First-Order Logic

- First-order logic (FOL) models the world in terms of
  - **Objects,** which are things with individual identities
  - **Properties** of objects that distinguish them from other objects
  - **Relations** that hold among sets of objects
  - **Functions,** which are a subset of relations where there is only one "value" for any given "input"
- Examples:
  - Objects: Students, lectures, companies, cars ...
  - Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - Properties: blue, oval, even, large, ...
  - Functions: father-of, best-friend, second-half, one-more-than ...

Sentences: Terms and Atoms

- A **term** (denoting a real-world individual) is:
  - A constant symbol: *John*, or
  - A variable symbol: *x*, or
  - An n-place function of n terms

    x and $f(x_1, ..., x_n)$ are terms, where each x_i is a term

    is-a(John, Professor)
  - A term with no variables is a **ground term**.
- An **atomic sentence** is an n-place predicate of n terms
  - Has a truth value (*t* or *f*)

Sentences: Terms and Atoms

- A **complex sentence** is formed from atomic sentences connected by the logical connectives:

  ¬P, P∨Q, P∧Q, P→Q, P↔Q where P and Q are sentences

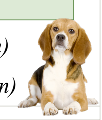
has-a(x, Bachelors) ∧ is-a(x, human)

does NOT SAY everyone with a bachelors' is human

What DOES it say?

has-a(John, Bachelors) ∧ is-a(John, human)

has-a(Mary, Bachelors) ∧ is-a(Mary, human)



Quantifiers

- **Universal quantification**
  - ∀x P(x) means that P holds for **all** values of *x* in its domain
  - States universal truths
  - E.g.: ∀*x dolphin(x) → mammal(x)*
- **Existential quantification**
  - ∃x P(x) means that P holds for **some** value of x in the domain associated with that variable
  - Makes a statement about some object without naming it
  - E.g., ∃*x mammal(x) ∧ lays-eggs(x)*

Sentences: Quantification

- **Quantified sentences** adds quantifiers ∀ and ∃

∀*x has-a(x, Bachelors) → is-a(x, human)*

∃*x has-a(x, Bachelors)*

∀*x* ∃*y Loves(x, y)*

Everyone who has a bachelors' is human.

There exists some who has a bachelors'.

Everybody loves somebody.

Sentences: Well-Formedness

- A **well-formed formula (wff)** is a sentence containing no "free" variables. That is, all variables are "bound" by universal or existential quantifiers.
- $(\forall x)P(x,y)$ has ***x*** bound as a universally quantified variable, but ***y*** is free.

Quantifiers: Uses

- Universal quantifiers **often** used with "implies" to form "rules":
  - (∀x) student(x) → smart(x)
  - "All students are smart"
- Universal quantification **rarely*** used to make blanket statements about every individual in the world:
  - (∀x)student(x)∧smart(x)
  - "Everyone in the world is a student and is smart"

*Deliberately, anyway

Quantifiers: Uses

- Existential quantifiers are **usually** used with "and" to specify a list of properties about an individual:
  (∃x) student(x) ∧ smart(x)
  "There is a student who is smart"
- A common mistake is to represent this English sentence as the FOL sentence:
  (∃x) student(x) → smart(x)
  - But what happens when there is a person who is *not* a student?

Quantifier Scope

- Switching the order of universal quantifiers *does not* change the meaning:
  - (∀x)(∀y)P(x,y) ↔ (∀y)(∀x) P(x,y)
- Similarly, you can switch the order of existential quantifiers:
  - (∃x)(∃y)P(x,y) ↔ (∃y)(∃x) P(x,y)
- Switching the order of universals and existentials *does* change meaning:
  - Everyone likes someone: (∀x)(∃y) likes(x,y)
  - Someone is liked by everyone: (∃y)(∀x) likes(x,y)

Connections between For All and Exists

We can relate sentences involving ∀ and ∃ using De Morgan's laws:

$(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$

$\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$

$(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$

$(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$

Quantified Inference Rules

- Universal instantiation
  - ∀x P(x) ∴ P(A)
- Universal generalization
  - P(A) ∧ P(B) … ∴ ∀x P(x)
- Existential instantiation
  - ∃x P(x) ∴P(F) ← **skolem constant F**
- Existential generalization
  - P(A) ∴ ∃x P(x)

Universal Instantiation (a.k.a. Universal Elimination)

- If (∀x) P(x) is true, then P(C) is true, where C is *any* constant in the domain of x
- Example:
  (∀x) eats(Ziggy, x) ⇒ eats(Ziggy, IceCream)
- The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only

Existential Instantiation (a.k.a. Existential Elimination)

- Variable is replaced by a **brand-new constant**
  - I.e., not occurring in the KB
- From (∃x) P(x) infer P(c)
  - Example:
    - (∃x) eats(Ziggy, x) → eats(Ziggy, Stuff)
  - "Skolemization"
- *Stuff* is a **skolem constant**
- Easier than manipulating the existential quantifier

Existential Generalization (a.k.a. Existential Introduction)

- If P(c) is true, then (∃x) P(x) is inferred.
- Example
  eats(Ziggy, IceCream) ⇒ (∃x) eats(Ziggy, x)
- All instances of the given constant symbol are replaced by the new variable symbol
- Note that the variable symbol cannot already exist anywhere in the expression

Translating English to FOL

Every gardener likes the sun.
∀x gardener(x) → likes(x,Sun)

You can fool some of the people all of the time.
∃x ∀t person(x) ∧time(t) → can-fool(x,t)

You can fool all of the people some of the time.
∀x ∃t (person(x) → time(t) ∧can-fool(x,t))
∀x (person(x) → ∃t (time(t) ∧can-fool(x,t))
Equivalent

All purple mushrooms are poisonous.
∀x (mushroom(x) ∧ purple(x)) → poisonous(x)

Translating English to FOL

No purple mushroom is poisonous.
¬∃x purple(x) ∧ mushroom(x) ∧ poisonous(x)
∀x (mushroom(x) ∧ purple(x)) → ¬poisonous(x)
Equivalent

There are exactly two purple mushrooms.
∃x ∃y mushroom(x) ∧ purple(x) ∧ mushroom(y) ∧ purple(y) ^ ¬(x=y) ∧ ∀z (mushroom(z) ∧ purple(z)) → ((x=z) ∨ (y=z))

Clinton is not tall.
¬tall(Clinton)

X is above Y iff X is on directly on top of Y or there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.
∀x ∀y above(x,y) ↔ (on(x,y) ∨ ∃z (on(x,z) ∧ above(z,y)))

Semantics of FOL

- **Domain M:** the set of all objects in the world (of interest)
- **Interpretation I:**
  - Assign each constant to an object in M
  - Define each function of n arguments as a mapping M^n => M
  - Define each predicate of n arguments as a mapping M^n => {T, F}
  - Therefore, every **ground predicate** with any instantiation will have a truth value
  - In general there is an infinite number of interpretations because |M| is infinite
- **Define logical connectives: ~, ^, v, =>, <=>** as in PL
- **Define semantics of (∀x) and (∃x)**
  - (∀x) P(x) is true iff P(x) is true under all interpretations
  - (∃x) P(x) is true iff P(x) is true under some interpretation

- **Model:** an interpretation of a set of sentences such that every sentence is *True*
- **A sentence is**
  - **Satisfiable** if it is true under some interpretation
  - **Valid** if it is true under all possible interpretations
  - **Inconsistent** if there does not exist any interpretation under which the sentence is true
- **Logical consequence:** S |= X if all models of S are also models of X

Axioms, Definitions and Theorems

- **Axioms**: facts and rules that attempt to capture all of the (important) facts and concepts about a domain
- Axioms can be used to prove **theorems**
  - Mathematicians don't want any unnecessary (dependent) axioms –ones that can be derived from other axioms
  - Dependent axioms can make reasoning faster, however
  - Choosing a good set of axioms for a domain is a design problem!
- A **definition** of a predicate is of the form "p(X) ↔ …" and can be decomposed into two parts
  - **Necessary** description: "p(x) → …"
  - **Sufficient** description "p(x) ← …"
  - Some concepts don't have complete definitions (e.g., person(x))

More on Definitions

- Examples: define father(x, y) by parent(x, y) and male(x)
  - parent(x, y) is a necessary (**but not sufficient**) description of father(x, y)
    - father(x, y) → parent(x, y)
  - parent(x, y) ^ male(x) ^ age(x, 35) is a **sufficient** (**but not necessary**) description of father(x, y):

    father(x, y) ← parent(x, y) ^ male(x) ^ age(x, 35)
  - parent(x, y) ^ male(x) is a **necessary and sufficient** description of father(x, y)

    parent(x, y) ^ male(x) ↔ father(x, y)

Higher-Order Logics

- FOL only allows to quantify over variables, and variables can only range over objects.
- HOL allows us to quantify over relations
- Example: (quantify over functions)
  - "two functions are equal iff they produce the same value for all arguments"
  - ∀f ∀g (f = g) ↔ (∀x f(x) = g(x))
- Example: (quantify over predicates)
  - ∀r transitive(r) → (∀xyz) r(x,y) ∧ r(y,z) → r(x,z))
- More expressive, but undecidable.

Expressing Uniqueness

- Sometimes we want to say that there is a single, unique object that satisfies a certain condition
- "There exists a unique x such that king(x) is true"
  - ∃x king(x) ∧ ∀y (king(y) → x=y)
  - ∃x king(x) ∧ ¬∃y (king(y) ∧ x≠y)
  - ∃! x king(x)
- "Every country has exactly one ruler"
  - ∀c country(c) → ∃! r ruler(c,r)
- Iota operator: "ι x P(x)" means "the unique x such that p(x) is true"
  - "The unique ruler of Freedonia is dead"
  - dead(ι x ruler(freedonia,x))