Propositional logic, pros and cons

- Propositional logic is declarative
- Propositional logic allows partial (disjunctive/negated) information
  - (unlike most data structures and databases)
- Propositional logic is compositional:
  - meaning of $B_{1,1} \land P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
Propositional logic, pros and cons

- Meaning in propositional logic is **context-independent**
  - (unlike natural language, where meaning depends on context)
- Propositional logic has very limited expressive power
  - (unlike natural language)
  - E.g., cannot say "pits cause breezes in adjacent squares"
    - except by writing one sentence for each square

Why not use Natural Language?

- It serves a different purpose:
  - Communication rather than representation
- It is not compositional
  - Context matters
- It can be ambiguous
  - Again, context matters

Create a new language

- Builds on propositional logic
- But is inspired by natural language!
First-order logic

- Whereas propositional logic assumes the world contains facts,
- first-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, colors, baseball games, wars, …
  - Relations: red, round, prime, brother of, bigger than, part of, comes between, …
  - Functions: father of, best friend, one more than, plus, …

Syntax of FOL: Basic elements

- Constants: KingJohn, 2, NUS,…
- Predicates: Brother, >,…
- Functions: Sqrt, LeftLegOf,…
- Variables: x, y, a, b,…
- Connectives: ¬, ⇒, ∧, ∨, ⇔
- Equality: =
- Quantifiers: ∀, ∃

Atomic sentences

Atomic sentence = predicate (term₁,…,termₙ)
or term₁ = term₂

Term = function (term₁,…,termₙ)
or constant or variable

- E.g., Brother(KingJohn,RichardTheLionheart)

  > (Length(LeftLegOf(Richard)),Length(LeftLegOf(KingJohn)))
Complex sentences
- Complex sentences are made from atomic sentences using connectives
  \(-S, S_1 \land S_2, S_1 \lor S_2, S_1 \Rightarrow S_2, S_1 \Leftrightarrow S_2\)

  E.g. \(\text{Sibling(KingJohn,Richard)} \Rightarrow \text{Sibling(Richard,KingJohn)}\)
  >\((1,2) \lor \leq \leq (1,2)\)
  \(<(1,2) \land \neg \geq (1,2)\)

Truth in first-order logic
- Sentences are true with respect to a model and an interpretation.
- Model contains objects (domain elements) and relations among them.
- Interpretation specifies referents for:
  - Constant symbols \(\rightarrow\) objects
  - Predicate symbols \(\rightarrow\) relations
  - Function symbols \(\rightarrow\) functional relations

  An atomic sentence \(\text{predicate(\text{term}_1,\ldots,\text{term}_n)}\) is true iff the objects referred to by \(\text{term}_1,\ldots,\text{term}_n\) are in the relation referred to by \(\text{predicate}\).

Models for FOL: Example

![Diagram showing relationships between objects and relations in a model for first-order logic.]}
Universal quantification

- ∀<variables> <sentence>
- Everyone in HR is smart:
  ∀x At(x,HR) ⇒ Smart(x)
- ∀x P is true in a model m iff P is true with x being each possible object in the model
- Roughly speaking, equivalent to the conjunction of instantiations of P
  - At(KingJohn,HR) ⇒ Smart(KingJohn)
  - At(Richard,HR) ⇒ Smart(Richard)
  - At(HR,HR) ⇒ Smart(HR)
  - ...

A common mistake to avoid

- Typically, ⇒ is the main connective with ∀
- Common mistake: using ∧ as the main connective with ∀:
  ∀x At(x,HR) ∧ Smart(x)
  means “Everyone is at HR and everyone is smart”

Existential quantification

- ∃<variables> <sentence>
- Someone at HR is smart:
  ∃x At(x,HR) ∧ Smart(x)
- ∃x P is true in a model m iff P is true with x being some possible object in the model
- Roughly speaking, equivalent to the disjunction of instantiations of P
  - At(KingJohn,HR) ∧ Smart(KingJohn)
  - At(Richard,HR) ∧ Smart(Richard)
  - At(HR,HR) ∧ Smart(HR)
  - ...

Another mistake to avoid

- Typically, \( \land \) is the main connective with \( \exists \).
- Common mistake: using \( \Rightarrow \) as the main connective with \( \exists \):
  \[ \exists x \, \text{At}(x, \text{HR}) \Rightarrow \text{Smart}(x) \]
  is true if there is anyone who is not at HR!

Properties of quantifiers

- \( \forall x \, \forall y \) is the same as \( \forall y \, \forall x \)
- \( \exists x \, \exists y \) is the same as \( \exists y \, \exists x \)
- \( \exists x \, \forall y \) is not the same as \( \forall y \, \exists x \)
- \( \exists x \text{ Loves}(x,y) \) "There is a person who loves everyone in the world"
- \( \forall y \text{ Loves}(x,y) \) "Everyone in the world is loved by at least one person"
- Quantifier duality: each can be expressed using the other
  \[ \forall x \text{ Likes}(x,\text{IceCream}) \iff \exists x \, \neg \text{Likes}(x,\text{IceCream}) \]
  \[ \exists x \text{ Likes}(x,\text{Broccoli}) \iff \forall x \, \neg \text{Likes}(x,\text{Broccoli}) \]

Equality

- \( \text{term}_1 = \text{term}_2 \) is true under a given interpretation if and only if \( \text{term}_1 \) and \( \text{term}_2 \) refer to the same object.
- E.g., definition of Sibling in terms of Parent:
  \[ \forall x, y \, \text{Sibling}(x,y) \iff \]
  \[ \neg (x = y) \land \exists m, f \, (m = f) \land \text{Parent}(m,x) \land \text{Parent}(f,x) \land \text{Parent}(m,y) \land \text{Parent}(f,y) \]
Using FOL

The kinship domain:

- Brothers are siblings
  \[ \forall x,y \text{ Brother}(x,y) \iff \text{Sibling}(x,y) \]

- One's mother is one's female parent
  \[ \forall m,c \text{ Mother}(c) = m \iff (\text{Female}(m) \land \text{Parent}(m,c)) \]

- "Sibling" is symmetric
  \[ \forall x,y \text{ Sibling}(x,y) \iff \text{Sibling}(y,x) \]

Some sentences are Axioms (i.e. definitions, facts) while others are Theorems derived from those.

Interacting with FOL KBs

- Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at \( t=5 \):
  \[ \text{Tell}(\text{KB}, \text{Percept}([\text{Smell}, \text{Breeze}, \text{None}], 5)) \]

- Ask (KB, \exists a \text{ BestAction}(a,5))

- I.e., does the KB entail some best action at \( t=5 \)?

- Answer: Yes, \( \{a/\text{Shoot}\} \) ← substitution (binding list)

- Given a sentence \( S \) and a substitution \( q \), \( S^q \) denotes the result of plugging \( q \) into \( S \); e.g.,
  \[ S = \text{Smarter}(x,y) \]
  \[ q = \{x/\text{Hillary}, y/\text{Bill}\} \]
  \[ S^q = \text{Smarter}([\text{Hillary}, \text{Bill}]) \]

- \( \text{Ask}(\text{KB}, S) \) returns some/all \( q \) such that \( \text{KB} \models S^q \)

KB for the wumpus world

- Perception
  \[ \forall t,s,b \text{ Percept}([s,b,\text{Glitter}],t) \iff \text{Glitter}(t) \]

- Reflex
  \[ \forall t \text{ Glitter}(t) \implies \text{BestAction}(\text{Grab},t) \]
Deducing hidden properties

- \( \forall x,y,a,b \text{ Adjacent}([x,y],[a,b]) \Rightarrow [a,b] \in \{[x+1,y], [x-1,y], [x,y+1], [x,y-1]\} \)

Properties of squares:
- \( \forall s,t \text{ At(Agent,s,t) \& Breeze(t) } \Rightarrow \text{ Breezy(s) } \)

Squares are breezy near a pit:
- Diagnostic rule---infer cause from effect
  \( \forall s \text{ Breezy(s) } \Rightarrow \exists \text{ Adjacent(r,s) } \& \text{ Pit(r) } \)
- Causal rule---infer effect from cause
  \( \forall r \text{ Pit(r) } \Rightarrow [\forall s \text{ Adjacent(r,s) } \Rightarrow \text{ Breezy(s) } ] \)

Summary

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers

- Increased expressive power:
  sufficient to define wumpus world