FIRST ORDER LOGIC

SHIRAZ KHURANA

COMBINING THE BEST OF FORMAL AND NATURAL LANGUAGES

- Propositional logic is a declarative, compositional semantics that is context-independent and unambiguous
- Need to build a more expressive logic, borrowing representational ideas from natural language while avoiding its drawbacks.
- In the syntax of natural language, the most obvious elements are nouns and noun phrases that refer to objects (squares, pits, wumpuses) and verbs and verb phrases that refer to relations among objects (is breezy, is adjacent to, shoots).
- Some of these relations are functions—relations in which there is only one "value" for a given "input."

FIRST ORDER LOGIC

- predicate logic, first order logic, elementary logic, restricted predicate calculus, restricted functional calculus, relational calculus, theory of quantification, theory of quantification with equality, etc.
- In propositional logic only the logical forms of compound propositions are analyzed. A simple proposition is an unanalyzed whole which is either true or false.
- There are certain arguments that seem to be perfectly logical, yet they cannot be expressed by using propositional calculus.
  - 1. All cats have tails.
  - 2. Tom is a cat.
  - From these two sentences, one should be able to conclude that
  - 3. Tom has a tail.
- To show that this argument is sound, we must be able to identify individuals, such as Tom, together with their properties and predicates. This is the objective of predicate calculus.

FIRST ORDER LOGIC

- Predicates are used to describe certain properties or relationships between individuals or objects. Example: In "Mary and Jane are sisters", the phrase "are sisters" is a predicate. The entities connected this way, Mary and Jane, are called terms.
- Terms play a similar role in predicate calculus as nouns and pronouns do in the English language.
- Quantifiers indicate how frequently a certain statement is true. Specifically, the universal quantifier is used to indicate that a statement is always true, whereas the existential quantifier indicates that a statement is sometimes true.
- Example: In "All cats have tails", the word "all" indicates that the statement "cats have tails" is universally true.
- Predicate calculus is a generalization of propositional calculus. Besides terms, predicates, and quantifiers, predicate calculus contains propositional variables, constants and connectives as part of the language.

SYNTAX

- The Domain (universe of discourse)
  - 1. Jane is Paul's mother.
  - 2. Jane is Mary's mother.
  - 3. Any two persons having the same mother are siblings.
  - 4. Paul and Mary are siblings.
- The truth of the statement "Jane is Paul's mother" can only be assessed within a certain context. There are many people named Jane and Paul, and without further information the statement in question can refer to many different people, which makes it ambiguous. To prevent such ambiguities, the concept of a domain or universe of discourse is used.
- The universe of discourse or domain is the collection of all persons, ideas, symbols, data structures, and so on, that affect the logical argument under consideration. The elements of the domain are called individuals.
- The truth of a statement may depend on the domain selected. The statement "there is a smallest number" is true in the domain of natural numbers, but false in the domain of integers.

SYNTAX

- An individual can be a person, a number, a data structure. Instead, object also used, such as in “the domain must contain at least one object”.
- To refer to a particular individual or object, identifiers must be used. These identifiers are called individual constants.
- If the universe of discourse consists of persons, the individual constants may be their names. In the case of natural numbers, the individual constants are the digits representing these numbers. Each individual constant must uniquely identify a particular individual and no other one.

PREDICATES

- Predicates make statements about individuals:
  - Mary and Paul are siblings.
  - Jane is the mother of Mary.
  - Tom is a cat. The sum of 2 and 3 is 5.
- In each of these statements, there is a list of individuals, which is given by the argument list, together with phrases that describe certain relations among, or properties of the individuals mentioned in the argument list. These properties or relations are referred to as predicates.
- In the statement "Mary and Paul are siblings", the argument list is given by Mary and Paul, in that order, whereas the predicate is described by the phrase "are siblings".
- "Tom is a cat" has an argument list with the single element "Tom" in it, and its predicate is described by "is a cat".
- The entries of the argument list are called arguments. The arguments can be either variables or individual constants

PREDICATES

- In predicate calculus, each predicate is given a name, which is followed by the list of arguments. The list of arguments is enclosed in parentheses.
- To express "Jane is the mother of Mary" one could choose an identifier, say "mother" to express the predicate "is the mother of", and one would write mother (Jane, Mary).
- We can use only single letters for predicate names and constants. For instance, M(j, m) instead of mother(Jane, Mary); that is, M as a name for the predicate "is the mother of", j for Jane and m for Mary.
- The order of arguments is important. Clearly, the statements mother(Mary, Jane) and mother(Jane, Mary) have a completely different meaning.
- The number of elements in the argument list of a predicate is called the arity of the predicate. For instance, mother(Jane, Mary) has arity 2. The arity of a predicate is fixed.

PREDICATES

- A predicate and cannot have two arguments in one case three in another. Alternatively, one can consider two predicates different if their arity is different. The following statement illustrates this:
  - The sum of 2 and 3 is 5.
  - The sum of 2, 3 and 4 is 9.
- To express these statements in predicate calculus, one can either use two predicate names such as "sum2" and "sum3" and write sum2(2, 3, 5) and sum3(2, 3, 4, 9) respectively, or one can use the same symbol, say "sum" with the implicit understanding that the name "sum" in sum(2, 3, 5) refers to a different predicate than in sum(2, 3, 4, 9).
- A predicate with arity n is often called an n-place predicate. A one-place predicate is called a property. The predicate "is a cat" is a one-place predicate, or a property.

PREDICATES

- A predicate name, followed by an argument list in parentheses is called an atomic formula . The atomic formulas can be combined by logical connectives like propositions.
- For instance, if cat(Tom) and hastail(Tom) are two atomic formulas, expressing that Tom is a cat and that Tom has a tail respectively, one can form cat(Tom) → hastail(Tom).
- If all arguments of a predicate are individual constants, then the resulting atomic formula must be either true or false.
- If the domain consists of Jane, Doug, Mary and Paul, we must know for each ordered pair of individuals whether the predicate "is the mother of" is true. This can be done in the form of a table. The method that assigns truth values to all possible combinations of individuals of a predicate is called an assignment.

	Doug	Jane	Mary	Paul
Doug	0	0	0	0
Jane	0	0	1	1
Mary	0	0	0	0
Paul	0	0	0	0

VARIABLES

- Often one does not want to associate the arguments of an atomic formula with a particular individual. To avoid this, variables are used.
- Variables are frequently chosen from the end of the alphabet; that is x, y and z, with or without subscripts, suggest variable names. Examples:
  - cat(x) $\rightarrow$ hastail(x)
  - dog(y) $\wedge$ brown(y)
  - grade(x) $\rightarrow$ ($x \geq 0$) $\wedge$ ($x \leq 100$)
- As in propositional calculus, formulas can be given names. For instance, one can define A as follows:
  - A = cat(x) $\rightarrow$ hastail(x). which means that when we write A we really mean "cat(x) $\rightarrow$ hastail(x)".

INSTANTIATION

- If A is a formula, one often must replace all occurrences of a particular variable by a term. For example, in the expression cat(x) → hastail(x), one may want to replace all instances of x by the term Tom which yields cat(Tom) → hastail(Tom).
- Let A represent a formula, x represents a variable, and t represent a term. Then S^x_t A represents the formula obtained by replacing all occurrences of x in A by t. S^x_t A is called an instantiation of A, and t is said to be an instance of x.

QUANTIFIERS

- Consider the following three statements:
- 1. All cats have tails. 2. Some people like their meat raw. 3. Everyone gets a break once in a while.
- All these statements indicate how frequently certain things are true.
- Let A represent a formula, and let x represent a variable. If we want to indicate that A is true for all possible values of x, we write $\forall xA$. Here, $\forall x$ is called universal quantifier, and A is called the scope of the quantifier. The variable x is said to be bound by the quantifier. The symbol $\forall$ is pronounced "for all".
- The quantifier and the bounded variable that follows have to be treated as a unit, and this unit acts somewhat like a unary connective. Statements containing words like "every", "each", and "everyone" usually indicate universal quantification. Such statements must typically be reworded such that they start with "for every x", which is then translated into $\forall x$.

EXAMPLE FOR ALL

- Example: Express "Everyone gets a break once in a while" in predicate calculus.
- Solution: We define B to mean "gets a break once in a while". Hence, B(x) means that x gets a break once in a while. The word everyone indicates that this is true for all x. This leads to the following translation: $\forall xB(x)$ Example: Express "All cats have tails" in predicate calculus

EXISTENTIAL QUANTIFIER

- Let A represent a formula, and let x represent a variable. If we want to indicate that A is true for at least one value x, we write $\exists xA$.
- This statement is pronounced "There exists an x such that A." Here, $\exists x$ is called the existential quantifier, and A is called the scope of the quantifier. The variable x is said to be bound by the quantifier.
- Statements containing such phrases as "some", and "at least one" suggest existential quantifiers. They should be rephrased as "there is an x such that" which is translated by $\exists x$.
- Example: Let P be the predicate "like their meat raw". Then $\exists xP(x)$ can be translated as "There exist people who like their meat raw" or "Some people like their meat raw."
- Example: If the universe of discourse (domain) is a collection of things, $\exists x$ blue(x) should be understood as "There exists objects that are blue" or "Some objects are blue."

QUANTIFIERS

- $\forall x$ and $\exists x$ have to be treated like unary connectives.
- The quantifiers are given a higher precedence than all binary connectives.
- For instance, if $P(x)$ and $Q(x)$ means that x is living and that x is dead, respectively, then one has to write $\forall x(P(x) \vee Q(x))$ to indicate that everything is either living or dead. $\forall xP(x) \vee Q(x)$ means that either everything is living, or x is dead.
- The variable x in a quantifier is just a placeholder, and it can be replaced by any other variable name not appearing elsewhere in the formula. For instance, $\forall xP(x)$ and $\forall yP(y)$ mean the same thing: they are logically equivalent. The expression $\forall yP(y)$ is a variant of $\forall xP(x)$.

QUANTIFIERS EXAMPLE

- Quantifiers may be nested, as demonstrated by the following example.
- Example: Translate the sentence "There is somebody who knows everyone" into the language of predicate calculus. To do this, use $K(x, y)$ to express the fact that x knows y.
- Solution: The best way to solve this problem is to go in steps.
- We write informally $\exists x$(x knows everybody) Here, "x knows everybody" is still in English and means that for all y is it true that x knows y.
- Hence x knows everybody = $\forall y K(x, y)$.
- We now add the existential quantifier and obtain $\exists x \forall y K(x, y)$.

QUANTIFIERS EXAMPLE

- The English statement "Nobody is perfect" also includes a quantifier, "nobody" which is the absence of an individual with a certain property.
- In predicate calculus, the fact that nobody has property P cannot be expressed directly.
- To express the fact that there is no x for which an expression A is true one can either use $\neg\exists xA$ or $\forall x\neg A$.
- If P represents the property of perfection, both $\neg\exists xP(x)$ and $\forall x\neg P(x)$ indicate that nobody is perfect. They correspond to "It is not the case that there is somebody who is perfect", "for everyone, it is not the case that he or she is perfect" respectively.
- The two methods to express that nobody is A must of course be logically equivalent, $\neg\exists xA \mid=\mid \forall x\neg A$.
- There are many quantifiers in English, such as "a few", "most", and "about a third", that are useful in daily language, but are not precise and cannot be used in logic.

BOUND AND FREE VARIABLES

- The variable appearing in the quantifier is said to be bound. For instance, in the expression $\forall x(P(x) \rightarrow Q(x))$, the variable x appears three times and each time x is a bound variable.
- Any variable that is not bound is said to be free. The same variable can occur both bound and free in a formula. For this reason, it is also important to indicate the position of the variable.
- Find the free variables in $\forall z(P(z) \land Q(x)) \lor \exists yQ(y)$.
- Solution: Only variable x is free. All occurrences of z are bound, and so are all occurrences of the variable y.
- The status of a variable changes as formulas are divided into sub formulas.
- For instance, in $\forall xP(x)$, x occurs twice, and it is bound both times. This formula contains P(x) as sub formula. In P(x) the variable x is free.

RESTRICTIONS OF QUANTIFIERS

- Sometimes, quantification is over a subset of the universe of discourse. Suppose, for instance, that animals form the universe of discourse. How can one express sentence such as "All dogs are mammals" and "Some dogs are brown"?
- Consider the first statement "All dogs are mammals". Since the quantifier should be restricted to dogs, one rephrases the statement as "If x is a dog, then x is a mammal" which immediately leads to
  - $\forall x(\text{dog}(x) \rightarrow \text{mammal}(x))$.
- Generally, the sentence $\forall x(P(x) \rightarrow Q(x))$ can be translated as "All individuals with property P also have property Q.".
- **if the universal quantifier is to apply only to individuals with a given property, we use the conditional to restrict the domain.**

RESTRICTIONS OF QUANTIFIERS

- Consider now the statement "Some dogs are brown". This means that there are some animals that are dogs and that are brown. Of course, the statement "x is a dog and x is brown" can be translated as dog(x) $\wedge$ brown(x).
- "There are some brown dogs" can be now translated as $\exists$x(dog(x) $\wedge$ brown(x)).
- The statement $\exists$x(P(x) $\wedge$ Q(x)) can in general be interpreted as "Some individuals with property P have also property Q."
- **To restrict application of the existential quantifier, we use the conjunction.**
- Consider statements containing the word "only" such as "only dogs bark".
- To convert this into predicate calculus, this must be reworded as "It barks only if it is a dog" or, equivalently "If it barks, then is is a dog". Therefore
- $\forall$x(barks(x) $\rightarrow$ dog(x)).

FIRST-ORDER LOGIC EXAMPLES

- For example, "Brothers are siblings" can be written as
  - $\forall x \; \forall y \; Brother(x, y) \Rightarrow Sibling(x, y)$.
- Consecutive quantifiers of the same type can be written as one quantifier with several variables. For example, to say that "siblinghood is a symmetric relationship", we can write
  - $\forall x, y \; Sibling(x, y) \Leftrightarrow Sibling(y, x)$.
- "Everybody loves somebody" means that for every person, there is someone that person loves:
  - $\forall x \; \exists y \; Loves(x, y)$.
  - The order of quantification is very important. It becomes clearer if we insert parentheses. $\forall x \; (\exists y \; Loves(x, y))$ says that everyone has a particular property, namely, the property that they love someone.
- "There is someone who is loved by everyone," we write
  - $\exists y \; \forall x \; Loves(x, y)$.
  - $\exists y \; (\forall x \; Loves(x, y))$ says that someone in the world has a particular property, namely the property of being loved by everybody.

CONNECTIONS BETWEEN ∀ AND ∃

- Everyone dislikes banana is the same as asserting there does not exist someone who likes banana.
- $\forall x \neg Likes(x, banana)$ is equivalent to $\neg \exists x\ Likes(x, banana)$.
- "Everyone likes ice cream" means that there is no one who does not like ice cream.
- $\forall x\ Likes(x, IceCream)$ is equivalent to $\neg \exists x \neg Likes(x, IceCream)$.
- $\forall x \neg P \equiv \neg \exists x P$
- $\neg \forall x P \equiv \exists x \neg P$
- $\forall x P \equiv \neg \exists x \neg P$
- $\exists x P \equiv \neg \forall x \neg P$

EQUALITY

- the equality symbol to signify that two terms refer to the same object. For example, Father (John) = Henry says that the object referred to by Father (John) and the object referred to by Henry are the same.
- Because an interpretation fixes the referent of any term, determining the truth of an equality sentence is simply a matter of seeing that the referents of the two terms are the same object.
- It can also be used with negation to insist that two terms are not the same object. To say that "Richard has at least two brothers", we would write
- $\exists$ x, y Brother (x, Richard) $\wedge$ Brother (y, Richard) $\wedge$ $\neg$(x = y) .
- The sentence $\exists$ x, y Brother (x, Richard) $\wedge$ Brother (y, Richard) **is wrong**

ASSERTIONS AND QUERIES IN FIRST-ORDER LOGIC

- Sentences are added to a knowledge base using TELL, exactly as in propositional logic. Such sentences are called assertions. For example, we can assert that John is a king, Richard is a person, and all kings are persons.
  - TELL(KB, King(John)) .
  - TELL(KB, Person(Richard)) .
  - TELL(KB, $\forall$ x King(x) $\Rightarrow$ Person(x)) .
- We can ask questions of the knowledge base using ASK.
- For example, ASK(KB, King(John)) returns true.

ASSERTIONS AND QUERIES IN FIRST-ORDER LOGIC

- Questions asked with ASK are called queries or goals. Any query that is logically entailed by the knowledge base should be answered affirmatively.
- For example, given the two preceding assertions, the query ASK(KB, Person(John)) should also return true. We can ask quantified queries, such as ASK(KB, $\exists$ x Person(x)) .
- If we want to know what value of x makes the sentence true, we will need a different function, ASKVARS, which we call with ASKVARS(KB,Person(x)) and which yields a stream of answers.
- In this case there will be two answers: {x/John} and {x/Richard}. Such an answer is called a substitution or binding list

THE KINSHIP DOMAIN

- The first example is the domain of family relationships, or kinship.
- This domain includes facts such as "Elizabeth is the mother of Charles" and "Charles is the father of William" and rules such as "One's grandmother is the mother of one's parent."
- The objects in our domain are people. We have two unary predicates, Male and Female.
- Kinship relations—parenthood, brotherhood, marriage, and so on—are represented by binary predicates: Parent, Sibling, Brother , Sister , Child, Daughter, Son, Spouse, Wife, Husband, Grandparent, Grandchild , Cousin, Aunt, and Uncle.
- We use functions for Mother and Father , because every person has exactly one of each of these

THE KINSHIP DOMAIN

- “one’s mother is one’s female parent”: $\forall\ m, c\ Mother(c) = m \Leftrightarrow Female(m) \land Parent(m, c)$.
- “One’s husband is one’s male spouse”: $\forall\ w, h\ Husband(h, w) \Leftrightarrow Male(h) \land Spouse(h, w)$.
- Male and female are disjoint categories: $\forall\ x\ Male(x) \Leftrightarrow \neg Female(x)$.
- Parent and child are inverse relations: $\forall\ p, c\ Parent(p, c) \Leftrightarrow Child(c, p)$.
- A grandparent is a parent of one’s parent: $\forall\ g, c\ Grandparent(g, c) \Leftrightarrow \exists\ p\ Parent(g, p) \land Parent(p, c)$.
- A sibling is another child of one’s parents: $\forall\ x, y\ Sibling(x, y) \Leftrightarrow x = y \land \exists\ p\ Parent(p, x) \land Parent(p, y)$

INFERENCE RULES FOR QUANTIFIERS

- all greedy kings are evil: $\forall\ x\ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$.
  - $King(John) \wedge Greedy(John) \Rightarrow Evil(John)$
  - $King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$
  - $King(Father\ (John)) \wedge Greedy(Father\ (John)) \Rightarrow Evil(Father\ (John))$
- The rule of Universal Instantiation says that we can infer any sentence obtained by substituting a ground term (a term without variables) for the variable.
- To write out the inference rule formally, the notion of substitutions is used. Let SUBST(θ,α) denote the result of applying the substitution θ to the sentence α.
- Then the rule is written $\forall\ v\ \alpha$ / SUBST({v/g}, α) for any variable v and ground term g.
- For example, the three sentences given earlier are obtained with the substitutions {x/John}, {x/Richard}, and {x/Father (John)}

INFERENCE RULES FOR QUANTIFIERS

- In the rule for Existential Instantiation, the variable is replaced by a single new constant symbol. The formal statement is as follows: for any sentence α, variable v, and constant symbol k that does not appear elsewhere in the knowledge base, $\exists\ v\ \alpha$ / SUBST({v/k}, α) .
- For example, from the sentence $\exists$ x Crown(x) $\wedge$ OnHead (x, John) we can infer the sentence
- Crown(C1) $\wedge$ OnHead (C1, John).
- an existentially quantified sentence can be replaced by one instantiation, a universally quantified sentence can be replaced by the set of all possible instantiations.
- Unification and Generalized Modus Ponens rules of propositional rule can be applied in predicate logic as well

FORWARD-CHAINING

- Start with the atomic sentences in the knowledge base and apply Modus Ponens in the forward direction, adding new atomic sentences, until no further inferences can be made.
- **First-order definite clauses:** First-order definite clauses closely resemble propositional definite clauses they are disjunctions of literals of which exactly one is positive. A definite clause either is atomic or is an implication whose antecedent is a conjunction of positive literals and whose consequent is a single positive literal.
- The following are first-order definite clauses: $King(x) \land Greedy(x) \Rightarrow Evil(x)$. $King(John)$. $Greedy(y)$.
- Unlike propositional literals, first-order literals can include variables, in which case those variables are assumed to be universally quantified. Not every knowledge base can be converted into a set of definite clauses because of the single-positive-literal restriction, but many can.

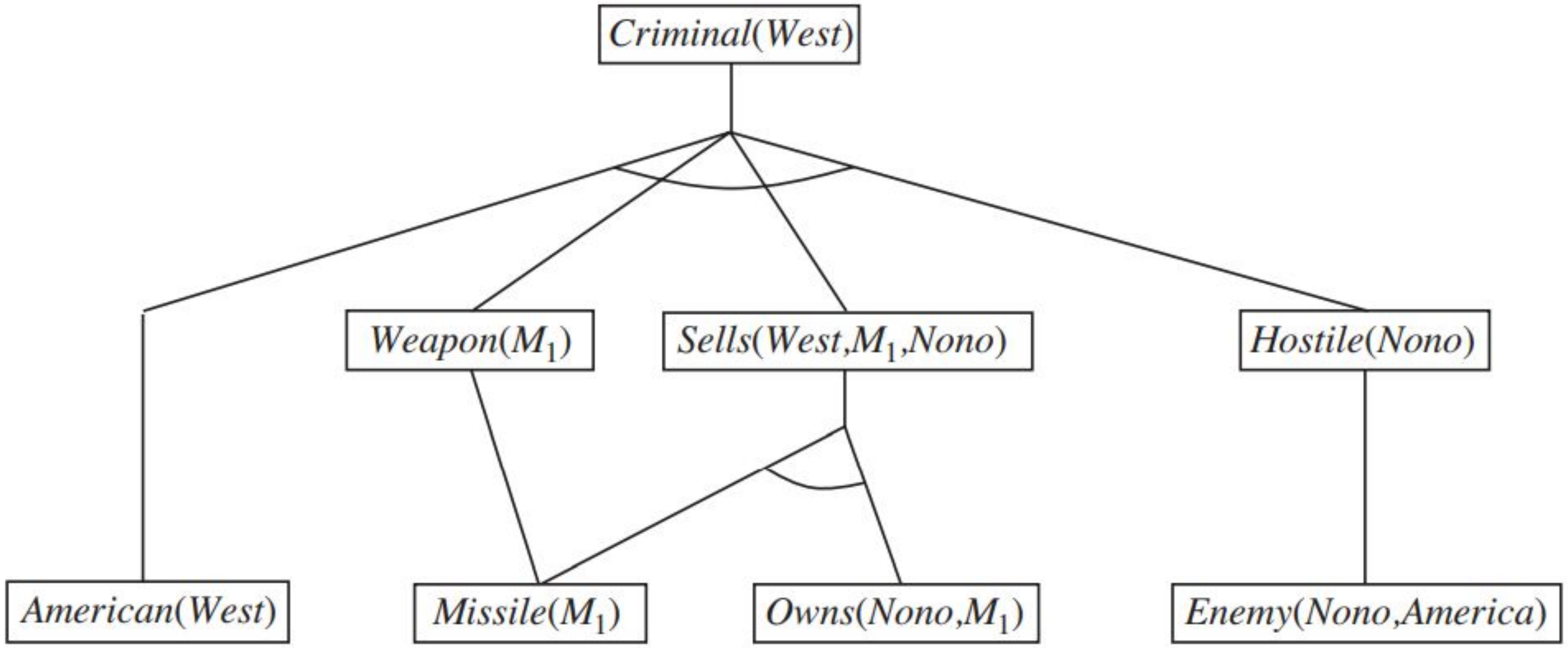
EXAMPLE PROBLEM

- The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- We will prove that West is a criminal.
- First, we will represent these facts as first-order definite clauses. Then how the forward-chaining algorithm solves the problem.
- "... it is a crime for an American to sell weapons to hostile nations":
  - $American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \Rightarrow Criminal(x)$.
- "Nono ... has some missiles."
  - $\exists\ x\ Owns(Nono, x) \land Missile(x)$
- is transformed into two definite clauses by Existential Instantiation, introducing a new constant
  - M1: Owns(Nono, M1)
  - Missile . (M1)

EXAMPLE PROBLEM

- "All of its missiles were sold to it by Colonel West":
  - $Missile(x) \land Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$.
- We will also need to know that missiles are weapons:
  - $Missile(x) \Rightarrow Weapon(x)$
- an enemy of America counts as "hostile":
  - $Enemy(x, America) \Rightarrow Hostile(x)$.
- West, who is American …":
  - $American(West)$.
- "The country Nono, an enemy of America …":
  - $Enemy(Nono, America)$.

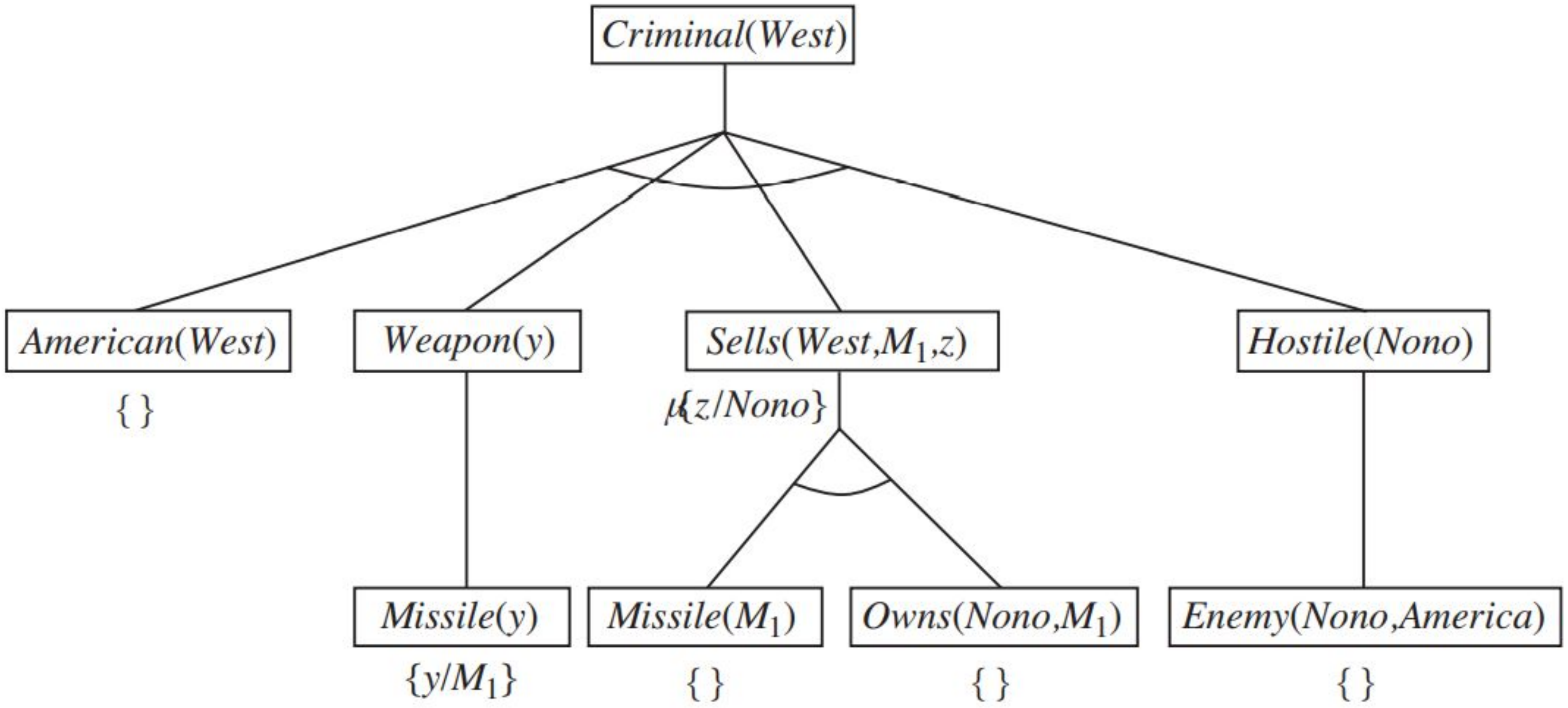
FORWARD-CHAINING



BACKWARD CHAINING

- These algorithms work backward from the goal, chaining through rules to find known facts that support the proof.
- if the knowledge base contains a clause of the form lhs ⇒ goal, where lhs (left-hand side) is a list of conjuncts.

BACKWARD CHAINING



CONJUNCTIVE NORMAL FORM FOR FIRST-ORDER LOGIC

- First-order resolution requires that sentences be in conjunctive normal form: a conjunction of clauses, where each clause is a disjunction of literals.
- Literals can contain variables, which are assumed to be universally quantified.
- For example, the sentence
- $\forall\ x\ American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \Rightarrow Criminal(x)$ becomes, in CNF,
- $\neg American(x) \lor \neg Weapon(y) \lor \neg Sells(x, y, z) \lor \neg Hostile(z) \lor Criminal(x)$.
- Every sentence of first-order logic can be converted into an inferentially equivalent CNF sentence.
- The procedure for conversion to CNF is similar to the propositional case. The principal difference arises from the need to eliminate existential quantifiers.

CONJUNCTIVE NORMAL FORM FOR FIRST-ORDER LOGIC

- "Everyone who loves all animals is loved by someone". $\forall\ x\ [\forall\ y\ Animal(y) \Rightarrow Loves(x, y)] \Rightarrow [\exists\ y\ Loves(y, x)]$.
- The steps are as follows: Eliminate implications: $\forall\ x\ [\neg\forall\ y\ \neg Animal(y) \lor Loves(x, y)] \lor [\exists\ y\ Loves(y, x)]$.
- Move ¬ inwards: $\forall\ x\ [\exists\ y\ \neg(\neg Animal(y) \lor Loves(x, y))] \lor [\exists\ y\ Loves(y, x)]$.
- $\forall\ x\ [\exists\ y\ \neg\neg Animal(y) \land \neg Loves(x, y)] \lor [\exists\ y\ Loves(y, x)]$.
- $\forall\ x\ [\exists\ y\ Animal(y) \land \neg Loves(x, y)] \lor [\exists\ y\ Loves(y, x)]$.
- The sentence now reads "Either there is some animal that x doesn't love, or (if this is not the case) someone loves x." Clearly, the meaning of the original sentence has been preserved.

CONJUNCTIVE NORMAL FORM FOR FIRST-ORDER LOGIC

- Standardize variables: For sentences like $(\exists x\, P(x)) \vee (\exists x\, Q(x))$ which use the same variable name twice, change the name of one of the variables. This avoids confusion later when we drop the quantifiers.
- Thus, we have $\forall x\, [\exists y\, Animal(y) \wedge \neg Loves(x, y)] \vee [\exists z\, Loves(z, x)]$.
- Skolemize: Skolemization is the process of removing existential quantifiers by elimination. translate $\exists x\, P(x)$ into $P(A)$, where A is a new constant. However, we can't apply Existential Instantiation to our sentence above because it doesn't match the pattern $\exists v\, \alpha$; only parts of the sentence match the pattern.
- If we apply the rule to the two matching parts, we get $\forall x\, [Animal(A) \wedge \neg Loves(x,A)] \vee Loves(B,x)$
- which has the wrong meaning entirely: it says that everyone either fails to love a particular animal A or is loved by some particular entity B.
- In fact, our original sentence allows each person to fail to love a different animal or to be loved by a different person. Thus, we want the Skolem entities to depend on x and z: $\forall x\, [Animal(F(x)) \wedge \neg Loves(x, F(x))] \vee Loves(G(z), x)$

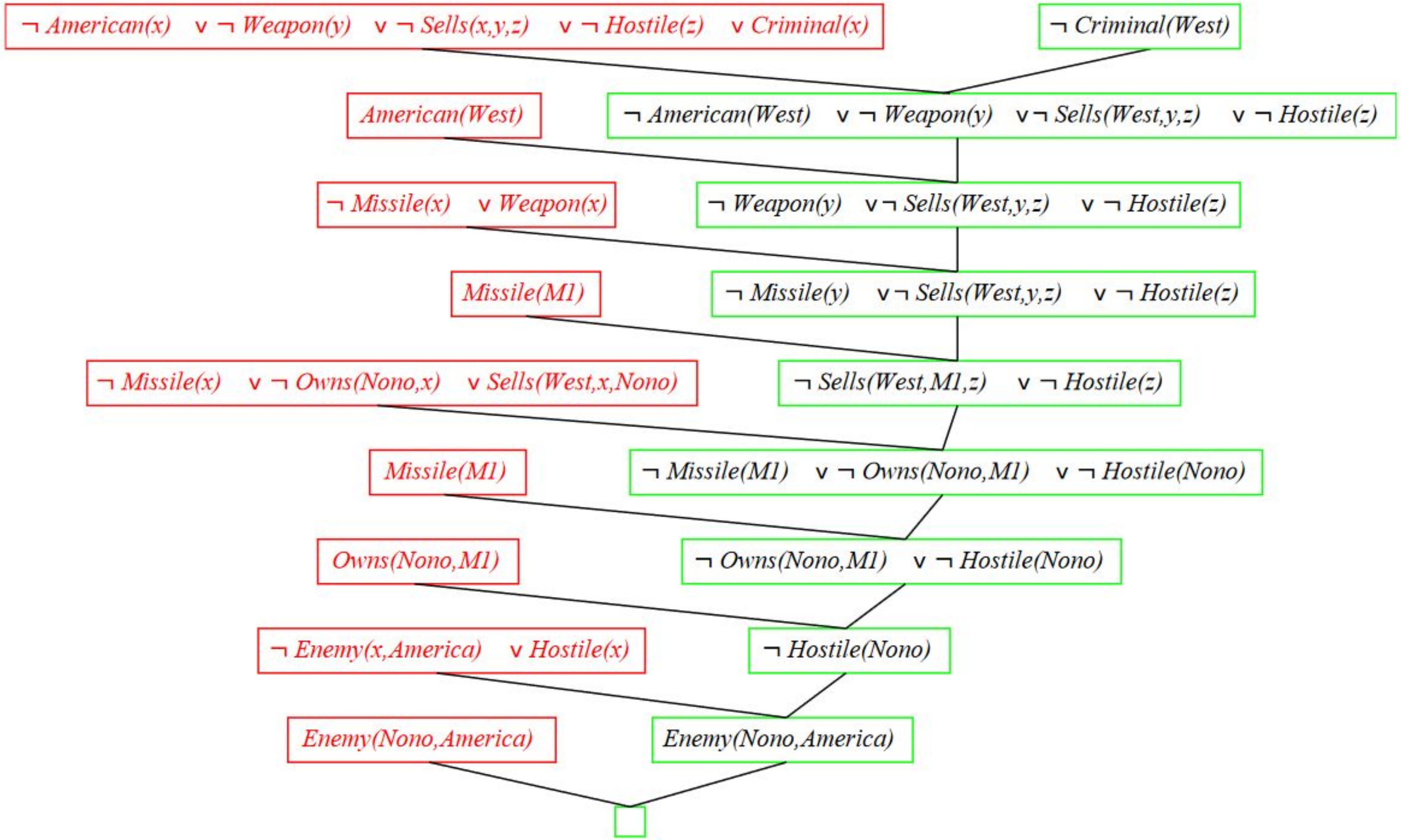
CONJUNCTIVE NORMAL FORM FOR FIRST-ORDER LOGIC

- Drop universal quantifiers: At this point, all remaining variables must be universally quantified. Moreover, the sentence is equivalent to one in which all the universal quantifiers have been moved to the left. We can therefore drop the universal quantifiers: $[Animal(F(x)) \wedge \neg Loves(x, F(x))] \vee Loves(G(z), x)$.
- Distribute $\vee$ over $\wedge$: $[Animal(F(x)) \vee Loves(G(z), x)] \wedge [\neg Loves(x, F(x)) \vee Loves(G(z), x)]$. This step may also require flattening out nested conjunctions and disjunctions.
- The sentence is now in CNF and consists of two clauses.

THE RESOLUTION INFERENCE RULE

- Two clauses, which are assumed to be standardized apart so that they share no variables, can be resolved if they contain complementary literals.
- Propositional literals are complementary if one is the negation of the other; first-order literals are complementary if one unifies with the negation of the other.
- [Animal(F(x)) $\vee$ Loves(G(x), x)] and [¬Loves(u, v) $\vee$ ¬Kills(u, v)] by eliminating the complementary literals Loves(G(x), x) and ¬Loves(u, v), with unifier $\theta = \{u/G(x), v/x\}$,
- to produce the resolvent clause [Animal(F(x)) $\vee$ ¬Kills(G(x), x)] .

THE RESOLUTION INFERENCE RULE



EXAMPLE PROBLEM

- Everyone who loves all animals is loved by someone. Anyone who kills an animal is loved by no one. Jack loves all animals. Either Jack or Curiosity killed the cat, who is named Tuna. Prove Did Curiosity kill the cat?
- A. $\forall\ x\ [\forall\ y\ Animal(y) \Rightarrow Loves(x, y)] \Rightarrow [\exists\ y\ Loves(y, x)]$
- B. $\forall\ x\ [\exists\ z\ Animal(z) \land Kills(x, z)] \Rightarrow [\forall\ y\ \neg Loves(y, x)]$
- C. $\forall\ x\ Animal(x) \Rightarrow Loves(Jack, x)$
- D. $Kills(Jack, Tuna) \lor Kills(Curiosity, Tuna)$
- E. $Cat(Tuna)$
- F. $\forall\ x\ Cat(x) \Rightarrow Animal(x)$
- G. $\neg Kills(Curiosity, Tuna)$

EXAMPLE PROBLEM

- A1. Animal(F(x)) ∨ Loves(G(x), x)
- A2. ¬Loves(x, F(x)) ∨ Loves(G(x), x)
- B. ¬Loves(y, x) ∨ ¬Animal(z) ∨ ¬Kills(x, z)
- C. ¬Animal(x) ∨ Loves(Jack, x)
- D. Kills(Jack, Tuna) ∨ Kills(Curiosity, Tuna)
- E. Cat(Tuna)
- F. ¬Cat(x) ∨ Animal(x)
- G. ¬Kills(Curiosity, Tuna)

EXAMPLE PROBLEM