Week1: A set of propositional formulas is **satisfiable** if some interpretation satisfies all formulas in the set.

A propositional formula is a **tautology** if every interpretation satisfies

F Is **equivalent** to G (symbolically,⇔) if, for every interpretation I Fi = Gi In other words, F ↔ G is a tautology

A set of formulas **entails** a formula (symbolically,Γ⊨G) if, every interpretation that satisfies all formulas in T satisfies G also.

F is a tautology iff ¬F is unsatisfiable. F is **equivalent** to G iff F ↔ G is a **tautology.** F is equivalent to G iff F entails G and G entails F!

A **literal** is either an atom or its negation A **clause** is of the form L1∨⋯∨ Ln(n≥1) where each Li is a literal

A formula is in **conjunctive normal form (CNF)** if it is of the form F1∧⋯∧Fm()≥1) where each Fi is a clause

Reasoning is a form of calculation over symbols standing for propositions rather than numbers. While **propositions** are abstract objects, their representations are concrete objects and can be easily manipulated

Deductive Reasoning: Usually, we are interested in deriving implicit, entailed facts from a given collection of explicitly represented facts that are logically 1. Sound (the derived proposition must be true, given that the premises are true), and 2. Complete (all true consequences can be derived).

Abductive Reasoning: Given a background theory, a set of explanations and an observation, find the most likely explanation

Default Reasoning: In the absence of evidence to the contrary, we jump to a conclusion

In **propositional logic**, an atom represents a proposition, which is either true or false. A **propositional signature** is a set of symbols called atoms.

A set T of formulas **Entails** a formula F For each int I For each formula G in T in check if I satis G If no, continue to next interpretation If yes: check if Satisfies If yes: continue to next interpretation If no: exit and report “Not Entailed”	**CLAUSIFY** (F) eliminate from F all connectives other than ¬, ∧, ∨ and distribute¬ over ∧ and ∨ until It applies to atoms only;distribute ∨ over ∧ until it applies to literals only; return the set of conjunctive terms of the	**Unit Propagation 1,** For any CNF F and atom A, F|A is obtained from F by replacing all occurrences of A by ⊤ and simplifying the result by removing -all clauses containing the disjunctive term ⊤, and - the disjunctive terms ¬⊤ in all remaining clauses	**Unit propagation 2.** Similarly, F|¬A is the result of replacing all occurrences of A by ⊥ and simplifying the result by removing -all clauses containing the disjunctive term ¬⊥, and -the disjunctive terms ⊥ in all remaining clauses.	A **propositional formul**a of is defined recursively as follows: -Every atom is a formula - Both 0 place connectives are formulas - If F is a formula then ¬F is a formula - For any binary connective ⊙, if F and G are formulas then (F⊙G) is a formula

Week 2: **FOL: F** is logically valid iff ¬F is unsatisfiable. Individuals are expressed by **object/function constants:** andy, paul , father (andy). Properties are expressed by **predicates.** S, I, Y are predicates. S(andy), Andy is a student.I(paul) Paul is an instructor. Y(andy,paul). Andy is younger than Paul. **A Herbrand** interpretation is a special case of first-order interpretation| A Herbrand interpretation of signature σ (containing at least one object constant) is an interpretation of σ such that -its universe (Herbrand Universe) is the set of all ground (i.e., variable-free) terms of σ - every ground term is interpreted as itself (tI=t)

Signature: 2 types symbols -function constants (with arity n): +/2, a/0, father/1 •function constants of arity 0 are called object constants -predicate constants (with arity n): even/1, >/2 • predicate constants of arity 0 are called propositional constants	**Other symbols-non sign** (object) variables: x, y, z, -the propositional connectives:⊥,⊤,¬ ,∧ ,∨ ,→ ,↔ - the universal quantifier ∀ and the existential quantifier ∃ - the parentheses and the comma	A **term** is meant to denote an individual. It is defined recursively: - an object constant is a term - an object variable is a term - for every function constant f of arity n (n>0), if t1,...,tn are terms then so is f(t1,...,tn)	**FOL(propos+below)** -every atomic formula of σ is a formula -if F is a formula then ∀x F and ∃x F are formulas. Whereas propositional logic assumes world contains facts, **First order logic** (like natural language) assumes the world contains Objects, Functions, Relations	N queens puzzle: 1 {queen(R,1..n)} 1 :- R=1..n. :-queen(R1,C), queen(R2,C), R1!=R2. :-queen(R1,C1), queen(R2,C2), R1!=R2, |R1-R2|=|C1-C2|. No two queens share the same row, column, or diagonal

Week 3: If two formulas are equivalent under propositional logic, then they have the **same minimal models.**, converse is not true?. A propositional rule is **definite** if a) its head is an atom, and b) its body (if it has one) does not contain negation. A **definite** program has a **unique minimal model**. A **stable model** of a definite program Π is **the minimal model** of Π. A **stable mode**l of a **positive** program Π is **a minimal model** of Π. A **critical part** of a propositional rule is a subformula of its head or body that begins with negation but is not part of another subformula that begins with negation. The **reduct** Π x of Π relative to an interpretation X is the positive propositional program obtained from Π by replacing each critical part ¬H of each of its rules - by ⊤ if X satisfies ¬H;-by ⊥ otherwise. **Equivalent propositional** programs (i.e., having the same models) may have different stable models.

Fibonacci Numbers Fib(0) = 0 Fib(1) = 1 Fib(n+2) = Fib(n) + Fib(n+1)for n≥0 fib(i,m):i-thFibonacci number is m fib(0,0). fib(1,1). fib(N+2,F1+F2) :-fib(N,F1),fib(N+1,F2), N=0..(n-2).	**Factorials** Fac(0) = 1 Fac(n+1) = (n+1)×Fac(n) %fac(i,m) :i-th factorial is mfac(0,1).fac(N+1, (N+1)*F) :-fac(N,F), N=1..n.fac(F) :-fac(N,F).	**Stable model of a clingo program** 1.Construct the propositional image 2. Find the reduct relative to {p(a),q(a)} 3. Check if {p(a),q(a)} is a minimal model of the reduct	Stable model of a propositional program Π 1. Guess an interpretation X 2.Find the reduct of Π relative to X (i.e.,Π") 3.Check if X satisfies Π X a.If yes, continue b.If no, conclude X is not a stable model of Π 4.Check if no other interpretation that is smaller than X satisfies Π". I.e., for each interpretation Y that is smaller than X, a. If Y satisfies Π", conclude X is not a stable model of Π B. Else continue 5.Conclude X is a stable model ofΠ

1:1 function domain(1;2;3). codomain(a;b;c;d;e). {f(X,Y): codomain(Y)} =1 :-domain(X). X=X1 :- f(X,Y), f(X1,Y).	% onto domain(1;2;3). codomain(a;b). {f(X,Y): codomain(Y)} = 1 :-domain(X). :-{f(X,Y): domain(X)} = 0, codomain(Y).	X equals 8 (soln 1) ∃y [Q(x,y) ⋀ Q(y,a)] ⋀ ¬∃y,z [Q(x,y)⋀Q(y,a)⋀Q(x,z)⋀Q(z,a)⋀y≠z] ∃y [Q(x,y) ⋀ Q(y,a)] means that there is a number y between x and a, thus x can only be {0,1,2, ...,7,8}. ¬∃y,z [Q(x,y)⋀Q(y,a)⋀Q(x,z)⋀Q(z,a)⋀y≠z] means that we cannot find 2 different numbers y and z between x and a. Thus x now can only be 8.	X equals 8 (soln 2) ∀y [P(y) ⋀ Q(y,a) → Q(y,x)] ⋀ ∃y [Q(x,y) ⋀ Q(y,a)] ∃y [Q(x,y) ⋀ Q(y,a)] means that there is a number y between x and a, thus x can only be {0,1,2, ...,7,8}. ∀y [P(y) ⋀ Q(y,a) → Q(y,x)] means that for all prime number y that is smaller than 10, y must be smaller than x. The possible values of y are {2,3,5,7}, and	Student taking CSE 579 2 solutions: 1. ∃x∃y(P(x)∧P(y)∧x≠y∧∀z (P(z) -> (z=x ∨ z=y))) 2. ∃x∃y∀z[P(x)∧P(y)∧x≠y∧((x≠z∧y≠z)-> ¬P(z))] **DPLL (F,U):** Unit-propagate (F,U); If F contains the empty clause then return; If F = T then exit with a model of U; L <- a literal containing an atom from F; DPLL(F|L, UU{L}); DPLL(F|notL,UU{notL})

			since they are all smaller than x, x can only be 8.	To solve satisfiability of F, call DPLL(F,∅)