Soundness and Completeness of Resolution

Resolution is **sound**:
If \( \varphi \vdash \psi \) then \( \varphi \models \psi \)

Resolution is **refutation complete**:
If \( \varphi \models \psi \) then \([\text{CNF}(\varphi) \land \text{CNF}(\lnot \psi)] \vdash ()\)

So:
(1) If it’s true, resolution will find it
(2) Everything resolution finds is true
Resolution Theorem Proving

To determine: $\text{KB} \models ? \beta$

Algorithm:

1. Convert $\text{KB}$ and $\neg \beta$ to CNF, conjoin them to get $\text{KB}'$
2. Apply resolution to $\text{KB}'$ until you get an empty clause
Resolution Theorem Proving

To determine: $KB \models ? \beta$

Algorithm:

1. Convert $KB$ and $\neg \beta$ to CNF, conjoin them to get $KB'$
2. Apply resolution to $KB'$ until you get an empty clause
Conversion to CNF

Summary

1. Remove implications:
   Replace \( \alpha \implies \beta \) with \( \neg \alpha \lor \beta \) everywhere
   [no more “\( \implies \)”]

2. Push in \( \neg \)
   Replace \( \neg (\alpha \lor \beta) \) with \( \neg \alpha \land \neg \beta \)
   [“\( \neg \)” only before propositional symbols]

   Replace \( \neg (\alpha \land \beta) \) with \( \neg \alpha \lor \neg \beta \)

3. Remove double negations:
   Replace \( \neg\neg \alpha \) with \( \alpha \)
   [“\( \neg \)” only singletons]

4. Distribute \( \lor \) over \( \land \):
   Rewrite \( \alpha \lor (\beta \land \gamma) \) as \( (\alpha \lor \beta) \land (\alpha \lor \gamma) \)
Resolution Theorem Proving

To determine: $KB \vdash \beta$

Algorithm:

1. Convert $KB$ and $\neg \beta$ to CNF, conjoin them to get $KB'$
2. **Apply resolution to $KB'$** until you get an empty clause
Resolution Inference Rule

\[
(p \lor \alpha_1 \lor ... \lor \alpha_a) \\
\frac{(\neg p \lor \beta_1 \lor ... \lor \beta_b)}{(\alpha_1 \lor ... \lor \alpha_a \lor \beta_1 \lor ... \lor \beta_b)}
\]

where each \( \alpha_i \) and \( \beta_j \) are literals (symbols or their negations) and \( p \) is a propositional symbol.
Resolution Inference Rule

\[(\neg p \lor \beta_1 \lor \ldots \lor \beta_b) \quad \text{where each } \alpha_i \text{ and } \beta_j \text{ are literals (symbols or their negations)} \quad \text{and} \quad p \text{ is a propositional symbol}\]

This really means

\[(\alpha_1 \lor \ldots \lor \alpha_{i-1} \lor p \lor \alpha_{i+1} \lor \ldots \lor \alpha_a) \quad \text{but this becomes more awkward notationally}\]
Resolution Inference Rule Example

\[(P \lor Q \lor \neg R)\]
\[\neg S \lor T \lor \neg Q \lor V\]
Resolution Inference Rule Example

\((P \lor Q \lor \neg R)\)
\((-S \lor T \lor \neg Q \lor V)\)
Resolution Inference Rule Example

\[(P \lor Q \lor \neg R)\]
\[\neg S \lor T \lor \neg Q \lor V\]
\[(P \lor \neg R \lor \neg S \lor T \lor V)\]
Resolution Inference Rule Example

WARNING!

(P ∨ Q ∨ ¬R)
(¬P ∨ ¬Q ∨ V)
Resolution Inference Rule Example

WARNING!

\((P \lor Q \lor \neg R)\)
\((-P \lor -Q \lor V)\)
Resolution Inference Rule Example

WARNING!

\((P \lor Q \lor \neg R)\)
\((\neg P \lor \neg Q \lor V)\)
\((Q \lor \neg R \lor \neg Q \lor V)\)
Resolution Inference Rule Example

WARNING!

\[(P \lor Q \lor \neg R) \land (\neg P \lor \neg Q \lor V)\]
Resolution Inference Rule Example

WARNING!

\((P \lor Q \lor \neg R)\)
\((\neg P \lor \neg Q \lor V)\)
\((P \lor \neg R \lor \neg P \lor V)\)
Resolution Inference Rule Example

WARNING!

\[
(P \lor Q \lor \neg R) \\
(\neg P \lor \neg Q \lor V) \quad \text{OR} \quad (\neg P \lor \neg Q \lor V) \\
(P \lor \neg R \lor \neg P \lor V) \\
(Q \lor \neg R \lor \neg Q \lor V) \\
\] 

NOT

\[
(\neg R \lor V)
\]
Resolution Theorem Proving

To determine: $\text{KB} \vdash \beta$

Algorithm:

1. Convert KB and $\neg \beta$ to CNF, conjoin them to get KB’
2. Apply resolution to KB’ until you get an empty clause
Resolution Proof Example

\[(my \implies i) \land (\neg my \implies (mo \land ml)) \land ((i \lor ml) \implies h) \land (h \implies ma) \models ma?\]

Compute CNF( \( (my \implies i) \land (\neg my \implies (mo \land ml)) \land ((i \lor ml) \implies h) \land (h \implies ma) \) )

Compute CNF(\(\neg ma \) )

\[KB' = (\neg my \lor i) \land (my \lor mo) \land (my \lor ml) \land (\neg i \lor h) \land (\neg ml \lor h) \land (\neg h \lor ma) \land (\neg ma)\]
Resolution Proof Example

\[(\neg \text{my} \lor \text{i}) \land (\text{my} \lor \text{mo}) \land (\text{my} \lor \text{ml}) \land (\neg \text{i} \lor \text{h}) \land (\neg \text{ml} \lor \text{h}) \land (\neg \text{h} \lor \text{ma}) \land (\neg \text{ma})\]
Resolution Proof Example

(¬ my ∨ i)  (my ∨ mo)  (my ∨ ml)  (¬i ∨ h)  (¬ml ∨ h)  (¬ h ∨ ma)  (¬ ma)
Resolution Proof Example

(¬my ∨ i) (my ∨ mo) (my ∨ ml) (¬i ∨ h) (¬ml ∨ h) (¬h ∨ ma) (¬ma)
Resolution Proof Example

$$(-my \lor i) \land (my \lor mo) \land (my \lor ml) \land (-i \lor h) \land (-ml \lor h) \land (-h \lor ma) \land (-ma) \land (i \lor ml)$$
Resolution Proof Example

\[ (\neg my \vee i) \quad (my \vee mo) \quad (my \vee ml) \quad (\neg i \vee h) \quad (\neg ml \vee h) \quad (\neg h \vee ma) \quad (\neg ma) \quad (i \vee ml) \]
Resolution Proof Example

\((\neg \text{my } \lor \text{i}) \ (\text{my } \lor \text{mo}) \ (\text{my } \lor \text{ml}) \ (\neg \text{i } \lor \text{h}) \ (\neg \text{ml } \lor \text{h}) \ (\neg \text{h } \lor \text{ma}) \ (\neg \text{ma}) \ (\text{i } \lor \text{ml})\)
Resolution Proof Example

\[ \neg my \lor i \quad (my \lor mo) \quad (my \lor ml) \quad \neg i \lor h \quad \neg ml \lor h \quad \neg h \lor ma \quad \neg ma \\
(i \lor ml) \\
(ml \lor h) \]
Resolution Proof Example

\((\neg my \lor i) \land (my \lor mo) \land (my \lor ml) \land (\neg i \lor h) \land (\neg ml \lor h) \land (\neg h \lor ma) \land (\neg ma)\)

\((i \lor ml)\)

\((ml \lor h)\)
Resolution Proof Example

(\neg my \lor i) \ (my \lor mo) \ (my \lor ml) \ (\neg i \lor h) \ (\neg ml \lor h) \ (\neg h \lor ma) \ (\neg ma) \\
(i \lor ml) \\
(ml \lor h)
Resolution Proof Example

$$\neg my \lor i \quad (my \lor mo) \quad (my \lor ml) \quad (\neg i \lor h) \quad (\neg ml \lor h) \quad (\neg h \lor ma) \quad (\neg ma)$$

$$i \lor ml$$

$$ml \lor h$$

$$h$$
Resolution Proof Example

(¬my ∨ i) (my ∨ mo) (my ∨ ml) (¬i ∨ h) (¬ml ∨ h) (¬h ∨ ma) (¬ma)
(i ∨ ml)

(ml ∨ h)

(h)
Resolution Proof Example

\((\neg \text{my} \lor i)\) \(\ (\text{my} \lor \text{mo})\) \(\ (\text{my} \lor \text{ml})\) \(\ (\neg i \lor h)\) \(\ (\neg \text{ml} \lor h)\) \(\ (\neg \text{h} \lor \text{ma})\) \(\ (\neg \text{ma})\) 

\((i \lor \text{ml})\) 

\((\text{ml} \lor h)\) 

\((h)\)
Resolution Proof Example

\((\neg my \lor i) \ (my \lor mo) \ (my \lor ml) \ (\neg i \lor h) \ (\neg ml \lor h) \ (\neg h \lor ma) \ (\neg ma)\)

\((i \lor ml)\)

\((ml \lor h)\)

\((h)\)

\((ma)\)
Resolution Proof Example

(¬ my \lor i) \quad (my \lor mo) \quad (my \lor ml) \quad (¬ i \lor h) \quad (¬ ml \lor h) \quad (¬ h \lor ma) \quad (¬ ma) \\
(i \lor ml) \\
(ml \lor h) \\
(h) \\
(ma)
Resolution Proof Example

\( (\neg my \lor i) \ (my \lor mo) \ (my \lor ml) \ (\neg i \lor h) \ (\neg ml \lor h) \ (\neg h \lor ma) \ (\neg ma) \)

\( (i \lor ml) \)

\( (ml \lor h) \)

\( (h) \)

\( (ma) \)
Resolution Proof Example

\((\neg my \lor i) (my \lor mo) (my \lor ml) (\neg i \lor h) (\neg ml \lor h) (\neg h \lor ma) (\neg ma)\)

\((i \lor ml)\)

\((ml \lor h)\)

\((h)\)

\((ma)\)

\((()\))
Premises

1. \((\neg \text{my} \lor \text{i})\)
2. \((\text{my} \lor \text{mo})\)
3. \((\text{my} \lor \text{ml})\)
4. \((\neg \text{i} \lor \text{h})\)
5. \((\neg \text{ml} \lor \text{h})\)
6. \((\neg \text{h} \lor \text{ma})\)

Negated Goal

7. \(\neg \text{ma}\)
Resolution Proof Example

\[(\neg \text{my} \lor \text{i}) \ (\text{my} \lor \text{mo}) \ (\text{my} \lor \text{ml}) \ (\neg \text{i} \lor \text{h}) \ (\neg \text{ml} \lor \text{h}) \ (\neg \text{h} \lor \text{ma}) \ (\neg \text{ma}) \]

\[(\text{i} \lor \text{ml}) \]

\[(\text{ml} \lor \text{h}) \]

\[(\text{h}) \]

\[(\text{ma}) \]

\[() \]
Resolution Proof Example

\[ (\neg \text{my} \lor \text{i}) \quad (\text{my} \lor \text{mo}) \quad (\text{my} \lor \text{ml}) \quad (\neg \text{i} \lor \text{h}) \quad (\neg \text{ml} \lor \text{h}) \quad (\neg \text{h} \lor \text{ma}) \quad (\neg \text{ma}) \quad \]
\[ (\text{i} \lor \text{ml}) \quad \]
\[ (\text{ml} \lor \text{h}) \quad \]
\[ (\text{h}) \quad \]
\[ (\text{ma}) \quad \]
\[ () \quad \]

Is this the only way to do it?
Resolution Proof Example

\((\neg \text{my} \lor \text{i}) \ (\text{my} \lor \text{mo}) \ (\text{my} \lor \text{ml}) \ (\neg \text{i} \lor \text{h}) \ (\neg \text{ml} \lor \text{h}) \ (\neg \text{h} \lor \text{ma}) \ (\neg \text{ma})\)

\((\text{i} \lor \text{ml})\)

\((\text{ml} \lor \text{h})\)

\((\text{h})\)

\((\text{ma})\)

\((()\)\)

Is this the only way to do it?

No. But we just need one way, not all ways. Any proof is fine.
Resolution Proof Example

\((\neg \text{my} \lor \text{i}) \quad (\text{my} \lor \text{mo}) \quad (\text{my} \lor \text{ml}) \quad (\neg \text{i} \lor \text{h}) \quad (\neg \text{ml} \lor \text{h}) \quad (\neg \text{h} \lor \text{ma}) \quad (\neg \text{ma})\)

\((\text{i} \lor \text{ml})\)

\((\text{ml} \lor \text{h})\)

\((\text{h})\)

\((\text{ma})\)

\((\text{})\)

Is this the only way to do it?

No. But we just need one way, not all ways. Any proof is fine.

Which one? Choose a search method.
Resolution as Search

States: The collection (conjunction) of what you know (clauses)

Operators: Take two clauses and resolve them

Initial State: $\text{CNF}(\varphi) \land \text{CNF}(\neg \psi)$

Goal State: State includes ($\Box$)
Resolution as Search

States: The collection (conjunction) of what you know (clauses)

Operators: Take two clauses and resolve them

Initial State: $\text{CNF}(\varphi) \land \text{CNF}(\neg \psi)$

Goal State: State includes ($\emptyset$)

Search space is finite
Resolution as Search

States: The collection (conjunction) of what you know (clauses)

Operators: Take two clauses and resolve them

Initial State: $\text{CNF}(\varphi) \land \text{CNF}(\neg\psi)$

Goal State: State includes ()

Search space is finite

You will either reach () or run out of things to resolve
Tractability of Resolution
Tractability of Resolution

1. Convert KB and $\neg \beta$ to CNF, conjoin them to get $KB'$
2. Apply resolution to $KB'$ until you get an empty clause
Tractability of Resolution

1. Convert KB and $\neg \beta$ to CNF, conjoin them to get KB’
2. Apply resolution to KB’ until you get an empty clause

$|\text{CNF}(\varphi)|$ can be exponential in $|\varphi|$

Exercise for the student:
See if you can figure out what sentences $\varphi$ can blow up exponentially when converted to CNF
Tractability of Resolution

1. Convert KB and $\neg \beta$ to CNF, conjoin them to get KB’
2. Apply resolution to KB’ until you get an empty clause
Tractability of Resolution

1. Convert KB and \( \neg \beta \) to CNF, conjoin them to get KB’
2. Apply resolution to KB’ until you get an empty clause

Resolution introduced 1937
popularized for AI by Davis and Putnam in 1960, Robinson in 1965
1985 (Haken): There are statements whose proof requires a number of steps exponential in the size of the problem
1988 (Chvatal and Szemeredi): With high probability any sufficiently large random CNF formula over n variables
   1. is unsatisfiable and
   2. requires an exponentially long resolution proof
Tractability of Resolution

What to do?
Tractability of Resolution

What to do?

Hope for the best
Tractability of Resolution

What to do?

Hope for the best
Consider restricted languages
Horn Clause Logic

• There are restricted forms of propositional logic for which $|\text{CNF}(\varphi)|$ is polynomial in $|\varphi|$

• Propositional Horn Clause Logic:

\[
\begin{align*}
(p_{1,1} \land \ldots \land p_{1,k_1}) & \Rightarrow q_1 \\
& \hspace{1cm} \vdots \\
(p_{n,1} \land \ldots \land p_{n,k_n}) & \Rightarrow q_n \\
& \hspace{1cm} r_1, r_2, \ldots
\end{align*}
\]

• $\text{CNF}(|\varphi|)$ is linear in $|\varphi|$ for Horn Clause Logic

• There are specialized inference methods for Horn Clause Logic that run in time linear in KB and $\beta$
Tractability of Resolution

What to do?
Hope for the best
Consider restricted languages
Tractability of Resolution

What to do?

Hope for the best
Consider restricted languages
Consider other inference operations
Other Inference Approaches for Propositional Logic

\( \varphi \) is satisfiable if there is an assignment of truth values to each of the propositional symbols in \( \varphi \) that makes \( \varphi \) true

\( \not \models \neg \varphi \)
Other Inference Approaches for Propositional Logic

\( \varphi \) is **satisfiable** if there is an assignment of truth values to each of the propositional symbols in \( \varphi \) that makes \( \varphi \) true

\[ \neg \neg \varphi \]

But we will search for a satisfying assignment directly
Example: N-Queens

• Encode in propositional logic (“Queens(N)“)
  • Qij: There is a queen in position (i,j)
  • One queen in each row, column, diagonal:
    • (Qi1 ∨ Qi2 ∨ ... ∨ QiN) for i ∈ {1,...,N}
    • (Q1j ∨ Q2j ∨ ... ∨ QNj) for j ∈ {1,...,N}
    • (Q11 ∨ Q22 ∨ ... ∨ QNN) etc. for each diagonal
  • No two attacking queens:
    • ¬ (Qab ∧ Qac) for all b ≠ c
    • ¬ (Qab ∧ Qdb) for all a ≠ d
    • ¬ (Qij ∧ Qkl) for all k=i±m, l=j±m, m ≠ 0
    • ¬ (Qij ∧ Qkl) for all i+j = k+l, i ≠ k
Example: N-Queens

• If N = 2 or 3
Example: N-Queens

• If $N = 2$ or $3$ there is no legal placement of questions
Example: N-Queens

• If N = 2 or 3 there is no legal placement of questions
  • Queens(2) and Queens(3) are unsatisfiable
Example: N-Queens

• If N = 2 or 3 there is no legal placement of questions
  • Queens(2) and Queens(3) are unsatisfiable
  • Could use resolution to prove it
Example: N-Queens

• If N = 2 or 3 there is no legal placement of questions
  • Queens(2) and Queens(3) are unsatisfiable
  • Could use resolution to prove it

• For N > 3
Example: N-Queens

• If N = 2 or 3 there is no legal placement of questions
  • Queens(2) and Queens(3) are unsatisfiable
  • Could use resolution to prove it

• For N > 3 there is always a legal placement of queens
Example: N-Queens

• If N = 2 or 3 there is no legal placement of questions
  • Queens(2) and Queens(3) are unsatisfiable
  • Could use resolution to prove it

• For N > 3 there is always a legal placement of queens
  • Queens(N) is satisfiable for N>3
Example: N-Queens

• If N = 2 or 3 there is no legal placement of questions
  • Queens(2) and Queens(3) are unsatisfiable
  • Could use resolution to prove it

• For N > 3 there is always a legal placement of queens
  • Queens(N) is satisfiable for N>3
  • Don’t just want to be told there is a placement, want an actual placement!
Searching for Truth Assignments

• Brute-force search ("backtracking") for CNF:
  • Assign a variable True or False
  • Is any clause violated?
    • If not, go on to another variable
    • If yes, backtrack to most recent variable assignment and flip it
    • If both options have been tried and failed, backtrack to the next most recent assignment
  • Complete
  • Davis-Putnam: Exploiting additional observations about CNF that simplify the problem in certain cases
  • Further efficiency: Grab bag of clever ideas
  • Guaranteed to find a solution, worst-case exponential
Searching for Truth Assignments

• Hill climbing: Surprisingly effective
  • Pick random assignment
  • If it doesn’t satisfy the sentence, flip one of the variable’s values
    • GSAT:
      • Flip whichever variable most increases the number of satisfied terms, or
      • With some probability flip a random variable
    • WalkSAT
      • Pick an unsatisfied clause at random
      • Flip which variable violates the fewest previously satisfied clauses, or
      • With some probability flip a random variable in the clause

• Significant real-world use
Tractability of Resolution

What to do?

Hope for the best
Consider restricted languages
Consider other inference operations
Logic for AI

- Propositional logic despite intractability, is often still too restrictive
Logic for AI

• Propositional logic despite intractability, is often still too restrictive

• Use first-order logic instead
Logic for AI

• Propositional logic despite intractability, is often still too restrictive

• Use first-order logic instead

(There many other logic variants that sample different tradeoff points between expressiveness and tractability that we won’t be able to explore)
Example

• Mother(Alice,Charlie)
• Father(Bob,Charlie)
Example

• Mother(Alice,Charlie)
• Father(Bob,Charlie)
• $\forall x,y$ Mother($x,y$) $\Rightarrow$ Parent($x,y$)
• $\forall x,y$ Father($x,y$) $\Rightarrow$ Parent($x,y$)
Example

- Mother(Alice, Charlie)
- Father(Bob, Charlie)
- $\forall x, y \text{ Mother}(x, y) \Rightarrow \text{Parent}(x, y)$
- $\forall x, y \text{ Father}(x, y) \Rightarrow \text{Parent}(x, y)$
- $\forall x, y \text{ Parent}(x, y) \Rightarrow \text{Ancestor}(x, y)$
Example

• Mother(Alice, Charlie)
• Father(Bob, Charlie)
• ∀x, y Mother(x, y) ⇒ Parent(x, y)
• ∀x, y Father(x, y) ⇒ Parent(x, y)
• ∀x, y Parent(x, y) ⇒ Ancestor(x, y)
• ∀x, y, z Ancestor(x, y) ∧ Ancestor(y, z) ⇒ Ancestor(x, z)
Example

• Mother(Alice,Charlie)
• Father(Bob,Charlie)
• ∀x,y Mother(x,y) ⇒ Parent(x,y)
• ∀x,y Father(x,y) ⇒ Parent(x,y)
• ∀x,y Parent(x,y) ⇒ Ancestor(x,y)
• ∀x,y,z Ancestor(x,y) ∧ Ancestor(y,z) ⇒ Ancestor(x,z)
• ∀x ∃y Mother(y,x)    ∀x ∃y Father(y,x)
Example

- \( \text{Mother}(\text{Alice}, \text{Charlie}) \)
- \( \text{Father}(\text{Bob}, \text{Charlie}) \)
- \( \forall x, y \text{ Mother}(x, y) \implies \text{Parent}(x, y) \)
- \( \forall x, y \text{ Father}(x, y) \implies \text{Parent}(x, y) \)
- \( \forall x, y \text{ Parent}(x, y) \implies \text{Ancestor}(x, y) \)
- \( \forall x, y, z \text{ Ancestor}(x, y) \land \text{Ancestor}(y, z) \implies \text{Ancestor}(x, z) \)
- \( \forall x \exists y \text{ Mother}(y, x) \quad \forall x \exists y \text{ Father}(y, x) \)
- \( \exists x \neg \exists y \text{ Mother}(x, y) \quad \exists x \neg \exists y \text{ Father}(x, y) \)
Example

- Mother(Alice, Charlie)
- Father(Bob, Charlie)
- $\forall x, y \text{ Mother}(x, y) \implies \text{Parent}(x, y)$
- $\forall x, y \text{ Father}(x, y) \implies \text{Parent}(x, y)$
- $\forall x, y \text{ Parent}(x, y) \implies \text{Ancestor}(x, y)$
- $\forall x, y, z \text{ Ancestor}(x, y) \land \text{Ancestor}(y, z) \implies \text{Ancestor}(x, z)$
- $\forall x \exists y \text{ Mother}(y, x)$ $\forall x \exists y \text{ Father}(y, x)$
- $\exists x \neg \exists y \text{ Mother}(x, y)$ $\exists x \neg \exists y \text{ Father}(x, y)$
- $\forall x (\text{Mother}(Alice, x) \implies \text{Father}(Bob, x)) \land (\text{Father}(Bob, x) \implies \text{Mother}(Alice, x))$
First-Order Logic

• Sentence $\rightarrow$ AtomicSentence $|$ ComplexSentence
First-Order Logic

• Sentence → AtomicSentence | ComplexSentence
• AtomicSentence → Predicate | Predicate(Arguments) | Term = Term
• Arguments → Term | Term, Arguments
• Term → Function(Arguments) | Constant | Variable
• Constant/Predicate/Function → <strings starting with an upper case letter>
• Variable → <strings comprised of lower case letters>
First-Order Logic

• Sentence → AtomicSentence | ComplexSentence
• AtomicSentence → Predicate | Predicate(Arguments) | Term = Term
• Arguments → Term | Term,Arguments
• Term → Function(Arguments) | Constant | Variable
• Constant/Predicate/Function → <strings starting with an upper case letter>
• Variable → <strings comprised of lower case letters>
First-Order Logic

- Sentence $\rightarrow$ AtomicSentence | ComplexSentence
- AtomicSentence $\rightarrow$ Predicate | Predicate(Arguments) | Term = Term
- Arguments $\rightarrow$ Term | Term,Arguments
- Term $\rightarrow$ Function(Arguments) | Constant | Variable
- Constant/Predicate/Function $\rightarrow$ <strings starting with an upper case letter>
- Variable $\rightarrow$ <strings comprised of lower case letters>
- ComplexSentence $\rightarrow$ (Sentence) | [Sentence] | $\neg$ Sentence
  | Sentence $\land$ Sentence | Sentence $\lor$ Sentence
  | Sentence $\Rightarrow$ Sentence | Quantifiers Sentence
- Quantifiers $\rightarrow$ Quantifier Variables | Quantifier Variables Quantifiers
- Quantifier $\rightarrow$ $\forall$ | $\exists$
- Variables $\rightarrow$ Variable | Variable,Variables
First-Order Logic

- Sentence → AtomicSentence | ComplexSentence
- AtomicSentence → Predicate | Predicate(Arguments) | Term = Term
- Arguments → Term | Term,Arguments
- Term → Function(Arguments) | Constant | Variable
- Constant/Predicate/Function → <strings starting with an upper case letter>
- Variable → <strings comprised of lower case letters>
- ComplexSentence → (Sentence) | [Sentence] | ¬ Sentence
  | Sentence ∧ Sentence | Sentence ∨ Sentence
  | Sentence ⇒ Sentence | Quantifiers Sentence
- Quantifiers → Quantifier Variables | Quantifier Variables Quantifiers
- Quantifier → ∀ | ∃
- Variables → Variable | Variable,Variables