IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE DA

IAE Academy free study material · Updated 5 October 2026

Artificial Intelligence Formula Sheet

INSIDE

- 46 formulas in 3 topics
- When to use each one
- Taken from the free topic notes on iaeacademy.in

Online, one page per topic: **iaeacademy.in/study/artificial-intelligence/formula-sheet**

FREE COUNSELLING

Preparing for GATE DA? Talk to IAE Academy.

Or call **+91 84204 36524**

Every day, 10:30 am to 7:30 pm

Chat on WhatsApp

Free. Not for sale. Pass it on.

Search: uninformed, informed and adversarial

NODES GENERATED BY ITERATIVE DEEPENING (WORST CASE)

$$N_{\text{IDS}} = \sum_{i=0}^{d} (d+1-i)\, b^i = (d+1) + d\,b + (d-1)\,b^2 + \cdots + 1 \cdot b^d$$

Uniform tree with branching factor b, goal at depth d, the last iteration generating every node to depth d, root counted in every iteration. Nodes at depth i are generated $d+1-i$ times.

BOUND ON THE ITERATIVE-DEEPENING OVERHEAD

$$N_{\text{IDS}} < \left(\frac{b}{b-1}\right)^2 b^d \qquad (b \geq 2)$$

From the sum above with $k = d - i$: $N_{\text{IDS}} = b^d \sum_{k=0}^{d} (k+1) b^{-k} < b^d\,(1 - 1/b)^{-2}$. For $b = 2$ the factor is 4; for $b = 10$ it is about 1.23.

A* EVALUATION FUNCTION

$$f(n) = g(n) + h(n)$$

A* always expands the frontier node with the smallest f. Greedy best-first search uses $f(n) = h(n)$ alone; uniform-cost search is A* with $h \equiv 0$.

ADMISSIBLE HEURISTIC

$$0 \leq h(n) \leq h^*(n) \quad \text{for every node } n$$

Never overestimates the true remaining cost. Enough for A* to be optimal in tree search, and in graph search when cheaper paths re-open expanded nodes.

CONSISTENT (MONOTONE) HEURISTIC

$$h(n) \leq c(n, a, n') + h(n') \quad \text{for every node } n \text{ and successor } n',\ h(\text{goal}) = 0$$

A triangle inequality on estimates. Every consistent heuristic is admissible; the converse is false. Along any path f never decreases when h is consistent.

MINIMAX VALUE

$$\text{MINIMAX}(s) = \begin{cases} \text{UTILITY}(s) & \text{if } s \text{ is terminal} \\ \max_a \text{MINIMAX}(\text{RESULT}(s,a)) & \text{if MAX moves at } s \\ \min_a \text{MINIMAX}(\text{RESULT}(s,a)) & \text{if MIN moves at } s \end{cases}$$

Two-player zero-sum game, perfect information, finite tree. The value is computed bottom-up.

MINIMAX COMPLEXITY

$$\text{time } O(b^m), \qquad \text{space } O(bm) \text{ (depth-first, all actions at once)}$$

m is the maximum depth of the game tree. Minimax is complete on a finite tree and optimal against an optimal opponent.

ALPHA-BETA CUT-OFF RULE

$$\text{prune the remaining children of a node as soon as } \alpha \geq \beta$$

Updates: at a MAX node $\alpha \leftarrow \max(\alpha, v)$; at a MIN node $\beta \leftarrow \min(\beta, v)$, where v is the best child value so far. Pruning when $\alpha = \beta$ is safe because it cannot change the root value. Some sources prune only when $\alpha > \beta$; that can change the number of pruned leaves but never the root value. These notes use $\alpha \geq \beta$.

ALPHA-BETA TIME COMPLEXITY

$$O\left(b^{m/2}\right) \text{ with best move ordering}, \qquad O\left(b^m\right) \text{ with worst ordering}$$

With perfect ordering alpha-beta can look about twice as deep as minimax in the same time; with random ordering the cost is roughly $b^{3m/4}$. The root value is the same for every ordering; only the number of pruned leaves changes.

LEAVES EVALUATED WITH PERFECT ORDERING

$$b^{\lceil m/2 \rceil} + b^{\lfloor m/2 \rfloor} - 1$$

Uniform game tree with branching factor b, all leaves at depth m, distinct leaf values, children ordered best-first for the player to move. Example: $b = 3$, $m = 2$ gives $3 + 3 - 1 = 5$ leaves (all three under the first MIN child, then one under each other child).

Propositional and predicate logic

IMPLICATION ELIMINATION

$$p \to q \;\equiv\; \neg p \lor q$$

The usual first step (after rewriting $\leftrightarrow$) when converting to CNF or DNF.

BICONDITIONAL

$$p \leftrightarrow q \;\equiv\; (p \to q) \land (q \to p) \;\equiv\; (p \land q) \lor (\neg p \land \neg q)$$

True exactly when p and q have the same truth value.

CONTRAPOSITIVE

$$p \to q \;\equiv\; \neg q \to \neg p$$

The converse $q \to p$ and the inverse $\neg p \to \neg q$ are NOT equivalent to $p \to q$ (they are equivalent to each other).

DE MORGAN'S LAWS

$$\neg(p \land q) \equiv \neg p \lor \neg q, \qquad \neg(p \lor q) \equiv \neg p \land \neg q$$

Double negation: $\neg\neg p \equiv p$.

NEGATING AN IMPLICATION AND A BICONDITIONAL

$$\neg(p \to q) \equiv p \land \neg q, \qquad \neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q \equiv (p \land \neg q) \lor (\neg p \land q)$$

$\neg(p \leftrightarrow q)$ is the exclusive or $p \oplus q$: true exactly when p and q differ. The negation of an implication is a conjunction, never another implication.

DISTRIBUTIVE LAWS

$$p \land (q \lor r) \equiv (p \land q) \lor (p \land r), \qquad p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$$

Each of $\land$ and $\lor$ distributes over the other; used to reach CNF and DNF.

EXPORTATION

$$(p \land q) \to r \;\equiv\; p \to (q \to r)$$

Moves a conjunction in the premise into a chain of implications.

IMPLICATIONS WITH A COMPOUND PART

$$(p \lor q) \to r \equiv (p \to r) \land (q \to r)$$
$$(p \land q) \to r \equiv (p \to r) \lor (q \to r)$$
$$p \to (q \land r) \equiv (p \to q) \land (p \to r)$$
$$p \to (q \lor r) \equiv (p \to q) \lor (p \to r)$$

Check any of these by rewriting both sides with $p \to q \equiv \neg p \lor q$. Note the second line: a conjunctive premise splits into an OR, not an AND.

ENTAILMENT AS VALIDITY

$$\varphi \models \psi \iff (\varphi \to \psi) \text{ is a tautology}$$

$\varphi \models \psi$ means every assignment making φ true also makes ψ true. With several premises: $\varphi_1, \ldots, \varphi_k \models \psi$ exactly when $(\varphi_1 \land \cdots \land \varphi_k) \to \psi$ is a tautology.

REFUTATION TEST FOR ENTAILMENT

$$\mathrm{KB} \models \alpha \iff \mathrm{KB} \wedge \neg\alpha \text{ is unsatisfiable}$$

The basis of proof by contradiction and of resolution refutation.

NORMAL FORMS FROM A TRUTH TABLE

$$\text{DNF: or of the minterms of the rows that are T;} \qquad \text{CNF: and of the maxterms of the rows that are F}$$

A minterm is true exactly on its row; a maxterm is false exactly on its row. In the maxterm, a variable that is T in the row appears negated and one that is F appears plain (for the row $p = \mathrm{T}, q = \mathrm{F}$ the maxterm is $\neg p \vee q$). A formula with no T rows has an empty DNF (a contradiction); one with no F rows has an empty CNF (a tautology).

RESOLUTION RULE

$$\frac{(A \vee \ell) \quad (B \vee \neg\ell)}{A \vee B}$$

Applies to two clauses holding a complementary pair of literals. The resolvent is entailed by its two parents (soundness). Resolution refutation is also complete for propositional logic: if a clause set is unsatisfiable, the empty clause can be derived from it.

NEGATING QUANTIFIERS

$$\neg\forall x\, P(x) \equiv \exists x\, \neg P(x), \qquad \neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$$

Negation flips the quantifier and moves inside; apply De Morgan's laws to what remains.

DISTRIBUTION OF QUANTIFIERS

$$\forall x\, (P \wedge Q) \equiv \forall x\, P \wedge \forall x\, Q, \qquad \exists x\, (P \vee Q) \equiv \exists x\, P \vee \exists x\, Q$$

When both P and Q contain x free, only these two distribute as equivalences.

MOVING A QUANTIFIER PAST Q (X NOT FREE IN Q)

$$\forall x\, (P(x) \vee Q) \equiv \forall x\, P(x) \vee Q$$
$$\exists x\, (P(x) \wedge Q) \equiv \exists x\, P(x) \wedge Q$$
$$\forall x\, (P(x) \wedge Q) \equiv \forall x\, P(x) \wedge Q$$
$$\exists x\, (P(x) \vee Q) \equiv \exists x\, P(x) \vee Q$$

Valid when x does not occur free in Q and the domain is non-empty.

MOVING A QUANTIFIER OUT OF AN IMPLICATION (X NOT FREE IN Q)

$$\forall x\, (Q \to P(x)) \equiv Q \to \forall x\, P(x)$$
$$\exists x\, (Q \to P(x)) \equiv Q \to \exists x\, P(x)$$
$$\forall x\, (P(x) \to Q) \equiv \exists x\, P(x) \to Q$$
$$\exists x\, (P(x) \to Q) \equiv \forall x\, P(x) \to Q$$

Valid when x does not occur free in Q and the domain is non-empty. A quantifier moved out of the conclusion stays the same; one moved out of the premise FLIPS ($\forall$ becomes $\exists$ and vice versa), because the premise sits under a hidden negation: $P(x) \to Q \equiv \neg P(x) \vee Q$.

ONE-WAY IMPLICATIONS

$$\forall x\, P \vee \forall x\, Q \models \forall x\, (P \vee Q), \qquad \exists x\, (P \wedge Q) \models \exists x\, P \wedge \exists x\, Q$$

The converse of each is NOT valid.

MORE ONE-WAY ENTAILMENTS

$$\forall x\, P(x) \models \exists x\, P(x), \qquad \forall x\, (P \to Q) \models \forall x\, P \to \forall x\, Q, \qquad \forall x\, (P \to Q) \models \exists x\, P \to \exists x\, Q$$

The first needs the non-empty domain. None of the converses is valid: for example, with domain $\{1, 2\}$, $P(1)$ true, $P(2)$ false and $Q(1)$ false, $\forall x\, P \to \forall x\, Q$ is true but $\forall x\, (P \to Q)$ is false.

QUANTIFIER ORDER

$$\exists y\,\forall x\,R(x,y) \models \forall x\,\exists y\,R(x,y), \qquad \text{but not conversely}$$

Like quantifiers commute: $\forall x \forall y \equiv \forall y \forall x$ and $\exists x \exists y \equiv \exists y \exists x$.

FIRST-ORDER RESOLUTION RULE

$$\frac{(A \vee \ell_1) \quad (B \vee \neg \ell_2) \quad \theta = \mathrm{MGU}(\ell_1, \ell_2)}{(A \vee B)\theta}$$

Clauses must share no variable names (standardise apart); the empty clause again signals unsatisfiability of the clause set. Resolution together with factoring (unifying two literals inside one clause) is refutation-complete for first-order logic: an unsatisfiable clause set always yields the empty clause. If the set is satisfiable the search may never stop, since first-order entailment is only semi-decidable.

Reasoning under uncertainty: conditional independence, variable elimination and sampling

PRODUCT RULE

$$P(A, B) = P(A \mid B)\,P(B) = P(B \mid A)\,P(A)$$

Each conditional is defined only when the event it conditions on has positive probability ($P(B) > 0$ for $P(A \mid B)$).

MARGINALISATION (SUM RULE)

$$P(X) = \sum_{y} P(X, y)$$

Sum over every value of each variable being removed.

BAYES' RULE

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}, \qquad P(B) = \sum_{a} P(B \mid a)P(a)$$

Requires $P(B) > 0$. Turns a likelihood $P(B \mid A)$ into a posterior $P(A \mid B)$; the denominator comes from marginalisation.

CONDITIONING BY NORMALISATION

$$P(X \mid e) = \alpha\,P(X, e), \qquad \alpha = \frac{1}{\sum_x P(x, e)}$$

Requires $P(e) > 0$. Compute the unnormalised values $P(x, e)$ for each x, then divide by their sum. α is just $1/P(e)$.

CHAIN RULE OF PROBABILITY

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1})$$

Holds for any distribution and any variable ordering (wherever the conditionals are defined); it is the starting point for the Bayesian-network factorisation.

INDEPENDENCE

$$A \perp\!\!\!\perp B \iff P(A, B) = P(A)\,P(B) \iff P(A \mid B) = P(A)$$

Must hold for every pair of values of A and B; the last form is for values with $P(B) > 0$.

CONDITIONAL INDEPENDENCE

$$A \perp\!\!\!\perp B \mid C \iff P(A, B \mid C) = P(A \mid C)\,P(B \mid C) \iff P(A \mid B, C) = P(A \mid C)$$

Must hold for every value of A, B and C with $P(C) > 0$ (and $P(B, C) > 0$ for the last form).

FACTORISATION OF THE JOINT IN A BAYESIAN NETWORK

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\big(X_i \mid \mathrm{Pa}(X_i)\big)$$

For any choice of CPTs this product is a valid joint distribution, and a joint factorises this way exactly when it satisfies the local Markov property for the DAG. Any joint entry is the product of one CPT entry per node.

NUMBER OF INDEPENDENT CPT PARAMETERS

$$\text{free parameters} = \sum_{i=1}^{n} \big(|\mathrm{dom}(X_i)| - 1\big) \prod_{X_j \in \mathrm{Pa}(X_i)} |\mathrm{dom}(X_j)|$$

For all-binary variables this is $\sum_i 2^{|\mathrm{Pa}(X_i)|}$, compared with $2^n - 1$ for the full joint.

MARKOV BLANKET

$$\mathrm{MB}(X) = \big(\mathrm{Pa}(X) \cup \mathrm{Children}(X) \cup \mathrm{Pa}(\mathrm{Children}(X))\big) \setminus \{X\}$$

Parents, children and the children's other parents (co-parents). Given its Markov blanket, X is conditionally independent of every other variable in the network.

INFERENCE BY ENUMERATION

$$P(Q \mid e) = \alpha \sum_{h} P(Q, e, h) = \alpha \sum_{h} \prod_{i} P\big(X_i \mid \mathrm{Pa}(X_i)\big)\Big|_{e}$$

Sum the network's joint over all hidden-variable assignments with evidence fixed, then normalise ($P(e) > 0$). Cost is exponential in the number of hidden variables.

ELIMINATING ONE VARIABLE

$$f'(\mathbf{y}) = \sum_{x} \prod_{k} f_k(x, \mathbf{y}_k)$$

Multiply only the factors that mention X, then sum X out; factors not mentioning X are left untouched and can be pulled outside the sum.

PRIOR (FORWARD) SAMPLING

$$\hat{P}(x_1, \ldots, x_n) = \frac{N(x_1, \ldots, x_n)}{N}$$

Each sample is drawn from the full joint, ignoring evidence; $N(\cdot)$ counts matching samples out of N.

REJECTION SAMPLING

$$\hat{P}(Q = q \mid e) = \frac{N(q, e)}{N(e)}$$

Generate prior samples, discard any that disagree with the evidence, and use the counts among the kept ones (undefined if $N(e) = 0$). The fraction kept is about $P(e)$, so rare evidence wastes almost all samples.

LIKELIHOOD WEIGHTING

$$w = \prod_{j} P\big(e_j \mid \mathrm{Pa}(E_j)\big), \qquad \hat{P}(Q = q \mid e) = \frac{\sum_{s:\, Q^{(s)} = q} w^{(s)}}{\sum_{s} w^{(s)}}$$

Evidence variables are fixed, not sampled; each sample is weighted by the probability of the evidence given its parents' values in that same sample (a parent may itself be an evidence variable, fixed at its observed value). Non-evidence variables are sampled as usual, so no sample is rejected. The method is consistent because the probability of drawing the non-evidence values z times the weight equals $P(z, e)$.

GIBBS SAMPLING: RESAMPLING ONE VARIABLE

$$P\big(x_i \mid \mathrm{mb}(X_i)\big) = \alpha\, P\big(x_i \mid \mathrm{pa}(X_i)\big) \prod_{Y_j \in \mathrm{Children}(X_i)} P\big(y_j \mid \mathrm{pa}(Y_j)\big)$$

All other variables are held at their current values; α normalises over the values x_i. Only the CPTs of X_i and of its children are needed, because X_i is independent of everything else given its Markov blanket.

FROM IAE ACADEMY

Preparing for GATE DA?

IAE Academy teaches GATE DA in Ballygunge, Kolkata, and online. Counselling is free: ask us about batches, fees or your preparation.

FREE COUNSELLING

Chat on WhatsApp

Call +91 84204 36524

Open every day, 10:30 am to 7:30 pm.

Scan to chat on WhatsApp

THE GATE DA COURSE

Classroom batches in Ballygunge and online batches. Fees and details are on the course page.

Scan: course page

See the course

FREE ON IAEACADEMY.IN

Notes and solved previous-year questions for Artificial Intelligence, topic by topic.

Scan: the notes

Open the notes

Found this useful? Send it to a friend preparing for GATE.

Send to a friend on WhatsApp

IAE Academy free study material. The formulas come from the free Artificial Intelligence notes on iaeacademy.in. Prepared and published by IAE Academy, Kolkata. Free to read and free to share, unaltered and with attribution.

IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE coaching in Kolkata since 2012 · EC, EE, IN, CS and IT, DA, BT and CE · in the classroom and online

227A Rash Behari Ave, Ballygunge, Kolkata 700019

+91 84204 36524 · iaeacademy.in · Updated 5 October 2026

Get directions