INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

# CSE643 – Artificial Intelligence

## Monsoon 2022 session

## Quiz-2

**Max marks: 10 (will be scaled down to 5 marks)** **07-Oct-22**

**Time: 10:20AM to 10:50 AM**

**INSTRUCTIONS:**

**Ensure that in the answer sheet you write your name and roll number clearly. No laptops and no mobiles. Closed book quiz. Submit your hard-copy to the TAs.**

**Q1:** Express the following in Conjunctive Normal Form and then use Resolution refutation to prove whether the hypothesis is true or false. Also draw the refutation graph.
(8 marks)

Every tiger can outrun every hyena. Some hyenas can outrun every zebra. The relationship of outrun is transitive. Machhli is the name of a tiger. Raja is the name of a zebra.

Hypothesis: Machhli can outrun Raja.

**Answers for Q1:**

1. ∀(x) ∀(y) (tiger(x) ^ hyena(y)) → outrun(x,y).
2. ∃(p) hyena(p) ^ (∀(q) zebra(q) → outrun(p,q)).
3. ∀(r) ∀(s) ∀(t) outrun(r,s) ^ outrun(s,t) → outrun(r,t).
4. tiger(Machhli).
5. zebra(Raja).

**In CNF:**

6. ¬tiger(x) v ¬hyena(y) v outrun(x,y).

(2) can be rewritten as hyena(C)) ^ (¬zebra(q) v outrun(C, q)). – skolem constant C for existentially quantified variable and it gives rise to two clauses below

7. hyena(C).

8. ¬zebra(q) v outrun(C, q).
9. ¬outrun(r,s) v ¬outrun(s,t) v outrun(r,t).

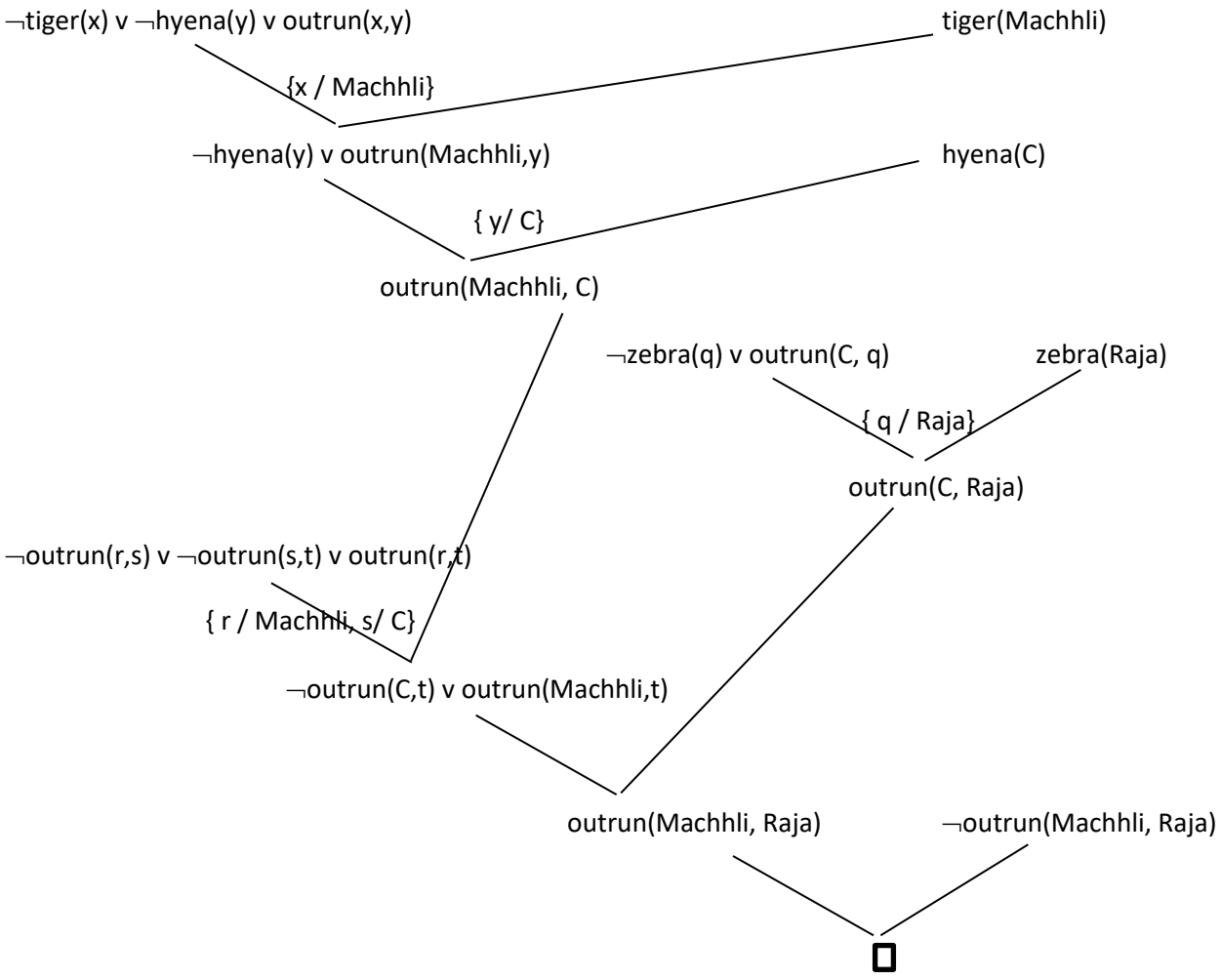
Hypothesis: outrun(Machhli, Raja). Assume its negation and add it to the set of clauses.

10. ¬outrun(Machhli, Raja).

**Resolution proof**

(a) From 4 and 6 we derive ¬hyena(y) v outrun(Machhli,y).
(b) From 7 and (a) we get outrun(Machhli, C).
(c) From 5 and 8 we derive outrun(C, Raja).
(d) From (b) and 9 we get ¬outrun(C,t) v outrun(Machhli,t).
(e) From (c) and (d) we get outrun(Machhli, Raja).
(f) From 10 and (e) we get a contradiction. Thus our negation of Hypothesis is FALSE, and hence outrun(Machhli, Raja) is TRUE.

**Resolution graph**



**Q2:** (2 marks)

i) What is ‘best’ in Best-first search?

ii) What is the difference between Best-first search and A* search?

**Answers for Q2**

i) Best-first search is a method that evaluates the estimated cost of reaching the goal node from the current child nodes and selects that child node that has the least cost from the heuristic estimates. Thus Best-first search focuses on expanding that node that is estimated closest to the goal. It evaluates nodes by using just the heuristic cost function; that is, f(n) = h(n). Best-first search is useful when we have many child nodes and need to find a quick path to the goal node without evaluating the cost of reaching a node. Thus it is selecting the ‘best’ among the children nodes to reach the goal.

ii) The difference between A* search and Best-first search is that A* considers the cost of reaching a child node in addition to the estimated cost of reaching the goal node from that child node, whereas Best-first does not consider the cost of reaching a particular node but only focuses on selecting that node that is estimated to reach the goal faster.