

1 Predicate Logic

50 points

1.1 First Order Logic

14 points

Translate the following sentences into first order logic:

- a. There is an ice cream flavor loved by everyone.

$$\exists flavor \forall x : Loves(x, flavor)$$

- b. It is not true that everyone loves some ice cream flavor.

$$\exists x \forall flavor : \neg Loves(x, icecream)$$

- c. Anyone who does not shave himself must be shaved by the barber.

Let *barber* be a constant.
$$\forall x : \neg(Shaves(x, x)) \rightarrow Shaves(barber, x)$$

- d. Whomever the barber shaves, must not shave himself.

Let *barber* be a constant.
$$\forall x : Shaves(barber, x) \rightarrow \neg(Shaves(x, x))$$

- e. Rudolph is a reindeer and Rudolph has a red nose.

Let *rudolph* be a constant.
$$Reindeer(rudolph) \wedge RedNose(rudolph)$$

- f. Anyone with a red nose is weird or is a clown.

$$\forall x : RedNose(x) \rightarrow (Weird(x) \vee Clown(x))$$

- g. No reindeer is a clown.

$$\forall x : Reindeer(x) \rightarrow \neg Clown(x)$$

Note: You are not required to specify which object a variable refers to, or what a function returns, if it is obvious from the context. For instance, it's obvious from the context that x is a person and $Likes(x, y)$ is true if x likes some y . However, if you think that the context is insufficient, you might start providing definitions.

1.2 Clausal Forms

6 points

Reduce the following sentences to clausal form:

a. $\forall x \forall y (R(x, y) \rightarrow (R(x, y) \wedge Q(y)))$

$$\begin{aligned} & \forall x \forall y (R(x, y) \rightarrow (R(x, y) \wedge Q(y))) \\ & \equiv \forall x \forall y (\neg R(x, y) \vee (R(x, y) \wedge Q(y))) \\ & \equiv \forall x \forall y ((\neg R(x, y) \vee R(x, y)) \wedge (\neg R(x, y) \vee Q(y))) \\ & \equiv \forall x \forall y (1 \wedge (\neg R(x, y) \vee Q(y))) \\ & \equiv \forall x \forall y ((\neg R(x, y) \vee Q(y))) \\ & \equiv \boxed{\neg R(x, y) \vee Q(y)} \end{aligned}$$

b. $\forall x \exists y \forall z (P(x, y, z) \rightarrow \exists u R(x, u, z))$

$$\begin{aligned} & \forall x \exists y \forall z (P(x, y, z) \rightarrow \exists u R(x, u, z)) \\ & \equiv \forall x \exists y \forall z (\neg P(x, y, z) \vee \exists u R(x, u, z)) \\ & \equiv \forall x \forall z (\neg P(x, F(x), z) \vee R(x, G(x, z), z)) \\ & \equiv \boxed{\neg P(x, F(x), z) \vee R(x, G(x, z), z)} \end{aligned}$$

c. $\forall x (\neg \exists y P(x, y) \wedge \neg (Q(x) \wedge \neg R(x)))$

$$\begin{aligned} & \forall x (\neg \exists y P(x, y) \wedge \neg (Q(x) \wedge \neg R(x))) \\ & \equiv \forall x (\neg \exists y P(x, y) \wedge (\neg Q(x) \vee R(x))) \\ & \equiv \forall x (\forall y \neg P(x, y) \wedge (\neg Q(x) \vee R(x))) \\ & \equiv \boxed{\neg P(x, y) \wedge (\neg Q(x) \vee R(x))} \end{aligned}$$

1.3 Knowledge Extraction via Resolution

30 points

Let *flee*, *stay*, *now*, *lion* be constants. You are given the following Knowledge Base:

S1 $\forall t \forall x (Observe(t, x) \wedge Danger(x) \rightarrow Suggestion(t, flee))$

S2 $\forall t (\neg \exists x (Observe(t, x) \wedge Danger(x)) \rightarrow Suggestion(t, stay))$

S3 $Danger(lion)$

S4 $Observe(now, lion)$

Use Resolution to show that the KB entails $\exists x Suggestion(now, x)$ and extract an answer.

S1:

$$\begin{aligned} & \forall t \forall x (Observe(t, x) \wedge Danger(x) \rightarrow Suggestion(t, flee)) \\ & \equiv \neg Observe(t, x) \vee \neg Danger(x) \vee Suggestion(t, flee) \end{aligned}$$

S2:

$$\begin{aligned} & \forall t (\neg \exists x (Observe(t, x) \wedge Danger(x)) \rightarrow Suggestion(t, stay)) \\ & \equiv (Observe(t, f(t)) \wedge Danger(f(t))) \vee Suggestion(t, stay) \end{aligned}$$

S3:

$$Danger(lion)$$

S3:

$$Observe(now, lion)$$

Negated Query:

$$\begin{aligned} & \neg \exists x Suggestion(now, x) \\ & \equiv \forall x \neg Suggestion(now, x) \\ & \equiv \neg Suggestion(now, x) \end{aligned}$$

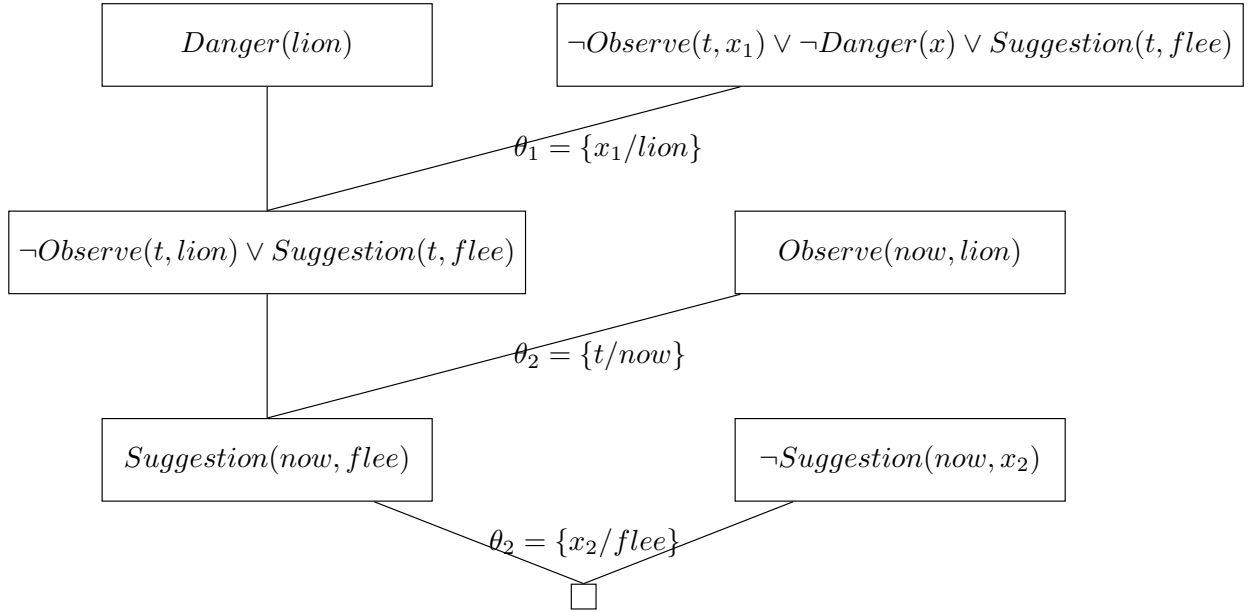


Figure 1: FOL-Resolution

The empty set at the end proves, that the $KB \wedge \neg Query$ is unsatisfiable, but that implies that the Query is satisfiable. The extracted answer is $x = flee$. It was not necessary to use the entire knowledge base, since a subset already proved the unsatisfiability.

2 Predicate Logic

50 points

2.1 Probability

30 points

Examiners running the game show “*Ring the golden bell*” have been taking bribes from some participants. A given participant is either allowed to stay or kicked off in each episode.

If the participant has been bribing the examiners she will be allowed to stay with probability $4/5$. If the participant has not been bribing the examiners, she will be allowed to stay with probability $1/3$.

Suppose that $1/4$ of the participants have been bribing the examiners. The same participants bribe the examiners in both rounds, i.e., if a participant bribes them in the first round, she bribes them in the second round too (and vice versa).

- a. If you pick a random participant who was allowed to stay during the first episode, what is the probability that she was bribing the examiners?

B = is bribing, $\neg B$ = is not bribing, S = is staying, $\neg S$ = is not staying.

We are looking for and want to compute $P(B|S)$.

Given:

- $P(B) = \frac{1}{4}$
- $P(S | B) = \frac{4}{5}$
- $P(S | \neg B) = \frac{1}{3}$

With the given information we can create the tree in Figure 2.

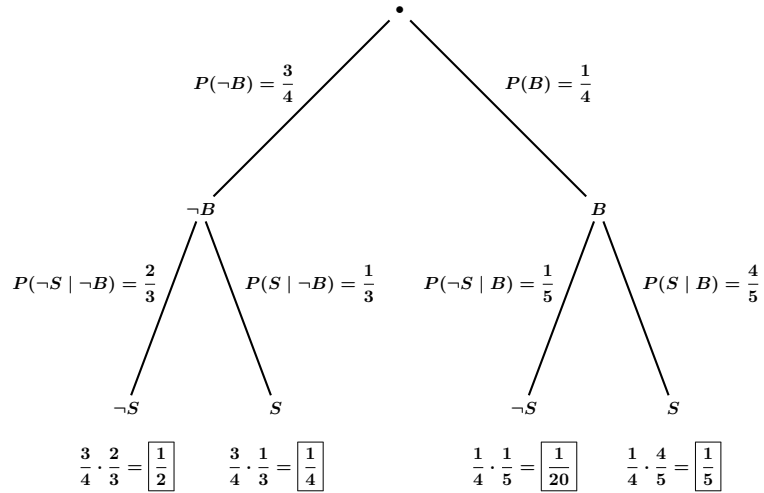


Figure 2: Probability tree

Before computing $P(B|S)$ we need to compute $P(S)$ first:

$$P(S) = P(\neg B) \cdot P(S | \neg B) + P(B) \cdot P(S | B) = \frac{1}{4} \cdot \frac{4}{5} + \frac{3}{4} \cdot \frac{1}{3} = \frac{1}{5} + \frac{1}{4} = \frac{4+5}{5 \cdot 4} = \frac{9}{20}$$

we computing $P(B | S)$ using Bayes' theorem:

$$P(B | S) = \frac{P(S|B) \cdot P(B)}{P(S)} = \frac{\frac{4}{5} \cdot \frac{1}{4}}{\frac{9}{20}} = \frac{4}{5} \cdot \frac{1}{4} \cdot \frac{20}{9} = \frac{4}{9}$$

- b. If you pick a random participant, what is the probability that she is allowed to stay during both of the first two episodes?

$$\begin{aligned}P(\text{stay 2 times}) &= P(\neg B) \cdot P(S \mid \neg B)^2 + P(B) \cdot P(S \mid B)^2 \\&= \frac{3}{4} \cdot \left(\frac{1}{3}\right)^2 + \frac{1}{4} \cdot \left(\frac{4}{5}\right)^2 \\&= \frac{3}{4} \cdot \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{4} \cdot \frac{4}{5} \cdot \frac{4}{5} \\&= \frac{1}{4 \cdot 3} + \frac{4}{5^2} \\&= \frac{1}{12} + \frac{4}{25} \\&= \frac{25 + 4 \cdot 12}{12 \cdot 25} \\&= \frac{73}{300}\end{aligned}$$

- c. If you pick random participant who was allowed to stay during the first episode, what is the probability that she gets kicked off during the second episode?

$$\begin{aligned}P(\text{kicked off second time}) &= P(\neg B) \cdot P(S \mid \neg B) \cdot P(\neg S \mid \neg B) \\&\quad + P(B) \cdot P(S \mid B) \cdot P(\neg S \mid B) \\&= \frac{3}{4} \cdot \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{4} \cdot \frac{4}{5} \cdot \frac{1}{5} \\&= \frac{1}{2 \cdot 3} + \frac{1}{5^2} \\&= \frac{1}{6} + \frac{1}{25} \\&= \frac{25 + 6}{6 \cdot 25} \\&= \frac{31}{150}\end{aligned}$$

2.2 Naïve Bayes' Classifier

20 points

Given this data table about buying a car. Predict the class of the following new example using Naïve Bayes' Classification:

age ≤ 30 , income = high, Married = yes, credit-rating = fair

RID	age	income	married	credit_rating	class: buy_car
1	≤ 30	high	no	fair	no
2	≤ 30	high	no	excellent	no
3	31, 32, ..., 40	high	no	fair	yes
4	> 40	medium	no	fair	yes
5	> 40	low	yes	fair	yes
6	> 40	low	yes	excellent	no
7	31, 32, ..., 40	low	yes	excellent	yes
8	≤ 30	medium	no	fair	no
9	≤ 30	low	yes	fair	yes
10	> 40	medium	yes	fair	yes
11	≤ 30	medium	yes	excellent	yes
12	31, 32, ..., 40	medium	yes	excellent	yes
13	31, 32, ..., 40	high	yes	fair	yes
14	> 40	medium	no	excellent	no

We want to predict the class of the given example:

$$e = (\underbrace{age \leq 30}_{e_0}, \underbrace{income = high}_{e_1}, \underbrace{married = yes}_{e_2}, \underbrace{credit = fair}_{e_3})$$

by classifying it into one of the two categories:

$$buy_car \in \{yes, no\}.$$

1. Prior over categories ($P(category)$):

$$P(buy_car = yes) = \frac{9}{14}, \quad P(buy_car = no) = \frac{5}{14}.$$

2. Conditional probabilities ($P(e_i | category)$):

- $buy_car = yes$:

$$P(\underbrace{age \leq 30}_{e_0} | buy_car = yes) = \frac{2}{9}, \quad P(\underbrace{income = high}_{e_1} | buy_car = yes) = \frac{2}{9},$$

$$P(\underbrace{married = yes}_{e_2} | buy_car = yes) = \frac{7}{9}, \quad P(\underbrace{credit = fair}_{e_3} | buy_car = yes) = \frac{6}{9}.$$

- $buy_car = no$:

$$\begin{aligned} P(\underbrace{age \leq 30}_{e_0} | buy_car = no) &= \frac{3}{5}, & P(\underbrace{income = high}_{e_1} | buy_car = no) &= \frac{2}{5}, \\ P(\underbrace{married = yes}_{e_2} | buy_car = no) &= \frac{1}{5}, & P(\underbrace{credit = fair}_{e_3} | buy_car = no) &= \frac{2}{5}. \end{aligned}$$

Using the Naïve Bayes formula

$$P(Category | e) = \alpha P(Category) \prod_j P(e_j | Category)$$

we can compute $P(buy_car = yes | e)$:

$$P(e | buy_car = yes) = \prod_j P(e_j | buy_car = yes) = \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{7}{9} \cdot \frac{6}{9} = \frac{56}{2187} \approx 0.0256,$$

$$P(buy_car = yes | e) = P(buy_car = yes) \cdot P(e | buy_car = yes) = \frac{9}{14} \cdot \frac{56}{2187} \approx 0.0165.$$

and $P(buy_car = no | e)$:

$$P(e | buy_car = no) = \prod_j P(e_j | buy_car = no) = \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} = \frac{12}{625} \approx 0.0192,$$

$$P(buy_car = no | e) = P(buy_car = no) \cdot P(e | buy_car = no) = \frac{5}{14} \cdot \frac{12}{625} \approx 0.00686.$$

Now we can predict the class of the given example:

$$P(buy_car = yes | e) = 0.0165 > 0.00686 = P(buy_car = no | e).$$

Predicted Category: $buy_car = yes$

The Naïve Bayes classifier predicts that the customer will buy a car.