Assignment 5

Introduction to Artificial Intelligence

Pablo Díaz Viñambres : *pablodi@ntnu.no*

November 4, 2022

1 Models and entailment in propositional logic

1.1 Modelling

For each of the following statements, determine whether it is true or not by building the complete model (truth table).

(a) $\neg A \wedge \neg B \models \neg B$

A	B	$\neg A$	$\neg B$	$\neg A \wedge \neg B$	$\neg A \wedge \neg B \implies \neg B$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Since $A \wedge \neg B \implies \neg B$ is true in every model, $A \wedge \neg B \models \neg B$.

(b) $\neg A \vee \neg B \models \neg B$

A	B	$\neg A$	$\neg B$	$\neg A \vee \neg B$	$\neg A \vee \neg B \implies \neg B$
0	0	1	1	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	1	0	0	0	1

Since $\neg A \vee \neg B \implies \neg B$ doesn't hold for $A = 0, B = 1$, the statement is false.

(c) $\neg A \wedge B \models A \vee B$

A	B	$\neg A$	$\neg A \wedge B$	$A \vee B$	$\neg A \wedge B \implies A \vee B$
0	0	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Since $\neg A \wedge B \implies A \vee B$ is true in every model, $\neg A \wedge B \models A \vee B$.

(d) $A \implies B \models A \iff B$

A	B	$A \implies B$	$A \iff B$	$A \implies B \implies A \iff B$
0	0	1	1	1
0	1	1	0	0
1	0	0	0	1
1	1	1	1	1

Since $A \implies B \implies A \iff B$ doesn't hold for $A = 0, B = 1$, the statement is false.

(e) $(A \implies B) \iff C \models A \vee \neg B \vee C$

A	B	C	$\neg B$	$A \implies B$	$(A \implies B) \iff C$	$A \vee \neg B \vee C$	$(A \implies B) \iff C \models A \vee \neg B \vee C$
0	0	0	1	1	0	1	1
0	0	1	1	1	1	1	1
0	1	0	0	1	0	1	1
0	1	1	0	1	1	0	0
1	0	0	1	0	1	1	1
1	0	1	1	0	0	1	1
1	1	0	0	1	0	1	1
1	1	1	0	1	1	1	1

Since $(A \implies B) \iff C \models A \vee \neg B \vee C$ doesn't hold for $A = 0, B = 1, C = 1$, the statement is false.

(f) $(\neg A \implies \neg B) \land (A \land \neg B)$ is satisfiable

A	B	$\neg A$	$\neg B$	$\neg \implies \neg B$	$A \land \neg B$	$(\neg A \implies \neg B) \land (A \land \neg B)$
0	0	1	1	1	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	1	1	0	1	0	0

Since $(\neg A \implies \neg B) \land (A \land \neg B)$ is true for $A = 1, B = 0$, the statement is satisfiable.

(g) $(\neg A \iff \neg B) \land (A \land \neg B)$ is satisfiable

A	B	$\neg A$	$\neg B$	$\neg \implies \neg B$	$A \land \neg B$	$(\neg A \implies \neg B) \land (A \land \neg B)$
0	0	1	1	1	0	0
0	1	1	0	0	0	0
1	0	0	1	0	1	0
1	1	1	0	1	0	0

Since $(\neg A \implies \neg B) \land (A \land \neg B)$ is false for every model, the statement is unsatisfiable.

1.2 Trouble in the lab

(a) Generate the vocabulary of the system.

Given a tank $i, 0 \leq i \leq 31$ we denote the three metrics for the occupation, toxicity and electrical charge as S_1^i, S_2^i and S_3^i respectively.

We also define C^i as the closed state of the tank and $O^i \implies \neg C^i$ as its open state.

(b) Express all closing conditions (C_1, C_2, C_3) for the gates as propositional logic statements.

(C_1) If species inside are safe (low toxicity and low electric charge).
For a tank i, this can be written as $S_1^i \land \neg S_2^i \land \neg S_3^i$.

(C_2) If the tank is unoccupied and water toxicity is high.
For a tank i, this can be written as $\neg S_1^i \land S_2^i$.

(C_3) If the electrical charge level in a tank is marked as 'dangerous', no matter if it is inhabited or its toxicity.
For a tank i, this can be written as S_3^i.

Therefore, the closing condition for a given tank can be written as a disjunction of conjunctions: $C_i \iff (S_1^i \land \neg S_2^i \land \neg S_3^i) \lor (\neg S_1^i \land S_2^i) \lor S_3^i$

(c) Write the complete model (truth table) of the system. For any given tank i, we have the following table:

S_1^i	S_2^i	S_3^i	$\neg S_1^i$	$\neg S_2^i$	$\neg S_3^i$	C_1^i	C_2^i	C_3^i	C^i	O^i
0	0	0	1	1	1	0	0	0	0	1
0	0	1	1	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1	0	1	0
0	1	1	1	0	0	0	1	1	1	0
1	0	0	0	1	1	1	0	0	1	0
1	0	1	0	1	0	0	0	1	1	0
1	1	0	0	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	1	1	0

(d) Write an example packet and explain what the packet means.

As an example, let's use the packet 00100110 and see we can deduce from its contents:

- Scan packet ID
  The last 5 bits of the the packet are 00110, therefore the $i = 6$.

- Examine sensor values.

  The first 3 bits of the packet are 001, therefore $S_1^6 = 0$, $S_2^6 = 0$, $S_3^6 = 1$. That is, tank 6 is not occupied by any creature, it doesn't have a high level of toxicity but it has a high electrical charge.

- Lookup system table and take corresponding action.

  Since putting the values of the sensors in the table gives us $C^6 = 1$, we close the doors of the tank.

2 Resolution in propositional logic

2.1 Conjunctive Normal Form

Convert each of the following sentences to their Conjunctive Normal Form (CNF).

(a) $A \vee (B \wedge C \wedge \neg D)$

$$\begin{aligned} &A \vee (B \wedge C \wedge \neg D) & \\ &(A \vee B) \wedge (A \vee C) \wedge (A \vee \neg D) & \textit{(Distributive law)} \end{aligned}$$

(b) $\neg(A \implies \neg B) \wedge \neg(C \implies \neg D)$

$$\begin{aligned} \neg(A \implies \neg B) \wedge \neg(C \implies \neg D) & \\ \neg(\neg A \vee B) \wedge \neg(\neg C \vee D) & \quad \textit{(Conversion of } \implies \textit{)} \\ (A \wedge \neg B) \wedge (C \wedge \neg D) & \quad \textit{(De Morgan's Rule)} \\ A \wedge \neg B \wedge C \wedge \neg D & \quad \textit{(Flatten)} \end{aligned}$$

(c) $\neg((A \implies B) \wedge (C \implies D))$

$$\begin{aligned} \neg((A \implies B) \wedge (C \implies D)) & \\ \neg((\neg A \vee B) \wedge (\neg C \vee D)) & \quad \textit{(Conversion of} \implies \textit{)} \\ (\neg(\neg A \vee B) \vee \neg(\neg C \vee D)) & \quad \textit{(De Morgan's Rule)} \\ (A \wedge \neg B) \vee (C \wedge \neg D) & \quad \textit{(De Morgan's Rule)} \\ (A \vee (C \wedge \neg D)) \wedge (\neg B \vee (C \wedge \neg D)) & \quad \textit{(Distributive law)} \\ ((A \vee C) \wedge (A \vee \neg D)) \wedge ((\neg B \vee C) \wedge (\neg B \vee D)) & \quad \textit{(Distributive law)} \\ (A \vee C) \wedge (A \vee \neg D) \wedge (\neg B \vee C) \wedge (\neg B \vee D) & \quad \textit{(Flatten)} \end{aligned}$$

(d) $(A \wedge B) \vee (C \implies D)$

$$\begin{aligned} (A \wedge B) \vee (C \implies D) & \\ (A \wedge B) \vee (\neg C \vee D) & \quad \textit{(Conversion of} \implies \textit{)} \\ (A \vee (\neg C \vee D)) \wedge (B \vee (\neg C \vee D)) & \quad \textit{(Distributive law)} \\ (A \vee \neg C \vee D) \wedge (B \vee \neg C \vee D) & \quad \textit{(Flatten)} \end{aligned}$$

(e) $A \iff (B \implies \neg C)$

$$\begin{aligned} A \iff (B \implies \neg C) & \\ (A \implies (B \implies \neg C)) \wedge ((B \implies \neg C) \implies A) & \quad \textit{(Conversion of } \iff \textit{)} \\ (A \implies (\neg B \vee C)) \wedge ((\neg B \vee C) \implies A) & \quad \textit{(Conversion of } \implies \textit{)} \\ (\neg A \vee (\neg B \vee C)) \wedge (\neg(\neg B \vee C) \vee A) & \quad \textit{(Conversion of } \implies \textit{)} \\ (\neg A \vee (\neg B \vee C)) \wedge ((B \wedge \neg C) \vee A) & \quad \textit{(De Morgan's Rule)} \\ (\neg A \vee \neg B \vee C) \wedge (B \vee A) \wedge (\neg C \vee A) & \quad \textit{(Distributive law)} \end{aligned}$$

2.2 Inference in propositional logic

Consider the following knowledge:

If the weather is both Sunny and Warm, then I Enjoy. If the weather is both Warm and Nice (not Raining), then I pick up Berries. If it is Raining, then I won't pick up Berries. If it is Raining, then I will get wet. It is Warm. It is Raining. It is Sunny.

Build a knowledge base and **prove or disprove** the following statements using resolution:

(Q_1) I won't pick up Berries

(Q_2) I will Enjoy

(Q_3) I will get wet

We first define our terms for the statements:

$$\begin{aligned} Wa &\equiv \text{'Warm'} & R &\equiv \text{'Raining'} \\ S &\equiv \text{'Sunny'} & E &\equiv \text{'Enjoy'} \\ B &\equiv \text{'Berries'} & We &\equiv \text{'Wet'} \end{aligned}$$

And the knowledge base can be expressed as:

$$S \wedge Wa \implies E \tag{1}$$
$$Wa \wedge \neg R \implies B \tag{2}$$
$$R \implies \neg B \tag{3}$$
$$R \implies We \tag{4}$$
$$Wa \tag{5}$$
$$R \tag{6}$$
$$S \tag{7}$$

To use resolution, we must first convert it to CNF:

$$\begin{aligned} &S \wedge Wa \implies E \\ &\neg(S \wedge Wa) \vee E && \textit{(Conversion of} \implies \textit{)} \\ &(\neg S \vee \neg Wa) \vee E && \textit{(De Morgan's Rule)} \\ &\neg S \vee \neg Wa \vee E && \textit{(Flatten)} \end{aligned}$$

$$\begin{aligned} &Wa \wedge \neg R \implies B \\ &\neg(Wa \wedge \neg R) \vee B && \textit{(Conversion of} \implies \textit{)} \\ &\neg Wa \vee R \vee B && \textit{(De Morgan's Rule)} \end{aligned}$$

$$\begin{aligned} &R \implies \neg B \\ &\neg R \vee \neg B && \textit{(Conversion of} \implies \textit{)} \end{aligned}$$

$$\begin{aligned} &R \implies We \\ &\neg R \vee We && \textit{(Conversion of} \implies \textit{)} \end{aligned}$$

Finally, we have our CNF Knowledge Base:

$(\neg S \vee \neg Wa \vee E) \wedge (\neg Wa \vee R \vee B) \wedge (\neg R \vee \neg B) \wedge (\neg R \vee We) \wedge Wa \wedge R \wedge S$

And to prove a statement α using the Resolution rule, we must prove show that $KB \wedge \neg\alpha$ is unsatisfiable.

(Q_1) I won't pickup berries

This question can be expressed as $\neg B$, let's see if $KB \wedge \neg(\neg B) \iff KB \wedge B$ is unsatisfiable:

$KB \wedge B$ contains $(\neg R \vee \neg B) \wedge R \wedge B$, which is not satisfiable. Then $KB \models \neg B$.

(Q_2) I will enjoy

This question can be expressed as E, let's see if $KB \wedge \neg E$ is unsatisfiable:

$KB \wedge \neg E$ contains $(\neg S \vee Wa \vee E) \wedge Wa \wedge S \wedge \neg E$, which is not satisfiable. Then $KB \models E$.

(Q_3) I will get wet

This question can be expressed as We, let's see if $KB \wedge \neg We$ is unsatisfiable:

$KB \wedge \neg We$ contains $(\neg R \vee We) \wedge R \wedge \neg We$, which is not satisfiable. Then $KB \models We$.

3 Representation in FOL

3.1 Predicates

Consider the following vocabulary:

1. *Occupation(p, o)* is a predicate where person p has occupation o.
2. *Customer(p1, p2)* is a predicate where person *p1* is a customer of person *p2*.
3. *Boss(p1, p2)* is a predicate where person *p1* is a boss of person *p2*.
4. *Doctor, Surgeon, Lawyer, Actor* are constants denoting an occupation.
5. *Emily, Joe* are constants denoting people.

Use the symbols above to **write the following statements in FOL**:

(a) Emily is either a surgeon or a lawyer
$Occupation(Emily, Surgeon) \vee Occupation(Emily, Lawyer)$

(b) Joe is an actor, but he also holds another job
$Occupation(Joe, Actor) \wedge (\, \exists\, j\; Occupation(Joe, j) \wedge j \neq Actor)$

(c) All surgeons are doctors
$\forall\, p\; Occupation(p, Surgeon) \implies Occupation(p, Doctor)$

(d) Joe does not have a lawyer (i.e. he's not a customer of any lawyer)
$\neg\, \exists\, p\; Occupation(p, Laywer) \wedge Customer(Joe, p)$

(e) Emily has a boss who is a lawyer
$\exists\, p\; Boss(p, Emily) \wedge Occupation(p, Lawyer)$

(f) There exists a lawyer all of whose customers are doctors
$\exists\, p_1\; Occupation(p_1, Lawyer) \wedge (\, \forall\, p_2\; Customer(p_1, p_2) \implies Occupation(p_2, Doctor))$

(g) Every surgeon has a lawyer
$Occupation(p_1, Surgeon) \implies (\, \exists\, p_2\; Occupation(p_2, Lawyer) \wedge Customer(p_1, p_2))$

3.2 Functions as predicates

Arithmetic assertions can be written using FOL. Use the predicates $(<, \leq, =, \neq)$, the usual arithmetic operations $(+, -, \times, /)$ as function symbols, biconditionals to create new predicates, and integer number constants to **express the following statements in FOL**:

(a) *Divisible(x, y)*: an integer number x is divisible by y if there is some integer z less than x such that $x = z \times y$.
$\forall\, x, y\; (\, \exists z\; z < x \wedge x = z \times y) \implies \textit{Divisible(x, y)}$

(b) *Even(x)*: a number is even if and only if it is divisible by 2.
$\forall\, x\; \textit{Even(x)} \iff Divisible(x, 2)$

(c) *Odd(x)*: a number is odd if it is not divisible by 2.
$\forall\, x\; \neg \textit{Divisible(x, 2)} \implies \textit{Odd(x)}$

(d) *Odd(x)*: a number is odd if it is the result of summing 1 to an even number.
$\forall\, x\; (\, \exists\, y\; \textit{Even(y)} \wedge x = 1 + y) \implies \textit{Odd(x)}$

(e) *Prime(x)*: a number is prime if is divisible only by itself.
$\forall\, x\; (\, \forall y\; \textit{Divisible(x, y)} \implies (y = 1 \vee y = x)) \implies \textit{Prime(x)}$

(f) There is only one even prime number.

$\exists\, x\; \mathit{Even(x)} \wedge \mathit{Prime(x)} \wedge (\neg \exists\, y \mathit{Even(y)} \wedge \mathit{Prime(y)} \wedge x \neq y)$

(g) Every integer number is equal to a product of prime numbers (Hint: you can use $\Pi_{i=1}^{k} p_k$ to express a product of numbers, or use $\ldots$ to express a repeating pattern like $p_1, \ldots,$ p_n meaning p_1, p_2, p_3 and on until p_n).

$\forall\, x\; (\exists p_1, p_2, \ldots, p_n\;\; \mathit{Prime(p_1)} \wedge \mathit{Prime(p_2)} \wedge \cdots \wedge \mathit{Prime(p_n)} \wedge x = \Pi_{i=1}^{k} p_k)$

4 Resolution in FOL

A new niche market has been discovered and some of your friends had the idea of developing a girl group K-Pop recommender system. Instead of using machine learning (as they think doing so may include bias in the data), they decided that going for a rule-based system was a better approach. This is what they said about it:

- All users who like *GG* are known as *Sone*, and therefore all *Sone* like *GG*
- All users who like *RV* are known as *Reveluvs*, and therefore all *Reveluvs* like *RV*
- All users who like *BP* are known as *Blinks*, and therefore all *Blinks* like *BP*
- All users who like both *Dance* and *Ballads* will always like *CH*
- All users who like both *Drama* and *Ballads* will always like *HE*
- For all users who identify as *Sone*, the following holds:
  - If they like *Electro*, they will always like *DJH*
  - If they like *Drama*, they will always like *SEO*
  - If they like *Ballads*, they will always like *TAE*

You start to think if going with Machine Learning could result in less bias than using such rules, but you love your friends and carry on with their business idea. Your task is then to:

(a) **Generate the knowledge base** by converting each rule to symbolic form.

For the translation to FOL, we will use predicates such as $GG(x)$ or $Dance(x)$ to describe that a user x likes a group, e.g. *GG*, or a genre, e.g. *Dance*.

- $\forall x\ GG(x) \iff Sone(x)$
- $\forall x\ RV(x) \iff Reveluvs(x)$
- $\forall x\ BP(x) \iff Blinks(x)$
- $\forall x\ Dance(x) \land Ballads(x) \implies CH(x)$
- $\forall x\ Drama(x) \land Ballads(x) \implies HE(x)$
- $\forall x\ Sone(x) \land Electro(x) \implies DJH(x)$
- $\forall x\ Sone(x) \land Drama(x) \implies SEO(x)$
- $\forall x\ Sone(x) \land Ballads(x) \implies TAE(x)$

(b) Using **resolution, prove or disprove** (by showing the complete process) that if a new user u_1 is a fan of *GG* and identifies as *Reveluv*, then *TAE* will be a good recommendation.

We first need to convert the knowledge base to its CNF:

(a) Eliminate biconditionals and conditionals

For the first three statements, we can simply interpret both symbols (e.g. $GG(x)$ and $Sone(x)$) as the same symbol or alias.

The other statements become:

- $\forall x\ \neg(Dance(x) \land Ballads(x)) \lor CH(x)$
- $\forall x\ \neg(Drama(x) \land Ballads(x)) \lor HE(x)$
- $\forall x\ \neg(Sone(x) \land Electro(x)) \lor DJH(x)$
- $\forall x\ \neg(Sone(x) \land Drama(x)) \lor SEO(x)$
- $\forall x\ \neg(Sone(x) \land Ballads(x)) \lor TAE(x)$

(b) Move $\neg$ inwards

- $\forall x\ (\neg Dance(x) \lor \neg Ballads(x)) \lor CH(x)$
- $\forall x\ (\neg Drama(x) \lor \neg Ballads(x)) \lor HE(x)$
- $\forall x\ (\neg Sone(x) \lor \neg Electro(x)) \lor DJH(x)$

- $\forall\, x\, (\neg Sone(x) \vee \neg Drama(x)) \vee SEO(x)$
- $\forall\, x\, (\neg Sone(x) \vee \neg Ballads(x)) \vee TAE(x)$

(c) Standarize variable names

We put unique quantifier variables in each statement

- $\forall\, x\, (\neg Dance(x) \vee \neg Ballads(x)) \vee CH(x)$
- $\forall\, y\, (\neg Drama(y) \vee \neg Ballads(y)) \vee HE(y)$
- $\forall\, z\, (\neg Sone(z) \vee \neg Electro(z)) \vee DJH(z)$
- $\forall\, a\, (\neg Sone(a) \vee \neg Drama(a)) \vee SEO(a)$
- $\forall\, b\, (\neg Sone(b) \vee \neg Ballads(b)) \vee TAE(b)$

(d) Skolemize

Since we don't have any existential quantifiers, this step is not necessary.

(e) Drop universal quantifiers

- $(\neg Dance(x) \vee \neg Ballads(x)) \vee CH(x)$
- $(\neg Drama(y) \vee \neg Ballads(y)) \vee HE(y)$
- $(\neg Sone(z) \vee \neg Electro(z)) \vee DJH(z)$
- $(\neg Sone(a) \vee \neg Drama(a)) \vee SEO(a)$
- $(\neg Sone(b) \vee \neg Ballads(b)) \vee TAE(b)$

(f) Flatten and distribute $\wedge$ over $\vee$:

We can put together all the sentences on our KB to form the final CNF form:

$$
\begin{aligned}
&(\neg Dance(x) \vee \neg Ballads(x) \vee CH(x)) \wedge \\
&(\neg Drama(y) \vee \neg Ballads(y) \vee HE(y)) \wedge \\
&(\neg Sone(z) \vee \neg Electro(z) \vee DJH(z)) \wedge \\
&(\neg Sone(a) \vee \neg Drama(a) \vee SEO(a)) \wedge \\
&(\neg Sone(b) \vee \neg Ballads(b) \vee TAE(b))
\end{aligned}
$$

Now, we need to prove that if user u satisfies $GG(u) \wedge Reveluv(u) \iff Sone(u) \wedge Reveluv(u)$ then $TAE(u)$ would be a good recommendation.

We append the negated, CNF form of this condition in the KB: $Sone(u) \wedge Reveluv(u) \wedge \neg TAE(u)$ and apply Resolution to prove the if recommendation is good by contradiction.

$$((\neg Sone(b) \vee \neg Ballads(b) \vee TAE(b)) \wedge Sone(u) \wedge Reveluv(u) \wedge \neg TAE(u)$$

which simplifies to $\neg Ballads(u) \wedge Reveluv(u)$ which can not be simplified further to reach a contradiction. We deduce that *TAE* is not a good recommendation for the user unless they also like *Ballads*.

(c) Considering the same user u_1, **prove or disprove** that *HE* will be a good recommendation.

We append the negated, CNF form of this condition in the KB: $Sone(u) \wedge Reveluv(u) \wedge \neg HE(u)$ and apply Resolution to prove the if recommendation is good by contradiction.

$$(\neg Drama(y) \vee \neg Ballads(y) \vee HE(y))) \wedge Reveluv(u) \wedge \neg HE(u)$$

which simplifies to $(\neg Drama(u) \wedge Sone(u)) \vee (\neg Ballads(u) \wedge Reveluv(u))$ which can not be simplified further to reach a contradiction. We deduce that *HE* is not a good recommendation for the user unless they also like *Drama* and *Ballads*.

(d) Given what you know, if another user u_2 claims to be a *Sone*, a *Reveluv*, a *Blink*, and likes *Drama*; what are the possible artists and genre recommendations the system will provide?

Since $\forall\, x\, Sone(x) \wedge Drama(x) \implies SEO(x)$ and the user is a *Sone* and likes *Drama*, then *SEO* would be a good recommendation for them. No other good recommendations can be found with this data.