Artificial Intelligence I

Lab 4 - Winter Semester 2023 / 2024
https://moodle.haw-landshut.de/course/view.php?id=10282

1. Knowledge Representation.

Three friends work in a company: a C++ programmer, a Java programmer and a Python programmer. Their names are Emil, Paul and Felix. The C++ programmer has no siblings (A); he is the youngest of the friends (B). Felix, who is married to Emil's sister (C), is older than the Java programmer (D).

Knowledge often exists in verbal form and must be transferred into a formal representation so that a knowledge link can take place (think about how this can be achieved here). For the solution, general knowledge has to be considered and modeled in addition to the existing specialized knowledge. The general knowledge is not fomulated, but assumed by the reader. *What belongs to general knowledge in the example given above?*

Who programs in which language? Use as short notation the equal sign = as "programs in the language". Thus the relation $X = Y$ says that the person X programs in the language Y. We understand the $\neq$ sign analogously. Now try to deduce who programs in which language with the help of the statements from the text (A)-(D) and the signs = and $\neq$.

2. Consider a vocabulary with only four propositions, A, B, C and D. How many models are there for the following sentences?

(a) $B \vee C$.

(b) $\neg A \vee \neg B \vee \neg C \vee \neg D$.

(c) $(A \Rightarrow B) \wedge A \wedge \neg B \wedge C \wedge D$.

3. Use resolution to prove the sentence $\neg A \wedge \neg B$ from the clauses

S1: $(\neg A \vee B \vee E) \wedge (\neg B \vee A) \wedge (\neg E \vee A)$.

S2: $E \Rightarrow D$.

S3: $B \wedge F \Rightarrow \neg C$.

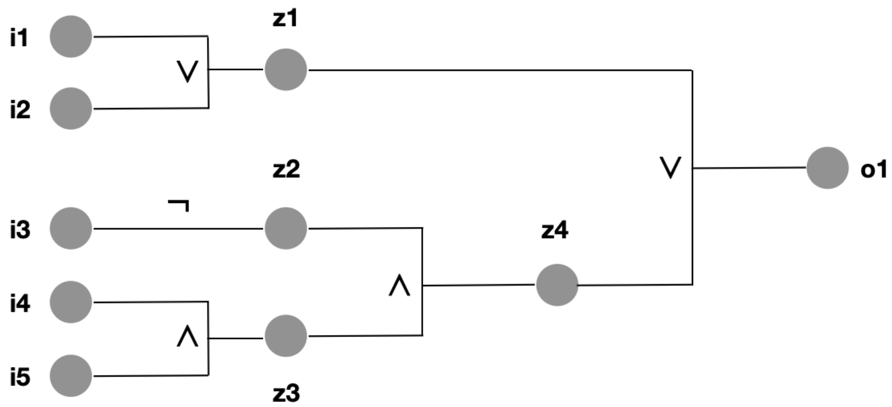
S4: $E \Rightarrow B$.

S5: $B \Rightarrow F$.

S6: $B \Rightarrow C$.

Convert the clauses to causal form first.

4. Let the following electronic circuit be given



Signals (0 for false and 1 for true) can be applied to the five input points. These are converted into an output signal by the links And, Or and Negation. The circuit can be described by propositional logic formulas.

2. a) $B \vee C \rightarrow \underline{\underline{12}}$

b) $\neg A \vee \neg B \vee \neg C \vee \neg D \rightarrow \underline{\underline{15}}$

c) $(A \Rightarrow B) \wedge A \wedge \neg B \wedge C \wedge D \rightarrow \underline{\underline{0}}$

$$\left(\begin{array}{cccc|c} A & B & C & D & b \\ 1 & 1 & 1 & 1 & F \end{array}\right)$$

a) $B \vee C \rightarrow 3 \times \text{True}$

B	C	$B \vee C$
0	0	F
0	1	T
1	0	T
1	1	T

$\}\ 3$

$\Downarrow$

Missing Propositions $\underbrace{A, D}_{2^2=4}$

$3 \cdot 2^2 = \underline{\underline{12}}$

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

$$\underbrace{\cancel{A \wedge \neg A} \vee A \wedge \underbrace{B \wedge \neg B}_{\emptyset} \wedge C \wedge D}_{\emptyset}$$

3. S1 $(\neg A \vee B \vee E) \wedge (\neg B \vee A) \wedge (\neg E \vee A)$

S2 $E \Rightarrow D = \neg E \vee D$

S3 $[B \wedge (F \Rightarrow \neg C)] = \cancel{B \wedge (\neg F \vee \neg C)} = \neg(B \wedge F) \vee \neg C = \neg B \vee \neg F \vee \neg C$

S4 $E \Rightarrow B = \neg E \vee B$

S5 $B \Rightarrow F = \neg B \vee F$

S6 $B \Rightarrow C = \neg B \vee C$

$$\left.\begin{array}{l} \neg A \lor B \lor E \\ \neg B \lor A \\ \neg E \lor A \\ \neg E \lor D \\ \neg B \lor \neg C \lor \neg F \\ \neg E \lor B \\ \neg B \lor F \\ \neg B \lor C \end{array}\right\} KB$$

$$\boxed{KB \land \neg\alpha}$$

$\alpha = \neg A \land \neg B$

$\neg\alpha = \neg(\neg A \land \neg B)$

$= \underline{\underline{A \lor B}}$

$$\boxed{KB \land A \lor B}$$

gesucht: $\neg A \lor \neg B \land A \lor B$

---

1 $\neg A \lor B \lor E$

2 $\neg B \lor A$

3 $\neg E \lor A$

4 $\neg E \lor D$

5 $\neg B \lor \neg C \lor \neg F$

6 $\neg E \lor B$

7 $\neg B \lor F$

8 $\neg B \lor C$

$A \lor B$

$1 \land 8$

$(\neg A \lor B \lor E) \land (\neg B \lor C)$

9) $\neg A \lor C \lor E$

$9 \land 5$

$(\neg A \lor C \lor E) \land (\neg B \lor \neg C \lor \neg F)$

10) $\neg A \lor \neg B \lor E \lor \neg F$

$10 \land 7$

$(\neg A \lor \neg B \lor E \lor \neg F) \land (\neg B \lor F)$

11) $\neg A \lor \neg B \lor \neg B \lor E$

$11 \land 6$

$(\neg A \lor \neg B \lor \neg B \lor E) \land (\neg E \lor B)$ $\quad \neg A \lor \neg B$

12) $\neg A \lor \neg B \lor \neg B \lor B$

$\downarrow$

Solution: $\underline{(A \lor B)} \land (\underline{\neg A \lor \neg B} \lor B \lor \neg B)$?

(1) $i1 \vee i2 \Leftrightarrow z1$

(2) $i3 \Leftrightarrow \neg z2$

(3) $i4 \wedge i5 \Leftrightarrow z3$

(4) $z2 \wedge z3 \Leftrightarrow z4$

(5) $z1 \vee z4 \Leftrightarrow o1$

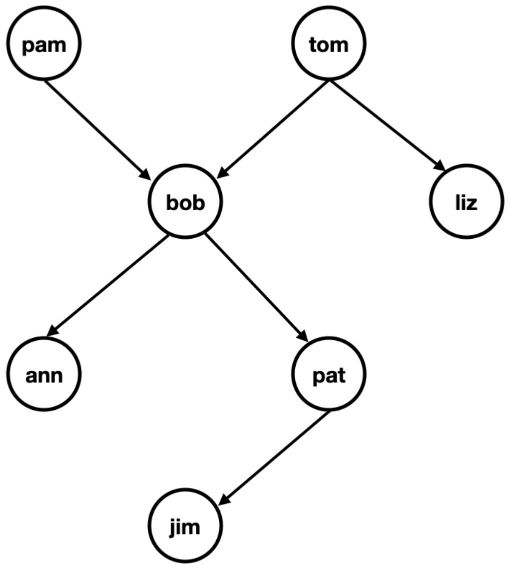
We claim that the output signal is always 1 when the input signal $i1 = 1$:

$$i1 \Rightarrow o1.$$

1. How many possible combinations would have to be considered if you wanted to prove this statement using a truth table?
2. How many combinations are necessary considering that the values of the intermediate nodes can be calculated from the input signals?
3. Transform statements (1)-(5) into CNF.
4. Use resolution to prove $i1 \Rightarrow o1$.

5. Knowledge Representation in Prolog.

Read A.2.1 - A.2.7 in P.M. Nugues, Language Processing with Perl and Prolog, then visit https://swish.swi-prolog.org/ and create a notebook based on an empty profile. Try to represent the following family relation



in Prolog by choosing `parent` as the name of the relation and e.g. stating the fact that Tom is a parent of Bob as: `parent(tom, bob)`.

(a) Execute the following queries
  - `?- parent(bob, pat)`
  - `?- parent(liz, pat)`
  - `?- parent(tom, ben)`
  - `?- parent(X, liz)`
  - `?- parent(bob, X)`
  - `?- parent(X, Y)`
  - `?- parent(Y, jim), parent(X, Y)`

(b) Now, ask the following questions in Prolog:
  - Who is Pat's parent?
  - Does Liz have a child?
  - Who is Pat's grandparent?

- Do Ann and Pat have a common parent?

(c) Extend your current program (KB) by adding the information on the sex of the people that occur in the `parent` relation, e.g. `female(pam)`, `male(tom)`. In addition, introduce the `offspring` relation as the inverse of the `parent` relation. Now, translate the following logical statements into Prolog rules

- `mother`: For all X and Y, X is the mother of Y if X is a parent of Y and X is a female.
- `sister`: For any X and Y, X is a sister of Y if both X and Y have the same parent, and X is a female.

(d) Define the relation `grandchild` using the `parent` relation.

(e) Define the relation `aunt` in terms of the relations `parent` and `sister`.