

Problem Set 1 Solutions

1. [Sentence Translation]

- (a) i. $\exists x(\text{Barber}(x) \wedge (\forall y(\text{Man-In-Town}(y) \wedge \neg\text{Shave}(y, y) \Rightarrow \text{Shave}(x, y))))$
ii. $\exists x(\text{Student}(x) \wedge \text{Failed}(x, \text{This-Class}, \text{Last-Year}) \wedge \forall y(\text{Student}(y) \wedge \text{Failed}(y, \text{This-Class}, \text{Last-Year}) \Rightarrow x = y))$
iii. $\neg\exists y(\text{Help}(y, I, \text{Logic-Problems})) \Rightarrow \neg\text{Finish}(I, \text{Homework})$
- (b) i. A. Intended meaning:
Go-To-Movie(I) \vee \neg Go-To-Movie(I)
B. Direct translation:
Go-To-Movie(I) \wedge \neg Go-To-Movie(I)
Direct translation is invalid.
- ii. A. Intended meaning:
 $(\neg\text{Grade}(I, \text{This-Class}, A) \Rightarrow \text{GPA-Below}(I, 3.5)) \wedge$
 $(\text{Grade}(I, \text{This-Class}, A) \Rightarrow \neg\text{GPA-Below}(I, 3.5))$
It is fine if you only included the first half of the conjunction.
B. Direct translation:
 $\text{Grade}(I, \text{This-Class}, A) \vee \text{GPA-Below}(I, 3.5)$
Direct translation is satisfiable.
- iii. A. Intended meaning:
 $\text{Interested-In}(You, \text{Academy-Award}) \Rightarrow \text{On-TV}(\text{Academy-Award}, \text{Tonight})$
 ~~$\text{On-TV}(\text{Academy-Award}, \text{Tonight})$~~
Direct translation is satisfiable.
B. Direct translation:
- iv. A. Intended meaning: $(\text{LivedHere}(You) \Rightarrow \text{BeHomeNow}(You)) \wedge$
 $(\text{BeHomeNow}(You) \Rightarrow \neg\text{BeHere}(You)) \wedge (\text{LivedHere}(You) \Rightarrow \neg\text{BeHere}(You))$
B. Direct translation:
Same as the Intended meaning.
Direct translation is satisfiable.

2. [Resolution Refutation]

Definitions of most of the predicates should be intuitive. To deal with years, you need to come up with a set of predicates (and possibly functions, if you wish) so that you can compare numbers, give chronological order of years, etc.

A possible solution is given below:

- (a) Man(Marcus)
(b) Pompeian(Marcus)

- (c) Born(Marcus, 40 A.D.)
- (d) $\forall x(\text{Man}(x) \Rightarrow \text{Mortal}(x))$
- (e) $\forall x(\text{Pompeian}(x) \Rightarrow \text{Dies}(x, 79 \text{ A.D.}))$
- (f) $\neg \exists x(\text{Mortal}(x) \wedge (\exists y(\text{Lives}(x, y) \wedge \text{GreaterThan}(y, 150))))$
- (g) Now(2004)
- (h) $\forall x \forall y(\text{Alive}(x, y) \Rightarrow \neg \text{Dead}(x, y))$
- (i) $\forall x \forall y \forall z((\text{Dies}(x, y) \wedge \text{Later}(z, y)) \Rightarrow \text{Dead}(x, z))$
- (j) Negated Goal: Alive(Marcus, 2004)

To complete resolution refutation, you need the following fact:

- (k) Later(2004, 79 A.D.)

After converting everything into CNF form, the sentences look as follows:

- (a) Man(Marcus)
- (b) Pompeian(Marcus)
- (c) Born(Marcus, 40 A.D.)
- (d) $\neg \text{Man}(x_1) \vee \text{Mortal}(x_1)$
- (e) $\neg \text{Pompeian}(x_2) \vee \text{Dies}(x_2, 79 \text{ A.D.})$
- (f) $\neg \text{Mortal}(x_3) \vee \neg \text{Lives}(x_3, y_1) \vee \neg \text{GreaterThan}(y, 150)$
- (g) Now(2004)
- (h) $\neg \text{Alive}(x_4, y_2) \vee \neg \text{Dead}(x_4, y_2)$
- (i) $\neg \text{Dies}(x_5, y_3) \vee \neg \text{Later}(z_1, y_3) \vee \text{Dead}(x_5, z_1)$
- (j) Alive(Marcus, 2004)
- (k) Later(2004, 79 A.D.)

Steps of resolution:

- (l) Dies(Marcus, 79 A.D.) (b) & (e)
- (m) $\neg \text{Later}(z_1, 79 \text{ A.D.}) \vee \text{Dead}(\text{Marcus}, z_1)$ (i) & (l)
- (n) Dead(Marcus, 2004) (k) & (m)
- (o) $\neg \text{Alive}(\text{Marcus}, 2004)$ (h) & (n)
- (p) Contradiction (j) & (o)

3. [Mine Sweeper and Logic] (60pt.)

The rules can be grouped into three categories:

i. (4pt.) All squares around an Adjacent0 square are safe.

$$\text{Adjacent0}([x, y]) \wedge \text{N}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{E}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{NE}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{SE}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{N}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{E}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{NE}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$$

$$\text{Adjacent0}([x, y]) \wedge \text{SE}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$$

ii. (6pt.) If a square is found to contain a bomb, then all squares adjacent to it and all squares two steps away from it are all safe. $\text{Bomb}([x, y]) \wedge \text{N}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{E}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{NE}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{SE}([u, v], [x, y]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{N}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{E}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{NE}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$ $\text{Bomb}([x, y]) \wedge \text{SE}([x, y], [u, v]) \Rightarrow \text{Safe}([u, v])$

$$\text{Bomb}([x, y]) \wedge \text{N}([u, v], [x, y]) \wedge \text{N}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{N}([u, v], [x, y]) \wedge \text{NE}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{NE}([u, v], [x, y]) \wedge \text{NE}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{NE}([u, v], [x, y]) \wedge \text{E}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{E}([u, v], [x, y]) \wedge \text{E}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{E}([u, v], [x, y]) \wedge \text{SE}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{SE}([u, v], [x, y]) \wedge \text{SE}([p, q], [u, v]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{SE}([u, v], [x, y]) \wedge \text{N}([u, v], [p, q]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{N}([x, y], [u, v]) \wedge \text{N}([u, v], [p, q]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{N}([x, y], [u, v]) \wedge \text{NE}([u, v], [p, q]) \Rightarrow \text{Safe}([p, q])$$

$$\text{Bomb}([x, y]) \wedge \text{NE}([x, y], [u, v]) \wedge \text{NE}([u, v], [p, q]) \Rightarrow$$

$$\text{Safe}([p, q]) \text{Bomb}([x, y]) \wedge \text{NE}([x, y], [u, v]) \wedge \text{E}([u, v], [p, q]) \Rightarrow$$

$$\text{Safe}([p, q]) \text{Bomb}([x, y]) \wedge \text{E}([x, y], [u, v]) \wedge \text{E}([u, v], [p, q]) \Rightarrow$$

$$\text{Safe}([p, q]) \text{Bomb}([x, y]) \wedge \text{E}([x, y], [u, v]) \wedge \text{SE}([u, v], [p, q]) \Rightarrow$$

$$\text{Safe}([p, q]) \text{Bomb}([x, y]) \wedge \text{SE}([x, y], [u, v]) \wedge \text{SE}([u, v], [p, q]) \Rightarrow$$

$$\text{Safe}([p, q]) \text{Bomb}([x, y]) \wedge \text{SE}([x, y], [u, v]) \wedge \text{N}([p, q], [u, v]) \Rightarrow$$

iii. (6pt.) If 7 of 8 squares around an Adjacent1 square are found to be safe, then the last square must contain a bomb. $\text{Adjacent1}([x, y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \text{Safe}([u_1, v_1]) \wedge \text{NE}([u_2, v_2], [x, y]) \wedge \text{Safe}([u_2, v_2]) \wedge \text{E}([u_3, v_3], [x, y]) \wedge \text{Safe}([u_3, v_3]) \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Safe}([u_4, v_4]) \wedge \text{N}([x, y], [u_5, v_5]) \wedge \text{Safe}([u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{Safe}([u_6, v_6]) \wedge \text{E}([x, y], [u_7, v_7]) \wedge \text{Safe}([u_7, v_7]) \wedge \text{SE}([x, y], [u_8, v_8]) \Rightarrow \text{Bomb}([u_8, v_8])$

$$\begin{aligned}
& \text{Adjacent1}([x, y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \text{Saft}([u_1, v_1]) \wedge \text{NE}([u_2, v_2], [x, y]) \wedge \\
& \text{Saft}([u_2, v_2]) \wedge \text{E}([u_3, v_3], [x, y]) \wedge \text{Saft}([u_3, v_3]) \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Saft}([u_4, v_4]) \\
& \wedge \text{N}([x, y], [u_5, v_5]) \wedge \text{Saft}([u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{Saft}([u_6, v_6]) \wedge \text{E}([x, y], \\
& [u_7, v_7]) \wedge \text{SE}([x, y], [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \Rightarrow \text{Bomb}([u_7, v_7]) \text{Adjacent1}([x, y]) \wedge \\
& \text{N}([u_1, v_1], [x, y]) \wedge \text{Saft}([u_1, v_1]) \wedge \text{NE}([u_2, v_2], [x, y]) \wedge \text{Saft}([u_2, v_2]) \wedge \text{E}([u_3, v_3], \\
& [x, y]) \wedge \text{Saft}([u_3, v_3]) \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Saft}([u_4, v_4]) \wedge \text{N}([x, y], [u_5, v_5]) \wedge \\
& \text{Saft}([u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{E}([x, y], [u_7, v_7]) \wedge \text{Saft}([u_7, v_7]) \wedge \text{SE}([x, y], \\
& [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \Rightarrow \text{Bomb}([u_6, v_6]) \text{Adjacent1}([x, y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \\
& \text{Saft}([u_1, v_1]) \wedge \text{NE}([u_2, v_2], [x, y]) \wedge \text{Saft}([u_2, v_2]) \wedge \text{E}([u_3, v_3], [x, y]) \wedge \text{Saft}([u_3, v_3]) \\
& \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Saft}([u_4, v_4]) \wedge \text{N}([x, y], [u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \\
& \text{Saft}([u_6, v_6]) \wedge \text{E}([x, y], [u_7, v_7]) \wedge \text{Saft}([u_7, v_7]) \wedge \text{SE}([x, y], [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \\
& \Rightarrow \text{Bomb}([u_5, v_5]) \text{Adjacent1}([x, y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \text{Saft}([u_1, v_1]) \wedge \\
& \text{NE}([u_2, v_2], [x, y]) \wedge \text{Saft}([u_2, v_2]) \wedge \text{E}([u_3, v_3], [x, y]) \wedge \text{Saft}([u_3, v_3]) \wedge \text{SE}([u_4, v_4], \\
& [x, y]) \wedge \text{N}([x, y], [u_5, v_5]) \wedge \text{Saft}([u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{Saft}([u_6, v_6]) \wedge \\
& \text{E}([x, y], [u_7, v_7]) \wedge \text{Saft}([u_7, v_7]) \wedge \text{SE}([x, y], [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \Rightarrow \\
& \text{Bomb}([u_4, v_4]) \text{Adjacent1}([x, y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \text{Saft}([u_1, v_1]) \wedge \text{NE}([u_2, v_2], \\
& [x, y]) \wedge \text{Saft}([u_2, v_2]) \wedge \text{E}([u_3, v_3], [x, y]) \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Saft}([u_4, v_4]) \wedge \\
& \text{N}([x, y], [u_5, v_5]) \wedge \text{Saft}([u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{Saft}([u_6, v_6]) \wedge \text{E}([x, y], \\
& [u_7, v_7]) \wedge \text{Saft}([u_7, v_7]) \wedge \text{SE}([x, y], [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \Rightarrow \text{Bomb}([u_3, v_3]) \\
& \text{Adjacent1}([x, y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \text{Saft}([u_1, v_1]) \wedge \text{NE}([u_2, v_2], [x, y]) \wedge \\
& \text{E}([u_3, v_3], [x, y]) \wedge \text{Saft}([u_3, v_3]) \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Saft}([u_4, v_4]) \wedge \text{N}([x, y], \\
& [u_5, v_5]) \wedge \text{Saft}([u_5, v_5]) \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{Saft}([u_6, v_6]) \wedge \text{E}([x, y], [u_7, v_7]) \wedge \\
& \text{Saft}([u_7, v_7]) \wedge \text{SE}([x, y], [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \Rightarrow \text{Bomb}([u_2, v_2]) \text{Adjacent1}([x, \\
& y]) \wedge \text{N}([u_1, v_1], [x, y]) \wedge \text{NE}([u_2, v_2], [x, y]) \wedge \text{Saft}([u_2, v_2]) \wedge \text{E}([u_3, v_3], [x, y]) \wedge \\
& \text{Saft}([u_3, v_3]) \wedge \text{SE}([u_4, v_4], [x, y]) \wedge \text{Saft}([u_4, v_4]) \wedge \text{N}([x, y], [u_5, v_5]) \wedge \text{Saft}([u_5, v_5]) \\
& \wedge \text{NE}([x, y], [u_6, v_6]) \wedge \text{Saft}([u_6, v_6]) \wedge \text{E}([x, y], [u_7, v_7]) \wedge \text{Saft}([u_7, v_7]) \wedge \text{SE}([x, y], \\
& [u_8, v_8]) \wedge \text{Saft}([u_8, v_8]) \Rightarrow \text{Bomb}([u_1, v_1])
\end{aligned}$$

Note that all rules must be Horn sentences.

(c) (5pt.) The first bomb you will find should be [6, 3]. From that, you could find [3, 1], [7, 0], [7, 6], and so on. No single order is correct, but there are wrong orders.

(d) (10pt.) You are supposed to store all the known facts and rules in some sort of data structure, and write an inference engine general enough so that it does not know anything about the predicate names, number of arguments of each predicate, etc.

You should stop your program when no new fact can be added to the knowledge base. If you terminate the program based on whether every square has been labeled Safe or Bomb, you will lost 2 points, because your inference engine is not general enough.

(e) (6pt.)

When all rules are Horn, forward chaining is possible. Forward chaining runs in linear time whereas resolution runs in exponential time.