KRR Lab 4

Introduction

0.1 Propositional resolution

Consider the following formula:

$$\varphi = (p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$$

The formula φ is true for any truth values assigned to p, q and r (in other words, the formula is valid). Based on this formula, we can write the resolution inference rule as follows:

$$\frac{\begin{array}{r} p \vee q \\ \neg p \vee r \end{array}}{q \vee r}$$

The fact that φ is valid means that the resolution inference rule is sound (i.e. cannot lead to false conclusions from true premises). The general form of resolution can be written as follows:

$$\frac{\begin{array}{r} p \vee \phi_1 \vee \phi_2 \vee \ldots \vee \phi_m \\ \neg p \vee \psi_1 \vee \psi_2 \vee \ldots \vee \psi_n \end{array}}{(\phi_1 \vee \phi_2 \vee \ldots \vee \phi_m) \vee (\psi_1 \vee \psi_2 \vee \ldots \vee \psi_n)}$$

where the ϕ's and ψ's are literals (propositions or negated propositions). When we compute the conclusion of applying the resolution inference rule presented above, it may happen that some of the literals are duplicated. We can use the fact that $\phi \vee \phi$ is equivalent to ϕ and remove any such duplicates.

In fact, it is quite common to use sets of literals instead of the disjunctions and write the resolution inference rule as follows:

$$\frac{\begin{array}{r} \{p, \phi_1, \phi_2, \ldots, \phi_m\} \\ \{\neg p, \psi_1, \psi_2, \ldots, \psi_n\} \end{array}}{\{\phi_1, \phi_2, \ldots, \phi_m\} \cup \{\psi_1, \psi_2, \ldots, \psi_n\}}$$

Note that p and $\neg p$ don't have to be the first elements from the premises in order for the rule to be applicable (the order of the elements from a set is not relevant).

A disjunction of literals (represented either as the actual disjunction or using the set notation) is called a **clause**. Given two clauses C_1 and C_2, we say that the two clauses **resolve** (can be used as premises for the resolution inference rule) iff there exists a literal in C_1 and a literal in C_2 such that one is the negation of the other (i.e. $p \in C_1$ and $\neg p \in C_2$, or $\neg p \in C_1$ and $p \in C_2$). The conclusion obtained by applying the resolution inference rule with C_1 and C_2 as premises is called the **resolvent** of the two clauses and can be written as $(C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\})$ (respectively $(C_1 \setminus \{\neg p\}) \cup (C_2 \setminus \{p\})$).

Note that it is possible to have several pairs of complement literals between two given clauses. In such cases, each pair generates a different resolvent for the two clauses. For example:

$$\frac{\begin{array}{c}\{p, \neg q\}\\ \{q, \neg p\}\end{array}}{\{q, \neg q\}} \qquad \frac{\begin{array}{c}\{p, \neg q\}\\ \{q, \neg p\}\end{array}}{\{p, \neg p\}}$$

In such cases, you should consider all the possible resolvents of the given clauses.

0.2 Example

We will consider the following premises:

$$\begin{aligned} P_1 &= p \wedge q \to r \\ P_2 &= p \wedge \neg q \to s \\ P_3 &= r \to t \\ P_4 &= s \to t \end{aligned}$$

and the conclusion

$$C = p \to t$$

In order to apply resolution for proving the conclusion based on the premises, we must first convert the premises and the conclusion to their clausal form. For this, we first convert each formula to its conjunctive normal form (conjunction of disjunctions – see the introduction of the previous lab if you don't remember this). We will also perform the conversion for the negation of the conclusion, because it is required for indirect proofs.

$$\begin{aligned} P_1 &= \neg p \vee \neg q \vee r \\ P_2 &= \neg p \vee q \vee s \\ P_3 &= \neg r \vee t \\ P_4 &= \neg s \vee t \\ C &= \neg p \vee t \\ \neg C &= p \wedge \neg t \end{aligned}$$

Next, we consider each disjunction from the conjunctive normal forms as a separate clause and write the clauses as sets of literals.

$$
\begin{aligned}
&P_1 = \{\neg p, \neg q, r\}\\
&P_2 = \{\neg p, q, s\}\\
&P_3 = \{\neg r, t\}\\
&P_4 = \{\neg s, t\}\\
&C = \{\neg p, t\}\\
&(\neg C)_1 = \{p\}\\
&(\neg C)_2 = \{\neg t\}
\end{aligned}
$$

Using the above clauses we can write a resolution-based proof for the conclusion. The proof presented below is generative, in the sense that the premises are used in order to generate the desired conclusion. For each clause we provide a justification: the clause is either a premise or it can be obtained from previous clauses by using resolution. In the latter case we also write the pair of complement literals that were used for the resolvent of the two clauses.

$$
\begin{aligned}
&1.\{\neg p, \neg q, r\} \leftarrow premise\\
&2.\{\neg p, q, s\} \leftarrow premise\\
&3.\{\neg r, t\} \leftarrow premise\\
&4.\{\neg s, t\} \leftarrow premise\\
&5.\{\neg p, \neg q, t\} \leftarrow 1, 3, (r, \neg r)\\
&6.\{\neg p, q, t\} \leftarrow 2, 4, (s, \neg s)\\
&7.\{\neg p, t\} \leftarrow 5, 6, (\neg q, q)
\end{aligned}
$$

We can also write an indirect proof, which is in fact a proof by contradiction. For this, we add the clauses that correspond to the negation of the conclusion to the set of premises. The goal in this case is to prove the contradiction, which corresponds to the empty clause (represented in our case by the empty set).

One possible indirect proof is presented below.

$$\begin{array}{r}
1.\{\neg p, \neg q, r\} \leftarrow \textit{premise} \\
2.\{\neg p, q, s\} \leftarrow \textit{premise} \\
3.\{\neg r, t\} \leftarrow \textit{premise} \\
4.\{\neg s, t\} \leftarrow \textit{premise} \\
5.\{p\} \leftarrow \textit{negated conclusion} \\
6.\{\neg t\} \leftarrow \textit{negated conclusion} \\
7.\{\neg q, r\} \leftarrow 1, 5, (\neg p, p) \\
8.\{q, s\} \leftarrow 2, 5, (\neg p, p) \\
9.\{t, \neg q\} \leftarrow 3, 7, (\neg r, r) \\
10.\{t, q\} \leftarrow 4, 8, (\neg s, s) \\
11.\{t\} \leftarrow 9, 10, (\neg q, q) \\
12.\{\} \leftarrow 6, 11, (\neg t, t)
\end{array}$$

1 Direct proof - 7p

Write a function that takes a set of premises and a conclusion and finds a direct proof. The general idea is to start with a list containing only the premises then test every pair of clauses to see whether they resolve or not and, if they do, add their resolvent to the set (only if it is not already there). The process ends either when the conclusion is obtained or when no more new clauses can be obtained. In the latter case you will write the following message instead of the proof: “There is not direct proof”.

Note that in order to be able to write the proofs as presented in the example, you should remember the premises that were used for each resolvent.

Bonus 3p: Write a function that finds all possible distinct proofs.

2 Indirect proof - 3p

Use the function from the previous task in order to generate indirect proofs for the given example. If the contradiction is not obtained, write the following message: “The conclusion does not necessarily follow from the premises”.

For both tasks the output should be written to a file and should be similar to the proofs provided in the example, using the syntax for propositional formulas as it was introduced in Lab 2.

Question: why is the failure message from the second task different from the one in the first task?