Logic for Computer Science - Week 6
Resolution

1 Introduction

Natural deduction is a proof system that was invented by Gentzen to be "as close as possible to actual reasoning".

However, a downside of natural deduction is that, given a sequent

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \varphi,$$

it is difficult for a computer program to find a formal proof of the sequent (i.e., it is not terribly efficient) due to the multitude of potential rules that can be applied.

Resolution is a proof system, just like natural deduction, but tailored to computers and not humans. In particular, it is easy for computers to find resolution proofs (at least, easier than finding natural deduction proofs). However, resolution proofs do not resemble human reasoning very well. There are programming languages (Prolog) that use (a variant of) resolution as their basic execution step.

2 Reminder on Conjunctive Normal Forms

Also unlike natural deduction, resolution works only of *clauses* instead of full propositional logic.

Recall that a literal is a formula that is either a propositional variable or the negation of a propositional variable. For example, the formulae $\mathtt{p}, \neg\mathtt{p}, \mathtt{q}, \neg\mathtt{q}, \neg\mathtt{r_1}$ are literals, but $\neg\neg\mathtt{p}$ and $\mathtt{p} \vee \mathtt{q}$ are not literals.

A clause is a disjunction of literals. For example, $p \vee q \vee r$ is a clause, $\neg p \vee p \vee \neg q$ is another clause and $\neg p \vee \neg p$ is yet another clause. Even p and $\neg p$ are clauses, as they are disjunctions of 1 literal(s). The disjunction of 0 literals is called the *empty* clause and is denoted by $\square$.

Note that $\square \in PL$ is a formula that is false in any truth assignment: $\hat{\tau}\Big(\square\Big) = 0$ for any truth assignment $\tau : A \to B$.

A formula in *CNF* is a conjunction of clauses. For example, the formula $(\mathtt{p} \vee \neg\mathtt{q}) \wedge (\neg\mathtt{p}' \vee \mathtt{q} \vee \neg\mathtt{r})$ is in CNF (conjunction of two clauses) and the formula $(\mathtt{p} \vee \neg\mathtt{p}) \wedge (\neg\mathtt{q} \vee \mathtt{p} \vee \mathtt{q}) \wedge (\mathtt{p})$ is also in CNF (conjunction of three clauses). As special cases, note that both $p \vee q$ and $p \wedge q$ are formulae in CNF, since the first is a conjunction of one clause and the second is conjunction of two clauses.

2.1 Clauses as Sets of Literals

We have that disjunction is:

1. associative: for all $\varphi_1, \varphi_2, \varphi_3 \in PL$, $(\varphi_1 \vee (\varphi_2 \vee \varphi_3)) \equiv ((\varphi_1 \vee \varphi_2) \vee \varphi_3)$;
2. commutative: for all $\varphi_1, \varphi_2 \in PL$, $(\varphi_1 \vee \varphi_2) \equiv (\varphi_2 \vee \varphi_1)$;
3. idempotent: for all $\varphi \in PL$, $\varphi \vee \varphi \equiv \varphi$.

Exercise 2.1. *Prove the three equivalences above.*

This means, for example, that the clauses $\mathtt{p} \vee \mathtt{p} \vee \mathtt{p} \vee \neg\mathtt{q} \vee \neg\mathtt{q}$, $\neg\mathtt{q} \vee \mathtt{p} \vee \neg\mathtt{q} \vee \mathtt{p}$, $p \vee \neg q$ are all equivalent. The three equivalences above justify the use of the notation:

For any literals $\varphi_1, \ldots, \varphi_n$:

$$\{\varphi_1, \varphi_2, \ldots, \varphi_n\} = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n.$$

That is, a clause is the set of its literals.

For example, the set of literals $\{\mathtt{p}, \neg\mathtt{q}\}$ denotes any of the clauses $\mathtt{p} \vee \neg\mathtt{q}$, $\mathtt{p} \vee \neg\mathtt{q} \vee \mathtt{p}$, $\neg\mathtt{q} \vee \neg\mathtt{q} \vee \neg\mathtt{q} \vee \mathtt{p}$ (all three of the clauses being equivalent).

2.2 CNFs as Sets of Clauses

Similarly to disjunction, conjunction also enjoys the following three properties:

1. associative: for all $\varphi_1, \varphi_2, \varphi_3 \in PL$, $(\varphi_1 \wedge (\varphi_2 \wedge \varphi_3)) \equiv ((\varphi_1 \wedge \varphi_2) \wedge \varphi_3)$;
2. commutative: for all $\varphi_1, \varphi_2 \in PL$, $(\varphi_1 \wedge \varphi_2) \equiv (\varphi_2 \wedge \varphi_1)$;
3. idempotent: for all $\varphi \in PL$, $\varphi \wedge \varphi \equiv \varphi$.

Exercise 2.2. *Prove the three equivalences above.*

The three equivalences above justify the following notation: given clauses $\varphi_1, \ldots, \varphi_n$, we write $\{\varphi_1, \ldots, \varphi_n\}$ instead of the CNF formula $\varphi_1 \wedge \ldots \wedge \varphi_n$.

For example, we write $\{\mathtt{p} \vee \neg\mathtt{q}, \mathtt{q} \vee \mathtt{r} \vee \mathtt{p}\}$ instead of $(\mathtt{p} \vee \neg\mathtt{q}) \wedge (\mathtt{q} \vee \mathtt{r} \vee \mathtt{p})$.

Going further, we will combine the two notations above and write, for example, $\{\{\mathtt{p}, \neg\mathtt{q}\}, \{\mathtt{q}, \mathtt{r}, \mathtt{p}\}\}$ for $(\mathtt{p} \vee \neg\mathtt{q}) \wedge (\mathtt{q} \vee \mathtt{r} \vee \mathtt{p})$. That is, we will write a CNF formula as a set of sets of literals.

Exercise 2.3. *Make sure you are comfortable with going from one notation to the other and vice-versa.*

Exercise 2.4. *Write the CNF formula* $\mathtt{p} \vee \mathtt{q}$ *as a set of sets of literals.*

Write the CNF formula $\mathtt{p} \wedge \mathtt{q}$ *as a set of sets of literals.*

Write the CNF formula $\square$ *as a set of sets of literals.*

3 Resolution

Resolution is a proof system with only one inference rule. The hypotheses and the conclusion of the inference rule are all clauses (unlike natural deduction, where the hypotheses and the conclusion were sequents).

Here is the resolution rule:

$$\textsc{Binary Resolution}\ \frac{C \cup \{a\} \qquad D \cup \{\neg a\}}{C \cup D}$$

In the rule above, C and D denote arbitrary clauses and $a \in A$ denotes an arbitrary propositional variable. As C is a clause and $a \in A$ is a propositional variable, it follows that $C \cup \{a\}$ is also a clause. In fact $C \cup \{a\}$ is the first hypothesis of the inference rule.

The second hypothesis is the clause $D \cup \{\neg a\}$. The conclusion is the clause $C \cup D$.

Here is an example of applying the resolution rule:

1. $\{\mathtt{p}, \mathtt{q}\}$; (premiss)
2. $\{\neg\mathtt{p}, \neg\mathtt{r}\}$; (premiss)
3. $\{\mathtt{q}, \neg\mathtt{r}\}$. (by Resolution, lines 1, 2, $a = \mathtt{p}$)

Given the fact that sets of literals are just clauses, it is equally correct to write the above as:

1. $\mathtt{p} \vee \mathtt{q}$; (premiss)
2. $\neg\mathtt{p} \vee \neg\mathtt{r}$; (premiss)
3. $\mathtt{q} \vee \neg\mathtt{r}$. (by Resolution, lines 1, 2, $a = \mathtt{p}$)

In fact, even the Resolution inference rule is sometimes given as

$$\textsc{Binary Resolution}\ \frac{C \vee a \qquad D \vee \neg a}{C \vee D,}$$

which is equally correct and perfectly equivalent to the other presentation of the inference rule given above. Make sure that you are comfortable with going between the two notations, as we will make heavy use of both of them.

Definition 3.1. *The clause $C \vee D$ resulting from applying resolution is called the* resolvent *of $C \vee a$ and $D \vee \neg a$.*

Note that the resolvent of two clauses is not unique. For example, the clauses $\mathtt{p} \vee \mathtt{q}$ and $\neg\mathtt{p} \vee \neg\mathtt{q} \vee \mathtt{r}$ have two resolvents: $\mathtt{p} \vee \neg\mathtt{p} \vee \mathtt{r}$ and respectively $\mathtt{q} \vee \neg\mathtt{q} \vee \mathtt{r}$. We may distinguish among the various resolvents by stating *on which propositional variable a* we have performed the resolution:

Definition 3.2. *The clause $C \vee D$ resulting from applying resolution is called the* resolvent *of $C \vee a$ and $D \vee \neg a$* on *a.*

For example, $\mathtt{p} \vee \neg\mathtt{p} \vee \mathtt{r}$ is the resolvent on $\mathtt{q}$ of the clauses $\mathtt{p} \vee \mathtt{q}$ and $\neg\mathtt{p} \vee \neg\mathtt{q} \vee \mathtt{r}$, while $\mathtt{q} \vee \neg\mathtt{q} \vee \mathtt{r}$ is their resolvent on $\mathtt{p}$.

Note that when we apply inference rules, we must follow them to the letter. For example, a common mistake in applying resolution is:

1. $\mathtt{p} \vee \mathtt{q} \vee \mathtt{r}$; (premiss)
2. $\neg\mathtt{p} \vee \neg\mathtt{q}$; (premiss)
3. $\mathtt{r}$. (**wrong** application of resolution)

On line 3, we may conclude either the resolvent on $\mathtt{p}$ or on $\mathtt{q}$, but it makes no sense to mix the two in order to have a single resolvent on both $\mathtt{p}$ and $\mathtt{q}$. Make sure you understand this point very well.

Here is the first correct variant:

1. $\mathtt{p} \vee \mathtt{q} \vee \mathtt{r}$; (premiss)
2. $\neg\mathtt{p} \vee \neg\mathtt{q}$; (premiss)
3. $\mathtt{q} \vee \mathtt{r} \vee \neg\mathtt{q}$; (resolvent of 1, 2 on $a = \mathtt{p}$)

and the second one:

1. $\mathtt{p} \vee \mathtt{q} \vee \mathtt{r}$; (premiss)
2. $\neg\mathtt{p} \vee \neg\mathtt{q}$; (premiss)
3. $\mathtt{p} \vee \mathtt{r} \vee \neg\mathtt{p}$. (resolvent of 1, 2 on $a = \mathtt{q}$)

4 Formal Proofs

Just as in the case of natural deduction,

Definition 4.1. *A* formal proof *of a clause φ from a set of clauses $\varphi_1, \varphi_2, \ldots, \varphi_n$ is a sequence of clauses $\psi_1, \psi_2, \ldots, \psi_m$ such that, for all $1 \leq i \leq m$:*

- *either $\psi_i \in \{\varphi_1, \ldots, \varphi_n\}$,*
- *or ψ_i is obtained by resolution from some clauses ψ_j, ψ_k that appear earlier in the formal proof: $1 \leq j, k < i$.*

Furthermore ψ_m must be the same as φ.

The clauses $\varphi_1, \ldots, \varphi_n$ are called the premisses of the formal proof and φ is the conclusion.

Here is an example of a proof of $\mathtt{p} \vee \neg\mathtt{q}$ from $\neg\mathtt{r} \vee \mathtt{p} \vee \neg\mathtt{r}'$, $\mathtt{r} \vee \mathtt{p}$ and $\mathtt{r}' \vee \neg\mathtt{q}$:

1. $\neg \mathtt{r} \vee \mathtt{p} \vee \neg \mathtt{r}'$; (premiss)
2. $\mathtt{r} \vee \mathtt{p}$; (premiss)
3. $\mathtt{r}' \vee \neg \mathtt{q}$; (premiss)
4. $\mathtt{p} \vee \neg \mathtt{r}'$; (resolution, 2, 1, $a = \mathtt{r}$)
5. $\mathtt{p} \vee \neg \mathtt{q}$. (resolution, 3, 4, $a = \mathtt{r}'$)

We sometimes also say *derivation (by resolution) of* φ instead of *formal proof of* φ. Here is a derivation of the empty clause from $\mathtt{p} \vee \neg \mathtt{q}, \mathtt{q}, \neg \mathtt{p}$:

1. $\mathtt{p} \vee \neg \mathtt{q}$; (premiss)
2. $\mathtt{q}$; (premiss)
3. $\neg \mathtt{p}$; (premiss)
4. $\mathtt{p}$; (resolution, 2, 1, $a = \mathtt{q}$)
5. $\Box$. (resolution 4, 3, $a = \mathtt{p}$)

A derivation of $\Box$ from $\varphi_1, \ldots, \varphi_n$ is sometimes also called a *refutation of* $\varphi_1, \ldots, \varphi_n$.

Exercise 4.1. *Starting with the same premisses, find a different proof by resolution of* $\Box$*.*

5 Soundness

Like natural deduction, resolution is sound. This section shows that this is indeed the case.

Lemma 5.1. *Let* $\varphi \in \{\varphi_1, \ldots, \varphi_n\}$*. We have that*

$$\varphi_1, \ldots \varphi_n \models \varphi.$$

Exercise 5.1. *Prove Lemma 5.1.*

Lemma 5.2. *Let* C, D *be two clauses and let* $a \in A$ *be a propositional variable. We have that*

$$C \cup \{a\}, D \cup \{\neg a\} \models C \vee D.$$

Exercise 5.2. *Prove Lemma 5.2.*

Theorem 5.1 (Soundness of Resolution)**.** *If there is a proof by resolution of* φ *from* $\varphi_1, \ldots, \varphi_n$*, then*

$$\varphi_1, \ldots, \varphi_n \models \varphi.$$

Proof. Let $\psi_1, \ldots, \psi_m$ be a proof by resolution of φ from $\varphi_1, \ldots, \varphi_n$.

We will prove by induction on $i \in \{1, 2, \ldots, m\}$ that

$$\varphi_1, \ldots, \varphi_n \models \psi_i.$$

Let $i \in \{1, 2, \ldots, m\}$ be an integer. We assume by the induction hypothesis that

$$\varphi_1, \ldots, \varphi_n \models \psi_l \text{ for any } l \in \{1, 2, \ldots, i-1\}$$

and we prove that

$$\varphi_1, \ldots, \varphi_n \models \psi_i.$$

By the definition of a formal proof by resolution, we must be in one of the following two cases:

1. $\psi_i \in \{\varphi_1, \ldots, \varphi_n\}$. In this case we have

   $$\varphi_1, \ldots, \varphi_n \models \psi_i$$

   by Lemma 5.1, which is what we had to show.

2. ψ_i was obtained by resolution from ψ_j, ψ_k with $1 \leq j, k < i$. In this case, ψ_j must be of the form $\psi_j = C \vee a$, ψ_k must be of the form $\psi_k = D \vee \neg a$ and $\psi_i = C \vee D$, where C, D are clauses and $a \in A$ is a propositional variable.

   By the induction hypotheses that $\varphi_1, \ldots, \varphi_n \models \psi_j$ and that $\varphi_1, \ldots, \varphi_n \models \psi_k$. Replacing ψ_j and ψ_k as detailed above, we have that

   $$\varphi_1, \ldots, \varphi_n \models C \vee a$$

   and that

   $$\varphi_1, \ldots, \varphi_n \models D \vee \neg a.$$

   We prove that

   $$\varphi_1, \ldots, \varphi_n \models C \vee D.$$

   Let τ be a model of φ_1, ..., and φ_n. We have that τ is a model of $C \vee a$ and of $D \vee \neg a$ by the semantical consequences above. By Lemma 5.2, it follows that τ is a model of $C \vee D$. But $\psi_i = C \vee D$ and therefore τ is a model ψ_i. As τ was any model of all of φ_1, ..., and φ_n, it follows that

   $$\varphi_1, \ldots, \varphi_n \models \psi_i,$$

   which is what we had to prove.

In both cases, we have established

$$\varphi_1, \ldots, \varphi_n \models \psi_i$$

for all $1 \leq i \leq m$. As $\psi_1, \ldots, \psi_m$ is a proof of φ, it follows that $\psi_m = \varphi$ and therefore, for $i = m$, we have

$$\varphi_1, \ldots, \varphi_n \models \varphi,$$

which is what we had to prove.

□

6 Completeness

A proof system must be sound in order to be of any use (otherwise, we could use it to prove something false).

However, it is also nice when a proof system is complete, in the sense of allowing to prove any true statement.

Unfortunately, resolution is not complete, as shown by the following example:

Theorem 6.1 (Incompleteness of Resolution)**.** *There exist clauses $\varphi_1, \ldots, \varphi_n, \varphi$ such that*

$$\varphi_1, \ldots, \varphi_n \models \varphi,$$

but there is no resolution proof of φ from $\varphi_1, \ldots, \varphi_n$.

Proof. Let $n = 2$, $\varphi_1 = \mathtt{p}$, $\varphi_2 = \mathtt{q}$ and $\varphi = \mathtt{p} \vee \mathtt{q}$. We clearly have that $\mathtt{p}, \mathtt{q} \models \mathtt{p} \vee \mathtt{q}$, but there is no way to continue the following resolution proof:

1. $\mathtt{p}$; (premiss)
2. $\mathtt{q}$; (premiss)
3. $\cdots$

because there is no negative literal anywhere. Therefore, only $\mathtt{p}$ and $\mathtt{q}$ can be derived by resolution from $\mathtt{p}, \mathtt{q}$. $\square$

However, resolution still has a weaker form of completeness called refutational completeness:

Theorem 6.2 (Refutational Completness of Resolution)**.** *If the CNF formula $\varphi_1 \wedge \ldots \wedge \varphi_n$ is unsatisfiable, then there is a derivation by resolution of $\square$ starting from the clauses φ_1, φ_2, ..., φ_n.*

The proof is beyond the scope of the course, but it is not too complicated in case you want to prove it yourself and I recommend this exercise for the more curious minds among you.

7 Proving Validity and Logical Consequences by Resolution

We may use the soundness and refutational completeness of resolution to construct an algorithm for validity checking and logical consequence testing.

Validity Testing Here is an example of how to test the validity of the formula $\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p}$.

First of all, recall that:

Theorem 7.1. *A formula is valid iff its negation is a contradiction.*

Exercise 7.1. *Prove the theorem above.*

Therefore, establishing validity of $\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p}$ is equivalent to establishing unsatisfiability of $\neg(\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p})$.

A formula is satisfiable iff its CNF is satisfiable.

Let us compute a CNF of $\neg(\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p})$:

$$\begin{aligned}
\neg(\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p}) \equiv & \quad \neg(\neg(\mathtt{p} \vee \mathtt{q}) \vee (\mathtt{q} \vee \mathtt{p})) \\
\equiv & \quad \neg\neg(\mathtt{p} \vee \mathtt{q}) \wedge \neg(\mathtt{q} \vee \mathtt{p}) \\
\equiv & \quad (\mathtt{p} \vee \mathtt{q}) \wedge (\neg\mathtt{q}) \wedge (\neg\mathtt{p}).
\end{aligned}$$

We have reached a CNF. Starting with the clauses that make up the CNF, we can derive $\square$:

1. $\mathtt{p} \vee \mathtt{q}$; (premiss)
2. $\neg\mathtt{q}$; (premiss)
3. $\neg\mathtt{p}$; (premiss)
4. $\mathtt{p}$; (resolution, 1, 2, $a = \mathtt{q}$)
5. $\square$. (resolution, 4, 3, $a = \mathtt{p}$)

We may reason by applying the following corrolary of the soundness theorem:

Corollary 7.1. *If $\square$ is derivable by resolution from $\varphi_1, \ldots, \varphi_n$, then the set of clauses $\{\varphi_1, \ldots, \varphi_n\}$ is inconsistent.*

Proof. By the soundness theorem, we have that $\varphi_1, \ldots \varphi_n \models \square$. Assume by contradiction that $\{\varphi_1, \ldots, \varphi_n\}$ is consistent; then there is a $\tau : A \rightarrow B$ such that $\hat{\tau}\Big(\varphi_1\Big) = \ldots = \hat{\tau}\Big(\varphi_n\Big) = 1$. As $\varphi_1, \ldots \varphi_n \models \square$, we also have that $\hat{\tau}\Big(\square\Big) = 1$. But, by definition, there is no such assignment (that makes $\square$ true), and therefore our assumption must have been false. Therefore $\{\varphi_1, \ldots, \varphi_N\}$ is not consistent. $\square$

Continuing our example, our set of clauses is inconsistent, which means that $\neg(\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p})$ is unsatisfiable, which further means that $(\mathtt{p} \vee \mathtt{q} \rightarrow \mathtt{q} \vee \mathtt{p})$ is valid (what we had to show in the first place).

Testing Logical Consequence How to prove by resolution that $\varphi_1, \ldots, \varphi_n \models \varphi$?

We use the following theorem, which reduces logical consequence to validity:

Theorem 7.2. *Let $\varphi_1, \ldots, \varphi_n, \varphi$ be any propositional formulae. We have that*

$$\varphi_1, \ldots, \varphi_n \models \varphi$$

iff

$$\varphi_1 \wedge \ldots \wedge \varphi_n \rightarrow \varphi$$

is valid.

Exercise 7.2. *Prove the theorem above.*

To establish a logical consequence by resolution, first apply the theorem above and then apply the method shown above for proving validity.

You may find good additional explanations on propositional resolution at `http://intrologic.stanford.edu/notes/chapter_05.html`.