Informatics 1 - Computation & Logic: Tutorial 7

Propositional Logic: Resolution and Inference

Week 9: 14-18 November 2016

Please attempt the entire worksheet in advance of the tutorial, and bring with you all work, including (if a computer is involved) printouts of code and test results. Tutorials cannot function properly unless you do the work in advance.

You may work with others, but you must understand the work; you can't phone a friend during the exam.

Assessment is formative, meaning that marks from coursework do not contribute to the final mark. But coursework is not optional. If you do not do the coursework you are unlikely to pass the exams.

Attendance at tutorials is **obligatory**; please let your tutor know if you cannot attend.

This tutorial comes in two parts. Part A is additional material on resolution—this may be useful if you need to develop your understanding of this topic. Part B concerns the new topic, inference.

If you have already mastered resolution you can skip straight to Part B.

Part A

In this section we revisit the use of resolution to determine the validity of an entailment, and consider an alternative treatment in which the entailment relation is generated by inference rules. For much of the tutorial we use two sets of constraints,

$\mathcal{A}$ and $\mathcal{B}$, as running examples:

$\mathcal{A}$	
(i)	$(C \lor \neg D) \to (A \lor B)$
(ii)	$(A \lor \neg B) \to (D \to C)$
(iii)	$(A \lor B) \to (C \lor D)$
(iv)	$D \to (B \to A)$
(v)	$B \to (C \to D)$
(vi)	$C \to (A \to B)$

$\mathcal{B}$	
(i)	$(C \lor \neg D) \to (A \lor B)$
(ii)	$(A \lor \neg B) \to (D \to C)$
(iii)	$(A \lor B) \to (C \lor D)$
(iv)	$(C \lor D) \to (A \lor \neg B)$
(v)	$A \to \neg C$

Satisfaction

1. For each set of constraints use a Karnaugh map to plot the states *excluded* by each constraint.

$\mathcal{A}$

CD \ AB	00	01	11	10
00				
01				
11				
10				

$\mathcal{B}$

CD \ AB	00	01	11	10
00				
01				
11				
10				

Can you see from your maps whether each set of constraints is satisfiable?

In tutorial 4, we introduced conjunctive normal form (CNF), and showed that,

> any collection of constraints expressed in propositional logic is equivalent to a conjunction of clauses, where each clause is a disjunction of literals.

CNF

2. (a) For each set of constraints, use Boolean algebra to derive an equivalent conjunctive normal form.

$\mathcal{A}$

$\mathcal{B}$

(b) For each CNF plot the states *excluded* by each clause on a Karnaugh map.

$\mathcal{A}$

CD \ AB	00	01	11	10
00				
01				
11				
10				

$\mathcal{B}$

CD \ AB	00	01	11	10
00				
01				
11				
10				

Resolution

In tutorial 5, we introduced resolution as a method for determining whether a given set of constraints, expressed in CNF, is consistent. We introduced the resolution rule and showed that it is **sound** — if any valuation satisfies both premises of the rule, then it satisfies the conclusion. In particular, if from some initial set of constraints (clauses), we can use resolution to derive the empty clause (which is the impossible constraint, not satisfied by any valuation), then every valuation is must be refuted by at least one of the initial constraints.

> So, if we can derive the empty clause then the inital set of constraints is inconsistent: there is no valuation that satisfies all the constraints.

3. Use resolution to show that one of the two sets of clauses $\mathcal{A}, \mathcal{B}$ is inconsistent.

To show that resolution is **complete** we must show that,

> If the inital set of constraints is inconsistent,
> then we can derive the empty clause.

It suffices to show that if we cannot derive the empty clause then there is a valuation that satisfies the initial set of clauses — because the existence of such a valuation shows that the set of clauses is consistent..

We say that a literal whose negation does not occur in any clause is **pure**. We can easily satisfy all clauses that contain a pure literal: if it is of the form $\neg A$ we let $\mathbf{V}(A) = \bot$; if it is of the form A we let $\mathbf{V}(A) = \top$.

In fact, if any valuation, $\mathbf{W}$, satisfies all of our constraints, then so does the valuation we obtain from $\mathbf{W}$ by making all pure literals true. So if we are only concerned with satisfiability, we can start by making all pure literals true, eliminate all clauses that containg any of them, and focus on finding a valuation of the remaining variables that satisfies the remaining clauses.

4. For each set of clauses, $\mathcal{A}, \mathcal{B}$, say how many resolution pairs there are for each variable.

	A	B	C	D
$\mathcal{A}$				
$\mathcal{B}$				

How many pairs would you find for a pure literal?

The Davis-Putnam resolution procedure is based on a step that simplifies such a set of clauses, $\mathcal{X}$, by using resolution to eliminate one variable (for example, A), by resolving all available pairs for resolution using that variable. We take away all the clauses that mention A and add the results of resolving each $A, \neg A$ pair—except for any trivial results, clauses that include both some literal and its negation are trivial constraints. This produces a set of clauses, $\mathcal{X}_{\setminus A}$ that don't mention A.

5. For each example, $\mathcal{A}, \mathcal{B}$, what clauses are in the set after resolution on A?

$\mathcal{A}_{\setminus A} =$	$\mathcal{B}_{\setminus A} =$

This set, $\mathcal{X}_{\setminus A}$, has the property that any valuation of the remaining variables that satifies this set of constraints, $\mathcal{X}_{\setminus A}$, can be extended, by providing a suitable value for A, to a valuation that satisfies all the constraints in $\mathcal{X}$.

6. For each example, $\mathcal{X} = \mathcal{A}, \mathcal{B}$, explain how, *if* you were given a valuation for the remaining variables, B, C, D, satisfying every clause in $\mathcal{X}_{\setminus A}$, you could choose a valuation for A that would satisfy every clause in $\mathcal{X}$.

$\mathcal{A}$	$\mathcal{B}$

This is the crucial property that allows us to construct a satisfying valuation if resolution fails to produce the empty clause. Unless we can produce the empty clause, resolution will end with every literal pure.

7. Suppose resolution fails to produce the empty clause,

   (a) How can you construct a counterexample to the remaining constraints?

   (b) It is possible that no clauses non-trivial remain. When does this happen? In this case, how do you construct a satisfying valuation?

8. For each example, $\mathcal{X} = \mathcal{A}, \mathcal{B}$ complete the procedure by resolving successively on all available pairs for each remaining variable B, C, D in turn.

   In each case, stop if at any stage you produce the empty clause.

$(\mathcal{A}_{\backslash A})_{\backslash B} =$	$(\mathcal{B}_{\backslash A})_{\backslash B} =$
$((\mathcal{A}_{\backslash A})_{\backslash B})_{\backslash C} =$	$((\mathcal{B}_{\backslash A})_{\backslash B})_{\backslash C} =$
$(((\mathcal{A}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D} =$	$(((\mathcal{B}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D} =$
Is there a satisfying valuation for $(((\mathcal{A}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D}$? If so, give a satisfying valuation for $\mathcal{A}$.	Is there a satisfying valuation for $(((\mathcal{B}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D}$? If so, give a satisfying valuation for $\mathcal{B}$.

Part B

Rules

In informatics we often use such rules to define sets of things inductively. This means that we start with some basic things and give rules that say how more complex things are produced from these.

A rule of the form:

$$\frac{\beta_1 \quad \cdots \quad \beta_n}{\alpha}$$

allows us to derive the conclusion α from the assumptions $\beta_1, \ldots, \beta_n$.

As a first example, consider defining the grammar of a language. A grammar tells us how we can construct sentences from different kinds of words.

We give the following rules:

$$\frac{}{\text{ideas} : \mathbf{N}} \quad \frac{}{\text{linguists} : \mathbf{N}} \quad \frac{}{\text{great} : \mathbf{A}} \quad \frac{}{\text{green} : \mathbf{A}} \quad \frac{}{\text{hate} : \mathbf{V}} \quad \frac{}{\text{generate} : \mathbf{V}}$$

$$\frac{X : \mathbf{V}}{X : \mathbf{VP}}\,(V) \qquad \frac{X : \mathbf{V} \quad Y : \mathbf{NP}}{XY : \mathbf{VP}}\,(VP) \qquad \frac{X : \mathbf{NP} \quad Y : \mathbf{VP}}{XY : \mathbf{S}}\,(S)$$

$$\frac{X : \mathbf{N}}{X : \mathbf{NP}}\,(N) \qquad \frac{X : \mathbf{A} \quad Y : \mathbf{NP}}{XY : \mathbf{NP}}\,(NP)$$

Here, "ideas:**N**" means that 'ideas' is a noun. Our rules allow us to infer that particular phrases belong to various grammatical categories: noun (**N**), adjective (**A**), verb (**V**), noun-phrase (**NP**), verb-phrase (**VP**), and sentence (**S**). The variables X, Y range over phrases, where phrases are non-empty lists of words. The rules are labelled, (V), (VP), etc., for ease of reference.

For example, we can show that, "great linguists generate green ideas" is a sentence. In symbols,

$$\text{great linguists generate green ideas} : \mathbf{S}$$

We do this by constructing a tree:

$$\dfrac{\dfrac{\dfrac{}{\text{great} : \mathbf{A}} \quad \dfrac{\dfrac{}{\text{linguists} : \mathbf{N}}}{\text{linguists} : \mathbf{NP}}\,(N)}{\text{great linguists} : \mathbf{NP}}\,(NP) \quad \dfrac{\dfrac{}{\text{generate} : \mathbf{V}} \quad \dfrac{\dfrac{}{\text{green} : \mathbf{A}} \quad \dfrac{\dfrac{}{\text{ideas} : \mathbf{N}}}{\text{ideas} : \mathbf{NP}}\,(N)}{\text{green ideas} : \mathbf{NP}}\,(NP)}{\text{generate green ideas} : \mathbf{VP}}\,(VP)}{\text{great linguists generate green ideas} : \mathbf{S}}\,(S)$$

9. (a) Which of the following are sentences for this grammar?
   - i. green linguists hate great ideas
   - ii. green green green linguists hate
   - iii. generate ideas
   - iv. green ideas generate hate

   (b) How might you extend the grammar to include the sentence, "colourless green ideas sleep furiously"?

(c) We say that a grammar is **sound** if it only generates grammatical sentences, and that it is **complete** if every grammatical sentence can be generated by the rules.

    i. Is it is possible to give a sound grammar for a natural language?
    ii. Is it possible to give a complete grammar for a natural language?
    iii. Is every grammatical sentence true?
    iv. Is it possible to write a grammar that will only generate true sentences?

10. Now consider the language whose sentences are expressions of propositional logic.

    (a) Is it is possible to give a sound and complete grammar propositional logic?
    (b) Is every grammatical sentence of propositional logic true?
    (c) Is it possible to write a grammar that will only generate tautologies?

We could also write a grammar for regular expressions.

11. Give a grammar for the language in the alphabet $\{$`[, ]`$\}$ that consists only of properly matched sets of parentheses such as `[[][][[][][]]]` (but not, for example, `[[]]][][[[][[]]]`.

A natural question is whether we can write a grammar for any regular language. We will not answer it here.

Entailment

We introduce some simple rules for generating valid entailments.

$$\frac{}{\Gamma, X \vdash X}\ (I) \qquad\qquad \frac{\Gamma \vdash X \quad \Delta, X \vdash Y}{\Gamma, \Delta \vdash Y}\ Cut$$

$$\frac{\Gamma \vdash X \quad \Gamma \vdash Y}{\Gamma \vdash X \wedge Y}\ (\wedge) \qquad \frac{\Gamma, X \vdash Z \quad \Gamma, Y \vdash Z}{\Gamma, X \vee Y \vdash Z}\ (\vee) \qquad \frac{\Gamma, X \vdash Y}{\Gamma \vdash X \to Y}\ (\to)$$

Here, Γ and Δ are variables that range over sets of expressions of propositional logic, and X, Y and Z are variables that range over expressions themselves. We read the 'turnstile' $\vdash$ symbol as *entails*.

Recall that an entailment is **valid** iff whenever a valuation $\mathbf{V}$ makes all of its premises (the formulae to the left of the turnstile) true, it also makes the conclusion, the formula to the right of the turnstile, true. A counterexample to an entailment is a valuation that make all of the premises, to the left of the turnstile true, while making the conclusion, to the right of the turnstile false. If there is a counterexample the entailment is invalid. If there is no counterexample then it is valid.

A rule is **sound** iff whenever all of its assumptions are valid then so is its conclusion.

12. Show that these rules are sound, by showing that if a valuation is not a counterexample to to any of the assumptions, then it is not a counterexample to the conclusion. Why is this sufficient to show the rule is sound?

The *immediate* rule (I) has no assumptions. The double line used for the other three structural rules means that the rule can be used in either direction. The entailment below the double line is valid iff *all* of the entailments above the line are valid. Read from top to bottom, they are called *introduction rules* $(^+)$, since they introduce a new connective into the argument. Read from bottom to top, they are *elimination rules* $(^-)$ since a connective is eliminated.

These rules are designed to allow us to produce *valid* entailments. We say that a valuation validates $\mathcal{A} \vdash X$ if it makes at least one of the assumptions $A \in \mathcal{A}$ *false* or it makes X *true*. The entailment is valid iffit is validated by **every** valuation. So it is valid iff any valuation that makes all the premisses in $\mathcal{A}$ true also makes X true.[1]

Using these rules we can prove validity. For example, the following proof tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \to B, C \vdash A \to B}\,(I)}{A \to B, C, A \vdash B}\,(\to^-) \qquad \dfrac{}{A \to B, C, A \vdash C}\,(I)}{A \to B, C, A \vdash B \wedge C}\,(\wedge^+)}{A \to B, C \vdash A \to (B \wedge C)}\,(\to^+)$$

shows that $A \to B, C \vdash A \to (B \wedge C)$ is valid.

We start with the goal of proving the bottom line — showing that it is valid. The fact that all of the rules are sound, and we can derive the goal starting from no assumptions shows that the goal is valid.

To find such a proof we start with the bottom line as our goal. Matching this goal with the conclusion of a rule allows us to replace the original goal with the assumptions of the rule. If we can derive these assumptions, then the rule we have just introduced allows us to derive the original goal.

This system is complete for the fragment of propositional logic without negation, but finding proofs is often tricky. When we mix introduction and elimination rules, and search for a proof, it is sometimes hard to tell whether we are making progress, or just going round in circles.

Sequent Calculus

As we saw in the case of DFA and NFA, it is sometimes helpful to place our objects of study in a wider context. Although every NFA is equivalent to a DFA, in many ways NFA are easier to construct, and to reason about.

Here we introduce an idea due to Gentzen. Instead of reasoning about entailments, with any (finite) number of premises and a single conclusion, we reason about sequents, which allow finitely many assumptions and finitely many conclusions.

Within this context, we can give an elegant set of rules, due to Gentzen, that eliminate the searching from propositional proof.

[1] Note that the rule (I) is certainly sound, since X occurs on both sides of the turnstile.

We now allow sequents that include multiple premisses *and* multiple conclusions: Γ, Δ vary over finite sets of expressions; A, B vary over expressions. The intended interpretation is that if *all* of the premises are true then at least one of the conclusions is true. Every entailment is a sequent, with a single conclusion.

A counterexample must make all of the premises true, and all of the conclusions false — for entailments, this is just as before. This seemingly minor change allowed Gentzen to introduce this beautifully symmetric set of rules:

$$\frac{}{\Gamma, A \vdash \Delta, A}\ (I)$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ (\wedge L) \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ (\vee R)$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ (\vee L) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ (\wedge R)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta}\ (\to L) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta}\ (\to R)$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ (\neg L) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ (\neg R)$$

These are all introduction rules. This means that a goal-directed proof will always produce simpler and simpler sequents (but maybe many many simpler sequents) as our trees grow upwards.

These rules have two crucial properties:

- ▷ **soundness** For each rule if a valuation is a counterexample to the conclusion then it is a counterexample to at least one of the assumptions. (From this, it follows that each rule is sound — if the assumptions are valid then so is the conclusion.)

- ▷ **completeness** For each of the rules, if any valuation is a counterexample to at least one of the assumptions, then it is a counterexample to the conclusion. (From this it follows that this system of rules is complete, since any proof attempt either succeeds with every leaf of the tree being reached by the immediate rule, with no assumptions, or fails with sequents containing only atomic propositions, such that the set of sequents to the left of the turnstile is disjoint from the set to the right. A valuation making everything to the left true and everything to the right false, makes such a sequent false.)

13. The following rules are suggested for xor ($\oplus$).
Do they have the soundeness and completeness properties?

$$\frac{\Gamma, A \vdash B, \Delta \quad \Gamma, B \vdash A, \Delta}{\Gamma, A \oplus B \vdash \Delta}\ (\oplus L) \qquad \frac{\Gamma, A, B \vdash \Delta \quad \Gamma \vdash A, B, \Delta}{\Gamma \vdash A \oplus B, \Delta}\ (\oplus R)$$

Can you suggest corresponding rules for $\leftrightarrow$, and check their properties?

14. For each of the entailments listed below, construct a proof tree, by applying the Gentzen rules until the leaves of your tree contain no connectives. Then say whether the entailment is valid. How can a proof attempt fail? How can you can construct a falsifying valuation from a failed proof attempt?

   (a) $\mathsf{B} \land \mathsf{C} \vdash (\mathsf{A} \to \mathsf{B}) \land (\mathsf{A} \to \mathsf{C})$

   (b) $\mathsf{A} \land (\mathsf{B} \land \mathsf{C}) \vdash (\mathsf{A} \land \mathsf{B}) \land \mathsf{C}$

   (c) $\mathsf{A} \to \mathsf{B}, \mathsf{A} \land \mathsf{C} \vdash \mathsf{B} \land C$

   (d) $\mathsf{A} \lor \mathsf{B} \to \mathsf{C},\ \mathsf{C} \to \mathsf{A} \vdash \mathsf{C} \to \mathsf{B}$

   (e) $\mathsf{A} \to \mathsf{C} \vdash \mathsf{A} \to (\mathsf{B} \lor \mathsf{C})$

15. Estimate the height and breadth of the proof trees you would obtain if you applied Gentzen's rules to the sets of constraints $\mathcal{A}, \mathcal{B}$, introduced at the start of this tutorial.

16. If you produced rules for $\leftrightarrow$ in your answer to Question 13 use these and the rules for $\oplus$ given there, to show that

$$(A \leftrightarrow B) \leftrightarrow C \vdash (A \oplus B) \oplus C$$

This tutorial exercise sheet was written by Paolo Besana, and extended by Thomas French Areti Manataki, and Michael Fourman. Send comments to `Michael.Fourman@ed.ac.uk`