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)	$(C \vee \neg D) \to (A \vee B)$
(ii)	$(A \vee \neg B) \to (D \to C)$
(iii)	$(A \vee B) \to (C \vee D)$
(iv)	$D \to (B \to A)$
(v)	$B \to (C \to D)$
(vi)	$C \to (A \to B)$

$\mathcal{B}$

(i)	$(C \vee \neg D) \to (A \vee B)$
(ii)	$(A \vee \neg B) \to (D \to C)$
(iii)	$(A \vee B) \to (C \vee D)$
(iv)	$(C \vee D) \to (A \vee \neg B)$
(v)	$A \to \neg C$

Satisfaction

1. For each set of constraints use the Karnaugh map provided to show which states are *excluded* by each constraint.

   For example, the constraint $\mathcal{A}(i)$ is $(C \vee \neg D) \to (A \vee B)$; the states excluded are those that make $C \vee \neg D$ true and make $A \vee B$ false. A state makes $C \vee \neg D$ true iff it is a row where CD = 00, 11, or 10; it makes $A \vee B$ false if it is in the column where AB = 00. So there are three states excluded by this constraint, as shown on the Karnaugh map.
   The states/valuations excluded by an implication $X \to Y$ are those that make X true and Y false. Some of our constraints exclude three states, some exclude two, and one excludes four.

   $\mathcal{A}$

CD \ AB	00	01	11	10
00	(i)	(iii)	(iii)	(iii)
01	(ii)	(iv)	(ii)	(ii)
11	(i)	(iv)		(vi)
10	(i)	(v)	(v)	(vi)

   $\mathcal{B}$

CD \ AB	00	01	11	10
00	(i)	(iii)	(iii)	(iii)
01	(ii)	(iv)	(ii)	(ii)
11	(i)	(iv)	(v)	(v)
10	(i)	(iv)	(v)	(v)

   Can you see from your maps whether each set of constraints is satisfiable?
   $\mathcal{A}$ is, $\mathcal{B}$ is not.
   Observe that a simple counting argument suffices to show that $\mathcal{A}$ is satifiable.

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.

$\mathcal{A}$	$\mathcal{B}$
i. a $\{\neg C, A, B\}$	i. a $\{\neg C, A, B\}$
b $\{D, A, B\}$	b $\{D, A, B\}$
ii. a $\{\neg A, \neg D, C\}$	ii. a $\{\neg A, \neg D, C\}$
b $\{B, \neg D, C\}$	b $\{B, \neg D, C\}$
iii. a $\{\neg A, C, D\}$	iii. a $\{\neg A, C, D\}$
b $\{\neg B, C, D\}$	b $\{\neg B, C, D\}$
iv. $\{\neg B, A, \neg D\}$	iv. a $\{\neg C, A, \neg B\}$
v. $\{\neg C, D, \neg B\}$	b $\{\neg D, A, \neg B\}$
vi. $\{\neg A, B, \neg C\}$	v. $\{\neg A, \neg C\}$

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

A clause with 1 literal excludes 1/2 of the 16 possible states; a clause with two literals excludes 1/4 of them; in general, a clause with n literals excludes $1/2^n$ of the possible states. On the Karnaugh map each clause excludes a rectangle (possibly wrapped around the back of the torus).

$\mathcal{A}$

CD \ AB	00	01	11	10
00	(i.b)	(iii.b)	(iii.a) (iii.b)	(iii.a)
01	(ii.b)	(iv)	(ii.a)	(ii.a) (ii.b)
11	(i.a)	(iv)		(vi)
10	(i.a) (i.b)	(v)	(v)	(vi)

$\mathcal{B}$

CD \ AB	00	01	11	10
00	(i.b)	(iii.b)	(iii.a) (iii.b)	(iii.a)
01	(ii.b)	(iv.b)	(ii.a)	(ii.a) (ii.b)
11	(i.a)	(iv.b) (iv.a)	(v)	(v)
10	(i.a) (i.b)	(iv.a)	(v)	(v)

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.

We know from the previous question that $\mathcal{B}$ is satisfiable.

We resolve the clauses in $\mathcal{B}$. Tautologous (and hence trivial) clauses are shown in **bold** face.

A	B	C	D
${}^{B}\{\mathbf{\neg C, B, \neg D, C}\}$	$\{\mathbf{\neg D, C, D}\}$	$\{\neg D\}$	$\{\}$
${}^{B}\{\mathbf{\neg C, B, C, D}\}$	$\{\mathbf{\neg D, C, \neg C}\}$	$\{\mathbf{\neg D, D}\}$	
${}^{B}\{\neg C, B\}$	${}^{C}\{\neg D, C\}$	$\{D\}$	
${}^{B}\{\mathbf{D, B, \neg D, C}\}$	$\{\mathbf{\neg C, C, D}\}$	+ 3 duplicates	
${}^{B}\{D, B, C\}$	${}^{C}\{\neg C\}$		
${}^{B}\{D, B, \neg C\}$	${}^{C}\{\neg C, \neg D\}$		
${}^{B}\{\mathbf{\neg C, \neg B, \neg D, C}\}$	${}^{C}\{D, C\}$		
${}^{B}\{\mathbf{\neg C, \neg B, C, D}\}$	$\{\mathbf{D, C, \neg D, \neg C}\}$		
${}^{B}\{\neg C, \neg B\}$	${}^{C}\{D, \neg C\}$		
${}^{B}\{\neg D, \neg B, C\}$	$\{\mathbf{D, \neg C, \neg D}\}$		
${}^{B}\{\mathbf{\neg D, \neg B, C, D}\}$	+ 6 duplicates		
${}^{B}\{\neg D, \neg B, \neg C\}$	(omitted)		

The empty clause is found, the clauses are 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, **W**, satisfies all of our constraints, then so does the valuation we obtain from **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
$\mathcal{A}$	9	12	12	12
$\mathcal{B}$	12	9	12	9

How many pairs would you find for a pure literal? $\boxed{0}$

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?

$\mathcal{A}_{\setminus A} =$ $\{\{B, \neg D, C\},$ $\{\neg B, C, D\},$ $\{\neg C, D, \neg B\},$ $\{\neg B, \neg D, C\},$ $\{D, B, C\},$ $\{\neg C, B\},$ $\{D, B, \neg C\}\}$	$\mathcal{B}_{\setminus A} =$ $\{\{B, \neg D, C\},$ $\{\neg B, C, D\},$ $\{\neg C, B\},$ $\{D, B, C\},$ $\{D, B, \neg C\},$ $\{\neg C, \neg D\},$ $\{\neg D, \neg B, C\},$ $\{\neg D, \neg B, \neg C\}\}$

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 $\mathcal{X}$.

If a valuation, **V**, satisfies $\mathcal{X}_{\setminus A}$ then either **V** makes all the clauses in $\mathcal{X}$ that include the literal $\neg A$ true, or **V** makes all the clauses in $\mathcal{X}$ that include the literal $\neg A$ true (or both). In either case, deleting all clauses satisfied by **V**, if A appears in the remaining clauses, it will be as a pure literal, either A or $\neg A$. We can extend **V** with the value for A required makes this literal true.

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.

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 $\mathcal{X}$.

$\mathcal{A}$	$\mathcal{B}$
From $\mathcal{A}$, delete all clauses satisfied by the valuation. Assuming that the valuation satisfies $\mathcal{A}_{\setminus \mathcal{A}}$, this will either leave A or $\neg A$ as pure literal, or all clauses containing A will be deleted. In the former case, you may add $A = \top$ or $A = \bot$ respectively to the valuation; in the latter, you may safely do either.	The same, but here we do not expect to find a satisfying valuation.

7. Suppose resolution fails to produce the empty clause,

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

First, make any pure literals in one's final resolution pool (in the examples above, $(((\mathcal{X}_{\setminus A})_{\setminus B})_{\setminus C})_{\setminus D})$ true.
From the preceding resolution pool, remove any clauses already satisfied by one's partial valuation. Make any resulting pure literals true. If, at any point, the whole pool is removed, any atoms previously in the pool but not in the valuation may be safely given either valuation.
Repeat until all atoms are given a valuation

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

This happens when you have more than one pair of complementary literals in the same clause. If resolving on some literal X terminates resolution by producing only trivial clauses, X and any unresolved atoms may safely be given any 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.

$(\mathcal{A}_{\backslash A})_{\backslash B} =$ $\{\{D, C\},$ $\{D, \neg C\},$ $\{\neg D, C\}\}$	$(\mathcal{B}_{\backslash A})_{\backslash B} =$ $\{\{\neg D, C\},$ $\{\neg C\},$ $\{\neg C, \neg D\},$ $\{D, C\},$ $\{D, \neg C\}\}$
$((\mathcal{A}_{\backslash A})_{\backslash B})_{\backslash C} =$ $\{\{D\}\}$	$((\mathcal{B}_{\backslash A})_{\backslash B})_{\backslash C} =$ $\{\{\neg D\},$ $\{D\}\}$
$(((\mathcal{A}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D} =$ $\{\{D\}\}$	$(((\mathcal{B}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D} =$ $\{\{\}\}$
Is there a satisfying valuation for $(((\mathcal{A}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D}$? Yes. If so, give a satisfying valuation for $\mathcal{A}$. $A = B = C = D = \top$	Is there a satisfying valuation for $(((\mathcal{B}_{\backslash A})_{\backslash B})_{\backslash C})_{\backslash D}$? No. 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:

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

9. (a) Which of the following are sentences for this grammar?

   i. green linguists hate great ideas   Yes

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

ii. green green green linguists hate Yes

$$\dfrac{\dfrac{}{\text{green} : \mathbf{A}} \quad \dfrac{\dfrac{}{\text{green} : \mathbf{A}} \quad \dfrac{\dfrac{}{\text{green} : \mathbf{A}} \quad \dfrac{\dfrac{}{\text{linguists} : \mathbf{N}}}{\text{linguists} : \mathbf{NP}}(N)}{\text{green linguists} : \mathbf{NP}}(NP)}{\text{green green linguists} : \mathbf{NP}}(NP)}{\text{green green green linguists} : \mathbf{NP}}(NP) \quad \dfrac{\dfrac{}{\text{hate} : \mathbf{V}}}{\text{hate} : \mathbf{VP}}(V)}{\text{green green green linguists hate} : \mathbf{S}}(S)$$

iii. generate ideas No

iv. green ideas generate hate No

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

$$\dfrac{}{\text{colourless} : \mathbf{A}} \quad \dfrac{}{\text{furiously} : \mathbf{AdV}} \quad \dfrac{X : \mathbf{VP} \quad Y : \mathbf{AdV}}{XY : \mathbf{VP}}(VP)$$

(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?

Yes, trivially; the empty grammar produces no ungrammatical sentences.

ii. Is it possible to give a complete grammar for a natural language?

Yes, trivially; one could produce grammar that generated all possible sequences of words for the given the vocabulary of the langauge. Of course, the real problem is, is it possible to produce a grammar that is both sound *and* complete for a natural language. This is very much a disputed issue; in the 1950's, Chomsky gave the production of complete, sound grammars of natural languages as a mission statement for the programme of Generative linguistics which has dominated the study of language from then until now, and so far, no-one has been able to do it convincingly.

iii. Is every grammatical sentence true?

No, nor even meaningful - witness "Colourless green ideas sleep furiously."

iv. Is it possible to write a grammar that will only generate true sentences?

Not if the truth conditions of some of its sentences refer to states of affairs in the world.

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

(a) Is it is possible to give a sound and complete grammar for propositional logic?

Yes, we start from atomic propositions and use simple rules to describe how the well-formed formulae (wffs), **W**, are built from simpler ones using the connectives.

$$\frac{}{A : \mathbf{W}} \text{ (where } A \text{ is a propositional letter)} \qquad \frac{X : \mathbf{W}}{\neg X : \mathbf{W}} (\neg)$$

$$\frac{X : \mathbf{W} \quad Y : \mathbf{W}}{X \vee Y : \mathbf{W}} (\vee) \quad \frac{X : \mathbf{W} \quad Y : \mathbf{W}}{X \wedge Y : \mathbf{W}} (\wedge) \quad \frac{X : \mathbf{W} \quad Y : \mathbf{W}}{X \to Y : \mathbf{W}} (\to)$$

Note that there is a subtle difference in the nature of the task here; in constructing a grammar for a natural language, the question of whether a sentence is grammatical is an empirical one, requiring verification using native speaker intuitions or large stores of language data. However, a grammar of propositional logic provides a *definition* of what counts as a Well-Formed Formula.

(b) Is every grammatical sentence of propositional logic true?

No.

(c) Is it possible to write a grammar that will only generate tautologies?

The answer is actually, Yes. However it is simpler to give rules that generate valid entailments, or valid sequents, which we will turn to shortly.

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, `[[]]][][[[][[]]]`.

$$\frac{}{[\,] : \mathbf{M}} \qquad \frac{\mathrm{x} : \mathbf{M} \quad \mathrm{y} : \mathbf{M}}{\mathrm{xy} : \mathbf{M}} \qquad \frac{\mathrm{x} : \mathbf{M}}{[\mathrm{x}] : \mathbf{M}}$$

We know that this language is not regular. So we can write a grammar for some language that is not regular. 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 a counterexample to the conclusion. then it is a counterexample to at least one of the conclusions. Why is this sufficient to show the rule is sound? A counterexample to an entailment makes all of its premises true and its the conclusion false. So an entailment is valid iff there is no counterexample.

    - ▷ $(\wedge)$: A valuation is a counterexample to the bottom line if it makes everything in Γ true and makes $(X \wedge Y)$ false. But, since this valuation makes $(X \wedge Y)$ false, it makes X false or it makes Y false. Thus it is a counterexample to one of the entaiments on the top line. Furthermore, any counterexample to either of these makes $(X \wedge Y)$ false, and so provides a counterexample to the bottom line.
    - ▷ $(\vee)$: A valuation is a counterexample to the bottom line if it makes everything in Γ true and makes $(X \vee Y)$ true. But, since this valuation makes $(X \vee Y)$ true, it makes X true or it makes Y true. Thus it is a counterexample to one of the entaiments on the top line. Furthermore, any counterexample to either of these makes $(X \vee Y)$ true, and so provides a counterexample to the bottom line.
    - ▷ $(\to)$: Since Γ is in the premises of the entailments above and below the lines, any counterexample, $\mathbf{V}$, to the entailment above or below the line must make everything in Γ true. $\mathbf{V}$ is a counterexample to the entailment above the line iff $\mathbf{V}(X) = \top$ and $\mathbf{V}(Y) = \bot$ iff $\mathbf{V}(X \to Y) = \bot$ iff $\mathbf{V}$ is a counterexample to the entailment below the line.

▷ (Cut): Since this rule does not have a double line, we only need to show that a counterexample to the conclusion is a counterexample to one of the assumptions. A counterexample to $\Gamma, \Delta \vdash Y$ makes everything in Γ, Δ true and Y false. if it makes X false it is a counteexample to the first assumption; if it makes X true it is a counterexample to the second.

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:

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

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.

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

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

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, provides a counterexample to such a sequent.)
  This is a straight-forward exercise in truth-table argumentation. A counterexample makes everything on the left true, and everything on the right false. So, for example, a counterexample to either of the assumptions of $(\to L)$ makes everything in Γ true, and everything in Δ false. A counter-example to the first

premise makes A false, while a counter-example to the second premise makes B true; in either case, $A \to B$ is true and we have a counter-example to the conclusion.

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?

- ▷ $(\oplus L)$, soundness: Observe that a counterexample to the conclusion must falsify Δ while making Γ and $A \oplus B$ true, meaning exactly one of A and B are true. A valuation that makes Γ true and Δ false will be a counterexample to the left premise, $\Gamma, A \vdash B, \Delta$, if A is true and B false, and will be a counterexample to $\Gamma, B \vdash A, \Delta$ if the reverse is the case; and one or other of these conditions is guaranteed by $A \oplus B$ being true.
- ▷ $(\oplus L)$, completeness: conversely, a valuation invalidating $\Gamma, A \vdash B, \Delta$ will make Γ and A true, and Δ and B false, which will make $A \oplus B$ true, invalidating $\Gamma, A \vdash B, \Delta$. Similar reasoning will get a similar result for the other branch.
- ▷ $(\oplus R)$, soundness: A counterexample to the conclusion must make Γ true, Δ false, and A and B both true or both false. If they are both true, this makes all the premises of $\Gamma, A, B \vdash \Delta$ true, while the conclusion is false, making it invalid. On the other hand, if they are both false, this makes all the conclusions of $\Gamma \vdash A, B, \Delta$ false, while the premise is true, making it invalid.
- ▷ $(\oplus R)$, completeness: Invalidating either formula above the line will make Γ true and Δ false, so that if $A \oplus B$ is false, $\Gamma \vdash A \oplus B, \Delta$ will be invalidated. A valuation that invalidates $\Gamma, A, B \vdash \Delta$ will make A and B both true, and one that invalidates $\Gamma \vdash A, B, \Delta$ will make them both false—either of which falsifies $A \oplus B$.

Regarding $\leftrightarrow$, consider the following:

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

Notice that the upper parts of $(\leftrightarrow L)$ and $(\oplus R)$ are the same, as are the upper parts of $(\leftrightarrow R)$ and $(\oplus L)$. This is to be expected, as $A \leftrightarrow B$ and $\neg(A \oplus B)$ are equivalent; thus, any valuation that invalidates $\Gamma, A \leftrightarrow B \vdash \Delta$ also invalidates $\Gamma \vdash A \oplus B, \Delta$, and any valuation that invalidates $\Gamma \vdash A \leftrightarrow B, \Delta$ also invalidates $\Gamma, A \oplus B \vdash \Delta$; and as such the proofs of soundness and completeness for $(\oplus L)$ and $(\oplus R)$ can easily be adapted to show soundness and completeness for $(\leftrightarrow L)$ and$(\leftrightarrow R)$.

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

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A, B, C \vdash B}\,(I)}{B, C \vdash A \to B}\,(\to R) \qquad \dfrac{\dfrac{}{A, B, C \vdash C}\,(I)}{B, C \vdash A \to C}\,(\to R)}{B, C \vdash (A \to B) \wedge (A \to C)}\,(\wedge R)}{B \wedge C \vdash (A \to B) \wedge (A \to C)}\,(\wedge L)$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A, B, C \vdash A}\,(I) \qquad \dfrac{}{A, B, C \vdash B}\,(I)}{A, B, C \vdash A \wedge B}\,(\wedge R) \qquad \dfrac{}{A, B, C \vdash C}\,(I)}{A, B, C \vdash (A \wedge B) \wedge C}\,(\wedge R)}{A, B \wedge C \vdash (A \wedge B) \wedge C}\,(\wedge L)}{A \wedge (B \wedge C) \vdash (A \wedge B) \wedge C}\,(\wedge L)$$

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

$$\dfrac{\dfrac{\dfrac{}{A, C \vdash A, B \wedge C}\,(I) \qquad \dfrac{\dfrac{}{A, B, C \vdash B}\,(I) \qquad \dfrac{}{A, B, C \vdash C}\,(I)}{A, B, C \vdash B \wedge C}\,(\wedge R)}{A \to B, A, C \vdash B \wedge C}\,(\to L)}{A \to B, A \wedge C \vdash B \wedge C}\,(\wedge L)$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{}{C \vdash B, A \vee B, C}\,(I) \qquad \dfrac{\dfrac{}{A, C \vdash B, A, B}\,(I)}{A, C \vdash B, A \vee B}\,(\vee R)}{C \to A, C \vdash B, A \vee B}\,(\to L) \qquad \dfrac{\dfrac{}{C \vdash B, C}\,(I) \qquad A, C \vdash B, \mathit{failure!}}{C \to A, C \vdash B}\,(\to L)}{A \vee B \to C, C \to A, C \vdash B}\,(\to L)}{A \vee B \to C, C \to A \vdash C \to B}\,(\to R)$$

$(A$	$\vee$	$B)$	$\to$	$C,$	C	$\to$	A	$\vdash$	C	$\to$	B
T	T	F	T	T	T	T	T		T	F	F

The valuation $A = true$, $B = false$, $C = true$ invalidates this inference.

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

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A, B, C}\,(I) \qquad \dfrac{}{C, A \vdash B, C}\,(I)}{A \to C, A \vdash B, C}\,(\to L)}{A \to C, A \vdash B \vee C}\,(\vee R)}{A \to C \vdash A \to (B \vee C)}\,(\to R)$$

Note that for these rules there are no cut formulae to find — and that the choice of rules to apply is limited by the context.

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.

    This requires some thought, as the rules we have here are designed to demonstrate the validity of entailments, whereas in Part 1, we were concerned with the satisfiability of constraints. How then can we express the claims that $\mathcal{A}$ and $\mathcal{B}$ are satisfiable as entailments? The key is to understand that

    $$\Gamma \vdash \qquad\qquad\qquad\qquad \vdash \Delta$$

    —are both perfectly meaningful entailments; the former is equivalent to the claim that the conjunction of Γ is contradictory, and the latter to the claim that the disjunction of Δ is tautologous. Thus we can, for instance, make an entailment $\mathcal{A}'$ in which the premises are the constraints of $\mathcal{A}$, and there are no conclusions. If the rules of Sequent Calculus *cannot* find prove it to be valid, the constraints are not contradictory, and so are satisfiable.

    We can give an upper bound for the height by counting connectives — we need one step to remove each connective. For $\mathcal{A}$ there are 16 (15 binary and one negation), for $\mathcal{B}$ we have the same count (but only 13 binary and 3 negations). Each binary connective may split the proof tree (depending on which of its rules applies), so worst case estimates for the breadth are 2^{15} and 2^{13}.

    These estimates strongly suggest that Gentzen's procedure is not as efficient as resolution.

    Can we give better estimates of the breadth?

    We might hope that around half of the time there would be no splitting, and reduce these estimates to $2^{7.5} \approx 90$ and $2^{6.5} \approx 181$. But we will see that this hope is over-optimistic.

    $\mathcal{A}'$ has six premises, each with $\rightarrow$ as the main connective. Dismissing the first of these will give a height of 2 and a breadth of 2. Dismissing each subsequent one will add 1 to the height and double the breadth, leaving a height of 7 and a breadth of $2^6 = 64$.

    This should be enough to convince us that Gentzen's procedure is not as efficient as resolution, which is the main lesson you should learn from this example.

Dave Cochrane has taken the analysis further (such an analysis goes beyond the requirements of this course, but may be of interest to a few intrepid explorers). Abandon hope all ye who enter here!

Once we have eliminated the 6 primary connectives, each branch will contain $0 \leq x \leq 6$ of the antecedents in their premises, and $6 - x$ of the consequents in their conclusions. The highest branches will be those which contain none of the antecedents of constraints (iv)-(vi), as those are single atoms (whereas all other premises and conclusions on each branch contain one two-place connective) and which contain the antecedents of both

(i) and (ii), as those contain a single negative literal each. Thus, each of these branches will have to dismiss eight more connectives, giving a maximum total height of 15.

Of the initial 64 branches, the $64 \times \frac{3}{8} = 24$ that contain a single premise from antecedents of (i)-(iii) will split into 2 more branches, as these are all disjuncts, and $(\vee L)$ is a branching rule. The $64 \times \frac{3}{8} = 24$ that have two will branch into 4, the $64 \times \frac{1}{8} = 24$ with 3 into 8, and the $64 \times \frac{1}{8} = 8$ with none will not branch any further. Thus, the breadth of the tree will be at most $(24 \times 2) + (24 \times 4) + (8 \times 8) + 8 = 48 + 96 + 64 + 8 = 216$.

$\mathcal{B}'$ has five premises, also each with $\to$ as the main connective; thus, once these are all dismissed the tree will have a height of 6 and a breadth of 32. At this point, each branch will have exactly 8 two-place connectives to dismiss, and a maximum of three negations, giving a maximmm total height of 17. (i)-(iv) have antecedents which, on the branches where $(\to L)$ makes them into premises, will branch; and there are no other connectives that require dismissal with a branching rule. Thus, $32 \times \frac{1}{16} = 2$ will not branch, $32 \times \frac{4}{16} = 8$ will branch into 2, $32 \times \frac{6}{16} = 12$ will branch into 4, $32 \times \frac{4}{16} = 8$ will branch into 8, and $32 \times \frac{1}{16} = 2$ will branch into 16, giving a maximum final breadth of $2 + (8 \times 2) + (12 \times 4) + (8 \times 8) + (2 \times 16) = 2 + 16 + 48 + 64 + 32 = 162$.

The actual height and breadth of each tree may turn out to be slightly less, if some branches reach the Immediate rule before all connectives are dismissed.

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

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

$$\dfrac{\dfrac{\text{See subtree 1...}}{(A \leftrightarrow B) \leftrightarrow C, A \oplus B, C \vdash}\,(\oplus L) \qquad \dfrac{\text{See subtree 2...}}{(A \leftrightarrow B) \leftrightarrow C \vdash A \oplus B, C}\,(\oplus R)}{(A \leftrightarrow B) \leftrightarrow C \vdash (A \oplus B) \oplus C}\,(\oplus R)$$

Subtree 1:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, C \vdash B}\,(I) \quad \dfrac{}{A, C \vdash B, A}\,(I)}{A \leftrightarrow B, A, C \vdash B}\,(\leftrightarrow L) \quad \dfrac{}{A, C \vdash A \leftrightarrow B, B, C}\,(I)}{(A \leftrightarrow B) \leftrightarrow C, A, C \vdash B}\,(\leftrightarrow L) \qquad \dfrac{\dfrac{\dfrac{}{A, B, C \vdash A}\,(I) \quad \dfrac{}{B, C \vdash A, B}\,(I)}{A \leftrightarrow B, B, C \vdash A}\,(\leftrightarrow L) \quad \dfrac{}{B, C \vdash A \leftrightarrow B, A, C}\,(I)}{(A \leftrightarrow B) \leftrightarrow C, B, C \vdash A}\,(\leftrightarrow L)}{(A \leftrightarrow B) \leftrightarrow C, A \oplus B, C \vdash}\,(\oplus L)$$

Subtree 2:

$$\dfrac{\dfrac{\dfrac{}{A \leftrightarrow B, C \vdash A, B, C}\,(I) \quad \dfrac{\dfrac{}{A \vdash A, B, C}\,(I) \quad \dfrac{}{B \vdash A, B, C}\,(I)}{\vdash A \leftrightarrow B, A, B, C}\,(\leftrightarrow R)}{(A \leftrightarrow B) \leftrightarrow C \vdash A, B, C}\,(\leftrightarrow L) \qquad \dfrac{\dfrac{}{A, B, A \leftrightarrow B, C \vdash C}\,(I) \quad \dfrac{\dfrac{}{A, B \vdash A, C}\,(I) \quad \dfrac{}{A, B \vdash B, C}\,(I)}{A, B \vdash A \leftrightarrow, C}\,(\leftrightarrow R)}{A, B, (A \leftrightarrow B) \leftrightarrow C \vdash C}\,(\leftrightarrow L)}{(A \leftrightarrow B) \leftrightarrow C \vdash A \oplus B, C}\,(\oplus R)$$

All branches terminate with the Immediate rule; the entailment is valid.

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`