

PART II D

Version 2.26: 2004/05/14 14:05:36.532 GMT-5

John Greiner
Ian Barland

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License *

Abstract

(Blank Abstract)

part II d

1 Inference

Truth tables and equivalences are useful and powerful tools, but they do not correspond to how we usually reason about things. What we will do now is look at more familiar reasoning and how to formalize that. For example, with Boolean algebra it is awkward to prove that $(a \wedge b)$ implies a . For that, it is necessary to reword the problem in terms of equivalences, as $((a \wedge b) \rightarrow a) \equiv \text{true}$. Our next tool provides a more straightforward way to reason about implications.

Example 1:

Given the following piece of a WaterWorld board, how would you conclude that G is unsafe?

Since H – has – 2, at least two of H 's three neighbors must be unsafe. But, since we know that one of these, J , isn't unsafe, then the two others, including G , must both be unsafe. Let's write this out more explicitly:

1	H – has – 2 would imply one of the following is true: $(P$ – unsafe and G – unsafe), or $(J$ – unsafe and P – unsafe),
2	H – has – 2 is true.
3	One of the following is true: $(P$ – unsafe and G – unsafe), or $(J$ – unsafe and P – unsafe), or $(G$ – unsafe and J – u
4	not J – unsafe
5	$(P$ – unsafe and G – unsafe)
6	G – unsafe

Whew! A lot of small steps are involved in even this small deduction. It's apparent we'd want to automate this as much as possible! Let's look at some other short examples, which we'll formalize in a moment.

* <http://creativecommons.org/licenses/by/1.0>

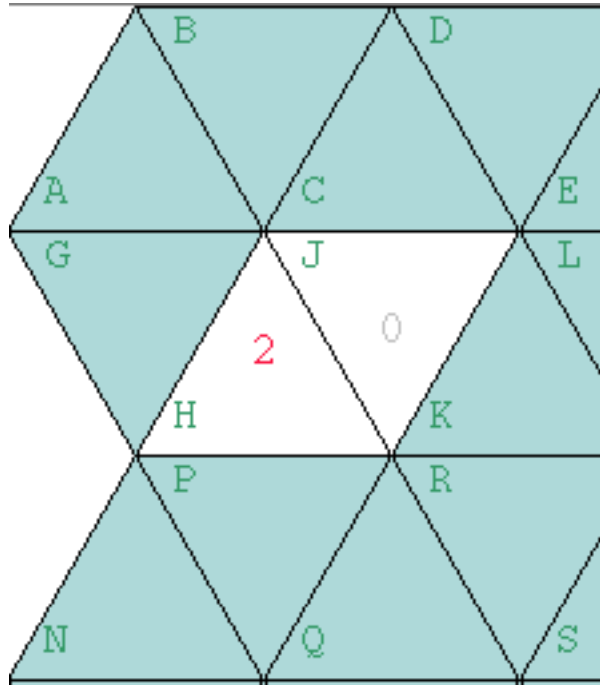


Figure 1: A glimpse of a WaterWorld board

Exercise 1:

How do you know that $A - \text{has} - 2$ proves $B - \text{unsafe}$?

Solution:

Intuitively, this is straightforward. Since $A - \text{has} - 2$, then both of its two neighbors, including B , must be unsafe. For this problem, let's be a bit more formal and use WFFs instead of prose in the steps.

1	$A - \text{has} - 2$	premise
2	$(A - \text{has} - 2 \rightarrow (B - \text{unsafe} \wedge F - \text{unsafe}))$	WaterWorld domain axiom, i.e., definition of $A - \text{has} - 2$
3	$(B - \text{unsafe} \wedge F - \text{unsafe})$	by lines 1,2
4	$B - \text{unsafe}$	by line 3

Exercise 2:

Similarly, how do you reason that $A - \text{has} - 1$ and $G - \text{safe}$ prove $B - \text{unsafe}$?

Solution:

Again a similar idea, if $A - \text{has} - 1$, then at least one of A 's two neighbors must be unsafe. But, since we know that one of these, G isn't unsafe, then the other, B , must be unsafe.

1	$(A - \text{has} - 1 \rightarrow ((B - \text{safe} \wedge G - \text{unsafe}) \vee (B - \text{unsafe} \wedge G - \text{safe})))$	WaterWorld domain axiom
2	$A - \text{has} - 1$	premise
3	$((B - \text{safe} \wedge G - \text{unsafe}) \vee (B - \text{unsafe} \wedge G - \text{safe}))$	by lines 1,2
4	$G - \text{safe}$	premise
5	$(B - \text{unsafe} \wedge G - \text{safe})$	by lines 3,4
6	$B - \text{unsafe}$	by line 5

1.1 Formal inference rules and proofs

In the above examples, we relied on common sense to know what new true formulas could be derived from previous ones. Unfortunately, common sense is imprecise and sometimes wrong. So, we need to formalize how we form proofs.

We now define a **formal proof** of θ from the **premises** ϕ, \dots, ψ , written

$$\phi, \dots, \psi \vdash \theta \quad (1)$$

. (A proof with no premises simply means there is nothing on the left of the turnstile: $\vdash \theta$.) For example, we'll show shortly that $H - \text{has} - 2 \vdash G - \text{unsafe}$. A proof consists of a sequence of WFFs, each with a justification for its truth. We will describe four permissible justifications for each step:

- A premise.
- An axiom.
- An inference rule.
- A subproof.

ASIDE: Officially we might want to use some turnstile subscripted with "ww", to mean "proves within the WaterWorld inference system", indicating our use of the WaterWorld domain axioms. If you're proving things about other domains, you'd use different domain axioms.

Example 2:

We can formalize the above examples to show each of the following:

- $H - \text{has} - 2 \vdash G - \text{unsafe}$
- $A - \text{has} - 2 \vdash B - \text{unsafe}$
- $A - \text{has} - 1, G - \text{safe} \vdash B - \text{unsafe}$

See below for formal proofs of some of these.

Stating an **axiom**, a simple assumed truth, is a rather trivial, boring way of coming up with a true formula. Some axioms are **domain axioms**: they pertain only to the domain you are considering, such as WaterWorld. In our case, we don't have any axioms that aren't domain axioms. (Another domain is arithmetic, with axioms about how multiplication distributes over addition, etc.)

Just using axioms is not enough, however. The interesting part is to deduce new true formulas from axioms and the results of previous deductions. (Bertrand Russell said, "The point of philosophy is to start with something so simple as not to seem worth stating, and

to end with something so paradoxical that no one will believe it.”) An **inference rule** formalizes what steps are allowed in proofs. We’ll use this list of valid inference rules¹ (.ps)² as our definition, but, this is just one set of possible inference rules, and other people could use slightly different ones.

First, let’s look at some simple examples, using the simpler inference rules.

Example 3:

We’ll formalize a previous exercise (Exercise 1) to show $A - \text{has} - 2 \vdash B - \text{unsafe}$.

1	$A - \text{has} - 2$	premise
2	$(A - \text{has} - 2 \rightarrow (B - \text{unsafe} \wedge F - \text{unsafe}))$	WaterWorld domain axiom
3	$(B - \text{unsafe} \wedge F - \text{unsafe})$	\rightarrow Elim, by lines 1,2, where $\phi=A - \text{has} - 2$, and $\psi=(B - \text{unsafe} \wedge F - \text{unsafe})$
4	$B - \text{unsafe}$	\wedge Elim (left), by line 3, where $\phi=B - \text{unsafe}$, and $\psi=F - \text{unsafe}$

That’s almost exactly like the steps we took in the previous informal proof, but now we’re a bit pickier about our justifications for each step.

Formally, when using a domain axiom, the justification is a combination of the name of that inference rule, the line numbers of which previous WFFs are being used, and a description of how those WFFs are used in that inference rule in this particular step. Later, we’ll often omit the description of exactly how the specific inference rule is used, since in many cases, that information is painfully obvious.

Example 4:

In this system, commutativity of \wedge and \vee are not among the inference rules. However, they do follow. For example, consider the following proof of $(A \wedge B) \vdash (B \wedge A)$

1	$(A \wedge B)$	premise
2	A	\wedge Elim (left), by line 1, where $\phi=A$
3	B	\wedge Elim (right), by line 1, where $\psi=B$
4	$(B \wedge A)$	\wedge Intro, by lines 3,2, where $\phi=B$ and $\psi=A$

Does this example (Example 4) also show that $(C \wedge D) \vdash (D \wedge C)$? Well, yes and no. That proof does *not* have anything to do with propositions C and D . But, clearly, we could create another nearly identical proof for $(C \wedge D) \vdash (D \wedge C)$, by substituting C and D for A and B , respectively. What about proving the other direction of commutativity: $(B \wedge A) \vdash (A \wedge B)$? Once again, the proof has exactly the same form, but substituting B and A for A and B , respectively. Stating such similar proofs over and over is technically necessary, but not very interesting. Instead, when the proof depends solely on the form of the formula and not on any axioms, we’ll use meta-variables to generalize.

Example 5:

Generalized \wedge commutativity. $(\chi \wedge \psi) \vdash (\psi \wedge \chi)$

¹<http://cnx.rice.edu/content/m10529/latest/>

²<http://www.teachLogic.org/Base/Printables/inference-rules.ps>

1	$(\chi \wedge v)$	premise
2	v	\wedge Elim (left), by line 1, where $\phi=\chi$
3	χ	\wedge Elim (right), by line 1, where $\psi=v$
4	$(\chi \wedge v)$	\wedge Intro, by lines 3,2, where $\phi=v$ and $\psi=\chi$

Exercise 3:

Similarly, associativity of \wedge and \vee are not among the inference rules. This is a particularly important detail, since our WaterWorld domain axioms frequently use formulas of the form $(a \wedge b \wedge c)$, which isn't technically legal according to our definition of WFFs. What we'd like to show is that $((\chi \wedge v) \wedge \omega) \vdash (\chi \wedge (v \wedge \omega))$ and $(\chi \wedge (v \wedge \omega)) \vdash ((\chi \wedge v) \wedge \omega)$ as well as the equivalent for \vee . Thus, when we see three, four, or more terms in a conjunction (or disjunction), we can legitimately group them as we see fit.

Solution:

Here, we'll show only $((\chi \wedge v) \wedge \omega) \vdash (\chi \wedge (v \wedge \omega))$ and leave the other direction (and \vee 's associativity) to the reader. These are all very similar to the previous commutativity example (Example 4).

1	$((\chi \wedge v) \wedge \omega)$	premise
2	$(\chi \wedge v)$	\wedge Elim (left), by line 1
3	χ	\wedge Elim (left), by line 2
4	v	\wedge Elim (right), by line 2
5	ω	\wedge Elim (right), by line 1
6	$(v \wedge \omega)$	\wedge Intro, by lines 4,5
7	$(\chi \wedge (v \wedge \omega))$	\wedge Intro, by lines 3,6

Note that we omitted the detailed explanation of how each rule applies, since this should be clear in each of these steps.

These deductions are straightforward and should be unsurprising, but perhaps not too interesting. These simple rules can carry us far and will be used commonly in other examples.

There is a subtle difference between implication (\rightarrow) and provability (\vdash). Both embody the idea that the truth of the right-hand-side follows from the left-hand-side. But, \rightarrow is a syntactic formula connective combining two WFFs into a larger WFF, while \vdash combines a list of propositions and a WFF into a statement about provability.

Exercise 4:

Show that, $\phi \vdash \psi$ is equivalent to $\vdash (\phi \rightarrow \psi)$ in that, we can show one if and only if we can show the other.

Solution:

First, if we know $\phi \vdash \psi$, then that means there is some written \vdash proof... we know $\vdash (\phi \rightarrow \psi)$, simply by \rightarrow Intro.

If we know $\vdash (\phi \rightarrow \psi)$, then if we add a premise ϕ , then ψ follows by \rightarrow Elim.

Note how this proof is about other proofs! (However, while we reason about this particular inference system, we're not using this system while proving things about it – this proof is necessarily outside the inference system.)

1.1.1 Subproofs

The reductio ad absurdum (RAA) rule, which in English means "reduction to absurdity", seems very strange. If we can prove that false is true, then we can prove the negation of our premise. Huh!?! What does it mean to prove false is true?

This is known as **proof-by-contradiction**. We start by making a single unproven assumption. We then try to prove that false is true. Clearly, that it nonsense, so we must have done something wrong. Assuming we didn't make any mistakes in the individual inference steps, then the only thing that could be wrong is the assumption. It must not hold. Therefore, we have just proven its negation.

This form of reasoning is often expressed via contrapositive. Consider the slogan "If you paid list price, you didn't buy it at SuperMegaMart." (We consider this a contrapositive, because the real fact the advertisers want to say is that if you buy it at SuperMegaMart, then you won't pay list price.), which we'll abbreviate ($\text{payFull} \rightarrow \neg\text{boughtAtSMM}$). You know this slogan is true, and you just made a SuperMegaMart purchase (boughtAtSMM), and are suddenly wanting a *proof* that you got a good deal. Well, suppose we didn't – that is, *suppose* payFull . Then (by the truth of the marketing slogan), we infer $\neg\text{boughtAtSMM}$. But this contradicts boughtAtSMM (that is, from $\neg\text{boughtAtSMM}$ and boughtAtSMM together we can prove that false is true). The problem must have been our pessimised assumption payFull ; clearly that couldn't have been true, and we're happy to know that $\neg\text{payFull}$.

Example 6:

Spot the proof-by-contradiction used in The Simpsons:

Bart, filing through the school records: "Hey, look at this – Skinner makes \$25,000 per year!"

Other kids: "Ooooh!"

Milhouse: "And he's 40 years old – that makes him a millionaire!"

Skinner, indignantly: "I wasn't a principal when I was 1!"

Milhouse: " *And*, he paints houses during the summer ... he's a billionaire!"

Skinner: "If I were a billionaire, would I still be living with my mother?" [Kids' laughter]

Skinner, to himself: "The kids just aren't responding to logic anymore!"

In the particular set of inference rules we have chosen to use, RAA is surprisingly important. It is one of the few ways to prove formulas of the form $\vdash \neg\phi$.

Example 7:

We'll prove $\vdash \neg(\chi \wedge \neg\chi)$.

1	subproof: $(\chi \wedge \neg\chi) \vdash \text{false}$		
1.a		$(\chi \wedge \neg\chi)$	premise for subproof
1.b		χ	$\wedge\text{Elim}$ (left), by line 1.1, where $\phi=\chi$, and $\psi=\neg\chi$
1.c		$\neg\chi$	$\wedge\text{Elim}$ (right), by line 1.1, where $\phi=\chi$, and $\psi=\neg\chi$
1.d		false	falseIntro, by lines 1.2,1.3, where $\phi=\chi$
2	$\neg(\chi \wedge \neg\chi)$		RAA, by line 1, where $\phi=(\chi \wedge \neg\chi)$

Exercise 5:

Here's another relatively simple example which uses RAA. Show that the **modus tollens** rule holds: $(\chi \rightarrow v), \neg v \vdash \neg\chi$

Solution:

1	$(\chi \rightarrow v)$		premise
2	$\neg v$		premise
3	subproof: $\chi \vdash \text{false}$		
3.a		χ	premise for subproof
3.b		v	\rightarrow Elim, by lines 1,3a
3.c		false	falseIntro, by lines 2,3b
4	$\neg\chi$		RAA, by line 3

Another use of subproofs is to organize proofs' presentations. Many proofs naturally break down into larger subparts, each with its own intermediate conclusion. These steps between these subparts are big enough to correspond to our intuition, but too big to correspond to individual inference rules. This gives additional useful structure to a proof, aiding our understanding.

Example 8:

Previously, we showed that \wedge commutes (Example 4). However, that conclusion is only directly applicable when the \wedge is at the "top-level", i.e., not nested inside some other connective. Here, we'll show that \wedge commutes inside \neg , or more formally, $\neg(\chi \wedge v) \vdash \neg(v \wedge \chi)$.

Our plan is as follows:

1. Assume the premise $\neg(\chi \wedge v)$.
2. Use commutativity to show that $((v \wedge \chi) \rightarrow (\chi \wedge v))$
3. Use modus tollens (pg ??) to obtain the conclusion.

We can organize the proof into corresponding subparts:

1	$\neg(\chi \wedge v)$		premise
2	subproof: $((v \wedge \chi) \rightarrow (\chi \wedge v))$		
2.a		$(v \wedge \chi) \vdash (\chi \wedge v)$	Previously proved \wedge comm
2.b		$((v \wedge \chi) \rightarrow (\chi \wedge v))$	\rightarrow Intro, by line 2.a
3	subproof: $\neg(v \wedge \chi)$		
3.a		$((v \wedge \chi) \rightarrow (\chi \wedge v)), \neg(\chi \wedge v) \vdash \neg(v \wedge \chi)$	Previously proved modus
3.b		$((((v \wedge \chi) \rightarrow (\chi \wedge v)) \wedge \neg(\chi \wedge v)) \rightarrow \neg(v \wedge \chi))$	\rightarrow Intro, by line 3a
3.c		$((v \wedge \chi) \rightarrow (\chi \wedge v)) \wedge \neg(\chi \wedge v)$	\wedge Intro, by lines 2,1
3.d		$\neg(v \wedge \chi)$	\rightarrow Elim, by lines 3b,3c

1.1.2 More examples

Now let's use these rules in a couple larger proofs, to show some more interesting results.

Example 9:

Let's redo the first example (Example 1)'s proof formally and show $(H - \text{has} - 2 \wedge J - \text{safe}) \vdash G - \text{unsafe}$. The inference rules we used informally above don't correspond exactly to those in our definition, so the formal proof is more complicated.

1	$(H - \text{has} - 2 \wedge J - \text{safe})$	
2	$H - \text{has} - 2$	
3	$J - \text{safe}$	
4	$(J - \text{safe} \rightarrow \neg J - \text{unsafe})$	
5	$\neg J - \text{unsafe}$	
6	$(H - \text{has} - 2 \rightarrow ((P - \text{unsafe} \wedge G - \text{unsafe}) \vee ((J - \text{unsafe} \wedge P - \text{unsafe}) \vee (G - \text{unsafe} \wedge J - \text{unsafe}))))$	
7	$((P - \text{unsafe} \wedge G - \text{unsafe}) \vee ((J - \text{unsafe} \wedge P - \text{unsafe}) \vee (G - \text{unsafe} \wedge J - \text{unsafe})))$	
8	subproof: $(J - \text{unsafe} \wedge P - \text{unsafe}) \vdash \text{false}$	
8.a		(J
8.b		J -
8.c		fal
9	$\neg (J - \text{unsafe} \wedge P - \text{unsafe})$	
10	subproof: $(G - \text{unsafe} \wedge J - \text{unsafe}) \vdash \text{false}$	
10.a		(G
10.b		J -
10.c		fal
11	$\neg (G - \text{unsafe} \wedge J - \text{unsafe})$	
12	$\neg ((J - \text{unsafe} \wedge P - \text{unsafe}) \vee (G - \text{unsafe} \wedge J - \text{unsafe}))$	
13	$(P - \text{unsafe} \wedge G - \text{unsafe})$	
14	$G - \text{unsafe}$	

Wow! This formalization is a lot longer than the original informal proof. That's a result of the particular set of inference rules we are using, that we can only make inferences in small steps. Also, here we were pickier about the distinction between "not safe" and "unsafe".

Example 10:

The previous example (Example 3) is a perfect candidate for adding structure to the proof by using additional subproofs.

A standard way of presenting proofs is by using **lemmas** to show parts of the proofs. Lemmas are simply formulas that typically aren't viewed as end results, but as intermediate steps in a larger proof. So, they are simply another way of presenting subproofs.

Example 11:

Consider the above figure (Figure 2). We'll show $(B - \text{has} - 1 \wedge (G - \text{has} - 1 \wedge J - \text{has} - 1)) \vdash K - \text{unsafe}$. We'll do this through the following series of lemmas:

- Lemma A: $\neg A - \text{unsafe}, G - \text{has} - 1 \vdash H - \text{unsafe}$
- Lemma B: $\neg A - \text{unsafe}, B - \text{has} - 1 \vdash C - \text{unsafe}$
- Lemma C: $H - \text{unsafe}, C - \text{unsafe}, J - \text{has} - 1 \vdash \text{false}$
- Lemma D: $A - \text{unsafe}, B - \text{has} - 1 \vdash C - \text{safe}$
- Lemma E: $A - \text{unsafe}, G - \text{has} - 1 \vdash H - \text{safe}$
- Lemma F: $C - \text{safe}, H - \text{safe}, J - \text{has} - 1 \vdash K - \text{unsafe}$

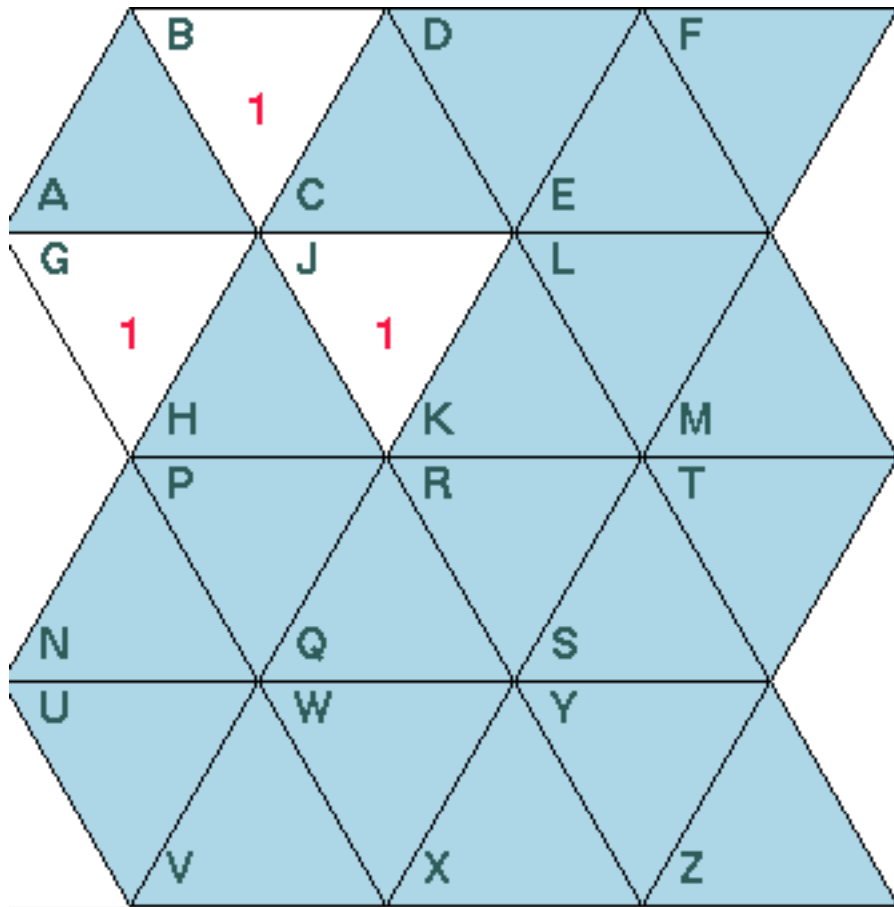


Figure 2: Example WaterWorld board

First, we'll show the main proof, assuming each of the lemmas. Then, proofs of each of the lemmas will follow.

1	$(B - \text{has} - 1 \wedge (G - \text{has} - 1 \wedge J - \text{has} - 1))$		premise
2	$B - \text{has} - 1$		\wedge Elim (left), by line 1
3	$(G - \text{has} - 1 \wedge J - \text{has} - 1)$		\wedge Elim (right), by line 1
4	$G - \text{has} - 1$		\wedge Elim (left), by line 3
5	$J - \text{has} - 1$		\wedge Elim (right), by line 3
6	subproof: $\neg A - \text{unsafe} \vdash \text{false}$		
6.a		$\neg A - \text{unsafe}$	premise for subproof
6.b		$H - \text{unsafe}$	Lemma A, by lines 6.1,4
6.c		$C - \text{unsafe}$	Lemma B, by lines 6.1,3
6.d		false	Lemma C, by lines 6.2,6.3,5
7	$A - \text{unsafe}$		RAA, by line 6
8	$C - \text{safe}$		Lemma D, by lines 7,2
9	$H - \text{safe}$		Lemma E, by lines 7,3
10	$K - \text{unsafe}$		Lemma F, by lines 8,9,5

And that's the desired proof! Now it just remains to show each of the six lemmas.

Lemma A: $\neg A - \text{unsafe} , G - \text{has} - 1 \vdash H - \text{unsafe}$

1	$\neg A - \text{unsafe}$		premise
2	$G - \text{has} - 1$		premise
3	subproof: $(A - \text{unsafe} \wedge H - \text{safe}) \vdash \text{false}$		
3.a		$(A - \text{unsafe} \wedge H - \text{safe})$	premise for
3.b		$A - \text{unsafe}$	\wedge Elim
3.c		false	falseIntro, b
4	$\neg (A - \text{unsafe} \wedge H - \text{safe})$		RAA, by li
5	5		
6	$(G - \text{has} - 1 \rightarrow ((A - \text{safe} \wedge H - \text{unsafe}) \vee (A - \text{unsafe} \wedge H - \text{safe})))$		WaterWorld
7	6		
8	$((A - \text{safe} \wedge H - \text{unsafe}) \vee (A - \text{unsafe} \wedge H - \text{safe}))$		\rightarrow Elim, by
9	$((A - \text{unsafe} \wedge H - \text{safe}) \vee (A - \text{safe} \wedge H - \text{unsafe}))$		\vee commute
10	$(A - \text{safe} \wedge H - \text{unsafe})$		CaseElim, b
11	$H - \text{unsafe}$		\wedge Elim (righ

Lemma B: $\neg A - \text{unsafe} , B - \text{has} - 1 \vdash H - \text{unsafe}$

1	$\neg A - \text{unsafe}$		premise
2	$B - \text{has} - 1$		premise
3	subproof: $(A - \text{unsafe} \wedge C - \text{safe}) \vdash \text{false}$		
3.a		$(A - \text{unsafe} \wedge C - \text{safe})$	premise for
3.b		$A - \text{unsafe}$	\wedge Elim (left)
3.c		false	falseIntro, b
4	$\neg(A - \text{unsafe} \wedge C - \text{safe})$		RAA, by lin
5	$(B - \text{has} - 1 \rightarrow ((A - \text{safe} \wedge C - \text{unsafe}) \vee (A - \text{unsafe} \wedge C - \text{safe})))$		WaterWorld
6	$((A - \text{safe} \wedge C - \text{unsafe}) \vee (A - \text{unsafe} \wedge C - \text{safe}))$		\rightarrow Elim, by
7	$((A - \text{unsafe} \wedge C - \text{safe}) \vee (A - \text{safe} \wedge C - \text{unsafe}))$		\vee commutes
8	$(A - \text{safe} \wedge C - \text{unsafe})$		CaseElim, b
9	$C - \text{unsafe}$		\wedge Elim (righ

Proof of the other lemmas are left as an exercise to the reader.

Note that we took a little shortcut: we used the lemmas as if they were inference rules. According to our previous definition of proofs, we technically should present the lemma as a subproof and then use an inference rule or two to show how that applies, as we've done in previous examples. This shorter form is common practice and a little easier to read.

In summary, we must state one of the following four possible reasons for each step in a proof, allowing subproofs.

- This step's WFF is a premise.
- This step's WFF is an axiom.
- This step's WFF follows from a inference rule applied to previous steps' WFFs. The reason includes a statement of which inference rule is used and how.
- This step's WFF follows from a subproof, where that subproof may temporarily introduce additional premises. The reason includes the entire subproof. When that subproof has been shown elsewhere, such as in class or another exercise, it may simply be cited, for brevity. Of course, subproofs may have additional embedded subproofs, in turn.

Technically, when using subproofs, one must be careful to rename variables, to avoid clashes. Rather than formalize this notion, we'll leave it as "obvious".

Throughout this discussion, we've implicitly assumed that if we've proven something, it must be true. But we should be careful: What if one of those listed inference rule isn't always valid? What if we introduced a new rule – You'd probably balk if we proposed introduction $((a \vee b) \rightarrow a)$, or even more degenerately false. But what about some more reasonable-sounding rule – how should we satisfy that our new rule won't introduce a consistency, when taken with the others? This brings us back to the questions of soundness and completeness of a proof system. Fortunately, the system presented here is both sound and complete (though proving this is beyond our current scope). However, we can rest assured, that for propositional logic, what we can prove really does correspond entirely to what is *true*.

Exercise 6:

If we omitted the RAA inference rule, would this new system be sound? Complete?

Solution:

It would be sound. We've already claimed (without proof) that the system with RAA is sound, and thus, that all proved formulas are indeed true. By omitting RAA, we cannot prove any additional formulas, so all those provable without RAA are still indeed true.

As pointed out, RAA is our primary way to prove negations without premises. There are negated formulas that are true, including the simple $\neg\text{false}$, that we cannot prove without RAA. So, without RAA, we cannot prove everything that is true. Thus, that restricted system is **incomplete**.

2 Proofs and programming

Proofs are organized a lot like programs. Based on some premises (inputs), we obtain some conclusion (output) after using a series of inference rules (basic computation like addition and other operations). Using subproofs, especially when citing previous proofs, is just like organizing our program into functions that can be used many times.

Naturally, since using inference rules is not only how people prove things, but also computers. A clear example is in type checking. The core idea of type checking a function application is "If function f takes an argument of type α and producing an output of type β , and expression exp is of type α , then $f(\text{exp})$ is of type β ." This type rule closely resembles $\rightarrow\text{Elim}$: "If a proven formula is $(a \rightarrow b)$ and other proven formula is a , then together, b is a proven formula." Furthermore, this similarity is highlighted by notation in many programming languages which would write the type of f as $\alpha \rightarrow \beta$. Type rules are simply inference rules for proving results about the types of programs, and in most typical programming languages these rules closely correspond to those we are using for logic. This correspondence is known as the Curry-Howard Isomorphism.

As with logic, we want type checkers to be sound and complete. Soundness here means that if the program passes type checking, when we execute the program (or single function) and get a value, that value is of the stated type. In other words, if our program type checks, then we are guaranteed that some kinds of errors will not happen at run-time. That also means that if our program would have a run-time type error, the type checker will correctly report that our program is erroneous. Completeness here means that if we execute the program (or single function) and get a value of a certain type, then our type checker indeed tells us that type.

Note that type checking is still an area of active research, since their job is made difficult in the presence of language features inheritance, multiple inheritance, dynamic class loading, etc. When people introduce new computer languages with new features, and want to claim that their new language is **type safe** (that no function ever will be applied to the wrong type at run-time), then the paper which introduces the language will contain such a proof.

3 Are we done yet?

Even these inference rules seem awkward, and you may have some more general ones in mind. Later, we'll see more interesting inference rules, such as Induction, which will give us a richer set of proofs. In general, a hard problem is finding a language that is both expressive enough to describe the domain succinctly, but also limited enough to automate reasoning. This is a very practical issue in type checking and other program analysis. While it can be easy to find program some errors automatically, it is very difficult or impossible to *guarantee* that you can find *all* errors (of some specific kind, like type errors).

One thing we would like to eliminate is the need (at least technically) to restate structurally identical proofs, as discussed for commutativity (Example 4). We will be able to add idea of generalizing such proofs directly into the logic and inference rules.

Despite the desire for more flexible reasoning, we'd also like to consider whether we have more inference rules than is necessary. Are some of them redundant? This is similar to the software rule that we should have a single point of control, or the similar idea that libraries should provide exactly one way of doing something. In general, this is not easy to ensure. We have shown that some potential additional inference rules, like commutativity and associativity, weren't necessary. But we haven't shown our core inference rules to be minimal. What do you think?

4 Distinctness of the approaches (optional)

You might be wondering: *When I'm trying to prove something using Propositional Inference Rules, can I use Propositional Logic Equivalences as Axioms like how I can use the WaterWorld Axioms?* No – the axioms are just individual formulas. The boolean-equivalences are always *pairs* of formulas, where you can re-write one from the other. (I think, you're wanting to turn each equivalence $\phi \equiv \psi$ into a rule "if you know ϕ , you can conclude ψ " and vice-versa. That technically doesn't cover re-writing little nested sub-parts of a big formula though, which is not part of the inference-rule approach. You could indeed create a new combo-system that involved both of these; in this class we're taking them one at a time, though.)

...it seems like propositional logic equivalences can just be considered a tool used by inference rules to proof things. For the inference-rule proofs, your justifications can be (a) premise, (b) domain-axiom [for waterworld], (c) a rule-of-inference, or (d) a subproof. But not an equivalence of the form " $\phi \equiv \psi$ ".

If not, those inference rules seem so limited in what it can prove. For whatever system we have, definitely the waterworld axioms are needed for reasoning specific to waterworld. Without it, you can only deal with statements true in **any** truth-setting (or, in any interpretation).

Proving things about waterworld with the boolean-equivalences or truth-tables is unwieldy – you don't try to show that (say) " $(A - \text{has} - 2 \rightarrow B - \text{unsafe})$ " is a tautology; rather you try to show that " $((\rho \wedge A - \text{has} - 2) \rightarrow B - \text{unsafe})$ ", where ρ is one big huge conjunction of *all* the rules (domain-axioms). More generally, you'd always be looking to show a big formula of the form " $((\rho \wedge [\text{Premises}]) \rightarrow [\text{desired-fact}])$ " as a tautology. (This is mentioned only briefly in the notes; I'm not stressing the use of equivalences to make waterworld deductions.)

You're right, that both inference-systems and boolean-equivalence-rewriting share similar approaches, and can be blended. In actual real-world reasoning systems, yes a combination of all three approaches might be used, but for this class we're treating them all separately (and discussing the completeness and soundness of each one separately).