1

Propositional logic

EXERCISES 1.1 (p.78)

1(a). We chose p to stand for "The sun shines today." and q denotes "The sun shines tomorrow." The corresponding formula is then

$$p \to \neg q\,.$$

NB: If we had chosen q to denote "The sun does not shine tomorrow." then the corresponding formula would be

$$p \to q\ .$$

1(d). We choose

p : "A request occurs."

q : "The request will eventually be acknowledged."

r : "The requesting process will eventually make progress."

The formula representing the declarative sentence is then

$$p \to (q \vee \neg r)\,.$$

NB: If we had chosen r to denote "The requesting process won't ever be able to make progress," the corresponding formula would be

$$p \to (q \vee r)\ .$$

1(h). We choose

r : "Today, it will shine."

s : "Today, it will rain."

The resulting formula is $(r \vee s) \wedge \neg(r \wedge s)$.

1(i). The proposition atoms we chose are

p : "Dick met Jane yesterday."

q : "Dick and Jane had a cup of coffee together."

r : "Dick and Jane had a walk in the park."

This results in the formula

$$p \to q \vee r$$

which reads as $p \to (q \vee r)$ if we recall the binding priorities of our logical operators.

2(a). $((\neg p) \wedge q) \to r$.

2(c). $(p \to q) \to (r \to (s \vee t))$.

2(e). $(p \vee q) \to ((\neg p) \wedge r)$.

2(g). The expression $p \vee q \wedge r$ is problematic since $\wedge$ and $\vee$ have the same binding priorities, so we have to insist on additional brackets in order to resolve this conflict.

EXERCISES 1.2 (p.78)

1(a). (Bonus) We prove the validity of $(p \wedge q) \wedge r,\ s \wedge t \vdash q \wedge s$ by

1	$(p \wedge q) \wedge r$	premise
2	$s \wedge t$	premise
3	$p \wedge q$	$\wedge e_1$ 1
4	q	$\wedge e_2$ 3
5	s	$\wedge e_1$ 2
6	$q \wedge s$	$\wedge i$ 4, 5

1(c) We prove the validity of $(p \wedge q) \wedge r \vdash p \wedge (q \wedge r)$ by

1	$(p \wedge q) \wedge r$	premise
2	$p \wedge q$	$\wedge e_1$ 1
3	r	$\wedge e_2$ 1
4	p	$\wedge e_1$ 2
5	q	$\wedge e_2$ 2
6	$q \wedge r$	$\wedge i$ 5, 3
7	$p \wedge (q \wedge r)$	$\wedge i$ 4, 6

1(e). One possible proof of the validity of $q \to (p \to r), \neg r, q \vdash \neg p$ is

1	$q \to (p \to r)$	premise
2	$\neg r$	premise
3	q	premise
4	$p \to r$	$\to$e $1, 3$
5	$\neg p$	MT $4, 2$

1(f). We prove the validity of $\vdash (p \wedge q) \to p$ by

1	$p \wedge q$	assumption
2	p	$\wedge e_1$ 1
3	$p \wedge q \to p$	$\to$i $1 - 2$

1(h). We prove the validity of $p \vdash (p \to q) \to q$ by

1	p	premise
2	$p \to q$	assumption
3	q	$\to$e $2, 1$
4	$(p \to q) \to q$	$\to$i $2 - 3$

1(i). We prove the validity of $(p \to r) \wedge (q \to r) \vdash p \wedge q \to r$ by

1	$(p \to r) \wedge (q \to r)$	premise
2	$p \wedge q$	assumption
3	p	$\wedge e_1$ 2
4	$p \to r$	$\wedge e_1$ 1
5	r	$\to$e $4, 3$
6	$p \wedge q \to r$	$\to$i $2 - 5$

1(j). We prove the validity of $q \to r \vdash (p \to q) \to (p \to r)$ by

1	$q \to r$	premise
2	$p \to q$	assumption
3	p	assumption
4	q	$\to$e $2, 3$
5	r	$\to$e $1, 4$
6	$p \to r$	$\to$i $3 - 5$
7	$(p \to q) \to (p \to r)$	$\to$i $2 - 6$

1(l). We prove the validity of $p \to q, r \to s \vdash p \vee r \to q \vee s$ by

1	$p \to q$	premise
2	$r \to s$	premise
3	$p \vee r$	assumption
4	p	assumption
5	q	$\to$e $1, 4$
6	$q \vee s$	$\vee$i$_1$ 5
7	r	assumption
8	s	$\to$e $2, 7$
9	$q \vee s$	$\vee$i$_2$ 8
10	$q \vee s$	$\vee$e $3, 4 - 6, 7 - 9$
11	$p \vee r \to q \vee s$	$\to$i $3 - 10$

1(n) We prove the validity of $(p \vee (q \to p)) \wedge q \vdash p$ by

1	$(p \vee (q \to p)) \wedge q$	premise
2	q	$\wedge$e$_2$ 1
3	$p \vee (q \to p)$	$\wedge$e$_1$ 1
4	p	assumption
5	$q \to p$	assumption
6	p	$\to$e $5, 2$
7	p	$\vee$e $3, 4 - 4, 5 - 6$

Note that one could have put line 2 in between lines 5 and 6 with

the corresponding renumbering of pointers. Would the proof above still be valid if we used rules $\wedge e_2$ and $\wedge e_1$ in the other ordering?

1(o). We prove the validity of $p \to q, r \to s \vdash p \wedge r \to q \wedge s$ by

1	$p \to q$	premise
2	$r \to s$	premise
3	$p \wedge r$	assumption
4	p	$\wedge e_1$ 3
5	q	$\to e$ 1, 4
6	r	$\wedge e_2$ 3
7	s	$\to e$ 2, 6
8	$q \wedge s$	$\wedge i$ 5, 7
9	$p \wedge r \to q \wedge s$	$\to i$ 3 – 8

1(r). We prove the validity of $p \to q \wedge r \vdash (p \to q) \wedge (p \to r)$ by

1	$p \to q \wedge r$	premise
2	p	assumption
3	$q \wedge r$	$\to e$ 1, 2
4	q	$\wedge e_1$ 3
5	$p \to q$	$\to i$ 2 – 4
6	p	assumption
7	$q \wedge r$	$\to e$ 1, 6
8	r	$\wedge e_2$ 7
9	$p \to r$	$\to i$ 6 – 8
10	$(p \to q) \wedge (p \to r)$	$\wedge i$ 5, 9

The reader may wonder why two separate, although almost identical, arguments have to be given. Our proof rules force this structure (= assuming p *twice*) upon us. If we proved q and r in the same box, then we would be able to show $p \to q \wedge r$ which is our premise and not what we are after.

1(v). We prove the validity of $p \vee (p \wedge q) \vdash p$ by

1	$p \vee (p \wedge q)$	premise
2	p	assumption
3	$p \wedge q$	assumption
4	p	$\wedge e_1$ 3
5	p	$\vee e$ $1, 2-2, 3-4$

1(x). We prove the validity of $p \to (q \vee r), q \to s, r \to s \vdash p \to s$ by

1	$p \to (q \vee r)$	premise
2	$q \to s$	premise
3	$r \to s$	premise
4	p	assumption
5	$q \vee r$	$\to e$ $1, 4$
6	q	assumption
7	s	$\to e$ $2, 6$
8	r	assumption
9	s	$\to e$ $3, 8$
10	s	$\vee e$ $5, 6-7, 8-9$
11	$p \to s$	$\to i$ $4-10$

1(y). We prove the validity of $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$ by

1	$(p \wedge q) \vee (p \wedge r)$	premise
2	$p \wedge q$	assumption
3	p	$\wedge e_1$ 2
4	q	$\wedge e_2$ 2
5	$q \vee r$	$\vee i_1$ 4
6	$p \wedge (q \vee r)$	$\wedge$i 3, 5
7	$p \wedge r$	assumption
8	p	$\wedge e_1$ 7
9	r	$\wedge e_2$ 7
10	$q \vee r$	$\vee i_2$ 9
11	$p \wedge (q \vee r)$	$\wedge$i 8, 10
12	$p \wedge (q \vee r)$	$\vee$e $1, 2-6, 7-11$

2(a). We prove the validity of $\neg p \to \neg q \vdash q \to p$ by

1	$\neg p \to \neg q$	premise
2	q	assumption
3	$\neg\neg q$	$\neg\neg$i 2
4	$\neg\neg p$	MT 1, 3
5	p	$\neg\neg$e 4
6	$q \to p$	$\to$i $2-5$

2(b). We prove the validity of $\neg p \vee \neg q \vdash \neg(p \wedge q)$ by

1	$\neg p \vee \neg q$	premise
2	$\neg p$	assumption
3	$p \wedge q$	assumption
4	p	$\wedge e_1$ 3
5	$\bot$	$\neg$e 4, 2
6	$\neg(p \wedge q)$	$\neg$i 3 – 5
7	$\neg q$	assumption
8	$p \wedge q$	assumption
9	q	$\wedge e_2$ 8
10	$\bot$	$\neg$e 9, 7
11	$\neg(p \wedge q)$	$\neg$i 8 – 10
12	$\neg(p \wedge q)$	$\vee$e 1, 2 – 6, 7 – 11

2(c). We prove the validity of $\neg p, p \vee q \vdash q$ by

1	$\neg p$	premise
2	$p \vee q$	premise
3	p	assumption
4	$\bot$	$\neg$e 3, 1
5	q	$\bot$e 4
6	q	assumption
7	q	$\vee$e 2, 3 – 5, 6 – 6

2(d). We prove the validity of $p \vee q, \neg q \vee r \vdash p \vee r$ by

1	$p \vee q$	premise
2	$\neg q \vee r$	premise
3	$\neg q$	assumption
4	$p \vee q$	copy 1
5	p	assumption
6	$p \vee r$	$\vee$i$_1$ 5
7	q	assumption
8	$\bot$	$\neg$e 7, 3
9	$p \vee r$	$\bot$e 8
10	$p \vee r$	$\vee$e $4, 5-6, 7-9$
11	r	assumption
12	$p \vee r$	$\vee$i$_2$ 11
13	$p \vee r$	$\vee$e $2, 3-10, 11-12$

Observe how the format of the proof rule $\vee$e forces us to use the copy rule if we nest two disjunctions as premises we want to eliminate.

2(e). We prove the validity of $p \to (q \vee r), \neg q, \neg r \vdash \neg p$ by

1	$p \to (q \vee r)$	premise
2	$\neg q$	premise
3	$\neg r$	premise
4	p	assumption
5	$q \vee r$	$\to$e $1, 4$
6	q	assumption
7	$\bot$	$\neg$e $6, 2$
8	r	assumption
9	$\bot$	$\neg$e $8, 3$
10	$\bot$	$\vee$e $5, 6-7, 8-9$
11	$\neg p$	$\neg$i $4-10$

2(f). We prove the validity of $\neg p \wedge \neg q \vdash \neg(p \vee q)$ by

1	$\neg p \wedge \neg q$	premise
2	$p \vee q$	assumption
3	p	assumption
4	$\neg p$	$\wedge e_1$ 1
5	$\bot$	$\neg e$ 3, 4
6	q	assumption
7	$\neg q$	$\wedge e_2$ 1
8	$\bot$	$\neg e$ 6, 7
9	$\bot$	$\vee e$ 2, 3 – 5, 6 – 8
10	$\neg(p \vee q)$	$\neg i$ 2 – 9

2(g). We prove the validity of $p \wedge \neg p \vdash \neg(r \to q) \wedge (r \to q)$ by

1	$p \wedge \neg p$	premise
2	p	$\wedge e_1$ 1
3	$\neg p$	$\wedge e_2$ 1
4	$\bot$	$\neg e$ 2, 3
5	$\neg(r \to q) \wedge (r \to q)$	$\bot e$ 4

2(i). We prove the validity of $\neg(\neg p \vee q) \vdash p$ by

1	$\neg(\neg p \vee q)$	premise
2	$\neg p$	assumption
3	$\neg p \vee q$	$\vee i_1$ 2
4	$\bot$	$\neg e$ 3, 1
5	$\neg\neg p$	$\neg i$ 2 – 4
6	p	$\neg\neg e$ 5

Note that lines 5 and 6 could be compressed to one line with the application of RAA.

3(d). We prove the validity of $\vdash \neg p \to (p \to (p \to q))$ by

1	$\neg p$	assumption
2	p	assumption
3	p	assumption
4	$\bot$	$\neg$e $3, 1$
5	q	$\bot$e 4
6	$p \to q$	$\to$i $3 - 5$
7	$p \to (p \to q)$	$\to$i $2 - 6$
8	$\neg p \to (p \to (p \to q))$	$\to$i $1 - 7$

Note that all three assumptions/boxes are required by the format of our $\to$i rule.

3(n). We prove the validity of $p \wedge q \vdash \neg(\neg p \vee \neg q)$ by

1	$p \wedge q$	assumption
2	$\neg p \vee \neg q$	assumption
3	$\neg p$	assumption
4	p	$\wedge e_1$ 1
5	$\bot$	$\neg$e $4, 3$
6	$\neg q$	assumption
7	q	$\wedge e_2$ 1
8	$\bot$	$\neg$e $7, 6$
9	$\bot$	$\vee$e $2, 3 - 5, 6 - 8$
10	$\neg(\neg p \vee \neg q)$	$\neg$i $2 - 9$

3(q). We prove the validity of $(p \to q) \vee (q \to r)$ by

1	$q \vee \neg q$	lemma
2	q	assumption
3	p	assumption
4	q	copy 2
5	$p \to q$	$\to$i $3-4$
6	$(p \to q) \vee (q \to r)$	$\vee i_1$ 5
7	$\neg q$	assumption
8	q	assumption
9	$\bot$	$\neg$e $8, 7$
10	r	$\bot$e 9
11	$q \to r$	$\to$i $8-10$
12	$(p \to q) \vee (q \to r)$	$\vee i_2$ 11
13	$(p \to q) \vee (q \to r)$	$\vee$e $1, 2-6, 7-12$

5(a). We prove the validity of $\vdash ((p \to q) \to q) \to ((q \to p) \to p)$ by

1	$(p \to q) \to q$	assumption
2	$q \to p$	assumption
3	$\neg p$	assumption
4	p	assumption
5	$\bot$	$\neg$e $4, 3$
6	q	$\bot$e 5
7	$p \to q$	$\to$i $4-6$
8	q	$\to$e $1, 7$
9	p	$\to$e $2, 8$
10	$\bot$	$\neg$e $9, 3$
11	$\neg\neg p$	$\neg$i $3-10$
12	p	$\neg\neg$e 11
13	$(q \to p) \to p$	$\to$i $2-12$
14	$((p \to q) \to q) \to ((q \to p) \to p)$	$\to$i $1-13$

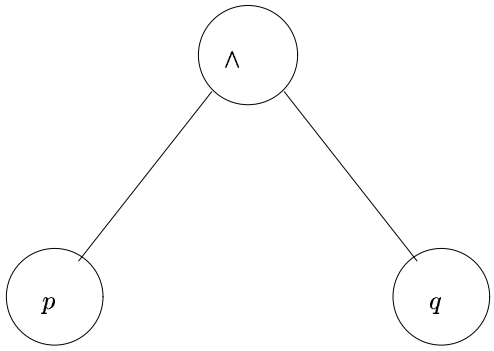
Let us motivate some of this proof's strategic choices. First, the opening of assumption boxes in lines 1 and 2 has nothing to do with strategy; it is simply dictated by the format of the formula we wish to prove. The assumption of $\neg p$ in line 3, however, is a strategic move with the desire to derive $\bot$ in order to get p in the end. Similarly, the assumption of p in line 4 is such a strategic move which tries to derive $p \to q$ which can be used to obtain $\bot$.

5(d). We prove the validity of $\vdash (p \to q) \to ((\neg p \to q) \to q)$ by

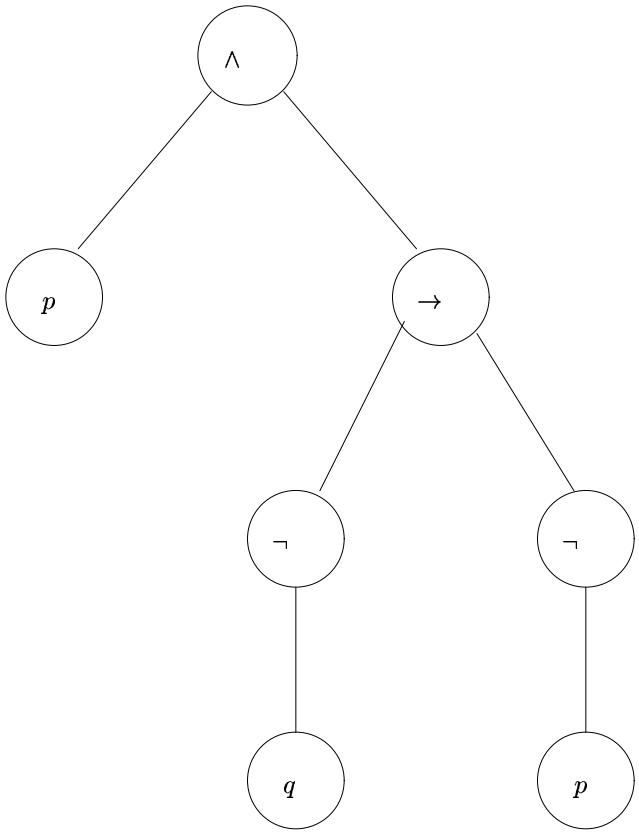
1	$p \to q$	assumption
2	$\neg p \to q$	assumption
3	$p \vee \neg p$	lemma
4	p	assumption
5	q	$\to$e 1, 4
6	$\neg p$	assumption
7	q	$\to$e 2, 6
8	q	$\vee$e $3, 4-5, 6-7$
9	$(\neg p \to q) \to q$	$\to$i $2-8$
10	$(p \to q) \to ((\neg p \to q) \to q)$	$\to$i $1-9$

EXERCISES 1.3 (p.81)

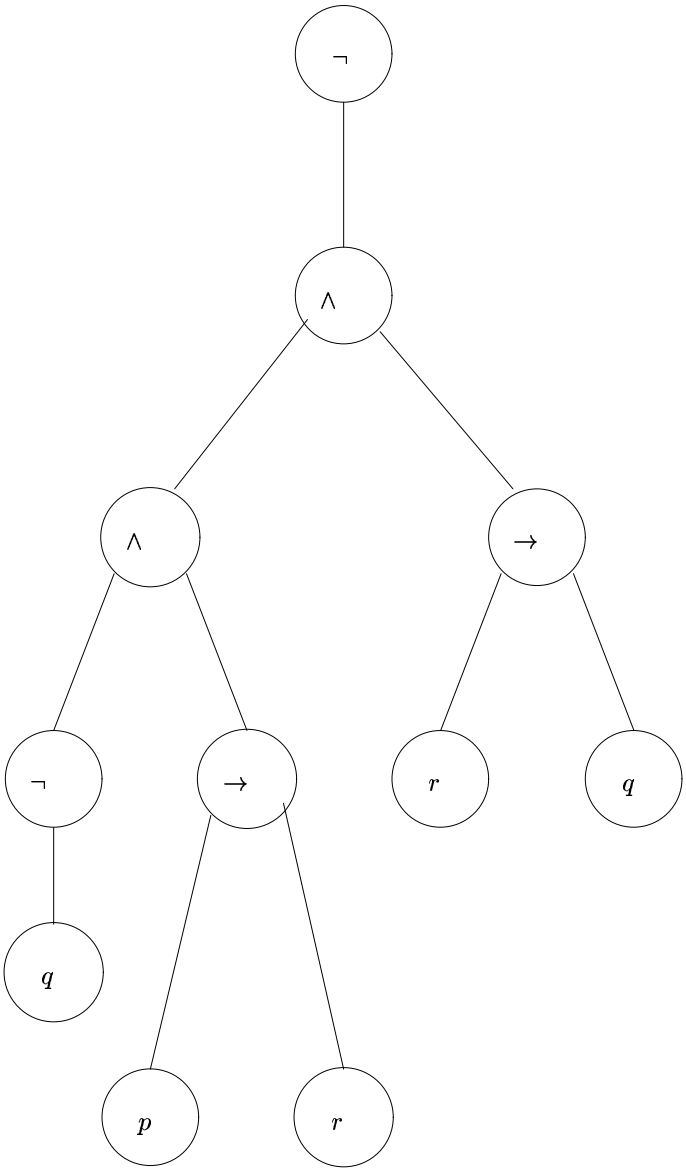
1(b). The parse tree of $p \wedge q$ is



1(d). The parse tree of $p \wedge (\neg q \to \neg p)$ is



1(f). The parse tree of $\neg((\neg q \wedge (p \to r)) \wedge (r \to q))$ is



2(a). The list of *all* subformulas of the formula $p \to (\neg p \vee (\neg\neg q \to (p \wedge q)))$ is

p

q

$\neg p$

$\neg q$

$\neg\neg q$

$p \wedge q$

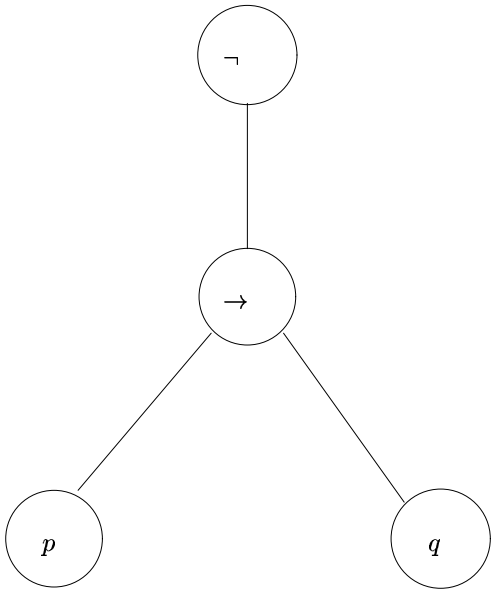
$\neg\neg q \to (p \wedge q)$

$\neg p \vee (\neg\neg q \to (p \wedge q))$

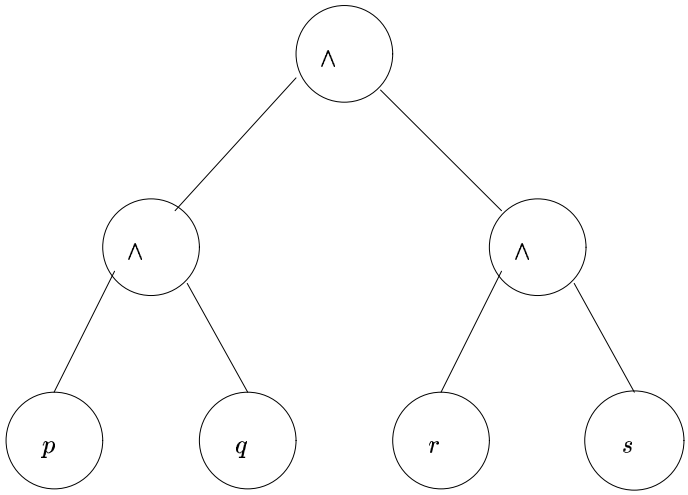
$p \to (\neg p \vee (\neg\neg q \to (p \wedge q)))$.

Note that $\neg q$ and $\neg\neg q$ are two different subformulas.

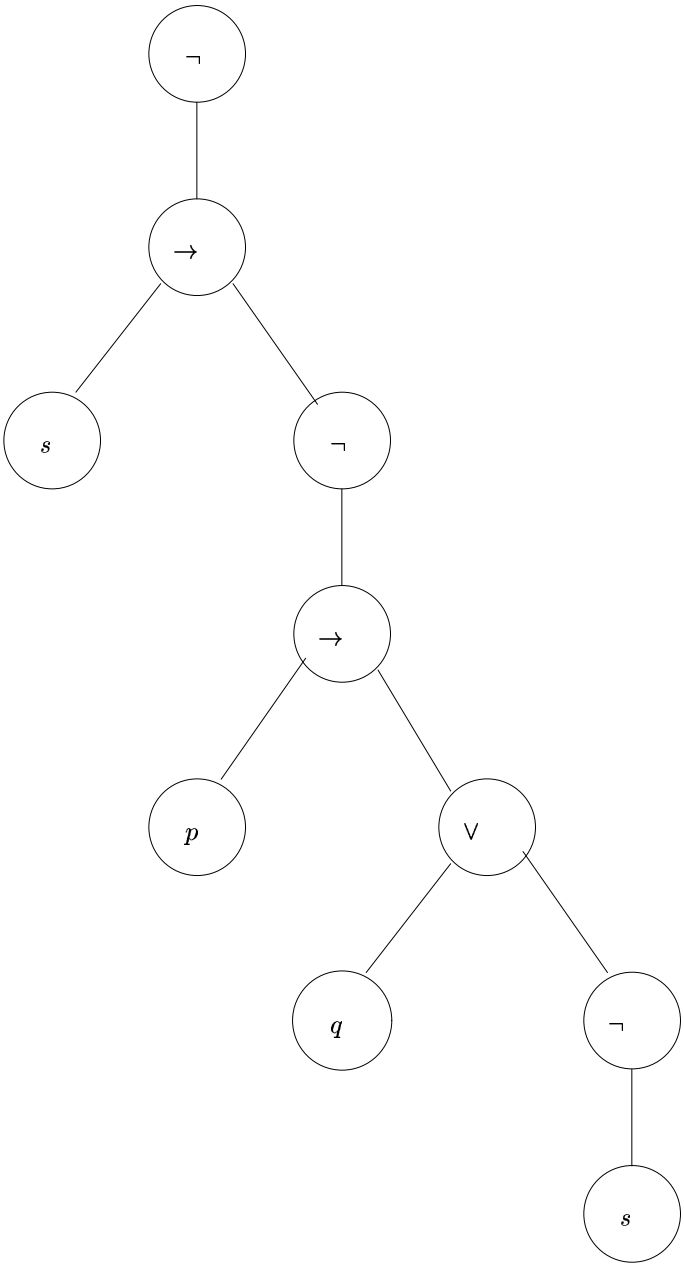
3(a). An example of a parse tree of a propositional logic formula which is a negation of an implication:



3(c). An example of a parse tree of a formula which is a conjunction of conjunctions is



4(a). The parse tree of $\neg(s \to (\neg(p \to (q \vee \neg s))))$ is



Its list of subformulas is

p

q

s

$\neg s$

$q \vee \neg s$

$p \rightarrow (q \vee \neg s)$

$\neg(p \rightarrow (q \vee \neg s))$

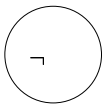
$s \rightarrow (\neg(p \rightarrow (q \vee \neg s)))$

$\neg(s \rightarrow (\neg(p \rightarrow (q \vee \neg s))))$.

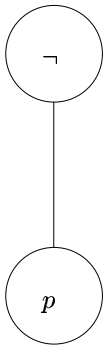
5. The logical formula represented by the parse tree in Figure 1.22 is

$$\neg(\neg(p \vee (q \wedge \neg p)) \rightarrow r).$$

7(a). The parse tree

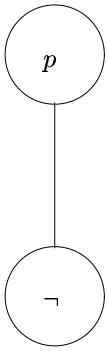


does not correspond to a well-formed formula, but its extension



does.

7(b). The parse tree



is *inherently* ill-formed; i.e. any extension thereof could not correspond to a well-formed formula. (Why?)

EXERCISES 1.4 (p.82)

1. The truth table for $\neg p \vee q$ is

p	q	$\neg p$	$\neg p \vee q$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

and matches the one for $p \to q$.

2(a). The complete truth table of the formula $((p \to q) \to p) \to p$ is given by

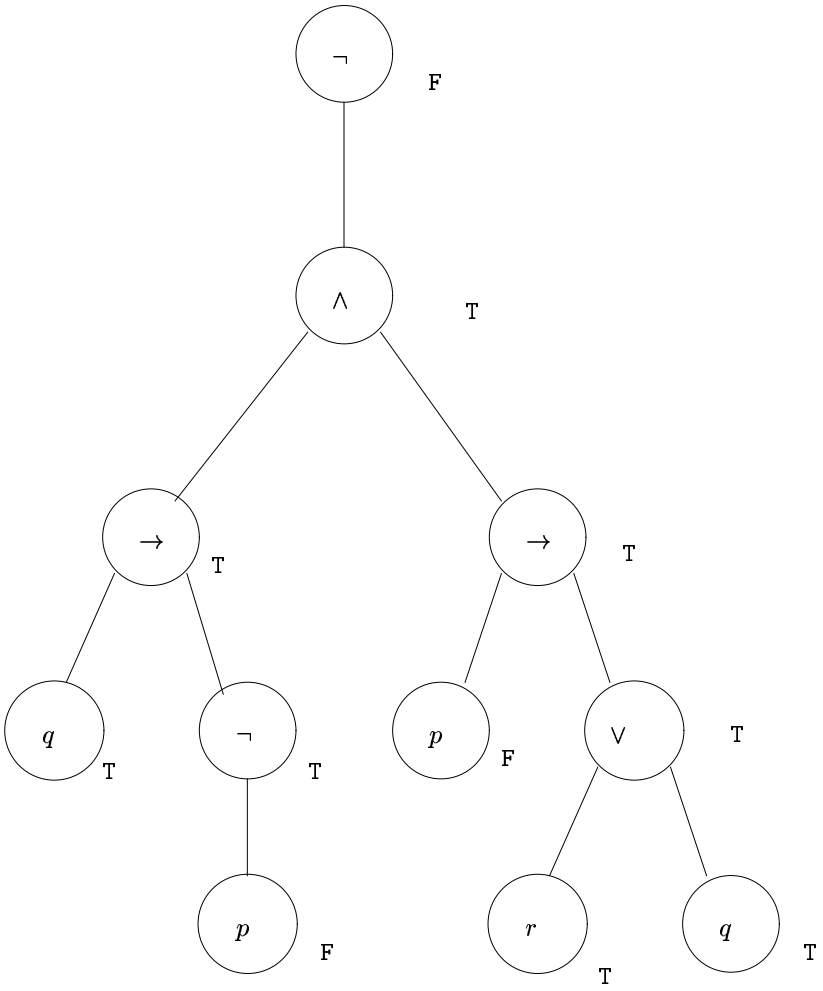
p	q	$p \to q$	$(p \to q) \to p$	$((p \to q) \to p) \to p$
T	T	T	T	T
T	F	F	T	T
F	T	T	F	T
F	F	T	F	T

Note that this formula is valid since it always evaluates to T.

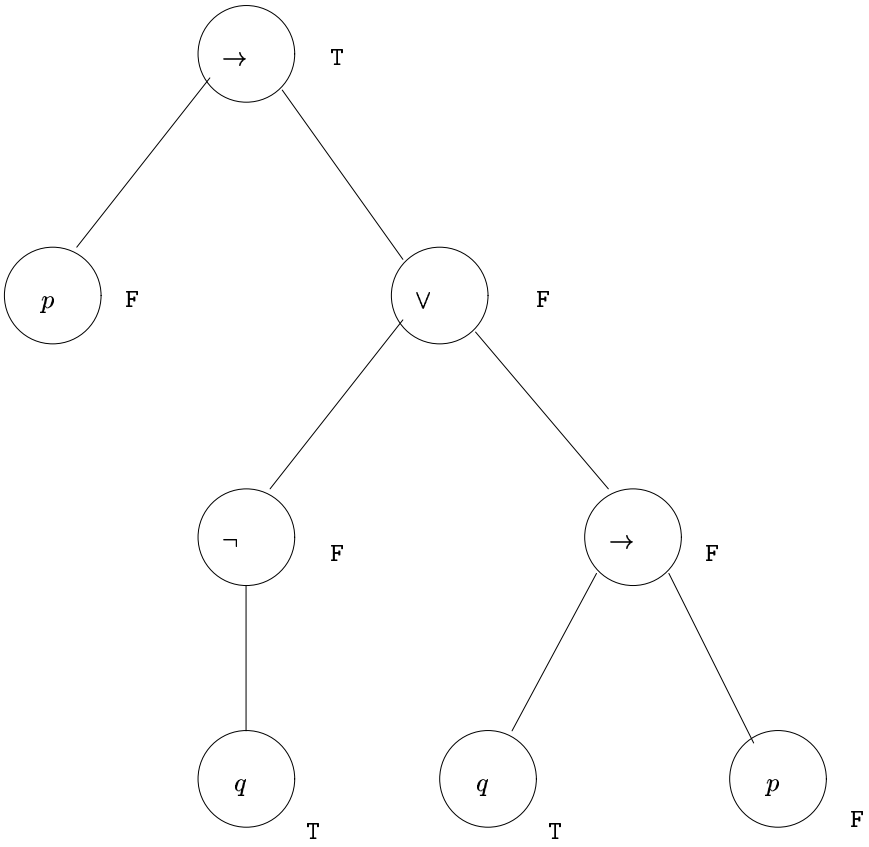
2(c). The *complete* truth table for $p \vee (\neg(q \wedge (r \to q)))$ is

p	q	r	$r \to q$	$q \wedge (r \to q)$	$\neg(q \wedge (r \to q))$	$p \vee (\neg(q \wedge (r \to q)))$
T	T	T	T	T	F	T
T	T	F	T	T	F	T
T	F	T	F	F	T	T
F	T	T	T	T	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	F
F	F	T	F	F	T	T
F	F	F	T	F	T	T

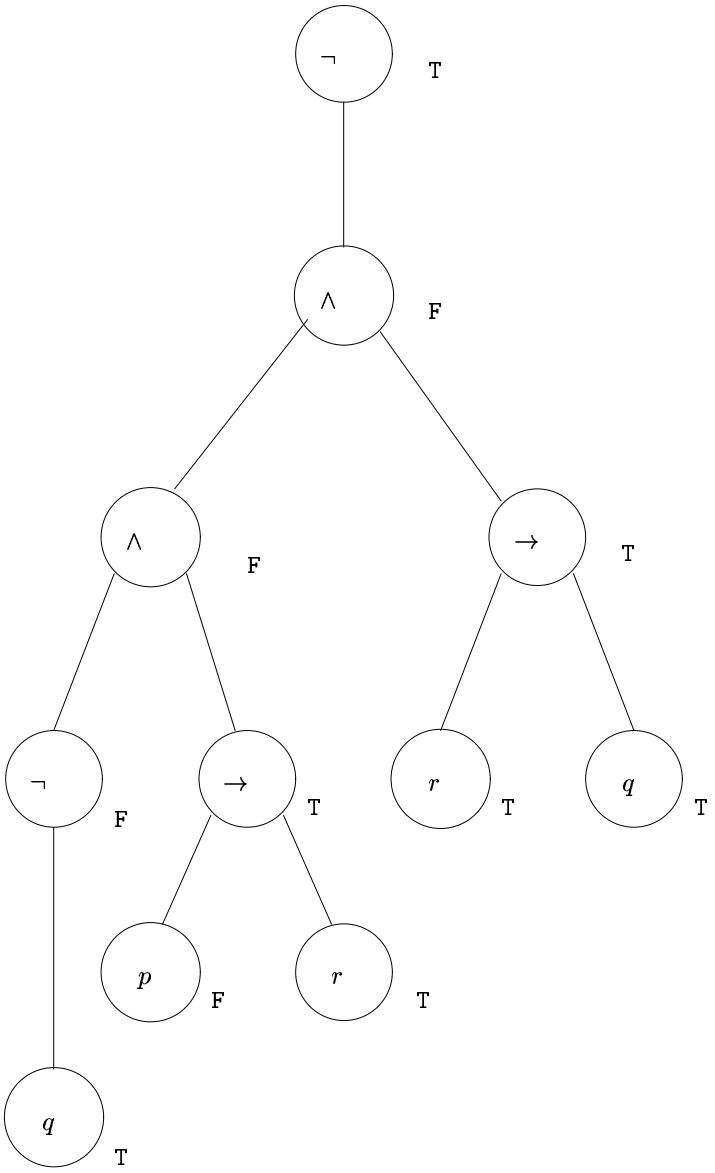
3(a). The requested truth value of the formula in Figure 1.10 on page 44 computed in a bottom-up fashion:



4(a). We compute the truth value of $p \to (\neg q \vee (q \to p))$ in a bottom-up fashion on its parse tree:



4(b). Similarly, we compute the truth value of $\neg((\neg q \wedge (p \to r)) \wedge (r \to q))$ in a bottom-up fashion on its parse tree:



5. The formula of the parse tree in Figure 1.10 on page 44 is not valid since it already evaluates to `F` for any assigment in which p and q evaluate to `F`. However, this formula is satisfiable: for example, if q and p evaluate to `T` then $q \to \neg p$ renders `F` and so the entire formula evaluates to `T`.

7(c). We prove

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6} \tag{1.1}$$

for all natural numbers $n \geq 1$ by mathematical induction on n.

1. **Base Case:** If $n = 1$, then the lefthand side of (1.1) equals 1^2 which is 1; the righthand side equals $1 \cdot (1+1) \cdot (2 \cdot 1 + 1)/6 = 2 \cdot 3/6 = 1$ as well. Thus, equation (1.1) holds for our base case.
2. **Inductive Step:** Our **induction hypothesis** is that (1.1) holds for n. Our task is to use that information to show that (1.1) also holds for $n+1$. The lefthand side for $n+1$ equals $1^2+2^2+3^2+\cdots+n^2+(n+1)^2$ which, by our induction hypothesis, equals

$$\frac{n \cdot (n+1) \cdot (2n+1)}{6} + (n+1)^2. \tag{1.2}$$

The righthand side of (1.1) for $n+1$ equals

$$\frac{(n+1) \cdot ((n+1)+1) \cdot (2(n+1)+1)}{6} \tag{1.3}$$

Thus, we are done if the expressions in (1.2) and (1.3) are equal. Using our algebra skills acquired in high-school, we see that both expressions equal

$$\frac{(n+1) \cdot (2n^2 + 7n + 6)}{6}. \tag{1.4}$$

7(d). We use mathematical induction to show that $2^n \geq n + 12$ for all natural numbers $n \geq 4$. Before we do that we inspect whether we could improve this statement somehow: if $n = 1, 2$ or 3 then 2^n is smaller than $n + 12$; e.g. $2^3 = 8$ is smaller than $3 + 12 = 15$. Thus, we can only hope to prove $2^n \geq n + 12$ for $n \geq 4$; there is no room for improvement.

1. **Base Case:** Our base case lets n be 4. Then $2^n = 2^4 = 16$ is greater or equal to $4 + 12 = 16$.
2. **Inductive Step:** We need to show that $2^{n+1} \geq (n+1) + 12$, where our **induction hypothesis** assumes that $2^n \geq n + 12$. Thus,

$$\begin{aligned} 2^{n+1} &= 2 \cdot 2^n \\ &\geq 2 \cdot (n+12) \qquad \text{by our induction hypothesis} \end{aligned}$$

$$\begin{aligned} &= [(n+1)+12]+(n+11) \\ &\geq (n+1)+12 \end{aligned}$$

guarantees that our claim holds for $n+1$. Note that the first step of this computation also uses that multiplication with a positive number is *monotone*: $x \geq y$ implies $2 \cdot x \geq 2 \cdot y$.

8. The Fibonacci numbers are defined by

$$\begin{aligned} F_1 &\stackrel{\text{def}}{=} 1 \\ F_2 &\stackrel{\text{def}}{=} 1 \\ F_{n+1} &\stackrel{\text{def}}{=} F_n + F_{n-1} \qquad \text{for all } n \geq 2. \end{aligned} \tag{1.5}$$

We use mathematical induction to prove that F_{3n} is even for all $n \geq 1$.

1. **Base Case:** For $n = 1$, we compute $F_{3\cdot 1} = F_3 = F_2 + F_1$ by (1.5), but the latter is just $1 + 1 = 2$ which is even.
2. **Inductive Step:** Our **induction hypothesis** assumes that F_{3n} be even. We need to show that $F_{3\cdot(n+1)}$ is even as well. This is a bit tricky as we have to decide on where to apply the inductive definition of (1.5):

$$\begin{aligned} F_{3\cdot(n+1)} &= F_{3n+3} \\ &= F_{(3n+2)+1} \\ &= F_{3n+2} + F_{(3n+2)-1} \qquad \text{by (1.5)} \\ &= F_{(3n+1)+1} + F_{3n+1} \\ &= (F_{3n+1} + F_{3n}) + F_{3n+1} \qquad \text{unfolding } F_{(3n+1)+1} \text{ with (1.5)} \\ &= 2 \cdot F_{3n+1} + F_{3n}. \end{aligned}$$

Since F_{3n} is assumed to be even and since $2 \cdot F_{3n+1}$ clearly is even, we conclude that $2 \cdot F_{3n+1} + F_{3n}$, and therefore $F_{3\cdot(n+1)}$, is even as well.
Note that it was crucial not to unfold F_{3n+1} as well; otherwise, we would obtain four summands but no inductive argument. (Why?)

10. Consider the assertion

"*The number* $n^2 + 5n + 1$ *is even for all* $n \geq 1$."

(a) We prove the **inductive step** of that assertion as follows: we

simply assume that $n^2 + 5n + 1$ is even and we need to show that $(n+1)^2 + 5(n+1) + 1$ is then even as well. But

$$\begin{aligned}(n+1)^2 + 5(n+1) + 1 &= (n^2 + 2n + 1) + (5n + 5) + 1 \\ &= (n^2 + 5n + 1) + (2n + 6)\end{aligned}$$

must be even since

- $n^2 + 5n + 1$ is assumed to be even,
- $2n + 6 = 2 \cdot (n + 3)$ is always even,
- and the sum of two even numbers is even again.

(b) However, the **Base Case** fails to hold for this assertion since $n^2 + 5n + 1 = 1^2 + 5 \cdot 1 + 1 = 1 + 5 + 1 = 7$ is odd if $n = 1$.

(c) Thus, the assertion is false, for it is simply not true for $n = 1$.

(d) We use mathematical induction to show that $n^2 + 5n + 1$ is *odd* for all $n \geq 1$.

1. **Base Case:** If $n = 1$, then we have already computed that $n^2 + 5n + 1 = 7$ is odd.
2. **Inductive Step:** Our **induction hypothesis** assumes that $n^2 + 5n + 1$ is odd. Above we already computed

$$(n+1)^2 + 5(n+1) + 1 = (n^2 + 5n + 1) + (2n + 6) \quad (1.6)$$

(this computation involves only high-school algebra and has nothing to do with possible induction hypotheses). Thus,

$$(n+1)^2 + 5(n+1) + 1$$

must be odd, for

- $n^2 + 5n + 1$ is assumed to be odd,
- $2n + 6$ is always even,
- and the sum of an odd and an even number is odd.

12(c). We prove that the sequent $p \to (q \to r) \vdash p \to (r \to q)$ is not valid. Using soundness of our natural deduction calculus with respect to the truth-table semantics, it suffices to find an assignment of truth values to p, q and r such that $p \to (q \to r)$ evaluates to `T`, but $p \to (r \to q)$ evaluates to `F`. The latter can only occur if $p = \mathtt{T}$ and $q \to r = \mathtt{F}$. So $p = q = \mathtt{T}$ and $r = \mathtt{F}$ is the only choice which can defeat the claimed validity of this sequent. However, for this choice we easily compute that $p \to (q \to r)$ evaluates to `T` as this amounts to $\mathtt{T} \to (\mathtt{F} \to \mathtt{T})$ which computes to `T`.

13(a) Let p denote "France is a country." and q denote "France won the Euro 2004 Soccer Championships." Then $p \vee q$ holds in the world we currently live in, but $p \wedge q$ does not.

13(b) Here one can choose for p and q the same meaning as in item 13(a): $\neg p \to \neg q$ holds as p is true, but $\neg q \to \neg p$ is false as $\neg q$ is true whereas $\neg p$ is false.

17(b) This formula is indeed valid. To make it evaluate to `F` we need that both disjuncts evaluate, in particular $\cdot \to p$, have to evaluate to `F`. This p needs to evaluate to `F` and $\neg(p \to q)$ needs to evaluate to `T`. But the latter contradicts that p is `F`.

EXERCISES 1.5 (p.87)

2(a) (Bonus) We have $q \vee (\neg p \vee r) \equiv \neg p \vee (q \vee r) \equiv p \to (q \vee r)$ since $\vee$ is associative and implication can be decomposed in this way. Since $\equiv$ is transitive, we conclude that $q \vee (\neg p \vee r) \equiv p \to (q \vee r)$.

(b) $q \wedge \neg r \to p$ is not equivalent to $p \to q \vee r$ since $q \wedge \neg r \to p \not\models p \to (q \vee r)$: for let q and r be `F` and p be `T`. Then $q \wedge \neg r \to p$ computes `T`, whereas $p \to q \vee r$ computes `F`.

(c) (Bonus) We have $p \wedge \neg r \to q \equiv \neg(p \wedge \neg r) \vee q \equiv (\neg p \vee \neg\neg r) \vee q \equiv q \vee (\neg p \vee r)$ which is the formula in (a). Thus, $p \wedge \neg r \to q$ is equivalent to $p \to q \vee r$. Note that this made use of the identity $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$ which is shown in item 10(a) of Exercises 1.13.

(d) We have $\neg q \wedge \neg r \to \neg p \equiv \neg(\neg q \wedge \neg r) \vee \neg p \equiv (\neg\neg q \vee \neg\neg r) \vee \neg p \equiv q \vee (\neg p \vee r)$ and, as in the previous item, we conclude that $\neg q \wedge \neg r \to \neg p$ is equivalent to $p \to q \vee r$.

6(d)i. To show $\phi \wedge (\phi \vee \eta) \equiv \phi$, we need to show $\phi \wedge (\phi \vee \eta) \models \phi$ and $\phi \models \phi \wedge (\phi \vee \eta)$. To see $\phi \wedge (\phi \vee \eta) \models \phi$, assume that $\phi \wedge (\phi \vee \eta)$ evaluates to `T`. Then ϕ has to evaluate to `T` as well. To see $\phi \models \phi \wedge (\phi \vee \eta)$, assume that ϕ evaluates to `T`. Then $\phi \vee \eta$ evaluates to `T` as well. Thus, $\phi \wedge (\phi \vee \eta)$ evaluates to `T`, too.

6(e)ii. To show $\phi \vee (\psi \wedge \eta) \models (\phi \vee \psi) \wedge (\phi \vee \eta)$, assume that $\phi \vee (\psi \wedge \eta)$ evaluates to `T`.

Case 1: Suppose that ϕ evaluates to `T`. Then $\phi \vee \psi$ and $\phi \vee \eta$ evaluate to `T`. Therefore, $(\phi \vee \psi) \wedge (\phi \vee \eta)$ evaluates to `T` as well.

Case 2: Suppose that $\psi \wedge \eta$ evaluates to `T`. Then both ψ and η evaluate to `T`. Thus, $\phi \vee \psi$ and $\phi \vee \eta$ evaluate to `T`. Therefore, $(\phi \vee \psi) \wedge (\phi \vee \eta)$ evaluates to `T`.

Note that these two cases are exhaustive. (Why?)
To show $(\phi \vee \psi) \wedge (\phi \vee \eta) \models \phi \vee (\psi \wedge \eta)$, assume that $(\phi \vee \psi) \wedge (\phi \vee \eta)$ evaluates to T. Then $\phi \vee \psi$ and $\psi \vee \eta$ evaluate to T.

Case 1: Suppose that ϕ evaluates to F. Then we conclude that ψ and η have to evaluate to T. Thus, $\psi \wedge \eta$ evaluates to T. Therefore, $\phi \vee (\psi \wedge \eta)$ evaluates to T.

Case 2: Suppose that ϕ evaluates to T. Then $\phi \vee (\psi \wedge \eta)$ evaluates to T as well.

6(g)ii. To show $\neg(\phi \vee \psi) \models \neg\phi \wedge \neg\psi$, assume that $\neg(\phi \vee \psi)$ evaluates to T. Then $\phi \vee \psi$ evaluates to F. Thus, ϕ and ψ evaluate to F. So $\neg\phi$ and $\neg\psi$ evaluate to T. Therefore, $\neg\phi \wedge \neg\psi$ evaluates to T as well.
To show $\neg\phi \wedge \neg\psi \models \neg(\phi \vee \psi)$, assume that $\neg\phi \wedge \neg\psi$ evaluates to T. Then $\neg\phi$ and $\neg\psi$ evaluate to T. Thus, ϕ and ψ evaluate to F. This entails that $\phi \vee \psi$ evaluate to F. Therefore, $\neg(\phi \vee \psi)$ evaluates to T.

7(a). The formula ϕ_1 in CNF which we construct from the truth table

p	q	ϕ_1
T	T	F
F	T	F
T	F	F
F	F	T

is

$$(\neg p \vee \neg q) \wedge (p \vee \neg q) \wedge (\neg p \vee q).$$

Note how these principal conjuncts correspond to the lines in the table above, where the ϕ_1 entry is F: e.g. the third line T | F | F results in the conjunct $(\neg p \vee q)$.

7(b). The formula ϕ_2 in CNF based on the truth table

p	q	r	ϕ_2
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

is

$$(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r).$$

Incidentally, if you are not sure about whether your answer is correct you can also try to verify it by ensuring that your answer has the same truth table as the above.

8. We write pseudo-code for a recursive function `IMPL_FREE` which expects a (parse tree of a) propositional formula as input and produces an equivalent formula as output such that the result contains no implications. Since our logic has five data constructors (= five ways of building a formula), namely atoms, negation, conjunction, disjunction, and implication, we have to consider five cases. Only in the case of implication do we have to do work beyond mere book-keeping:

function `IMPL_FREE` (ϕ) :
/* computes a formula without implication equivalent to ϕ */
begin function
case
ϕ is an atom: **return** ϕ
ϕ is $\neg\phi_1$: **return** $\neg$`IMPL_FREE` (ϕ_1)
ϕ is $\phi_1 \wedge \phi_2$: **return** `IMPL_FREE` $(\phi_1) \wedge$ `IMPL_FREE` (ϕ_2)
ϕ is $\phi_1 \vee \phi_2$: **return** `IMPL_FREE` $(\phi_1) \vee$ `IMPL_FREE` (ϕ_2)
ϕ is $\phi_1 \to \phi_2$: **return** `IMPL_FREE` $(\neg\phi_1 \vee \phi_2)$
end case
end function

There are quite a few other solutions possible. For example, we could have optimized this code in the last case by saying

return $\neg$(`IMPL_FREE` ϕ_1) $\vee$ (`IMPL_FREE` ϕ_2)

which would save us a couple of computation steps. Furthermore, it is possible to overlap the patterns of the first two cases by returning ϕ if ϕ is a literal in the first clause. Notice how the two clauses agree in this case. Such a programming style, while correct, it not recommended as it makes it harder to read somebody else's code.

9. We use our algorithm `IMPL_FREE` together with the functions `NNF` and `CNF` to compute `CNF(NNF(IMPL_FREE` ϕ`))`, where ϕ is the formula

$\neg(p \to (\neg(q \wedge (\neg p \to q))))$. In the solution below we skipped some of the obvious and tedious intermediate computation steps. Removing all implications results in:

$$\begin{aligned}
\texttt{IMPL_FREE}\ \phi &= \neg\texttt{IMPL_FREE}\ (p \to (\neg(q \wedge (\neg p \to q)))) \\
&= \neg(\neg p \vee \texttt{IMPL_FREE}\ (\neg(q \wedge (\neg p \to q)))) \\
&= \neg(\neg p \vee \neg(q \wedge \texttt{IMPL_FREE}\ (\neg p \to q))) \\
&= \neg(\neg p \vee \neg(q \wedge (\neg\neg p \vee q)))
\end{aligned}$$

Computing the corresponding negation normalform yields:

$$\begin{aligned}
\texttt{NNF}\ \neg(\neg p \vee \neg(q \wedge (\neg\neg p \vee q))) &= (\texttt{NNF}\ \neg\neg p) \wedge (\texttt{NNF}\ \neg\neg(q \wedge (\neg\neg p \vee q))) \\
&= (\texttt{NNF}\ p) \wedge (\texttt{NNF}\ (q \wedge (\neg\neg p \vee q))) \\
&= p \wedge (q \wedge (\texttt{NNF}\ (\neg\neg p \vee q))) \\
&= p \wedge (q \wedge ((\texttt{NNF}\ \neg\neg p) \vee (\texttt{NNF}\ q))) \\
&= p \wedge (q \wedge (p \vee q)) \\
&= p \wedge q \wedge (p \vee q)
\end{aligned}$$

which is already in conjunctive normalform so we won't have to call `CNF` anymore. Notice that this formula is equivalent to the CNF $p \wedge q$.

15(a) The marking algorithm first marks r, q, and u through clauses of the form $\top \to \cdot$. Then it marks p through clause $r \to p$ and s through clause $u \to s$. The w never gets marked and so $p \wedge q \wedge w$ can never trigger a marking for $\bot$. Thus the algorithm determines that the formula is satisfiable.

15(g). Applying the marking algorithm to $(\top \to q) \wedge (\top \to s) \wedge (w \to \bot) \wedge (p \wedge q \wedge s \to \bot) \wedge (v \to s) \wedge (\top \to r) \wedge (r \to p)$, it marks q, s, and r in the first iteration of the while-loop. In the second iteration, p gets marked and in the third iteration an inconsistency is found: $(p \wedge q \wedge s \to \bot)$ is a subformula of the original formula and p, q, s are all marked. Thus, the Horn formula is not satisfiable.

17. (Bonus) Note that Horn formulas are already of the form $\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_m$, where each ψ_i is of the form $p_1 \wedge p_2 \wedge \ldots \wedge p_{i_k} \to q_i$. We know that the latter formula is equivalent to $\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_{i_k}) \vee q_i$, i.e. it is equivalent to $\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_{i_k} \vee q_i$. Thus, we may convert any Horn formula into a CNF where each conjuction clause has at

most one positive literal in it. Note that this also covers some special cases such as $\top \to q \equiv \neg\top \vee q \equiv \bot \vee q \equiv q$.

EXERCISES 1.6 (p.90)

2. We illustrate the arguments for some of these rules.

- For example, if a $\wedge$-node has to evaluate to `T` in order for the overall formula to be satisfiable, then both its arguments have to evaluate to `T` in order for the overall formula to be satisfiable.
- If both arguments have to evaluate to `T` in order for the overall formula to be satisfiable, then the $\wedge$-node needs to evaluate to `T` as well for the overall formula to be satisfiable.
- If we know that the overall formula can only be satisfiable if an $\wedge$-node evaluates to `F` and one of its arguments evaluates to `T`, then we know that the overall formula can only be satisfiable if all this is true and the other argument evaluates to `F`.
- Etc.

4. There are many equivalent DAGs one could construct for this. We chose as a formula $\neg((p \vee q) \wedge (p \to r) \to r)$ which can be translated as $\neg(\neg p \wedge \neg q) \wedge (\neg(p \wedge \neg r) \wedge \neg r)$ into the SAT solver's adequate fragment. One can then draw the corresponding DAG and will see that the linear SAT solver computes a satisfiablity witness for which p and r evaluate to `F` and q evaluates to `T`.

11. A semi-formal argument could go like this: the linear solver will compute all constraints it can infer from the current permanent markings so the order of evaluation won't matter for it as all constraints will be found and so it will always detect a contradiction if present. The cubic solver may test nodes in a different order but if *any* test creates a new permanent constraint, the cubic solver will test again all unconstrained notes so the set of constraints can only increase. Moreover, a contradiction found by testing in some order will also be found by testing in some other order as the eventual discovery of constraints is independent of the order of tests by the argument just made.

2

Predicate logic

EXERCISES 2.1 (p.157)

1. (a) $\forall x(P(x) \to A(m,x))$

(b) $\exists x(P(x) \wedge A(x,m))$

(c) $A(m,m)$

(d) $\forall x(S(x) \to (\exists y(L(y) \wedge \neg B(x,y))))$, or, equivalently, $\neg\exists x(S(x) \wedge \forall y(L(y) \to B(x,y)))$.

(e) $\forall x(L(x) \to \exists y(S(y) \wedge \neg B(y,x)))$, or, equivalently, $\neg\exists y(L(y) \wedge \forall x(S(x) \to B(x,y)))$.

(f) $\forall x(L(x) \to \forall y(S(y) \to \neg B(y,x)))$, or, equivalently, $\forall x \forall y(L(x) \wedge S(y) \to \neg B(y,x))$.

3. (a) $\forall x(\texttt{Red}(x) \to \texttt{Inbox}(x))$

(b) $\forall x(\texttt{Inbox}(x) \to \texttt{Red}(x))$

(c) $\forall x(\texttt{Animal}(x) \to (\neg\texttt{Cat}(x) \vee \neg\texttt{Dog}(x)))$, or, equivalently, $\neg\exists x(\texttt{Animal}(x) \wedge \texttt{Cat}(x) \wedge \texttt{Dog}(x))$.

(d) $\forall x(\texttt{Prize}(x) \to \exists y(\texttt{Boy}(y) \wedge (\texttt{Win}(y,x))))$

(e) $\exists y(\texttt{Boy}(y) \wedge \forall x(\texttt{Prize}(x) \to \texttt{Win}(y,x)))$ Note carefully the difference between (d) and (e), which look so similar when expressed in natural language. Item (d) says that every prize was won by a boy, possibly a different one for each prize. But item (e) says it is the same boy that won all the prizes.

EXERCISES 2.2 (p.158)

1(a)ii. The string $f(x, g(y,z), d)$ is not a term since f requires two arguments, not three. Likewise, g requires three arguments, not two.

1(a)iii. $\surd$

1(c). 1. There is only one term of height 1 (without variables); it is simply d.

2. There are two terms of height 2, namely $f(d,d)$ and $g(d,d,d)$.
3. There are quite a few terms of height 3: the root node has to be f or g. Depending on that choice, we have 2 or 3 arguments. Since the overall term has to have height 3 it follows that at least one of these arguments has to be a term of height 2. Thus, we may list all these terms systematically:
   - $f(d, f(d,d))$
   - $f(f(d,d), d)$
   - $f(g(d,d,d), d)$
   - $f(d, g(d,d,d))$
   - $f(f(d,d), f(d,d))$
   - $f(g(d,d,d), g(d,d,d))$
   - $f(f(d,d), g(d,d,d))$
   - $f(g(d,d,d), f(d,d))$

   are all the terms with f as a root. The terms with g as a root are of the form $g(\#1, \#2, \#3)$, where at least one of the three arguments has height 2. Thus, $g(d,d,d)$ is not allowed, but other than that each argument can be d, $f(d,d)$, or $g(d,d,d)$. This gives us $3 \cdot 3 \cdot 3 - 1 = 26$ different terms of height 3. The first ten are
   - $g(d,d,g(d,d,d))$
   - $g(d,g(d,d,d),d)$
   - $g(g(d,d,d),d,d)$
   - $g(d,d,f(d,d))$
   - $g(d,f(d,d),d)$
   - $g(f(d,d),d,d)$
   - $g(d,f(d,d),f(d,d))$
   - $g(d,f(d,d),g(d,d,d))$
   - $g(d,g(d,d,d),f(d,d))$
   - $g(d,g(d,d,d),g(d,d,d))$

   and you are welcome to write down the remaining 16 terms if you wish. In conclusion, there are $1+2+8+26 = 37$ different terms already of height less than 4.

3(a). (i) $\surd$

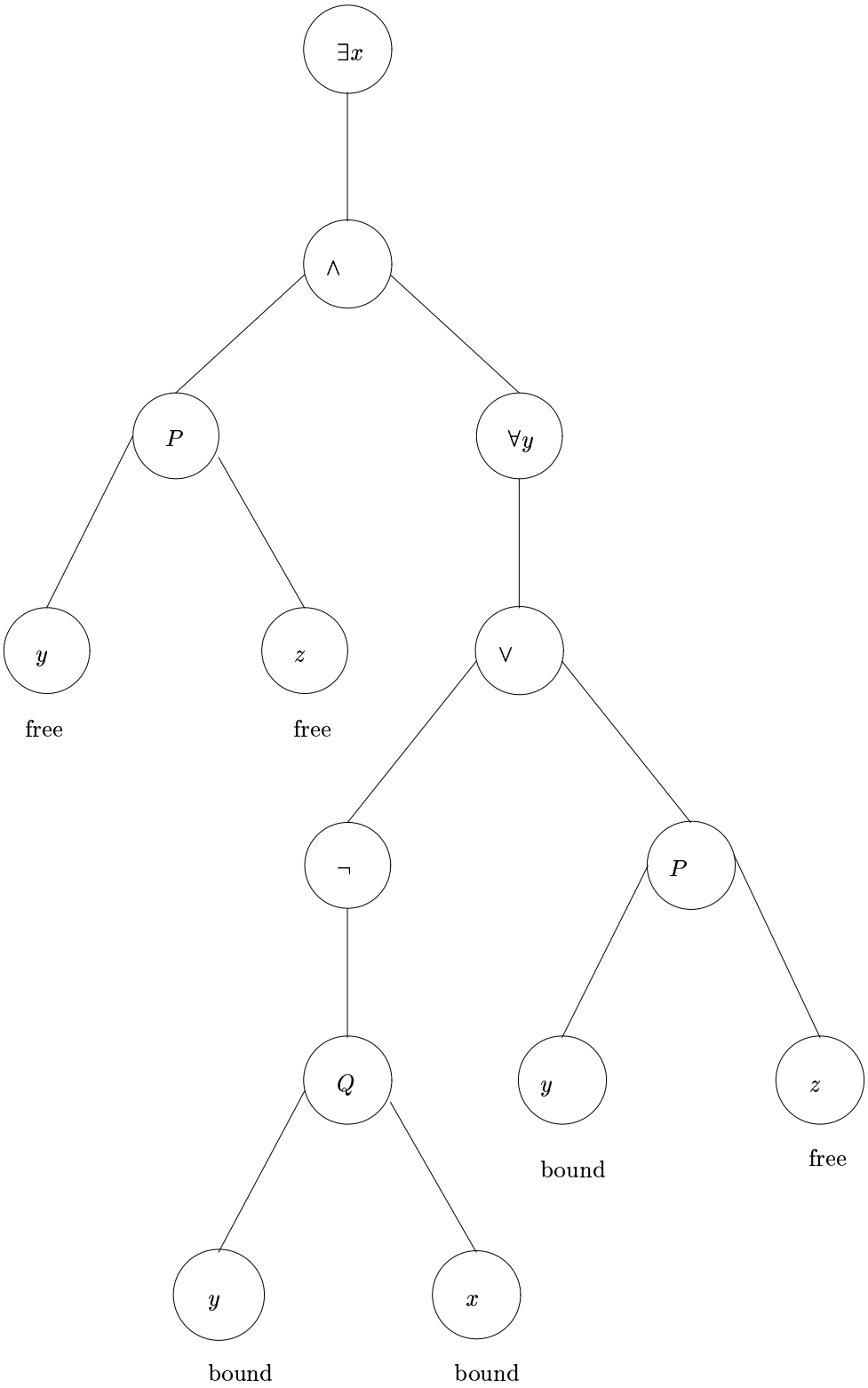
(ii) $\surd$

(iii) $\times$; $f(m)$ denotes a term, not a formula.

(iv) $\times$; in predicate logic, we can't nest predicates: the missing argument in $B(?, y)$ has to be a *term*, but $B(m, x)$ is a *formula*.

(v) $\times$; again, we can't nest predicates (B, S are predicates).
(vi) $\surd$
(vii) $\surd$
(viii) $\times$; again, we can't nest predicates.

4(a). The parsetree is


∃x
∧
P
∀y
y
z
∨
free
free
¬
P
Q
y
z
bound
free
y
x
bound
bound

4(b). From the parsetree of the previous item we see that all occurrences of z are free, all occurrences of x are bound, and the leftmost occurrence of y is free, whereas the other two occurrences of y are bound.

4(d).(i) – $\phi[w/x]$ is simply ϕ again since there are no free occurrences of x in ϕ that could be replaced by w;
– $\phi[w/y]$ is $\exists x(P(w,z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ since we replace the sole free occurrence of y with w;
– $\phi[f(x)/y]$ is $\exists x(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ since we replace the sole free occurrence of y with $f(x)$;
– $\phi[g(y,z)/z]$ $\exists x(P(y,g(y,z)) \wedge \forall y(\neg Q(y,x) \vee P(y,g(y,z))))$ since we replace all (free) occurrences of z with $g(y,z)$.

(ii) All of them, for there are no free occurrences of x in ϕ to begin with.

(iii) The terms w and $g(y,z)$ are free for y in ϕ, since the sole free occurrence of y only has $\exists x$ as a quantifier above it. For that very reason, $f(x)$ is *not* free for y in ϕ, since the x in $f(x)$ would be captured by $\exists x$ in that substitution process.

4(f). Now, the scope of $\exists x$ is only the formula $P(y,z)$, since the inner quantifier $\forall x$ binds the x (and overrides the binding of $\exists x$ in the formula $\neg Q(x,x) \vee P(x,z)$).

EXERCISES 2.3 (p.160)

3(a). A formula $\phi_{=3}$ whose models are exactly those which have three distinct elements is

$$\exists x\, \exists y\, \exists z\, (((\neg(x=y) \wedge \neg(x=z)) \wedge \neg(y=x)) \\ \wedge \forall w\, (((w=x) \vee (w=y)) \vee (w=z))).$$

3(c). As in item 3 above, we may generally construct formulas $\phi_{=n}$ for each $n = 1, 2, 3, \ldots$ such that the models of $\phi_{=n}$ are exactly those with n distinct elements. Then a model is finite iff it satisfies $\phi_{=n}$ *for some* $n \geq 1$. Therefore,

$$\bigvee_{n \geq 1} \phi_{=n}$$

would do the trick, but, alas, this conjunction is *infinite* and not part of the syntax of predicate logic.

7(a). We prove the validity of $\forall x \forall y P(x,y) \vdash \forall u \, \forall v \, P(u,v)$ by

1	$\forall x \forall y P(x,y)$	assum
2	x_0	
3	y_0	
4	$\forall y \, P(x_0,y)$	$\forall x$ e 1
5	$P(x_0,y_0)$	$\forall y$ e 4
6	$\forall v \, P(x_0,v)$	$\forall v$ i 3–5
7	$\forall u \, \forall v \, P(u,v)$	$\forall u$ i 2–6

7(c). We prove the validity of $\exists x \, \forall y \, P(x,y) \vdash \forall y \, \exists x \, P(x,y)$ by

1	$\exists x \, \forall y \, P(x,y)$	assum
2	y_0	
3	x_0	
4	$\forall y \, P(x_0,y)$	assum
5	$P(x_0,y_0)$	$\forall y$ e 4
6	$\exists x \, P(x,y_0)$	$\exists x$ i 5
7	$\exists x P(x,y_0)$	$\exists x$ e 1, 3–6
8	$\forall y \, \exists x \, P(x,y)$	$\forall y$ i 2–7

Note that we have to open a y_0-box first, since we mean to show a formula of the form $\forall y \, \psi$. Then we could open an x_0-box to use $\exists x$ e.

9(a). We prove the validity of $\exists x \, (S \to Q(x)) \vdash S \to \exists x \, Q(x)$ by

1	$\exists x \, (S \to Q(x))$	prem
2	S	assum
3	x_0	
4	$S \to Q(x_0)$	assum
5	$Q(x_0)$	$\to$e 4, 2
6	$\exists x \, Q(x)$	$\exists x$ i 5
7	$\exists x \, Q(x)$	$\exists x$ e 1, 3–6
8	$S \to \exists x \, Q(x)$	$\to$i 2–7

9(d). We prove the validity of $\forall x\, P(x) \to S \vdash \exists x\, (P(x) \to S)$ by

1	$\forall x\, P(x) \to S$	prem
2	$\neg\exists x\, (P(x) \to S)$	assum
3	x_0	
4	$\neg P(x_0)$	assum
5	$P(x_0)$	assum
6	$\bot$	$\neg$e 5, 4
7	S	$\bot$e 6
8	$P(x_0) \to S$	$\to$i 5–7
9	$\exists x\, (P(x) \to S)$	$\exists x$ i 8
10	$\bot$	$\neg$e 9, 2
11	$\neg\neg P(x_0)$	$\neg$i 4–10
12	$P(x_0)$	$\neg\neg$e 11
13	$\forall x\, P(x)$	$\forall x$ i 3–12
14	S	$\to$e 1, 13
15	$P(t)$	assum
16	S	copy 14
17	$P(t) \to S$	$\to$i 15–16
18	$\exists x\, (P(x) \to S)$	$\exists x$ i 17
19	$\bot$	$\neg$e 18, 2
20	$\neg\neg\exists x\, (P(x) \to S)$	$\neg$i 2–19
21	$\exists x\, (P(x) \to S)$	$\neg\neg$e 20

9(k). We prove the validity of $\forall x\,(P(x) \wedge Q(x)) \vdash \forall x\,P(x) \wedge \forall x\,Q(x)$ by

1	$\forall x\,(P(x) \wedge Q(x))$	prem
2	x_0	
3	$P(x_0) \wedge Q(x_0)$	$\forall x$ e 1
4	$P(x_0)$	$\wedge \mathrm{e}_1$ 3
5	$\forall x\,P(x)$	$\forall x$ i 2–4
6	x_0	
7	$P(x_0) \wedge Q(x_0)$	$\forall x$ e 1
8	$Q(x_0)$	$\wedge \mathrm{e}_2$ 7
9	$\forall x\,Q(x)$	$\forall x$ i 6–8
10	$\forall x\,P(x) \wedge \forall x\,Q(x)$	$\wedge$i 5, 9

9(l). We prove the validity of $\forall x\,P(x) \vee \forall x\,Q(x) \vdash \forall x\,(P(x) \vee Q(x))$ by

1	$\forall x\,P(x) \vee \forall x\,Q(x)$			prem
2	x_0			
3	$\forall x\,P(x)$	assum	$\forall x\,Q(x)$	assum
4	$P(x_0)$	$\forall x$ e 3	$Q(x_0)$	$\forall x$ e 3
5	$P(x_0) \vee Q(x_0)$	$\vee \mathrm{i}_1$ 4	$P(x_0) \vee Q(x_0)$	$\vee \mathrm{i}_2$ 4
6	$P(x_0) \vee Q(x_0)$			$\vee$e 1, 3–5, 3–5
7	$\forall x\,(P(x) \vee Q(x))$			$\forall x$ i 2–6

9(m). We prove the validity of $\exists x\,(P(x) \wedge Q(x)) \vdash \exists x\,P(x) \wedge \exists x\,Q(x)$ by

1	$\exists x\,(P(x) \wedge Q(x))$	prem
2	x_0	
3	$P(x_0) \wedge Q(x_0)$	assum
4	$P(x_0)$	$\wedge \mathrm{e}_1$ 3
5	$\exists x\,P(x)$	$\exists x$ i 4
6	$Q(x_0)$	$\wedge \mathrm{e}_2$ 3
7	$\exists x\,Q(x)$	$\exists x$ i 6
8	$\exists x\,P(x) \wedge \exists x\,Q(x)$	$\wedge$i 5, 7
9	$\exists x\,P(x) \wedge \exists x\,Q(x)$	$\exists x$ e 1, 2–8

9(n). We prove the validity of $\exists x\, F(x) \vee \exists x\, G(x) \vdash \exists x\, (F(x) \vee G(x))$ by

1	$\exists x\, F(x) \vee \exists x\, G(x)$				prem
2	$\exists x\, F(x)$	assum		$\exists x\, G(x)$	assum
3	x_0			x_0	
4	$F(x_0)$	assum		$G(x_0)$	assum
5	$F(x_0) \vee G(x_0)$	$\vee \mathrm{i}_1$ 4		$F(x_0) \vee G(x_0)$	$\vee \mathrm{i}_2$ 4
6	$\exists x\, (F(x) \vee G(x))$	$\exists x$ i 5		$\exists x\, (F(x) \vee G(x))$	$\exists x$ i 5
7	$\exists x\, (F(x) \vee G(x))$	$\exists x$ e 2, 3–6		$\exists x\, (F(x) \vee G(x))$	$\exists x$ e 2, 3–6
8	$\exists x\, (F(x) \vee G(x))$				$\vee$e 1, 2–7, 2–7

9(p). We prove the validity of $\neg\forall x \neg P(x) \vdash \exists x P(x)$ by

1	$\neg\forall x\, \neg P(x)$	prem
2	$\neg\exists x\, P(x)$	assum
3	x_0	
4	$P(x_0)$	assum
5	$\exists x\, P(x)$	$\exists x$ i 4
6	$\bot$	$\neg$e 5, 2
7	$\neg P(x_0)$	$\neg$i 4–6
8	$\forall x\, \neg P(x)$	$\forall x$ i 3–7
9	$\bot$	$\neg$e 8, 1
10	$\exists x\, P(x)$	RAA 2–9

9(q). We prove the validity of $\forall x\, \neg P(x) \vdash \neg\exists x P(x)$ by

1	$\forall x\, \neg P(x)$	prem
2	$\exists x\, P(x)$	assum
3	x_0	
4	$P(x_0)$	assum
5	$\neg P(x_0)$	$\forall x$ e 1
6	$\bot$	$\neg$e 4, 5
7	$\bot$	$\exists x$ e 2, 3–6
8	$\neg\exists x P(x)$	$\neg$i 2–7

9(r). We prove the validity of $\neg\exists x\, P(x) \vdash \forall x\, \neg P(x)$ by

1		$\neg\exists x\, P(x)$	prem
2	x_0		
3		$P(x_0)$	assum
4		$\exists x\, P(x)$	$\exists x$ i 3
5		$\bot$	$\neg$e 4, 1
6		$\neg P(x_0)$	$\neg$i 3–5
7		$\forall x\, \neg P(x)$	$\forall x$ i 2–6

Note that we mean to show $\forall x \neg P(x)$, so the x_0-box has to show the instance $\neg P(x_0)$ which is achieved via a proof by contradiction.

11(a). We prove the validity of $P(b) \vdash \forall x\, ((x = b) \to P(x))$ by

1		$P(b)$	prem
2	x_0		
3		$x_0 = b$	assum
4		$P(x_0)$	=e 3, 1
5		$(x_0 = b) \to P(x_0)$	$\to$i 3–4
6		$\forall x\, ((x = b) \to P(x))$	$\forall x$ i 2–5

11(c). We prove the validity of $\exists x \exists y (H(x, y) \vee H(y, x)), \neg\exists x H(x, x) \vdash$

$\exists x \exists y \neg (x = y)$ by

1	$\exists x\, \exists y\, (H(x,y) \vee H(y,x))$	prem
2	$\neg \exists x\, H(x,x)$	prem
3	x_o	
4	$\exists y\, (H(x_0,y) \vee H(y,x_0))$	assum
5	y_0	
6	$H(x_0,y_0) \vee H(y_0,x_0)$	assum
7	$x_0 = y_0$	assum
8	$H(x_0,y_0)$	assum
9	$H(y_0,y_0)$	=e 7, 8
10	$\exists x\, H(x,x)$	$\exists x$ i 9
11	$\bot$	$\neg$e 10, 2
12	$H(y_0,x_0)$	assum
13	$H(y_0,y_0)$	=e 7, 12
14	$\exists x\, H(x,x)$	$\exists x$ i 13
15	$\bot$	$\neg$e 14, 2
16	$\bot$	$\vee$e 6, 8−11, 12−15
17	$\neg(x_0 = y_0)$	$\neg$i 7−16
18	$\exists y\, \neg(x_0 = y)$	$\exists y$ i 17
19	$\exists x\, \exists y\, \neg(x = y)$	$\exists x$ i 18
20	$\exists x\, \exists y\, \neg(x = y)$	$\exists y$ e 4, 5−19
21	$\exists x\, \exists y\, \neg(x = y)$	$\exists x$ e 1, 3−20

12. We prove the validity of $S \to \forall x\, Q(x) \vdash \forall x\, (S \to Q(x))$ by

1	$S \to \forall x\, Q(x)$	prem
2	x_0	
3	$\quad S$	assum
4	$\quad \forall x\, Q(x)$	$\to$e 1, 3
5	$\quad Q(x_0)$	$\forall x$ e 4
6	$S \to Q(x_0)$	$\to$i 3–5
7	$\forall x\, (S \to Q(x))$	$\forall x$ i 2–6

13(a). We show the validity of

$$\forall x\, P(a, x, x), \forall x\, \forall y\, \forall z\, (P(x, y, z) \to P(f(x), y, f(z))) \vdash P(f(a), a, f(a))$$

by

1	$\forall x\, P(a, x, x)$	prem
2	$\forall x\, \forall y\, \forall z\, (P(x, y, z) \to P(f(x), y, f(z)))$	prem
3	$P(a, a, a)$	$\forall x$ e 1
4	$\forall y\, \forall z\, (P(a, y, z) \to P(f(a), y, f(z)))$	$\forall x$ e 2
5	$\forall z\, (P(a, a, z) \to P(f(a), a, f(z)))$	$\forall y$ e 4
6	$P(a, a, a) \to P(f(a), a, f(a))$	$\forall z$ e 5
7	$P(f(a), a, f(a))$	$\to$e 6, 3

13(b). We show the validity of

$$\forall x P(a, x, x), \forall x\, \forall y\, \forall z\, (P(x, y, z) \to P(f(x), y, f(z))) \vdash \exists z\, P(f(a), z, f(f(a)))$$

by

1	$\forall x P(a, x, x)$	prem
2	$\forall x\, \forall y\, \forall z\, (P(x, y, z) \to P(f(x), y, f(z)))$	prem
3	$P(a, f(a), f(a))$	$\forall x$ e 1
4	$\forall y\, \forall z\, (P(a, y, z) \to P(f(a), y, f(z)))$	$\forall x$ e 2
5	$\forall z\, (P(a, f(a), z) \to P(f(a), f(a), f(z)))$	$\forall y$ e 4
6	$P(a, f(a), f(a)) \to P(f(a), f(a), f(f(a)))$	$\forall z$ e 5
7	$P(f(a), f(a), f(f(a)))$	$\to$e 6, 3
8	$\exists z\, P(f(a), z, f(f(a)))$	$\exists z$ i 7

Note that we just had to add one more line to the proof of the previous item and instantiate x, y and z with $f(a)$ instead of a in lines 3, 4 and 5, respectively.

13(c). We prove the validity of

$$\forall y\, Q(b,y), \forall x\, \forall y\, (Q(x,y) \to Q(s(x),s(y))) \vdash \exists z\, (Q(b,z) \wedge Q(z,s(s(b))))$$

by

1	$\forall y\, Q(b,y)$	prem
2	$\forall x\, \forall y\, (Q(x,y) \to Q(s(x),s(y)))$	prem
3	$\forall y\, (Q(b,y) \to Q(s(b),s(y)))$	$\forall x$ e 2
4	$Q(b,s(b)) \to Q(s(b),s(s(b)))$	$\forall y$ e 3
5	$Q(b,s(b))$	$\forall x$ e 1
6	$Q(s(b),s(s(b)))$	$\to$e 4, 5
7	$Q(b,s(b)) \wedge Q(s(b),s(s(b))$	$\wedge$i 5, 6
8	$\exists z\, (Q(b,z) \wedge Q(z,s(s(b)))$	$\exists x$ i 7

EXERCISES 2.4 (p.163)

1. The truth value of $\forall x\, \forall y\, Q(g(x,y),g(y,y),z)$ depends only on the values that valuations assign to its free variables, i.e to the occurrence of z. We choose A to be the set of integers, $g^{\mathcal{M}}(a,a')$ is the result of subtracting a' from a, and a triple of integers (a,b,c) is in $Q^{\mathcal{M}}$ if, and only if, c equals the product of a and b. That way $g(y,y)$ is interpreted as 0 and, consequently, our formula says that 0 equals the value assigned to z by our valuation. So for $l(z) \stackrel{\text{def}}{=} 0$ the formula holds in our model, whereas for $l'(z) \stackrel{\text{def}}{=} 1$ it is false.

5(a). The formula $\forall x\, \forall y\, \exists z\, (R(x,y) \to R(y,z))$ is not true in the model $\mathcal{M}$: for example, we have $(b,a) \in R^{\mathcal{M}}$, but there is no $m \in A$ with $(a,m) \in R^{\mathcal{M}}$ contrary to what the formula claims. Thus, we may "defeat" this formula by choosing b for x and a for y to construct a counter-witness to the truth of this formula over the given model.

5(b). The formula $\forall x\, \forall y\, \exists z\, (R(x,y) \to R(y,z))$ *is* true in the model $\mathcal{M}'$: we may list all elements of R in a "cyclic" way as (a,b) (b,c) (c,b), the cycle being the last two pairs. Thus, for any choice of x and y we can find some z so that the implication $R(x,y) \to R(y,z)$ is true.

6. • We choose a model with A being the set of integers. We define $(n,m) \in P^{\mathcal{M}}$ if, and only if, n is less than or equal to m ($n \leq m$). Evidently,

this interpretation of P is reflexive (every integer is less than or equal to itself) and transitive: $n \leq m$ and $m \leq k$ imply $n \leq k$. However, $2 \leq 3$ and $3 \not\leq 2$ show that this interpretation cannot be symmetric.

- We choose as set A the sons of J. S. Bach. We interpret $P(x, y)$ as "x is a brother of y". Clearly, this relation is transitive and symmetric, but not reflexive.
- We define $A \stackrel{\text{def}}{=} \{a, b, c\}$ and

$$P^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a,a), (b,b), (c,c), (a,c), (a,b), (b,a), (c,a)\}.$$

Note that this interpretation is reflexive and symmetric. We also have that (b, a) and (a, c) are in $P^{\mathcal{M}}$. Thus, we would need $(b, c) \in P^{\mathcal{M}}$ to secure transitivity of $P^{\mathcal{M}}$. Since this is not the case, we infer that this interpretation of P is not transitive.

8. To show the semantic entailment $\forall x\, P(x) \vee \forall x\, Q(x) \vDash \forall x\, (P(x) \vee Q(x))$ let $\mathcal{M}$ be any model such that $\mathcal{M} \vDash \forall x\, P(x) \vee \forall x\, Q(x)$. Then $\mathcal{M} \vDash \forall x\, P(x)$ or $\mathcal{M} \vDash \forall x\, Q(x)$. Thus, either all elements of A are in $P^{\mathcal{M}}$, or they are all in $Q^{\mathcal{M}}$. In any case, every element of A is in the union $P^{\mathcal{M}} \cup Q^{\mathcal{M}}$. Therefore, $\mathcal{M} \vDash \forall x\, (P(x) \vee Q(x))$ follows.

9(b). To see that $\phi \wedge \eta \vDash \psi$ does not imply $\phi \vDash \psi$ and $\eta \vDash \psi$ in general consider the special case where ψ is just ϕ again. Then $\phi \wedge \eta \vDash \phi$ and $\phi \vDash \phi$ are clearly the case, but we cannot guarantee that $\eta \vDash \phi$ holds as well. For example, we could pick η to be $\exists x\, P(x)$ and ϕ to be $\forall x\, P(x)$.

11(b). The collection of formulas $\forall x\, \neg S(x, x)$, $\forall x\, \exists y\, S(x, y)$, $\forall x\, \forall y\, \forall z\, ((S(x, y) \wedge S(y, z)) \rightarrow S(x, z))$ says that the interpretation of S is irreflexive, serial (see Chapter 5), and transitive. There are plenty of such relations around. For example, consider $S(x, y)$ to be interpreted over the natural numbers as meaning "x is strictly less than y" $(x < y)$. Clearly, this is irreflexive and transitive. It is also serial since $n < n + 1$ for all natural numbers n.

11(d). The collection of formulas $\exists x\, S(x, x)$, $\forall x\, \forall y\, (S(x, y) \rightarrow (x = y))$ says that S should be interpreted as equality and that at least one element is equal to itself. But since the interpretation of equality over any model is reflexive ($(a, a) \in =^{\mathcal{M}}$ for all $a \in A$) this is true for all models which interpret equality in the standard way (and where A is non-empty, which we assumed in our definition of models).

12(b) This is valid. To prove it, use $\exists y\, \mathrm{i}$ with dummy variable y_0 at the toplevel and within its box use $\rightarrow \mathrm{i}$ to show $(\forall x P(x)) \rightarrow P(y_0)$ and then close the $\exists y\, \mathrm{i}$ box to conclude the desired formula.

12(g) This formula claims that any relation that is serial and anti-symmetric has no minimal element. Let the model be the set of natural numbers with where S is interpreted as "less than or equal." Then this relation is serial and anti-symmetric but 0 is a minimal element. So this formula is not valid.

EXERCISES 2.5 (p.164)

1(b). We interpret the sequent

$$\forall x(P(x) \to R(x)), \forall x(Q(x) \to R(x)) \vdash \exists x(P(x) \wedge Q(x))$$

by

$$\begin{aligned} P(x) &: \text{“}x \text{ is a cat.”} \\ Q(x) &: \text{“}x \text{ is a dog.”} \\ R(x) &: \text{“}x \text{ is an animal.”} \end{aligned}$$

Then $\forall x(P(x) \to R(x))$ and $\forall x(Q(x) \to R(x))$ are obviously true, but $\exists x(P(x) \wedge Q(x))$ is false, despite recent efforts in microbiology.

1(d). We interpret the sequent $\forall x \exists y S(x,y) \vdash \exists y \forall x S(x,y)$ over the integers by

$$S(x,y) \quad : \quad \text{“}y \text{ equals } 2 \cdot x\text{.”}$$

Clearly, every x has a y, namely $2 \cdot x$, such that $S(x,y)$ holds. But there is no integer y which equals $2 \cdot x$ *for all choices of* x.

1(f). For the sequent $\exists x\,(\neg P(x) \wedge Q(x)) \vdash \forall x\,(P(x) \to Q(x))$ no proof is possible since we can interpret P and Q over the integers such that $P(x)$ is saying "x is odd" and $Q(x)$ is saying that "x is even". In this model we have $\mathcal{M} \vDash \exists x\,(\neg P(x) \wedge Q(x))$, e.g. take 2 as a value of x, but we cannot have $\mathcal{M} \vDash \forall x\,(P(x) \to Q(x))$ since not all odd numbers are even (in fact, none of them are!).

1(g). For the sequent $\exists x\,(\neg P(x) \vee \neg Q(x)) \vdash \forall x\,(P(x) \vee Q(x))$ no proof is possible since we may interpret P and Q over the same domain of integers, but now $P(x)$ says that "x is divisible by 2", whereas $Q(x)$ says that "x is divisible by 3". Then we have $\mathcal{M}' \vDash \exists x\,(\neg P(x) \vee \neg Q(x))$ for this model, e.g. take 9 as the value of x; however, we cannot have $\mathcal{M}' \vDash \forall x\,(P(x) \vee Q(x))$ since not all integers are divisible by 2 or 3 (e.g. choose x to be 13).

EXERCISES 2.6 (p.165)

1(a) The formula $\exists P\phi$ was meant to be the one that specifies unreachability. Therefore, this does not hold here as s_1 is reachable from s_3 through the path $s_3 \to s_0 \to s_1$.

4. There are infinitely many such paths (you are welcome to think about the degree of infinity here). For example, we can travel from s_3 to s_0 and then cycle between s_1 and s_0 for any finite number of times before we travel from s_1 to s_2.

5(a) We can specify such a formula by $\exists P(\forall x \forall y(P(x,y) \to R(x,y))) \wedge (\forall x P(x,x)) \wedge (\forall x \forall y(P(x,y) \to P(y,x)))$.

5(e) We specify this by saying there is some P which contains exactly all pairs that have a directed R-path of length 2 and that is transitive. So this may read as $\exists P(\forall x \forall y(P(x,y) \Leftrightarrow \exists z(R(x,z) \wedge R(z,y)))) \wedge (\forall x \forall y \forall z(P(x,y) \wedge P(y,z) \to P(x,z)))$.

6. We use an instance of the law of the excluded middle. Either A is in A or it isn't. If A is in A, then by definition of A it is not in A, contradiction. If A is not in A, then by definition of A it is actually of member of A, a contradiction. So no matter what case, we derive a contradiction.

8(a) This formula claims that there is a binary relation P with the property that any P-edge excludes its reverse P-edge and that any R-edge has a reverse P-edge. This cannot be in the given model. For example, there are R-edges from s_1 to s_0 and vice versa so they would imply P-edges from s_0 to s_1 and vice versa, respectively. But this contradicts the requirement on P-edges, namely that a P-edge from s_0 to s_1 excludes the possibility of a P-edge from s_1 to s_0.

9(b) We can achieve this by using universal second-order logic to quantify over all transitive relations that contain R and to then make the desired statement about all such transitive relations: $\forall P(\forall x \forall y(R(x,y) \to P(x,y))) \wedge (\forall x \forall y \forall z(P(x,y) \wedge P(y,z) \to P(x,z))) \to (\forall x P(x,x))$.

9(e) The equivalence relation that identifies no two distinct elements is maximal in this sense and so $\forall x \forall y R(x,y) \leftrightarrow (x = y)$ is a correct encoding in predicate logic.

EXERCISES 2.7 (p.166)

1(a) Both `Person_1` and `Person_2` are members of `cast`, but `Person_2` loves `Person_0` which is `alma` and the `Person_1` loves `alma` and `Person_2`. Therefore this is a counterexample to `OfLovers` as no person in this graph loves him or herself.

1(c). A model with only two persons is such that one of them has to be `alma` and then the other person has to love `alma`. But the only way to then get a counterexample is to then have another love relationship which, necessarily, has to involve self-love.

2(b)
```
fun SevenNodesEachHavingFiveReachableNodes(G : Graph) {
  with G {
    # nodes = 7
    all n : nodes | # n.^edges = 5
  }
} run SevenNodesEachHavingFiveReachableNodes for 7 but 1 Graph
```

4(a-e) Here is a file `ColoredGraphs.als` with a possible solution:

```
module ColoredGraphs

sig Color {}
sig Node {
  color : Color
}

sig AboutColoredGraphs {
  nodes : set Node,
  edges : nodes -> nodes
}{ edges = ~ edges && all x : nodes, y : x.edges | not x.color = y.color }

fun TwoColorable(g : AboutColoredGraphs) {
  with g {
    # nodes.color = 2
    # nodes >= 3
    all x : nodes | some x.edges -- to see more interesting graphs
  }
} run TwoColorable for 4 but 1 AboutColoredGraphs


fun ThreeColorable(g : AboutColoredGraphs) {
  with g {
    # nodes.color = 3
    # nodes >= 3
    all x : nodes | some x.edges
  }
} run ThreeColorable for 4 but 1 AboutColoredGraphs
```

```
fun FourColorable(g : AboutColoredGraphs) {
  with g {
    # nodes.color = 4
    # nodes >= 3
    all x : nodes | some x.edges
  }
} run FourColorable for 4 but 1 AboutColoredGraphs

-- we cannot specify that a graph is 3-colorable but not
2-colorable, for in order to do this -- we need to say ``there is
a 3-coloring'' that works for this graph and ``all 2-colorings''
-- don't work; the presence of the existential and universal
quantifiers over relations are -- what makes this not expressible
in Alloy or any other tool that attempts to reduce this to -- SAT
solving for propositional logic
```

5(a-f) Here is a file `KripkeModels.als` with a possible solution:

```
module KripkeModel sig Prop {}

sig State {
  labels : set Prop -- those propositions that are true at this state
}

sig StateMachine {
  states : set State,
  init, final : set states,
  next : states -> states
}{ some init }

fun Reaches(m : StateMachine, s : set State) {
  with m {
    s = init.*next
  }
} run Reaches for 3 but 1 StateMachine

fun DeadlockFree(m : StateMachine) {
  with m {
    init.*next & { x : states | no x.next } in final
```

```
  }
} run DeadlockFree for 3 but 1 StateMachine

fun Deterministic(m : StateMachine) {
  with m {
    all x : init.*next | sole x.next
  }
} run Deterministic for 3 but 1 StateMachine

fun Reachability(m : StateMachine, p : Prop) {
  with m {
    some init.*next & { s : states | p in s.labels }
  }
} run Reachability for 3 but 1 StateMachine

fun Liveness(m : StateMachine, p : Prop) {
  with m {
    all x : init.*next | some x.*next & { s : states | p in s.labels }
  }
} run Liveness for 3 but 1 StateMachine

assert Implies {
  all m : StateMachine, p : Prop | Liveness(m,p) => Reachability(m,p)
} check Implies for 3 but 1 StateMachine

assert Converse {
  all m : StateMachine, p : Prop | Reachability(m,p) => Liveness(m,p)
} check Converse for 3 but 1 StateMachine

fun SimulationForFiveDf(m : StateMachine) {
  with m {
    # states = 3
    some x : states | not sole x.next
    all x : states | no x.next => x in final
    all x,y : states | { p : Prop | p in x.labels } =
                       { p : Prop | p in y.labels } => x = y
  }
} run SimulationForFiveDf for 3 but 1 StateMachine
```

6(a) We write a signature for groups along with the necessary constraints (group axioms):

```
sig Group {
  elements: set Element,
  unit: elements,
  mult: elements -> elements ->! elements,
  inv: elements ->+ elements
}{ all a,b,c: elements | c.((b.(a.mult)).mult) =
(c.(b.mult)).(a.mult)
   all a: elements | a = unit.(a.mult) && a = a.(unit.mult)
   all a: elements | unit = a.((a.inv).mult) && (a.inv).(a.mult) = unit
}
```

where `elements` models G, `unit` models e, and `mult` models $\star$. Please note

- that the three axioms for multiplication are encoded as constraints attached to the signature; we display multiplication in a "curried" form);
- the role of `!` which ensures that any two group elements have a unique result of their multiplication; and
- the role of `+` stating that each element has at least one inverse.

6(b) The following `fun`-statement generates a group with three elements:

```
fun AGroup(G: Group) {
  # G.elements = 3
} run AGroup for 3 but 1 Group
```

6(c) We declare

```
assert Inverse {
  all G: Group, e: G.elements | sole e.(G.inv)
} check Inverse for 5 but 1 Group
```

and no counter-example is found within that scope; `sole S` denotes that `S` contains at most one element. The small-scope hypothesis therefore suggests that inverses are unique in all finite groups. In this case, the small-scope hypothesis got it right: inverses are unique in groups, even in infinite ones.

6(d) i. We declare

```
fun Commutes (G: Group) {
  all a,b: G.elements | a.(b.(G.mult)) = b.(a.(G.mult))
} run Commutes for 3 but 1 Group

assert Commutative {
  all G: Group | Commutes(G)
} check Commutative for 5 but 1 Group
```

ii. Analyzing the assertion above we find no solution. The small-scope hypothesis therefore suggests that all finite groups are commutative. This time, the small-scope hypothesis got it wrong! There are finite groups that are not commutative.

iii. In fact, increasing the scope from 5 to 6 reveals a violation to our goal. Please run this analysis yourself and inspect the navigable tree to determine where and how commutativity is broken.

6(e) Yes, the assertions are formulas that make a claim about *all* groups. So a counter-example exists iff it exists for a single group. We already achieved the restriction to one group with the `but 1 Group` in the `check` and `run` directives.