Remark 3.7 Notice that, in clauses 9–13 above, the future includes the present. This means that, when we say 'in all future states,' we are including the present state as a future state. It is a matter of convention whether we do this, or not. As an exercise, you may consider developing a version of LTL in which the future excludes the present. A consequence of adopting the convention that the future shall include the present is that the formulas $\mathrm{G}\, p \to p$, $p \to q \mathrel{\mathrm{U}} p$ and $p \to \mathrm{F}\, p$ are true in every state of every model.

Definition 3.8 Suppose $\mathcal{M} = (S, \to, L)$ is a model, $s \in S$, and ϕ an LTL formula. We write $\mathcal{M}, s \vDash \phi$ if, for every execution path π of $\mathcal{M}$ starting at s, we have $\pi \vDash \phi$.

If $\mathcal{M}$ is clear from the context, we may abbreviate $\mathcal{M}, s \vDash \phi$ by $s \vDash \phi$.

Here are some examples of LTL formulas:

- $(((\mathrm{F}\,p) \wedge (\mathrm{G}\,q)) \rightarrow (p \,\mathrm{W}\, r))$
- $(\mathrm{F}\,(p \rightarrow (\mathrm{G}\,r)) \vee ((\neg q) \,\mathrm{U}\, p))$, the parse tree of this formula is illustrated in Figure 3.1.
- $(p \,\mathrm{W}\, (q \,\mathrm{W}\, r))$
- $((\mathrm{G}\,(\mathrm{F}\,p)) \rightarrow (\mathrm{F}\,(q \vee s)))$.

BINDING PRIORITIES

Convention 3.2 The unary connectives (consisting of $\neg$ and the temporal connectives X, F and G) bind most tightly. Next in the order come U, R and W; then come $\wedge$ and $\vee$; and after that comes $\rightarrow$.

- $\mathrm{F}\,p \wedge \mathrm{G}\,q \rightarrow p \,\mathrm{W}\, r$
- $\mathrm{F}\,(p \rightarrow \mathrm{G}\,r) \vee \neg q \,\mathrm{U}\, p$
- $p \,\mathrm{W}\, (q \,\mathrm{W}\, r)$
- $\mathrm{G}\,\mathrm{F}\,p \rightarrow \mathrm{F}\,(q \vee s)$.

The following are *not* well-formed formulas:

- $\mathrm{U}\,r$ – since U is binary, not unary
- $p\,\mathrm{G}\;q$ – since G is unary, not binary.

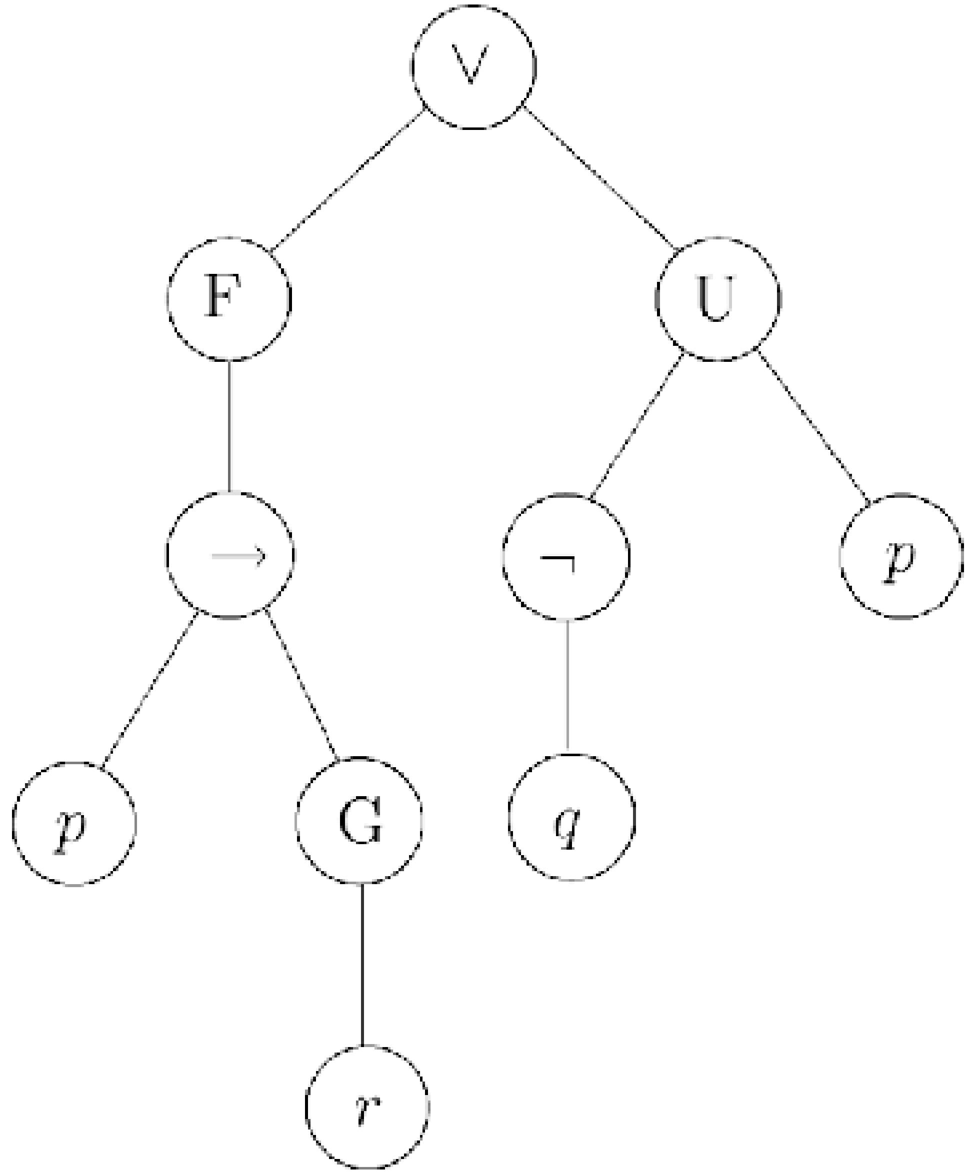

Figure 3.1. The parse tree of $(\mathrm{F}\,(p \to \mathrm{G}\,r) \vee ((\neg q)\ \mathrm{U}\ p))$.

Definition 3.3 A subformula of an LTL formula ϕ is any formula ψ whose parse tree is a subtree of ϕ's parse tree.

The subformulas of $p\ \mathrm{W}\ (q\ \mathrm{U}\ r)$, e.g., are p, q, r, $q\ \mathrm{U}\ r$ and $p\ \mathrm{W}\ (q\ \mathrm{U}\ r)$.

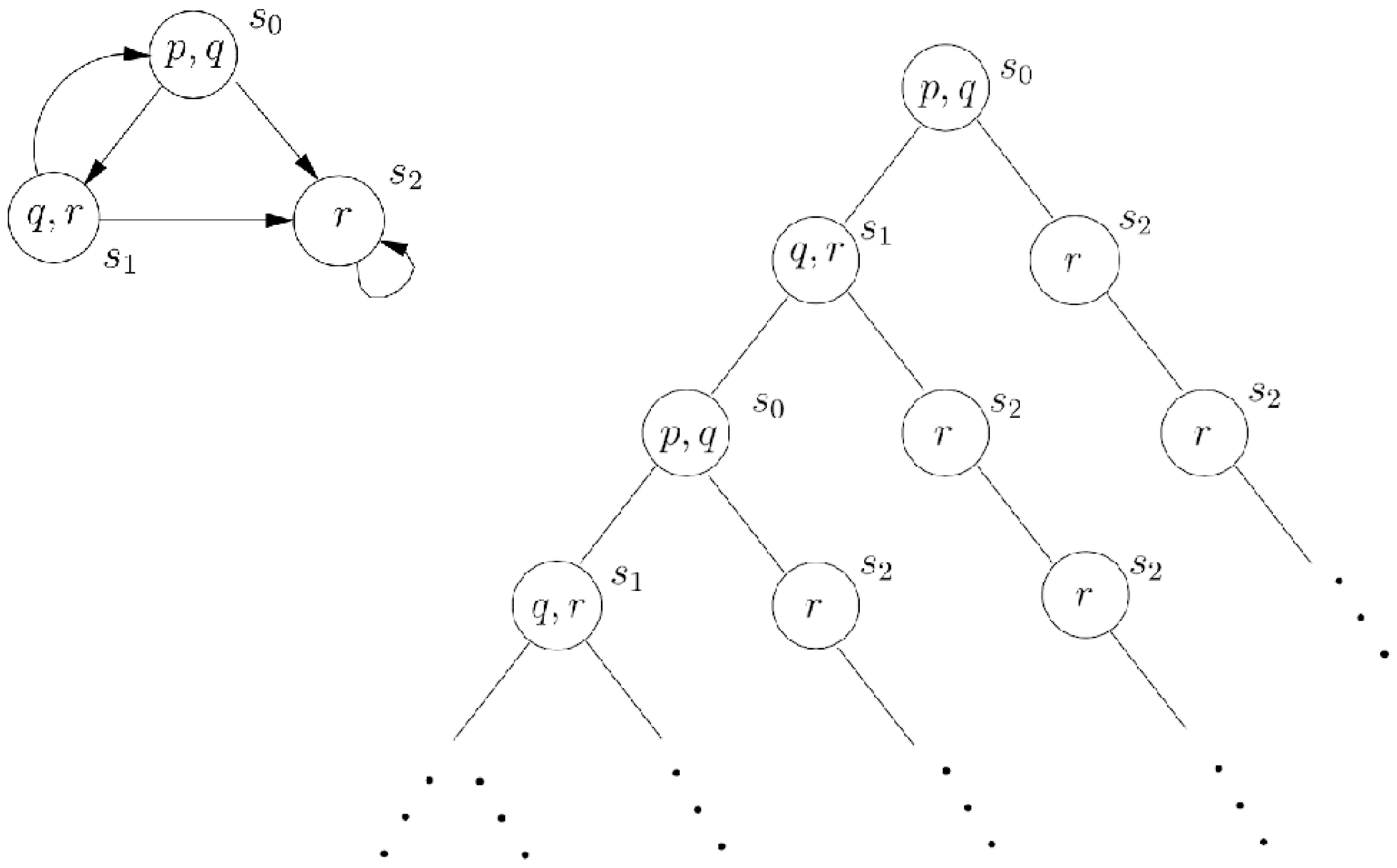

Figure 3.5. Unwinding the system of Figure 3.3 as an infinite tree of all computation paths beginning in a particular state.

EXAMPLES

1. $\mathcal{M}, s_0 \vDash p \wedge q$ holds since the atomic symbols p and q are contained in the node of s_0: $\pi \vDash p \wedge q$ for *every* path π beginning in s_0.
2. $\mathcal{M}, s_0 \vDash \neg r$ holds since the atomic symbol r is *not* contained in node s_0.
3. $\mathcal{M}, s_0 \vDash \top$ holds by definition.
4. $\mathcal{M}, s_0 \vDash \mathrm{X}\, r$ holds since all paths from s_0 have either s_1 or s_2 as their next state, and each of those states satisfies r.
5. $\mathcal{M}, s_0 \vDash \mathrm{X}\,(q \wedge r)$ does not hold since we have the rightmost computation path $s_0 \to s_2 \to s_2 \to s_2 \to \dots$ in Figure 3.5, whose second node s_2 contains r, but not q.
6. $\mathcal{M}, s_0 \vDash \mathrm{G}\,\neg(p \wedge r)$ holds since all computation paths beginning in s_0 satisfy $\mathrm{G}\,\neg(p \wedge r)$, i.e. they satisfy $\neg(p \wedge r)$ in each state along the path. Notice that $\mathrm{G}\,\phi$ holds in a state if, and only if, ϕ holds in all states reachable from the given state.
7. For similar reasons, $\mathcal{M}, s_2 \vDash \mathrm{G}\, r$ holds (note the s_2 instead of s_0).
8. For any state s of $\mathcal{M}$, we have $\mathcal{M}, s \vDash \mathrm{F}\,(\neg q \wedge r) \to \mathrm{F}\,\mathrm{G}\, r$. This says that if any path π beginning in s gets to a state satisfying $(\neg q \wedge r)$, then the path π satisfies $\mathrm{F}\,\mathrm{G}\, r$. Indeed this is true, since if the path has a state satisfying $(\neg q \wedge r)$ then (since that state must be s_2) the path does satisfy $\mathrm{F}\,\mathrm{G}\, r$. Notice what $\mathrm{F}\,\mathrm{G}\, r$ says about a path: eventually, you have continuously r.

9. The formula $\mathrm{G}\,\mathrm{F}\,p$ expresses that p occurs along the path in question infinitely often. Intuitively, it's saying: no matter how far along the path you go (that's the G part) you will find you still have a p in front of you (that's the F part). For example, the path $s_0 \to s_1 \to s_0 \to s_1 \to \ldots$ satisfies $\mathrm{G}\,\mathrm{F}\,p$. But the path $s_0 \to s_2 \to s_2 \to s_2 \to \ldots$ doesn't.
10. In our model, if a path from s_0 has infinitely many ps on it then it must be the path $s_0 \to s_1 \to s_0 \to s_1 \to \ldots$, and in that case it also has infinitely many rs on it. So, $\mathcal{M}, s_0 \vDash \mathrm{G}\,\mathrm{F}\,p \to \mathrm{G}\,\mathrm{F}\,r$. But it is not the case the other way around! It is not the case that $\mathcal{M}, s_0 \vDash \mathrm{G}\,\mathrm{F}\,r \to \mathrm{G}\,\mathrm{F}\,p$, because we can find a path from s_0 which has infinitely many rs but only one p.

Definition 3.9 We say that two LTL formulas ϕ and ψ are semantically equivalent, or simply equivalent, writing $\phi \equiv \psi$, if for all models $\mathcal{M}$ and all paths π in $\mathcal{M}$: $\pi \vDash \phi$ iff $\pi \vDash \psi$.

In propositional logic, we saw that $\wedge$ and $\vee$ are duals of each other, ing that if you push a $\neg$ past a $\wedge$, it becomes a $\vee$, and vice versa:

$$\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi \qquad \neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi.$$

Similarly, F and G are duals of each other, and X is dual with itself:

$$\neg \mathrm{G}\, \phi \equiv \mathrm{F}\, \neg\phi \qquad \neg \mathrm{F}\, \phi \equiv \mathrm{G}\, \neg\phi \qquad \neg \mathrm{X}\, \phi \equiv \mathrm{X}\, \neg\phi.$$

Also U and R are duals of each other:

$$\neg(\phi \; \mathrm{U} \; \psi) \equiv \neg\phi \; \mathrm{R} \; \neg\psi \qquad \neg(\phi \; \mathrm{R} \; \psi) \equiv \neg\phi \; \mathrm{U} \; \neg\psi.$$