$(p \wedge q) \to (r \vee s)$.
We use the same principle as in elementary algebra: deal with the brackets first. Suppose we denote the output of $p \wedge q$ by x and the output of $r \vee s$ by y. Then the required output is $x \to y$. The AND-gate gives output x for inputs p, q; the OR-gate gives output y for inputs r, s; the network shown in Figure 3.9 gives output $x \to y$, for inputs x and y.

Concatenating these components gives the network shown in Figure 3.10, where the final output $z = (p \wedge q) \to (r \vee s)$. □

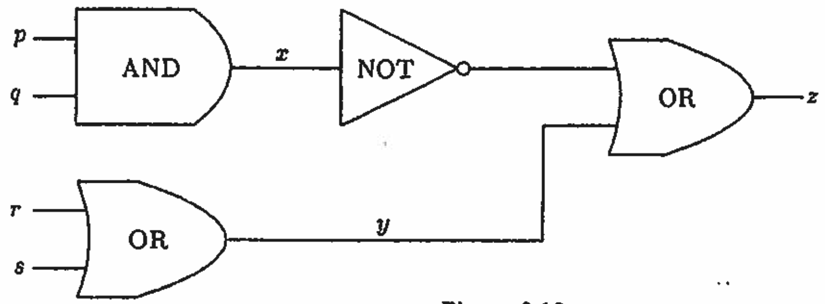

Figure 3.10.

3.4.2 Output of a given network

Suppose we are given the diagram of a logic network. We can reverse the process described above to obtain an expression for the output.

Example 3.14 We determine the output of the logic network shown in Figure 3.11. To do this, we work from left to right across the diagram, determining the output of each gate in turn. Note that the filled circles represent junctions where the inputs branch, whereas other points where lines cross represent bridges, where the inputs are insulated from one another.
We obtain:

1. Output $r = \neg q$.
2. Output $s = p \wedge \neg q$.
3. Output $t = p \wedge q$.
4. Output $w = (p \wedge \neg q) \vee (p \wedge q)$.

Hence the output of this network is $(p \wedge \neg q) \vee (p \wedge q)$. □

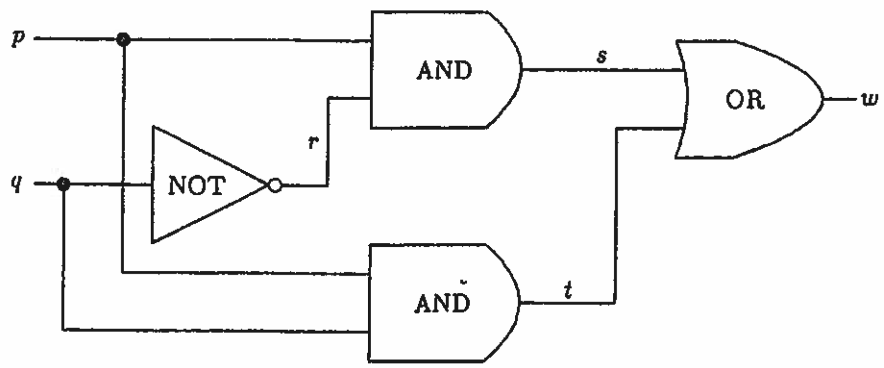

Figure 3.11.

We can next ask whether it is possible to simplify the network shown in Figure 3.11, by finding a network with fewer gates that gives the same output for every combination of inputs p and q. There