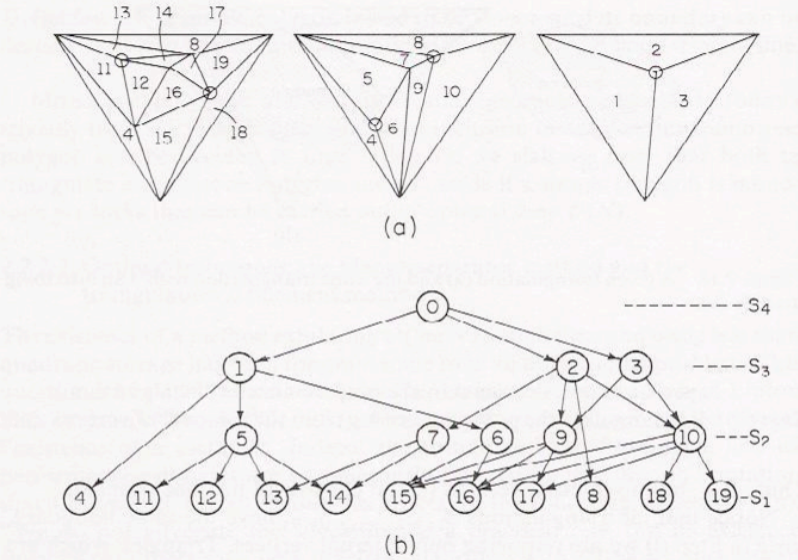

Figure 2.19 A sequence of triangulations (a) and the corresponding search-directed acyclic graph (b).

have reached a node in the path (i.e., we have located z in the corresponding triangle) we test z for inclusion in all of its descendants; since z is included in exactly one descendant, this advances the path one more arc. We may also view the search as the successive location of z in triangulations $S_{h(N)}, S_{h(N)-1}, \ldots$; since S_{i-1} is a refinement of S_i, this justifies the name of the technique.

Less informally, we assume that all descendants of a node v of T be arranged in a list $\Gamma(v)$, and let TRIANGLE(v) denote the triangle corresponding to node v. We then have the following search algorithm:

```
    procedure POINT-LOCATION
begin if (z ∉ TRIANGLE(root)) then print "z belongs to unbounded region"
    else begin v := root;
            while (Γ(v) ≠ ∅) do
                for each u ∈ Γ(v) do if (z ∈ TRIANGLE(u)) then v := u;
            print v
        end
end.
```

As mentioned earlier, the choice of the set of triangulation vertices to be removed in constructing S_i from S_{i-1} is crucial to the performance of the technique. Suppose we are able to choose this set so that, denoting by N_i the number of vertices of S_i, the following properties hold: