



## 11.2 REGION CONTOURS

**Fig. 11.9**

Combined region labeling and contour following (after [23]). The image is traversed from the top left to the lower right a row at a time. In (a), the first point  $A$  on the outer edge of the region is found. Starting from point  $A$ , the pixels on the edge along the outer contour are visited and labeled until  $A$  is reached again. In (b), the first point  $B$  on an inner contour is found. The pixels along the inner contour are visited and labeled until arriving back at  $B$  (c). In (d), an already labeled point  $C$  on an inner contour is found. Its label is propagated along the image row within the region.

### 11.2.3 Implementation

The complete implementation of the algorithm in Java (ImageJ) can be found in Appendix D (beginning on page 532). The implementation closely follows the description in Algs. 11.3 and 11.4 but illustrates several additional details:<sup>4</sup>

- First the image  $I$  (`pixelArray`) and the associated label map  $LM$  (`labelArray`) are enlarged by adding one pixel around the borders. The new pixels are marked as *background* (0) in the image  $I$ . This simplifies contour following and eliminates the need to handle a number of special situations.
- As contours are found they are stored in an object of the class `ContourSet`, separated into outer and inner contours. The contours themselves are represented by the classes `OuterContour` and `InnerContour`, with a common superclass `Contour`. Every contour consists of an ordered sequence of coordinate points of the class `Node`

<sup>4</sup> In the following description the names in parentheses after the algorithmic symbols denote the corresponding identifiers used in the Java implementation.