1. Trivial. Of the given data points, only two of them can be classified: (B,G,I,K,N) into class c1 and (B,F,J,K,M) into class c2. The other two cannot be classified in the induced tree (this example points out the importance of having default classifications).

2. Trivial. You will notice that all conditions have at most two attribute-value tests in their conjunctions. Here you will be able to classify all four test data points (the predicted classes will slightly vary depending on the choices made in forming the decision list).

3. If your answer was that ‘you will have one rule per class if the region indicated by the class is contiguous’ you have merely found a necessary condition, but not a sufficient one. A sufficient condition requires that the entire space of classifications be captured by recursive binary splitting. For instance, the figure on the left has contiguous regions for the classes but cannot be modeled so that each class gets exactly one rule. It is possible to get a recursive binary split into the five regions for the figure on the right.

4. If the test involved a categorical attribute and we branched on the value of this attribute, we will never test this attribute along the same path again, since the entropy with respect to this attribute would be zero. If the attribute was categorical and we formed a boolean (or set containment) condition of the form var = val? or var ∈ {⋯}? we might have occasion to test this attribute again on a given path. If the test involved a numeric or ordinal attribute of the form var < val, it will most likely be required to test this attribute again, especially for cases when the region of interest cannot be captured by recursive binary splitting. A more general statement to make is that there is an interaction between the attribute and the nature of the attribute-value test being applied. Unless they together cause the class conditional entropy with respect to the attribute to fall to zero, it might be required to test the attribute again. Of course, whether that attribute is chosen again in the path depends on the information content of other attributes and the bias of the induction algorithm.

5. Let us consider a decision tree algorithm that uses gain (improvement in entropy) as the evaluation criterion for splits. To make the mined decision tree resemble a decision list, we need to restrict our node conditions to only attribute-value tests that yield a boolean answer (e.g., a region containment test).

Notice that a decision list algorithm conducts bump hunting (looking for peaks that can be removed from the data) whereas decision tree algorithms look for regularities in terms of partitions. To bring a marriage between the two, the most informative partition capturable by the decision tree algorithm must have the most conspicuous peak on its true branch. Recursing this argument, we will find that the partition mined by the decision tree algorithm must be
locally decomposable into the sequence of peaks, in descending order of their evaluation criterion (e.g., support, or confidence). This is necessary because the left branch of the decision tree should always have an entropy of zero. Another way of saying this is that the removal of data by the decision list algorithm corresponds to the best possible specialization in the decision tree algorithm.