```
BinaryTree (E, T): trait
  % One of the many interesting tree structures
  introduces
    [__]: E → T
    [__, __]: T, T → T
    content: T → E
    first, second: T → T
    isLeaf: T → Bool
  asserts
    T generated by [__], [__, __]
    T partitioned by content, first, second, isLeaf
    ∀ e: E, t1, t2: T
      content([e]) == e;
      first([t1, t2]) == t1;
      second([t1, t2]) == t2;
      isLeaf([e]);
      ¬isLeaf([t1, t2])
  implies converts isLeaf
```