

- 那Redux模式是活的吗？
- 下面是一个Reducer的实现～上面的两个case就是state的抽象（上框），而下面的就是根据Action和现state去调用函数，并返回未来的状态（下框）

```
enum KeyboardState {  
    // 行号和处于行的状态  
  
    case startline(lineNumber : Int)  
    case inline(lineNumber : Int)  
  
    // apply是一个状态机函数， 它根据旧状态，用户的行动，确定新的状态  
    func apply(inparser : Parser, thisaction : KeyboardAction) -> KeyboardState {  
        switch self {  
            case .startline( _):  
                switch thisaction {  
                    case .editinline(let lineNumber, let newstring):  
                        // 这里随便写了parser的调用的函数  
                        inparser.chageOutline(lineid: lineNumber, pageid: 0, newstring: newstring)  
                        inparser.drawSingleSlide()  
                        return .inline(lineNumber: lineNumber)  
                    default:  
                        inparser.drawSingleSlide()  
                        return .startline(lineNumber: 0)  
                }  
            default:  
                switch thisaction {  
                    case .deleteline(lineNumber: 0):  
                        inparser.drawSingleSlide()  
                        return .startline(lineNumber: 0)  
                    default:  
                        inparser.drawSingleSlide()  
                }  
        }  
    }  
}
```

- 所以函数由Reducer调用，而调用哪一个函数由当前的状态决定