```
allRolls :: DiceChoice -> DiceState -> [ DiceState ]
allRolls choices (vs. n) = case pop choices vs of
  Nothing -> [ ([], n-1) ]
  Just ((chosen. v). (choices. vs)) -> do
    (roll.  ) <- allRolls choices (vs. n-1)
    [ (d:roll,  n-1) | d <- rollList ]
```

```
allRolls :: DiceChoice -> DiceState -> [ DiceState ]
allRolls choices (vs. n) = case pop choices vs of
  Nothing -> fmap (, n-1) [ [] ]
  Just ((chosen. v). (choices. vs)) -> do
    (roll.  ) <- allRolls choices (vs. n-1)
    fmap (, n-1) [ d:roll | d <- rollList ]
```