Based on

```
(define (solvable? m)
  (local [(define (fn-for-pos p)
           (... p
               (fn-for-lop (pos-subs p))))

         (define (fn-for-lop lop)
           (cond [(empty? lop) (...)]
                  [else
                   (...   (fn-for-pos (first lop))
                         (fn-for-lop (rest lop)))))))]

    (fn-for-pos ...)))
```

arb-tree

encapsulated

```
(define (genrec-fn d)
  ;; base case:
  ;; reduction:
  ;; argument:
  (cond [(trivial? d) (trivial-answer d)]
        [else
         (... d
              (genrec-fn (next-problem d)))]))
```

genrec

```
(if (not (false? (fn-for-pos (first lop))
    (fn-for-pos (first lop))
    (fn-for-pos (rest lop)))))])))]
```

try-catch