diagnosticate(dctor,observation)

```
1: {ricordando che dctor = diagnosticatore}
2: sHashMap ← addInitialDctorStateToHashMap(dctor)
3: obs ← tail(observation)
4: while obs ≠ NIL do
5:    newHashmap ← createHashMap()
6:    for each state in sHashMap do
7:       checkOutTransitions(newHashMap, state)
8:    end for
9:    obs ← prev[obs]
10: end while
11: sHashMap ← newHashMap
12: diagnosis ← NIL
13: initialized ← FALSE
14: for each state in sHashMap do
15:    lab ← NIL
16:    {delta contiene l'espressione regolare calcolata fino a state}
17:    if delta[state] ≠ NIL then
18:       deltaLab ← createNewRelevanceLabel(delta[state])
19:       addLabelToDiagnosis(state, deltaLab, initialized, diagnosis)
20:    end if
21: end for
22: return diagnosis
```

addInitialDctorStateToHashMap(dctor)

```
1: sHashmap ← createHashMap()
2: initial ← initial[dctor]
3: lookup ← createLookup(id[initial])
4: item ← createItem(lookup, initial)
5: hashMapInsert(sHashmap, item)
6: return sHashMap
```

checkOutTransitions(hashMap)

```
1: transitionsOut ← trOut[state]
2: while transitionsOut ≠ NIL do
3:    if id[obs] = id[obs[transitionsOut]] then
4:       newLabel ← concatenateLabel(value[state], rel[transitionsOut])
5:       dest ← dest[transitionsOut]
6:       destItem ← hashMapSearch(hashmap, id[dest])
7:       if destItem ≠ NIL then
8:          value[dest] ← alternateLabel(value[dest], newLabel)
9:       else
10:         lookup ← createLookup(id[dest])
11:         it ← createItem(lookup, dest)
12:         hashMapInster(hashmap, it)
13:      end if
14:   end if
15:   transitionsOut ← next[transitionsOut]
16: end while
```

addLabelToDiagnosis(state,deltaLab,initialized,diagnosis)

```
1: if final[state] = TRUE then
2:    if initialized = TRUE then
3:       l ← concatenateLabel(value[state], deltaLab)
4:       newLabel ← alternateLabel(diagnosis, l)
5:       diagnosis ← newLabel
6:    else
7:       diagnosis ← concatenateLabel(value[state], deltaLab)
8:       initialized ← TRUE
9:    end if
10: end if
```