TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Theoretische Informatik, Lehrstuhl für Automatentheorie

Theoretische Informatik und Logik

4. Übungsblatt

Sommersemester 2017

```
-- (c) 2017, Toni Dietze

main :: IO ()
main = putStr $ unlines $ map show $ head $ pkp instanceTwo

type PKPPair a = ([a], [a])

pkp :: Eq a => [PKPPair a] -> [[PKPPair a]]
pkp ps = [reverse ps' | (ps', [], []) <- pkpCands ps]


pkpCands :: Eq a => [PKPPair a] -> [([PKPPair a], [a], [a])]
pkpCands ps = tail cands
  where
    cands = ([], [], []) : concatMap (\ c -> fltr $ map (step c) ps) cands
    fltr = filter (\ (_, ls, rs) -> null ls || null rs)
    step (ps', ls', rs') p@(ls, rs) = (p : ps', ls'', rs'')
      where
        (ls'', rs'') = dropEqPrefix (ls' ++ ls) (rs' ++ rs)

dropEqPrefix :: Eq a => [a] -> [a] -> ([a], [a])
dropEqPrefix (x : xs) (y : ys) | x == y = dropEqPrefix xs ys
dropEqPrefix xs ys = (xs, ys)

instanceTwo :: [PKPPair Char]
instanceTwo = [("bba", "b"),
               ("ba", "aba"),
               ("ba", "baa"),
               ("ab", "bba")]
```

Lösung (ba,baa) (ab,bba) (ba,aba) (ab,bba) (ab,bba) (ba,baa) (bba,b) (ba,baa) (ab,bba) (ba,aba) (ab,bba) (ba,aba) (ab,bba) (ab,bba) (ba,aba) (ab,bba) (ab,bba) (ba,baa) (bba,b) (ab,bba) (ab,bba) (ba,baa) (bba,b) (ba,aba) (ab,bba) (bba,b) (bba,b) (ba,aba) (ab,bba) (ab,bba) (ab,bba) (ba,baa) (bba,b) (ba,baa) (bba,b) (bba,b) (bba,b) (ba,aba) (ab,bba) (ba,aba) (ab,bba) (bba,b) (ba,baa) (bba,b) (ab,bba) (ab,bba) (ba,baa) (bba,b) (ab,bba) (bba,b) (bba,b) (ba,aba) (ab,bba) (bba,b) (bba,b) (ba,aba) (bba,b) (bba,b) (ba,aba) (bba,b) (ba,baa) (bba,b) (ab,bba) (bba,b) (bba,b) (ba,aba) in 0.85 Sekunden auf einem alten Lenovo X220.