

Lec 23 More DP

Min Edit Dist Problem

Minimise the numbers of insertions and deletions to go from S:str to T:str

Ex. ABCADA → ABADC in 3 edits

Alg:

MED(S, T) =

let

$$\begin{aligned}
 \text{MED}'(i, j) &= \text{case } (i, j) \text{ of} && \begin{array}{l} \text{len-}i \text{ prefix of } S \\ \text{.. } j \text{ .. of } T \end{array} \\
 (0, j) &\Rightarrow j \\
 (i, 0) &\Rightarrow i \\
 (i, j) &\Rightarrow \begin{cases} \text{MED}'(i-1, j-1) & \text{if } S[i-1] = S[j-1] \\ \min \left\{ \begin{array}{l} \text{MED}'(i-1, j) + 1 \\ \text{MED}'(i, j-1) + 1 \end{array} \right\} & \text{else} \end{cases}
 \end{aligned}$$

in

MED'(|S|, |T|)

end

△ Exponential, but allows subinstance result reuse here

Analysis

(|S|+1)(|T|+1) unique subinstances
 each subinstance has constant local work
 so O(|S||T|) work, O(|S|+|T|) span

Bottom up impl

	0	1	2	3	4	5	6
0 -	0	1	2	3	4	5	6
1 A	1	0	1	2	3	4	5
2 B	2	1	0	1	2	3	4
3 A	3	2	1	2	1	2	3
4 D	4	3	2	3	2	1	2
5 C	5	4	3	2	3	2	3
6 A	6	5	2	3	2	3	2

Memorisation imp

(Magic) Memoised version of f

```
fun f g (i,j) = case (i,j) of
  (0,j) => j
| (i,0) => i
| (i,j) => { g (i-1,j-1) if S[i-1] = S[j-1]
            min { g (i-1,j) + 1
                  g (i,j-1) + 1 } else
```

← Memoiser lib

```
val MED' = memoiser.memoise (f)
```

Memorisation lib

```
fun memoise f =
  let
```

```
    val cache = ref (Table.empty ())
    fun g a = (case find (!cache, a) of
      SOME r => r
    | NONE => let
      val r = f g a
      val _ = cache := insert (!cache, a, r)
    in r end
```

```
  in
```

```
    g
  end
```

← Not thread safe ⚠

$g: \alpha \rightarrow \beta$

$f: (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$

$memoise: ((\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta)$

Ex. memoised fibb

```
fun f g n =
  if n <= 1 then 1 else
  g (n-1) + g (n-2)
```

```
val fib' = memoiser.memoise f
```