CS799 Ridiculously Advanced Systems
Project Report

Peter A. Wiseguy

8th July, 2018

Abstract

This document is a simple template for a typical term or semester paper (lab/course report, "Übungsbericht", etc.) based on the HagenbergThesis LaTeX package.[1] The structure and chapter titles have been formulated to provide a good starting point for a typical *project report*. This document uses the custom class hgbreport which is based on LaTeX's standard report document class with `chapter` as the top structuring element. If you wish to write this report in German you should substitute the line

```
\documentclass[english]{hgbreport}
```

at the top of this document by

```
\documentclass[german]{hgbreport}.
```

To omit the default **title page** (as in this document) use the `notitlepage` option, e.g.,

```
\documentclass[notitlepage,english]{hgbreport}.
```

Also, you may want to place the text of the individual chapters in separate files and include them using `\include{..}`.

Use the abstract to provide a short summary of the contents in the document.

[1]See https://github.com/Digital-Media/HagenbergThesis for the most current version and additional examples. This repository also provides a good introduction and useful hints for authoring academic texts with LaTeX.

Chapter 1

Algorithm test

Algorithm 1.1: Bikubische Interpolation in 2D. Die Funktion $w_{\text{cub}}()$ steht für die eindimensionale kubische Interpolationsfunktion (Zeile 8).

1: BICUBICINTERPOLATION(I, x, y) ▷ $x, y \in \mathbb{R}$
Input:
Returns the interpolated value of the image I at the continuous position (x, y).
2: $\quad val \leftarrow 0$
3: $\quad$ **for** $j \leftarrow 0, \ldots, 3$ **do** ▷ iterate over 4 lines
4: $\quad\quad v \leftarrow \lfloor y \rfloor - 1 + j$
5: $\quad\quad p \leftarrow 0$
6: $\quad\quad$ **for** $i \leftarrow 0, \ldots, 3$ **do** ▷ iterate over 4 columns
7: $\quad\quad\quad u \leftarrow \lfloor x \rfloor - 1 + i$
8: $\quad\quad\quad p \leftarrow p + I(u, v) \cdot w_{\text{cub}}(x - u)$
9: $\quad\quad$ **end for**
10: $\quad\quad val \leftarrow val + p \cdot w_{\text{cub}}(y - v)$
11: $\quad$ **end for**
12: $\quad$ **return** val
13: **end**

Algorithm 1.2: Finds a minimum makespan role assignment. This function is the MMDR $O(n^5)$ polynomial time implementation, as described by McAlpine et al. It rearranges target positions T so that their index corresponds with the indices of their assigned agents.

RoleAssignment(A, T)
Input: $A = (\mathbf{a}_0, \ldots, \mathbf{a}_{n-1})$, $T = (\mathbf{t}_0, \ldots, \mathbf{t}_{n-1})$, where $\mathbf{a}_i = (x_i, y_i)$, $\mathbf{t}_j = (x_j, y_j)$.
Output: $T' = (\mathbf{t}'_0, \ldots, \mathbf{t}'_{n-1})$, a permutation of T.
StateNN[1]
State
StateNN (no argument)
StateNN[1]
StatexIndent
State
Statex
StateY
item[]
▷ no need to explore more. we just want to stop over here.
$\mathit{edgesSorted} \leftarrow$ SortAscendingDist(Edges)
$\mathit{lastDistance} \leftarrow -1$
$\mathit{rank} \leftarrow 0$
$\mathit{currentIndex} \leftarrow 0$
Process sorted edges (INDENTATION PROBLEM): 2
for $e \in \mathit{edgesSorted}$ **do**
▷ no need to explore more. we just want to stop over here. 2
if $||e|| > \mathit{lastDistance}$ **then** 32.84998pt 3
State 49.27496pt 4
▷ no need to explore more 49.27496pt
StateY
StateNN
StatexIndent
Statex 4
item[] 4
$\mathit{rank} \leftarrow \mathit{currentIndex}$ 4
end if
$\mathit{lastDistance} \leftarrow ||e||$ 3
$||e|| \leftarrow 2^{\mathit{rank}}$
$\mathit{currentIndex} \leftarrow \mathit{currentIndex} + 1$
finally 3
end for
$\mathit{perfectMatching} \leftarrow$ HungarianAlg$(\mathit{edgesSorted})$ ▷ returns a set of edges
end