Consider the following Java-JDT plugin name in German:

*BOSTON UNIVERSITY*

*COLLEGE OF ENGINEERING*

*Dissertation*

# A BU THESIS LATEX TEMPLATE

*by*

## JOE CANDIDATE

*B.S., Some University, 2010*
*M.S., Another University, 2012*

*Submitted in partial fulfillment of the*

*requirements for the degree of*

*Doctor of Philosophy*

*2015*

$$Approved\ by$$

First Reader _____

First M. Last, PhD
Professor of Electrical and Computer Engineering

Second Reader _____

First M. Last
Associate Professor of ...

Third Reader _____

First M. Last
Assistant Professor of ...

*Facilis descensus Averni;*
*Noctes atque dies patet atri janua Ditis;*
*Sed revocare gradum, superasque evadere ad auras,*
*Hoc opus, hic labor est.*                    *Virgil (from Don's thesis!)*

# Acknowledgments

*Here go all your acknowledgments. You know, your advisor, funding agency, lab mates, etc., and of course your family.*

*As for me, I would like to thank Jonathan Polimeni for cleaning up old LaTeX style files and templates so that Engineering students would not have to suffer typesetting dissertations in MS Word. Also, I would like to thank IDS/ISS group (ECE) and CV/CNS lab graduates for their contributions and tweaks to this scheme over the years (after many frustrations when preparing their final document for BU library). In particular, I would like to thank Limor Martin who has helped with the transition to PDF-only dissertation format (no more printing hardcopies – hooray !!!)*

*The stylistic and aesthetic conventions implemented in this LaTeX thesis/dissertation format would not have been possible without the help from Brendan McDermot of Mugar library and Martha Wellman of CAS.*

*Finally, credit is due to Stephen Gildea for the MIT style file off which this current version is based, and Paolo Gaudiano for porting the MIT style to one compatible with BU requirements.*

*Janusz Konrad*

*Professor*

*ECE Department*

# A BU THESIS LATEX TEMPLATE

## JOE CANDIDATE

*Boston University, College of Engineering, 2015*

*Major Professors:*   *First M. Last, PhD*
*Professor of Electrical and Computer Engineering*
*Secondary appointment*

*First M. Last, PhD*
*Professor of Computer Science*

## ABSTRACT

*Have you ever wondered why this is called an* abstract*? Weird thing is that its legal to cite the abstract of a dissertation alone, apart from the rest of the manuscript.*

# Contents

# List of Tables

# List of Figures

# List of Abbreviations

The list below must be in alphabetical order as per BU library instructions or it will be returned to you for re-ordering.

| | | |
|---|---|---|
| *CAD* | . . . . . . . . . . . . . | *Computer-Aided Design* |
| *CO* | . . . . . . . . . . . . . | *Cytochrome Oxidase* |
| *DOG* | . . . . . . . . . . . . . | *Difference Of Gaussian (distributions)* |
| *FWHM* | . . . . . . . . . . . . . | *Full-Width at Half Maximum* |
| *LGN* | . . . . . . . . . . . . . | *Lateral Geniculate Nucleus* |
| *ODC* | . . . . . . . . . . . . . | *Ocular Dominance Column* |
| *PDF* | . . . . . . . . . . . . . | *Probability Distribution Function* |
| $\mathbb{R}^2$ | . . . . . . . . . . . . . | *the Real plane* |

# Chapter 1

# Introduction

## 1.1 Motivation

*Hi*

## 1.2 Problem at hand

*Hello*

## 1.3 Structure of thesis

*Works*

## 1.4 Conclusion

*Nope*

# Chapter 2

# Related Work

## 2.1 Introduction, Query optimization

*A simple SQL query looks like this*

```
1    SELECT *
2    FROM my_table
3    WHERE condition
```

**2.2   Relational algebra,**

**2.3   Converting SQL queries to parse trees**

**2.4   Converting Parse trees into logical expression**

**2.5   Explain difficulties/ Time complexity**

**2.6   Optimzation using relation algebra**

**2.7   Introduction to Data Streams**

**2.8   Data stream windowing**

**2.9   Query Processing of data streams(Combine the DBMS and DSMS)**

**2.10   Challenges of query optimization on data streams**

**2.11   Conclusion and discussion**

**2.12   SQL Query compiler**

*The steps involved are*

**2.12.1   Parsing**

*In a very general sense, given an SQL query, SQL converts it into a parse tree based on SQL grammar.*

**2.12.2   Preprocessing**

*This step has several functions.*

*If a "view" is used in the query as a relation, then each instance has to be replaced by the parse tree.*

*The preprocessor also has to conduct semantic checking, that is, check if relations used exist, check for ambiguity, and type checking. If a parse tree passes the preprocessing then it is said to be **valid***

### 2.12.3 Logical Query Plan

*The first step is to modify the parse tree into using operators and operators of relational algebra.*

*The next step is to convert expression obtained from the above substitution and modify it into an expression which can be converted to most efficient physical query plan.*

*To improve the algebraic expression obtained, few common steps taken are pushing down selections and projections carefully, carefully placing duplicate eliminations, combining selections, showing associativity and commutivity in the expression to help with enumeration.*

*At the end when we have the expression ready, we enumerate the physical plans and calculate their cost of execution and select the method with the lowest cost.*

### 2.12.4 Cost Estimation

*We need to consider what algorithm each operator in the expression is going to use, such as join, sort, scanning and more. Also need to consider the order for the associative and commutative operators, because at the end the operators are binary and how the output of one operator is provided as an input to the next/ outter operator in the expression.*

## 2.13   System R

## 2.14   Deep Reinforcement learning

*Markov decision process(MDP) is used to formalize various types of stochastic processes. In MDPs, the goal of the agent is to make a sequence of actions to optimize/ maximize an objective function.*

*Formally a MDP is a $5-tuple$*

$$\langle S, A, P(s,a), R(s,a), s_0 \rangle$$

$S \to$ *Set of all possible states the agent can be in.*

$A \to$ *Set of all possible actions the agent can take.*

$P(s,a) \to$ *A probability distribution of going to various states given current state and action.* $s^1 \sim P(s,a)$

$R(s,a) \to$ *Reward for taking action a on state s.*

$s_0 \to$ *Describes the initial state of the system/ agent.*

*The performance of the agent is measured using the rewards collected along the way through various states. So the objective of an MDP is to find a policy $\pi : S \to A$, a function that maps states to actions, in order to maximize the expected value:-*

$$\operatorname*{argmax}_{\pi} \mathbb{E}\left[ \sum_{t=0}^{T-1} R(s_t, a_t) \right]$$

*subject to $s_{t+1} = P(s_t, a_t), a_t = \pi s_t$*

*This method does not reduce the search space, and unlikely greedy solution, this will lead to an optimal solution. This method does not reduce the search space, and unlikely greedy solution, this will lead to an optimal solution.*

*Reinforcement learning(RL) is a technique which optimizes MDPs iteratively, by running a simulation in each iteration and changing the policy to find an optimal one based on the cumulative reward.*

## 2.15  Relations

*A common method/ data structure used to formalize joins*

**Query Graph** $\rightarrow$ *A query graph $G$ is an undirected graph, where each relation $R$ is a vertex and each join predicate $\rho$ defines an edge between 2 vertices. Let $\kappa_G$ denote the number of connected components in $G$*

*A join of relation $R_1, R_2$, in the graph corresponds to remove the vertices $v_{R_1}, v_{R_2}$, replacing them with a vertex $v_{R_1+R_2}$, the edges of the form $(v_{R_1}, v)\&(v_{R_2}, v)$ are replaced by $(v_{R_1+R_2}, v)$. Note each reduction reduces number of vertices by one, so this process is repeated until there are $\kappa_G$ number of vertices left.*

**Join Optimization Problem** $\rightarrow$ *Let $G$ be a query graph and $J$ be a join cost model. Find sequence, $c_1 \circ c_2 \circ ... \circ c_n$ resulting in $|V| = \kappa_G$ to minimize*

$$\min_{c_1, c_2, ... c_n} \sum_{i=1}^{n} J(c_i)$$

$$\text{subject to } G_{i+1} = c(G_i)$$

*Using these definitions, we define a MDP.*

$$\langle \{G_0, G_1, ... G_T\}, c, P(G, c), -J, G \rangle$$

*We are still not certain about how the cost function $J$ is structured. We are still not certain about how the cost function $J$ is structured. We are still not certain about how the cost function $J$ is structured.*

*We are still not certain about how the cost function $J$ is structured.*

# Chapter 3

# Stream Optimization

## 3.1 Query Optimization of Data Streams

# Chapter 4

# Implementation

## 4.1   Query Optimization of Data Streams

# Chapter 5

# Stream Optimization

## 5.1   Query Optimization of Data Streams

# Chapter 6

# Stream Optimization

## 6.1 Query Optimization of Data Streams

# Appendix A

# Proof of xyz

*This is the appendix.*

# CURRICULUM VITAE

## Joe Graduate

*Basically, this needs to be worked out by each individual, however the same format, margins, typeface, and type size must be used as in the rest of the dissertation.*