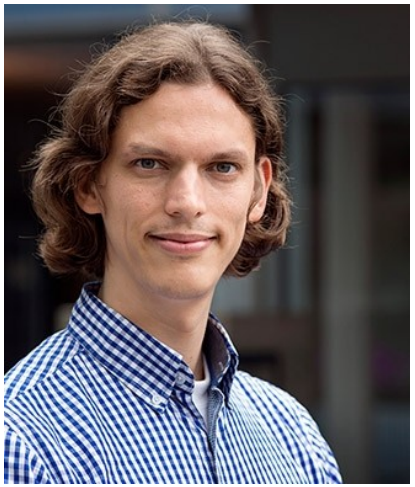
Bachelor Class 2015-2016

Siegfried Nijssen

17 November 2015

Universiteit Leiden

Coordinators



Siegfried Nijssen
s.nijssen@liacs.leidenuniv.nl
http://www.liacs.leidenuniv.nl/~nijssensgr

Amr Ali-Eldin
a.m.t.ali-eldin@liacs.leidenuniv.nl

The Bachelor Class in a Nutshell

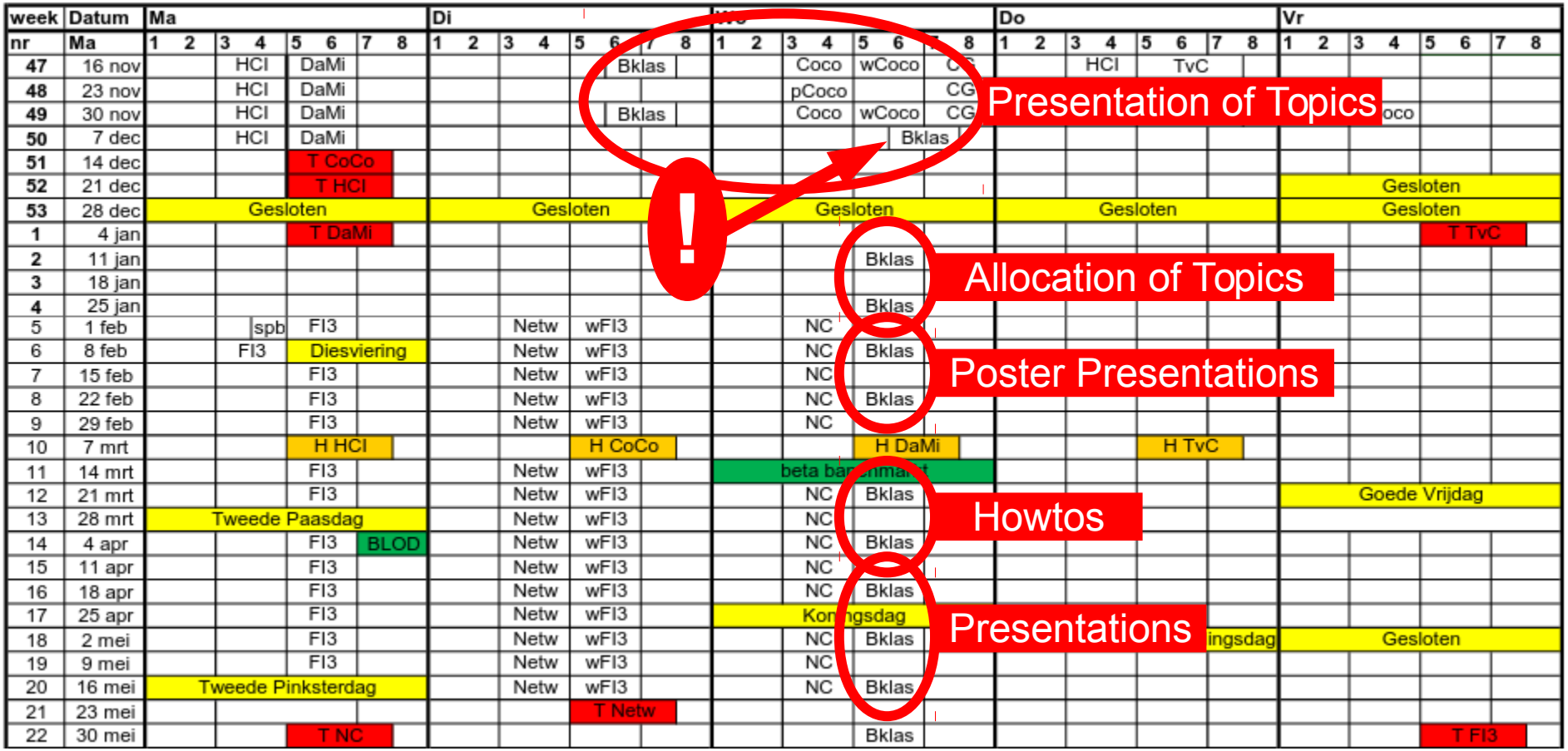
1) Overview of research groups and bachelor project topics

2) Hints & tips: how to write a thesis, how to manage your project, …

3) Progress reports:

- Poster presentation about the problem you are studying
- Intermediate presentation about how you are solving the problem
- Final presentation about your results

Overview of the Bachelor Class



Detailed Information

Rules for Participation

- Everybody is welcome to attend...
- Active participation is only allowed if:
  - You have finished your propedeuse
  - At the start of second semester, at most 2 courses from the second year and the first semester are missing

  Contact Jeannette De Graaf or Ronniy Joseph for exceptions
- You have to participate in order to graduate, including attending all classes and giving all presentations
  - If your 2 missing courses are in the spring semester, you may get permission to finish your project in the autumn and skip the intermediate presentation. Contact Jeannette or Ronniy!

Rules for Participation

- If you cannot be present for a class, you should send a mail **in advance** providing a good motivation

- After 2 missed classes for no good announced reason, you need to make an appointment with the study advisor

- Every class, you need to put your name on an attendance form

Bachelor Dossier

- Includes:
  - Spring seminar
  - Autumn seminar
  - Bachelor class
  - Bachelor thesis, presentation
- 18EC for informatica
  16EC for bachelor thesis and bachelorclass
- 16EC for informatica & economie
  14EC for bachelor thesis and bachelorclass

Choosing a Topic

- All research groups of LIACS will present topics till **December 9**
- The topics will be on the website; by **December 20** you will need to **rank** at least **6 topics**, from at least **3** different supervisors (→ link will be on the website)
- By **January 20** we will finalize the allocation of students to supervisors
- By **February 3** you have to hand in (on paper and online) a **contract** (→ document will be on the website) http://liacs.leidenuniv.nl/~nijssensgr/bachelorklas-2014-2015/contract.html

Research at LIACS

- Algorithms and Software Technology (AST)
  - Games
  - Formal methods
  - Optimization
  - Data science
- Computer systems and Imagery & Media (CSI)
  - Imaging, information retrieval
  - Bioinformatics
  - High performance computing
  - Embedded systems

AST: Games

- *Walter Kosters, Hendrik Jan Hoogeboom, Aske Plaat, Jaap van den Herik*
- Related to artificial intelligence, complexity
- Example projects:
  - Hanabi: A co-operative game of fireworks
  - Compact Decision Trees for Dou Shou Qi Tablebases
  - Predicting the Outcome of the Game Othello
  - Using Outcome Weights in Monte-Carlo Tree Search for Multiplayer 3D Hex
  - A Difficulty Measure for Light Up Puzzles
  - Strategies for Klondike Solitaire
  - Solving Jungle Checkers
  - An Analysis of Dominion
- Popularity warning: typically, many students interested

AST: Formal Methods

- *Marcello Bonsangue, Jetty Kleijn, Farhad Arbab, Frank de Boer, Rudy van Vliet, Luuk Groenenwegen*
- Related to concepts of logic, programming languages, fundamentals of computer science, programming and correctness, software engineering, theory of concurency
- Example projects:
  - Context Free Guarded Languages: A system for determining Guarded Strings
  - The Constraint-Relation Modelling Language and its relation to Petri Nets
  - Equivalence checking of regular expressions using non-deterministic finite automata
  - An On-Line Parsing Algorithm for conjunctive grammars
  - Reducing copying and network traffic in Reo circuits
  - Testing of Channel Based Service Connectors

AST: Optimization

- *Thomas Bäck, Michael Emmerich*
- Related to artificial intelligence, computational intelligence, natural computing
- Example projects:
  - Mining Bitcoins with Natural Computing Algorithms
  - An Evolutionary Algorithm for Finding Diverse Sets of Molecules with User-Defined Properties
  - Multi-objective Generation of Bicycle Routes
  - A Genetic Algorithm for the Travelling Salesman Problem with Area Constraints

CSI: High Performance Computing

- *Harry Wijshoff, Kristian Rietveld*
- Related to operating systems, networks, digital techniques, computer architecture, compiler construction
- Example projects:
  - Deploying Phenotype Analysis On LLSC
  - Deploying Single Particle Analysis on the LLSC
  - A Framework for Cross-Platform Dynamically Loaded Libraries
  - Implementing I/O Infrastructure Improvements for S.M.A.C.K.

CSI: Bioinformatics

- *Fons Verbeek, Kathy Wolstencroft, Sacha Goultiaev*
- Related to human computer interaction, data mining, topics from high computing, software engineering
- Example projects:
  - Deploying Phenotype Analysis On LLSC
  - Deploying Single Particle Analysis on the LLSC
  - Integrating data modeling with data analysis in Taverna workflows
  - Ontology viewer: from proof-of-concept to layered software

CSI: Embedded Systems

- *Todor Stefanov*
- Related to digital techniques, computer architecture, compiler construction
- Example projects:
  - Exploring scheduling alternatives for a Computer Vision application on embedded MPSoCs
  - Auto-vectorization using polyhedral compilation for an embedded ARM platform

CSI: Imaging & Media

- *Michael Lew, Erwin Bakker*
- Related to data mining, computer graphics, artificial intelligence
- Example projects:
  - An algorithm for morphing audio
  - Combined Neural Networks for Movie Recommendation
  - Image Similarity Using Color Histograms
  - Video rating and sorting with a genuine approach
  - A comparison of search engine user interfaces
  - Finding correspondence in stereo image pairs using an adaptive window comparison algorithm
  - Detailed crowd simulation and spatial hashing for large-scale collision detection
  - Video Recommendation, A comparison between collaborative filtering algorithms
- Popularity warning: typically, many students interested

AST: Data Science

- *Joost Kok, Aske Plaat, Jaap van den Herik, Siegfried Nijssen, Peter Lucas, Stefan Manegold, Thomas Bäck, Michael Emmerich, Matthijs van Leeuwen, Frank Takes, Cor Veenman, Wojtek Kowalczyk, Arno Knobbe*
- Related to data mining, databases, statistics, artificial intelligence
- Example projects:
  - Combining graph mining and deep learning in molecular activity prediction
  - Mining a scientific conference
  - Data mining the Peptide Sequenome
  - Analysis and Visualisation of Data of an Outdoor Sports Mobile Application
  - Inference in Markov Networks
  - Data triangulation: combining Ebola datasets for gaining retrospective insights

AST: Data Science Applications

- Industry (steel factories, car manufacturers, aircraft manufacturers)
  Michael Emmerich, Thomas Bäck, Matthijs van Leeuwen
- Banking, insurance
  Wojtek Kowalczyk, Arno Knobbe
- Sports
  Arno Knobbe, Joost Kok
- Biology, chemistry
  Siegfried Nijssen, Michael Emmerich
- Hospital
  Aske Plaat, Peter Lucas, Joost Kok, Siegfried Nijssen
- Law enforcement
  Aske Plaat, Cor Veenman
- Traffic
  Aske Plaat, Arno Knobbe, Siegfried Nijssen
- Social media
  Frank Takes, Aske Plaat, Siegfried Nijssen

AST: Data Science Fundamentals

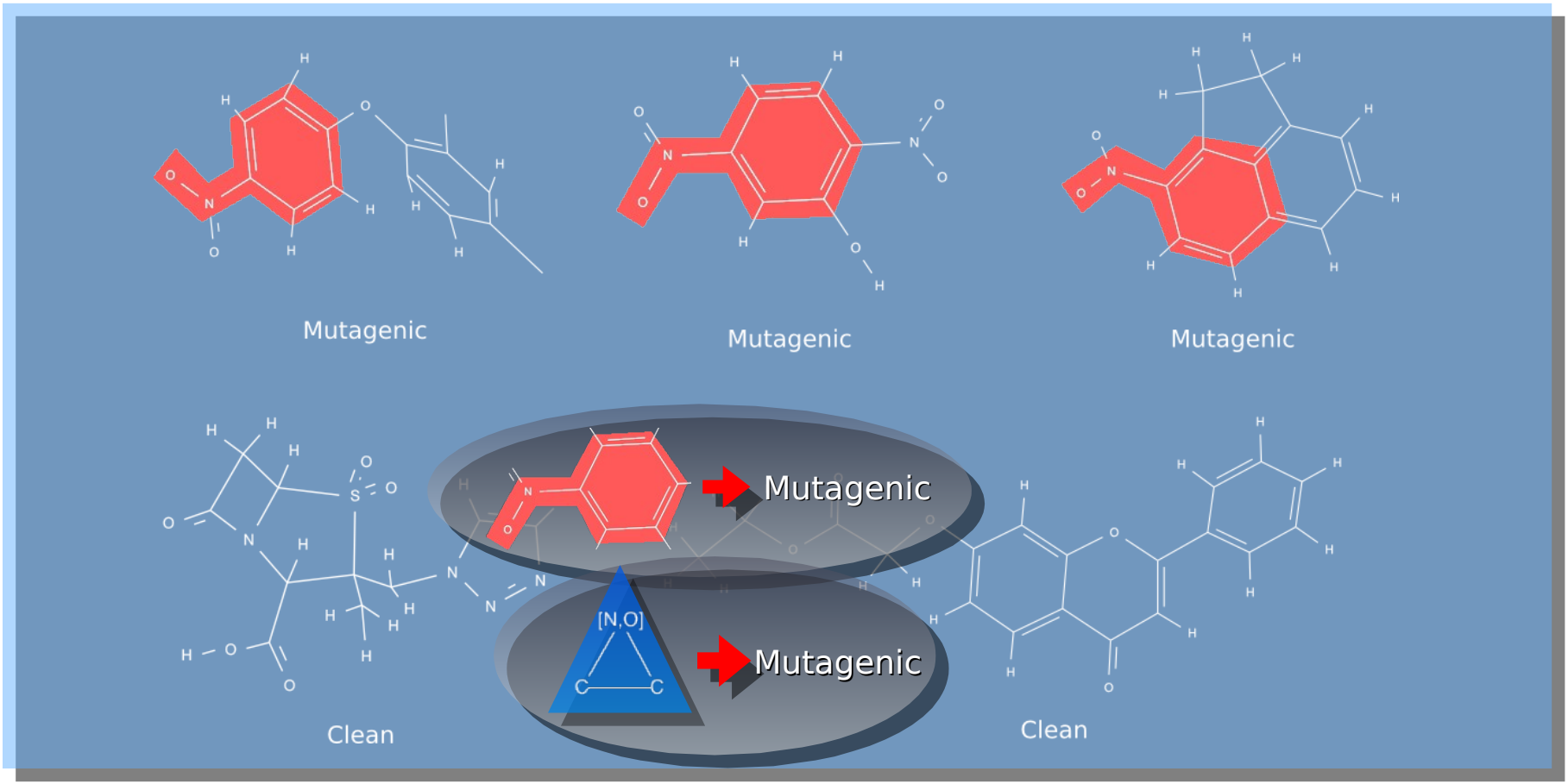
- Database systems
  Stefan Manegold, Michael Emmerich
- Neural networks
  Wojtek Kowalczyk, Siegfried Nijssen, Walter Kosters
- Pattern mining algorithms (itemset mining, subgroup discovery)
  Siegfried Nijssen, Matthijs van Leeuwen, Arno Knobbe
- Supervised machine learning algorithms
  Siegfried Nijssen
- Gaussian processes
  Michael Emmerich
- Bayesian networks
  Peter Lucas
- Graph algorithms
  Frank Takes, Siegfried Nijssen

Siegfried Nijssen

- Master in computer science (Leiden, 2000)
- PhD in computer science (Leiden, 2006)
- Post doc in Leuven (KU Leuven)
- Docent (Leiden)

- Machine learning
- Data mining
- Artificial intelligence

Graph Mining



Graph Mining

- Can we efficiently predict whether a molecule as active?
- Questions:
  - How to represent molecules?
  - How to search for features?
  - Which classifier to use?
- Requirements:
  - An interest in efficient programming in C++
  - An interest in graph theory
  - An interest in data mining

Pattern Mining

Market basket data

support()=3

Pattern Mining

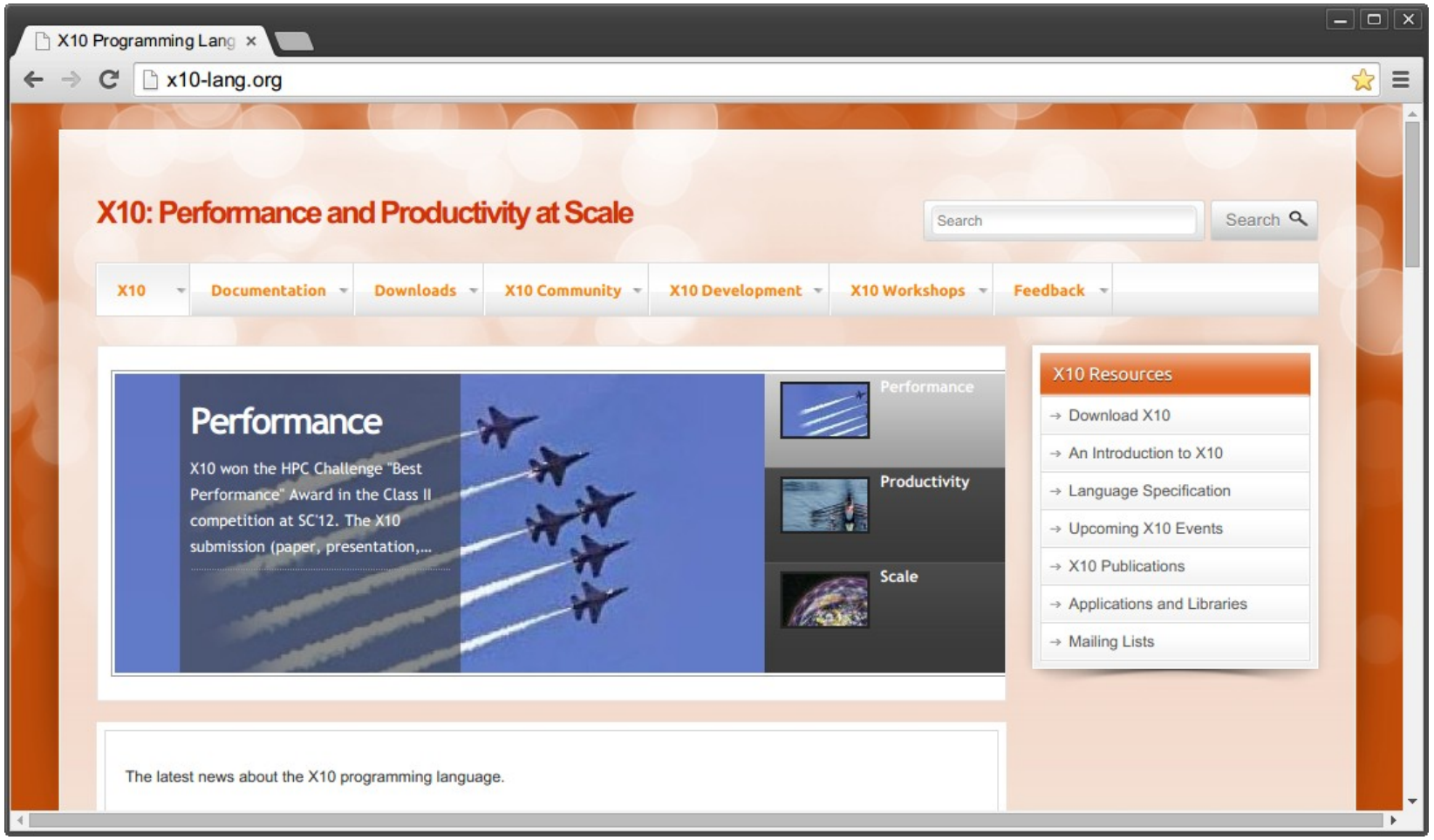
- Situation comparable to having a specialized system for each possible database query

• Apriori • FP-Growth • Eclat • SD-Apriori • DDPMine • Gaston • gSpan • FFSM	• TreeMiner • LCM • MaxMiner • DualMiner • Molfea • CorrMine • EclatV • Mafia	• kDCI • ARMOR • AIM • COFI-tree • DCI closed • WinePI • MinePI • … … …

A 4th Generation Language for Data Mining

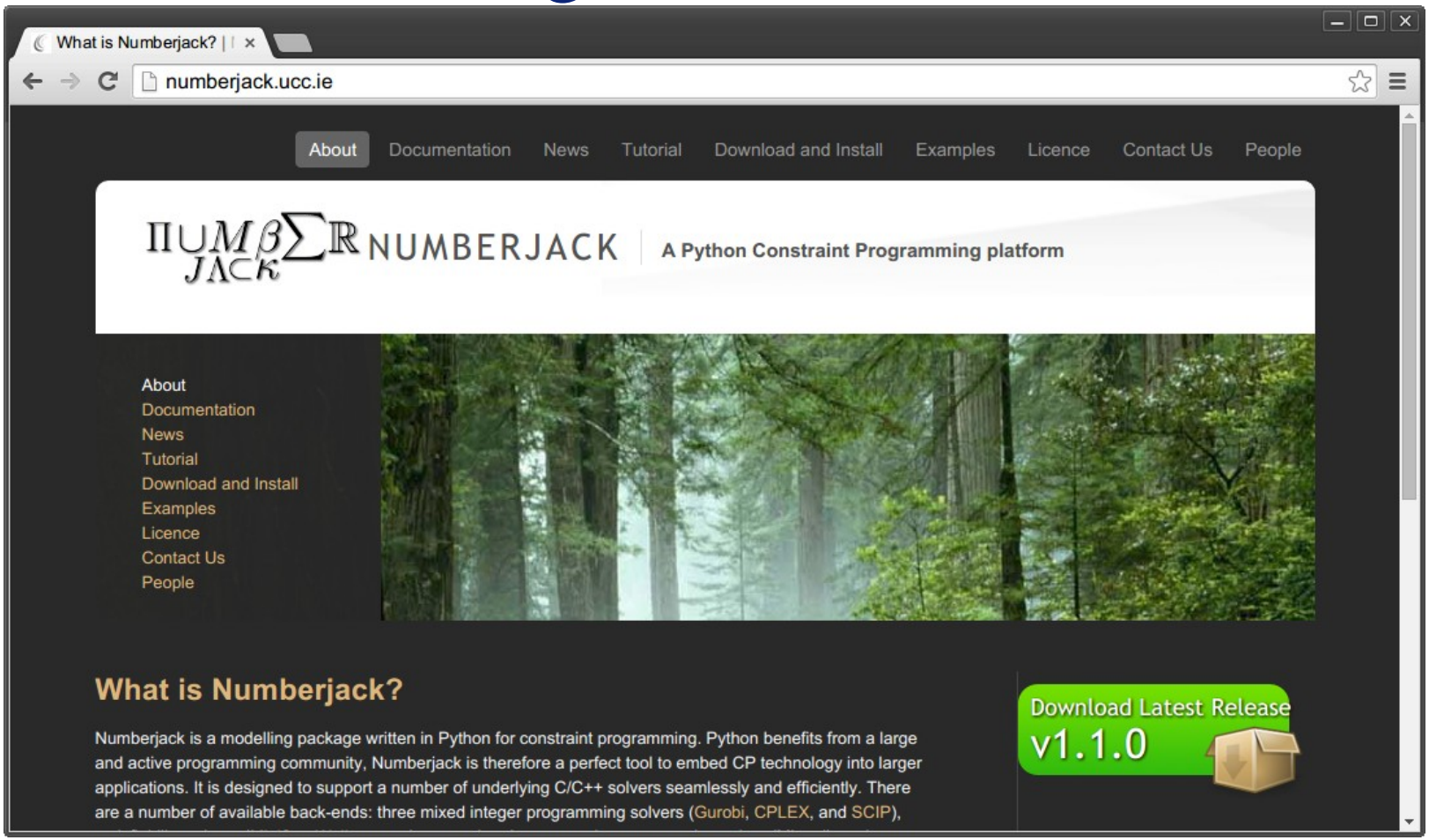
- “An SQL for data mining”

A 4th Generation Language for Data Mining



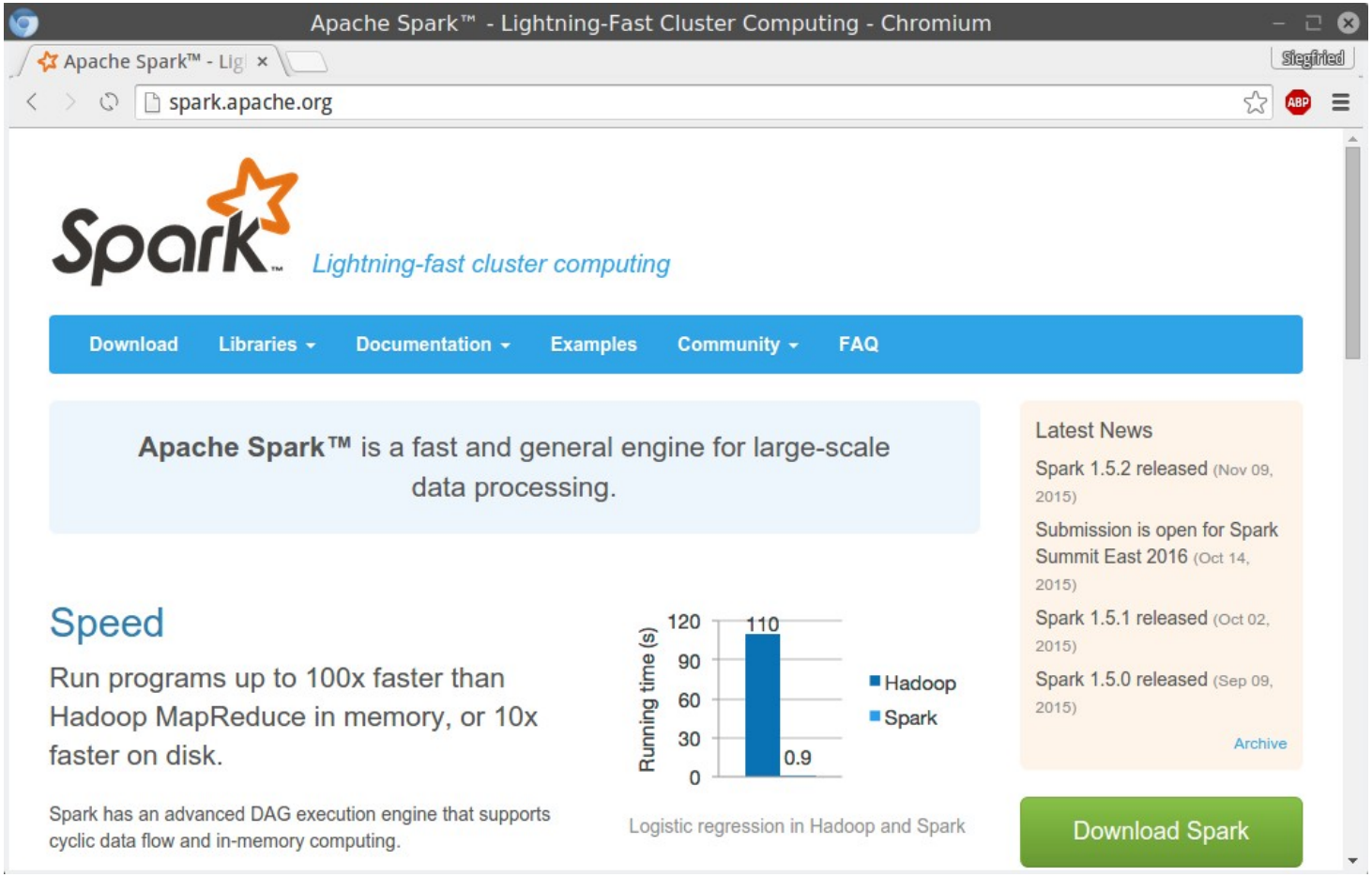
(IBM's X10 language)

A 4th Generation Language for Data Mining



(Numberjack CP system)

A 4th Generation Language for Data Mining

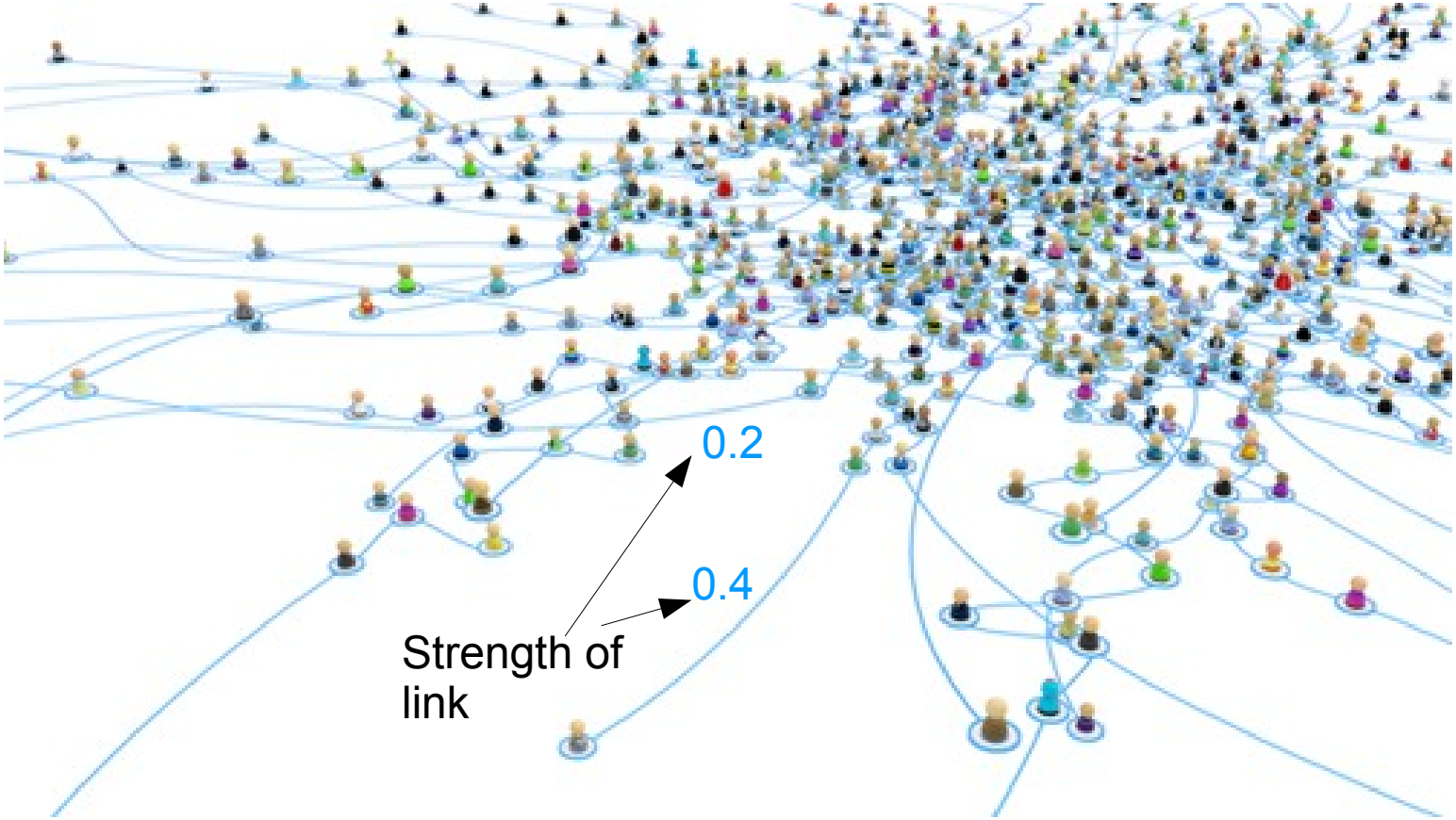


- Apache Spark

A 4th Generation Language for Data Mining

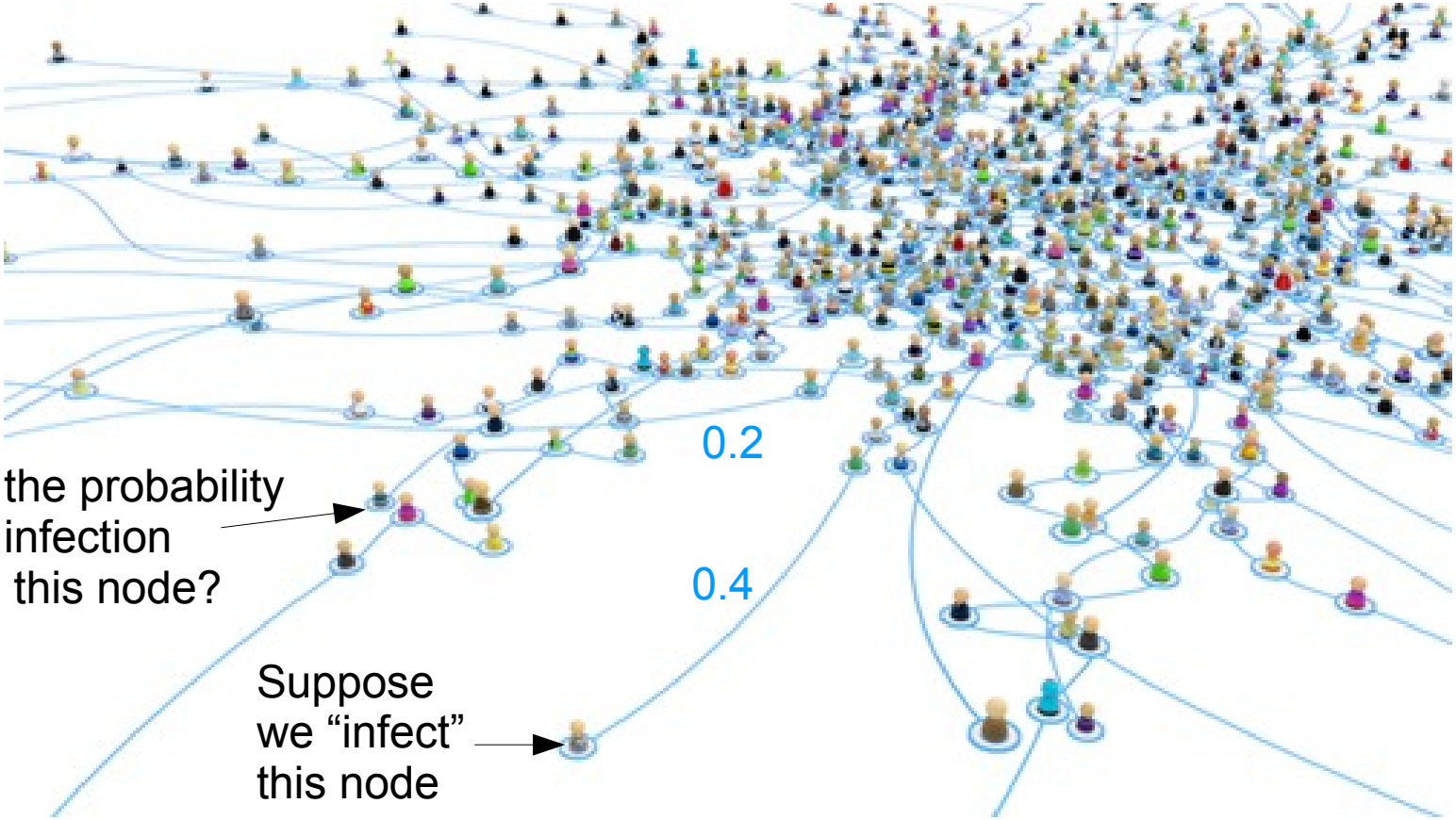
- Can an effective declarative data mining system be built in Python, based on *Numberjack*, *X10*, or *Apache Spark*?
- Requirements:
  - An interest in programming in developing and learning new languages
  - An interest in algorithms
  - An interest in artificial intelligence
  - An interest in data mining

Inference in Probabilistic Networks



Social network, protein interaction network, … with uncertain or unreliable links

Inference in Probabilistic Networks

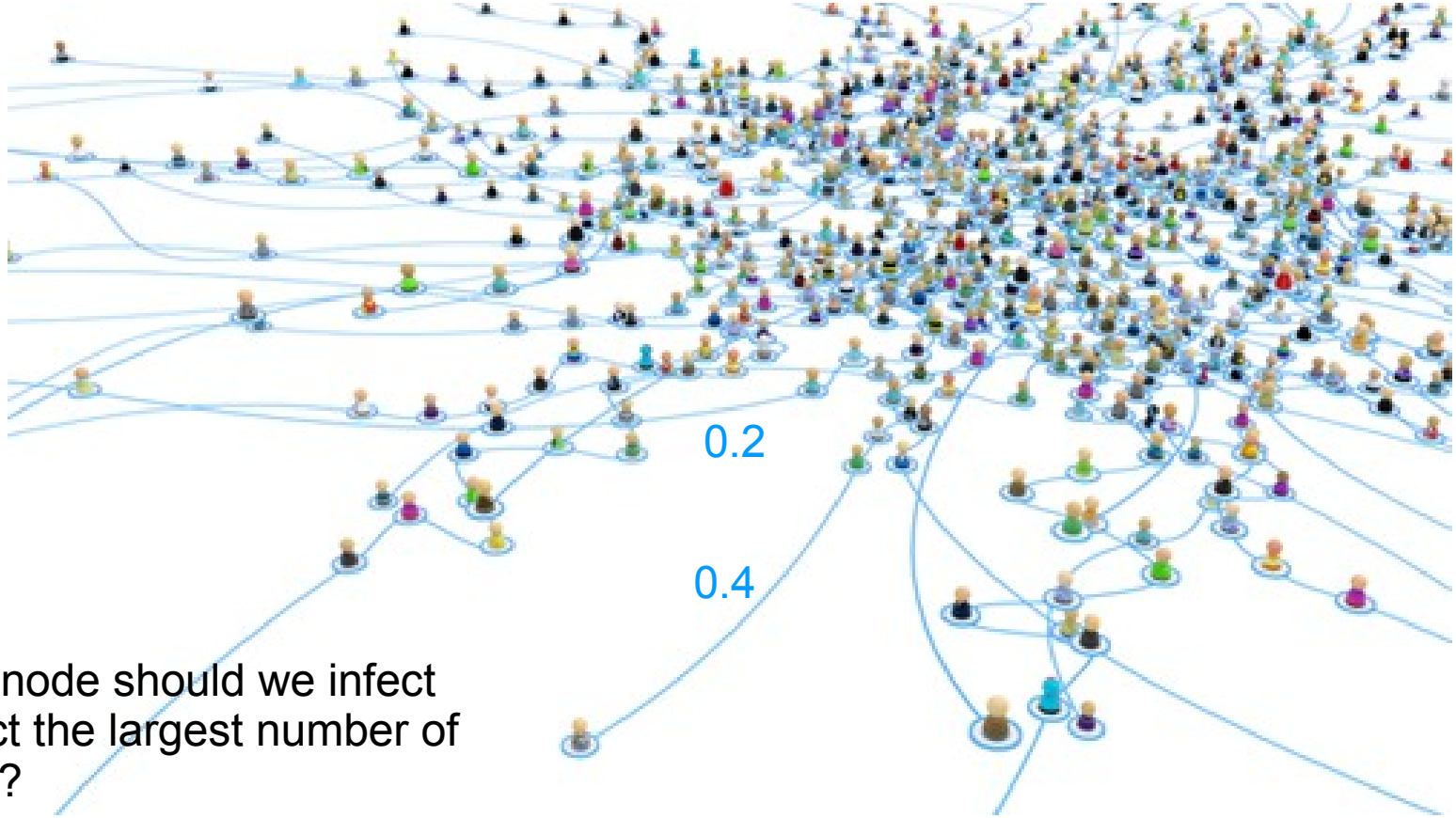


What is the probability that the infection reaches this node?

Suppose we “infect” this node

Social network, protein interaction network, … with uncertain or unreliable links

Inference in Probabilistic Networks



Which node should we infect to infect the largest number of people?

Social network, protein interaction network, … with uncertain or unreliable links

Inference in Probabilistic Networks

- Questions:
  - How to efficiently calculate probabilities?
  - How to efficiently find the state with the largest probability?
  - How to represent problems in networks?
- Requirements:
  - An interest in algorithms
  - An interest in probabilistic reasoning, artificial intelligence