

Bachelorclass 2014-2015

Siegfried Nijssen

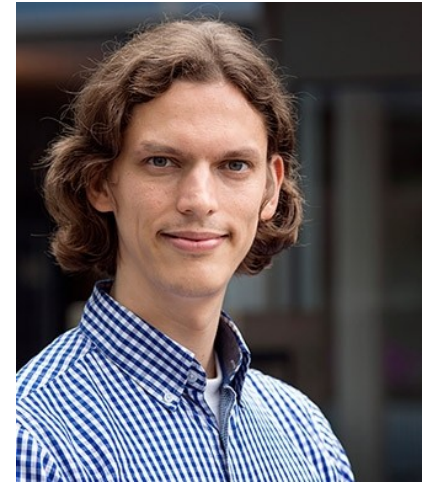
18 November 2014



Universiteit
Leiden

Coordinators

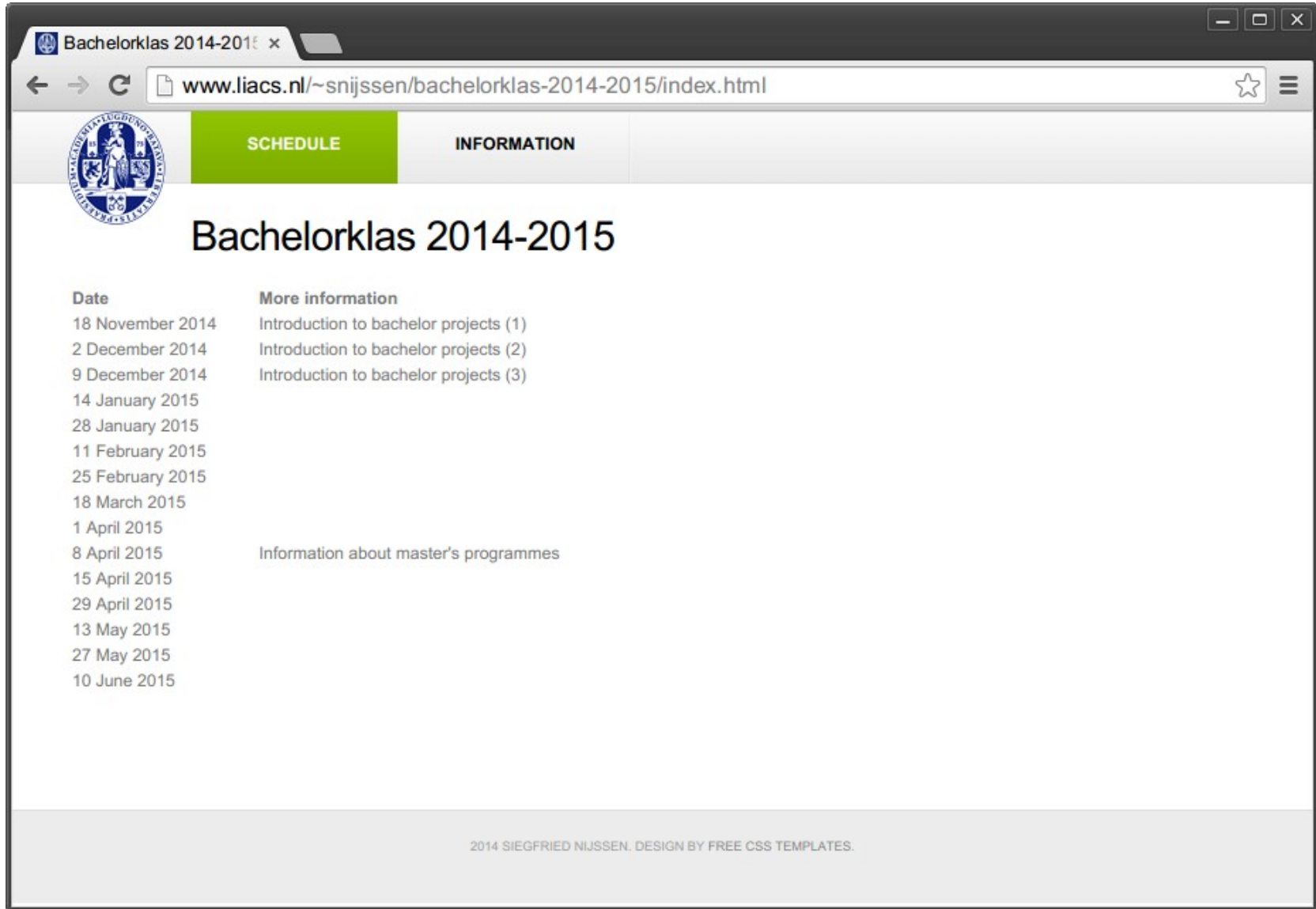
- Siegfried Nijssen
s.nijssen@liacs.leidenuniv.nl
<http://www.liacs.nl/~snijssen>
- Erik Schultes
schultes@hedgehogresearch.info



Overview of the Bachelorclass

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

Detailed Schedule



Bachelorklas 2014-2015

www.liacs.nl/~snijssen/bachelorklas-2014-2015/index.html

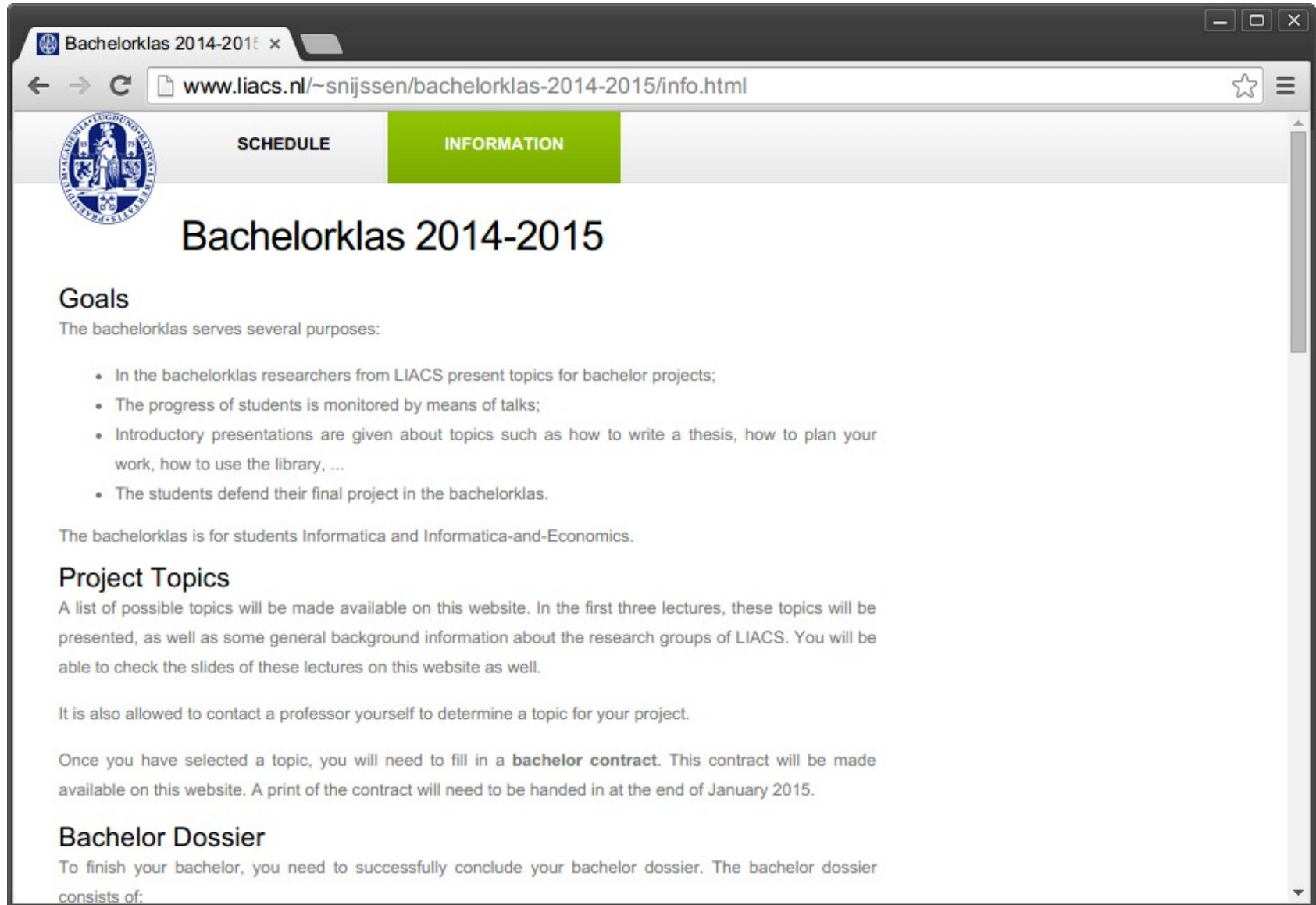
SCHEDULE **INFORMATION**

Bachelorklas 2014-2015

Date	More information
18 November 2014	Introduction to bachelor projects (1)
2 December 2014	Introduction to bachelor projects (2)
9 December 2014	Introduction to bachelor projects (3)
14 January 2015	
28 January 2015	
11 February 2015	
25 February 2015	
18 March 2015	
1 April 2015	
8 April 2015	Information about master's programmes
15 April 2015	
29 April 2015	
13 May 2015	
27 May 2015	
10 June 2015	


2014 SIEGFRIED NIJSSEN. DESIGN BY FREE CSS TEMPLATES.

Detailed Information



Bachelorklas 2014-2015 x

www.liacs.nl/~snijssen/bachelorklas-2014-2015/info.html

 SCHEDULE INFORMATION

Bachelorklas 2014-2015

Goals

The bachelorklas serves several purposes:

- In the bachelorklas researchers from LIACS present topics for bachelor projects;
- The progress of students is monitored by means of talks;
- Introductory presentations are given about topics such as how to write a thesis, how to plan your work, how to use the library, ...
- The students defend their final project in the bachelorklas.

The bachelorklas is for students Informatica and Informatica-and-Economics.

Project Topics

A list of possible topics will be made available on this website. In the first three lectures, these topics will be presented, as well as some general background information about the research groups of LIACS. You will be able to check the slides of these lectures on this website as well.

It is also allowed to contact a professor yourself to determine a topic for your project.

Once you have selected a topic, you will need to fill in a **bachelor contract**. This contract will be made available on this website. A print of the contract will need to be handed in at the end of January 2015.

Bachelor Dossier

To finish your bachelor, you need to successfully conclude your bachelor dossier. The bachelor dossier consists of:

Rules for Participation

- Everybody is welcome to attend...
- Active participation is only allowed if:
 - You have finished your propzedeuse
 - 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 Judith Havelaar 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 Judith!

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 sign an attendance form

Bachelor Dossier

- Includes:
 - Spring seminar
 - Autumn seminar
 - Bachelorclass
 - 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 before Christmas
- These topics will also be available on the website
- You can also approach lecturers if you have an idea yourself
- End of January you have to hand in a **contract** that will be available on the website

Research at LIACS

Two clusters:

- Algorithms and Software Technology (AST)
- Computer systems and Imagery & Media (CSI)

Research at LIACS

Algorithms and Software Technology (AST)

- Natural computing (evolutionary algorithms, genetic algorithms, multi-objective optimization)
Thomas Back
Michael Emmerich
- Data science (data mining, databases)
Joost Kok
Aske Plaat
Siegfried Nijssen
Peter Lucas
Stefan Manegold
Erik Schultes
- Search algorithms (games)
Walter Kusters
Hendrik Jan Hoogeboom
- Formal methods in computer science (automata, petri nets, models for parallelism)
Marcello Bonsangue
Jetty Kleijn
Farhad Arbab
Frank de Boer

Research at LIACS

Computer systems and Imagery & Media (CSI)

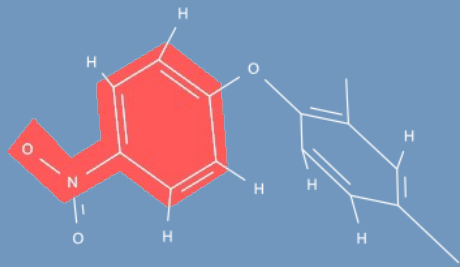
- Imaging, information retrieval
Michael Lew
Erwin Bakker
- Bioinformatics (imaging, pipeline tools, user interfaces)
Fons Verbeek
Kathy Wolstencroft
Alexander Goultiaev
- High performance computing (compilers, parallelism)
Harry Wijshoff
(but also Farhad Arbab)
- Embedded systems
Todor Stefanov

Siegfried Nijssen

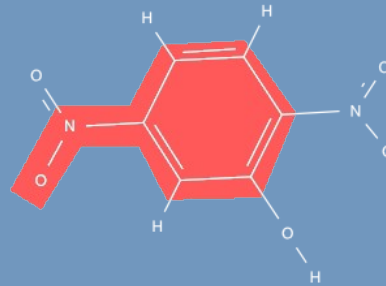
- Master in computer science (Leiden, 2000)
- PhD in computer science (Leiden, 2006)
- Post doc in Leuven (KU Leuven)
- Docent (Leiden)

- Machine learning
- Data mining
- Artificial intelligence

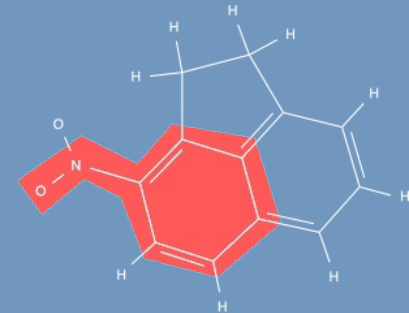
Graph Mining



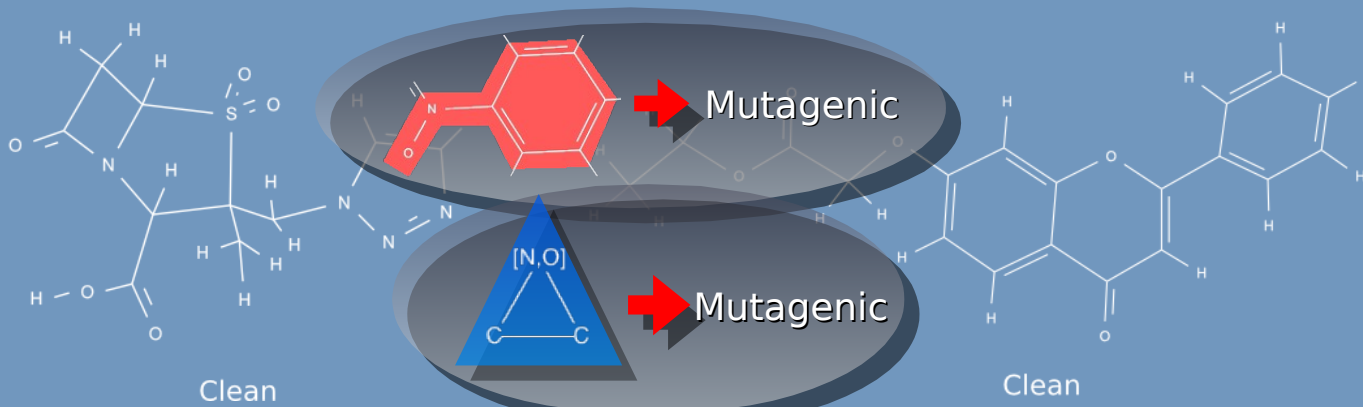
Mutagenic



Mutagenic



Mutagenic



Graph Mining

- Can we efficiently modify a graph mining system such that it supports *multiple* node labels?

O, hydrogen donor

O, hydrogen acceptor

N, hydrogen donor

N, hydrogen acceptor

...

- Requirements:
 - An interest in efficient programming in C++
 - An interest in graph theory

Pattern Mining

Market basket data

$$\text{support}(\text{Pampers, Beer}) = 3$$

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
-

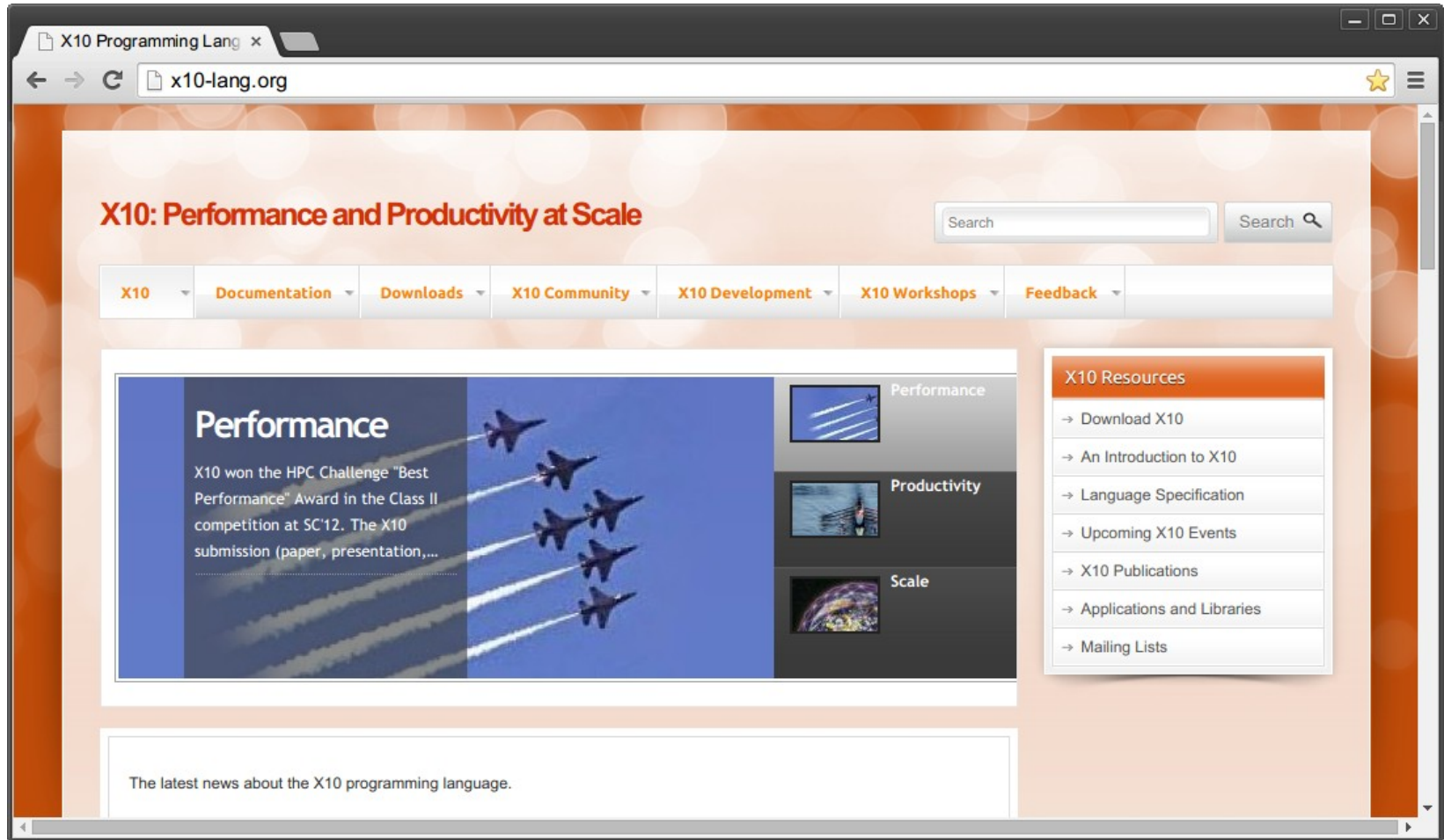
A 4th Generation Language for Pattern Mining

- “An SQL for pattern mining”

```
int: NrI;  
int: NrT;  
int: Freq;  
  
array [1 .. NrT] of set of 1 .. NrI : TDB;  
  
var set of 1..NrI: Items ;  
  
constraint card ( cover ( Items , TDB ) ) >= Freq ;  
  
solve satisfy;
```

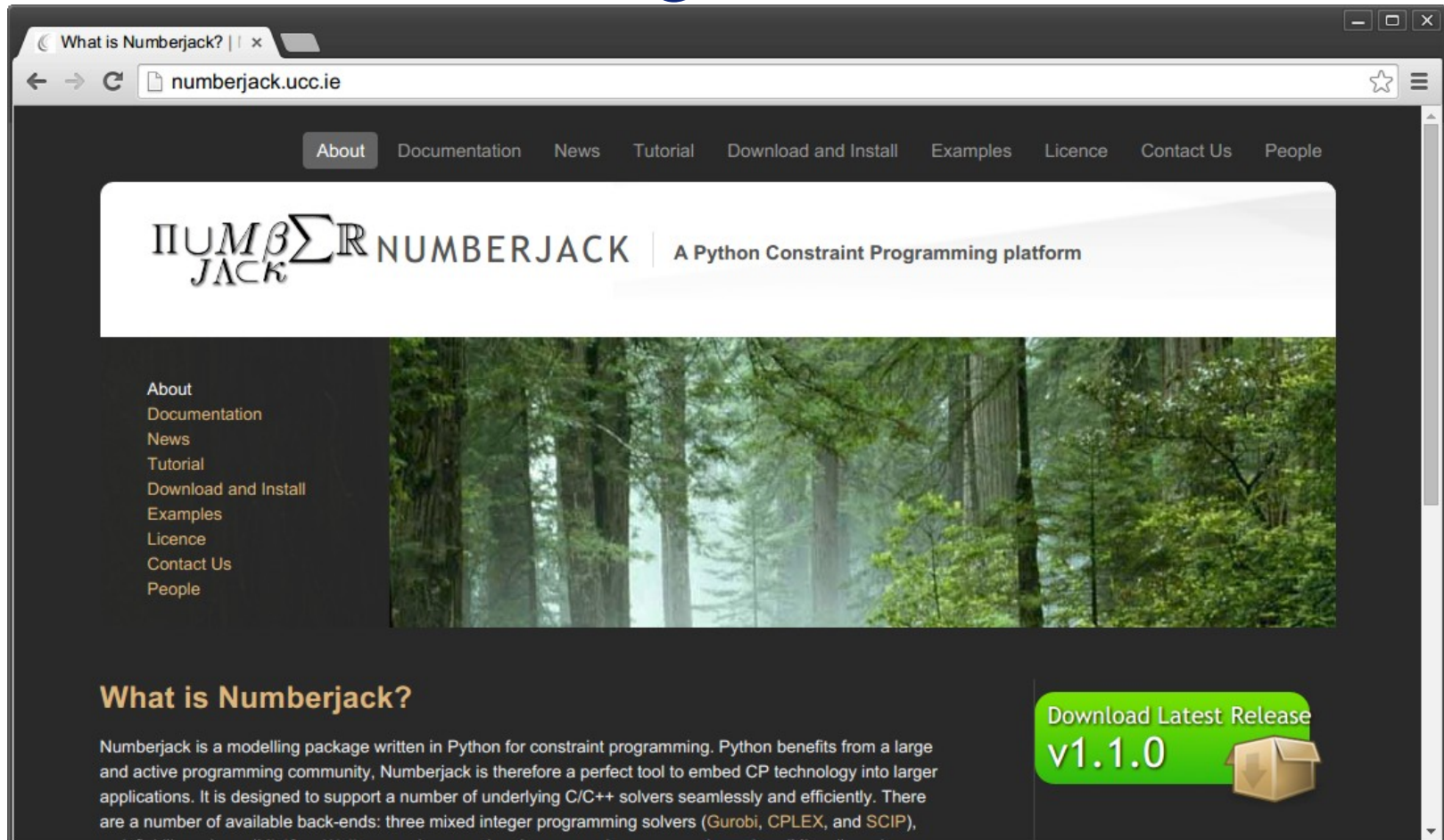
Constraint programming language MiniZinc

A 4th Generation Language for Pattern Mining



(IBM's X10 language)

A 4th Generation Language for Pattern Mining



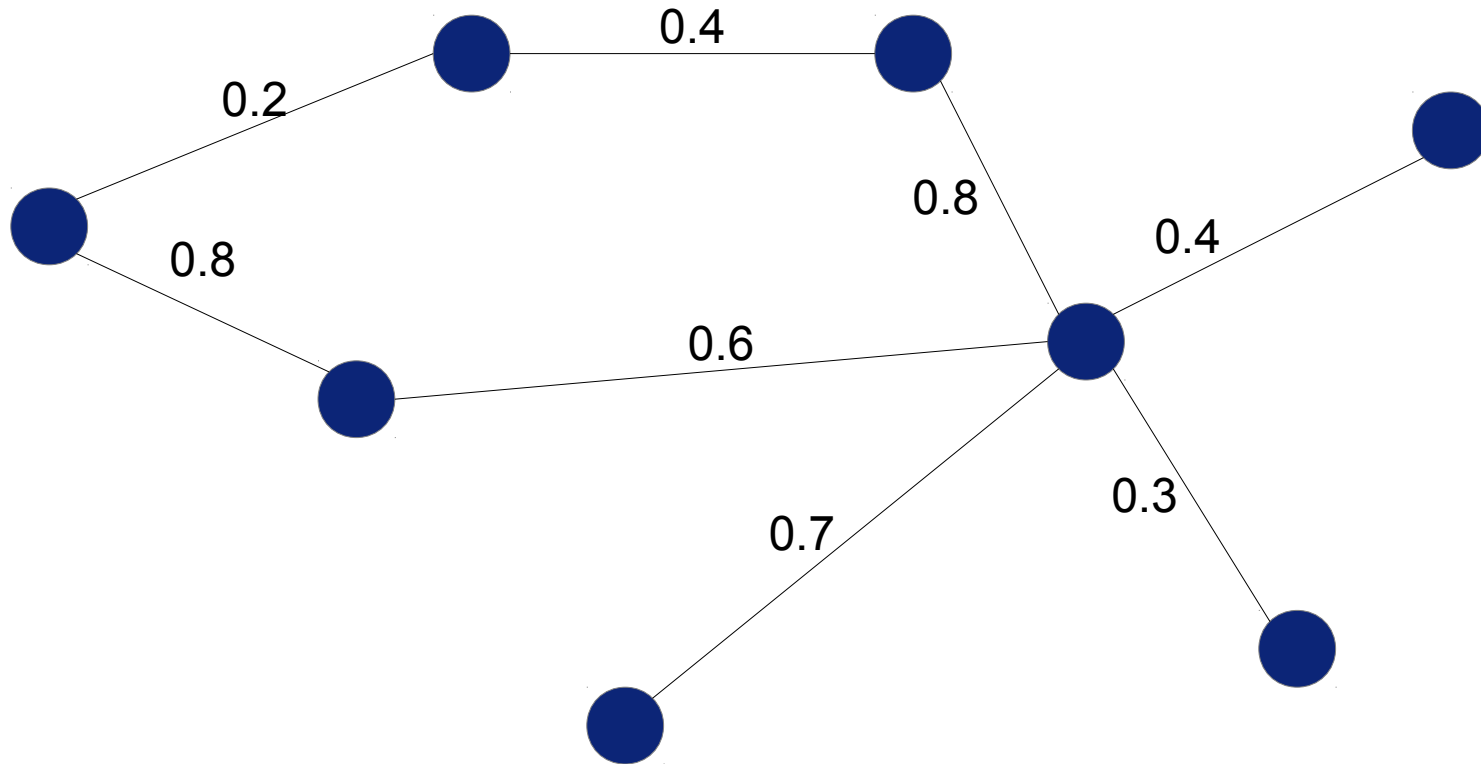
The screenshot shows the homepage of the Numberjack website. The browser address bar displays 'numberjack.ucc.ie'. The navigation menu includes 'About', 'Documentation', 'News', 'Tutorial', 'Download and Install', 'Examples', 'Licence', 'Contact Us', and 'People'. The main header features the logo 'NUMBERJACK' with mathematical symbols Π , β , Σ , and R integrated into the letters 'U', 'M', 'B', and 'S' respectively, and the tagline 'A Python Constraint Programming platform'. A large image of a forest is displayed below the header. On the left side, there is a vertical list of navigation links: 'About', 'Documentation', 'News', 'Tutorial', 'Download and Install', 'Examples', 'Licence', 'Contact Us', and 'People'. The main content area has a heading 'What is Numberjack?' followed by a paragraph: 'Numberjack is a modelling package written in Python for constraint programming. Python benefits from a large and active programming community, Numberjack is therefore a perfect tool to embed CP technology into larger applications. It is designed to support a number of underlying C/C++ solvers seamlessly and efficiently. There are a number of available back-ends: three mixed integer programming solvers (Gurobi, CPLEX, and SCIP),'. To the right of this text is a green button that says 'Download Latest Release v1.1.0' with a cardboard box icon.

(Numberjack CP system)

A 4th Generation Language for Pattern Mining

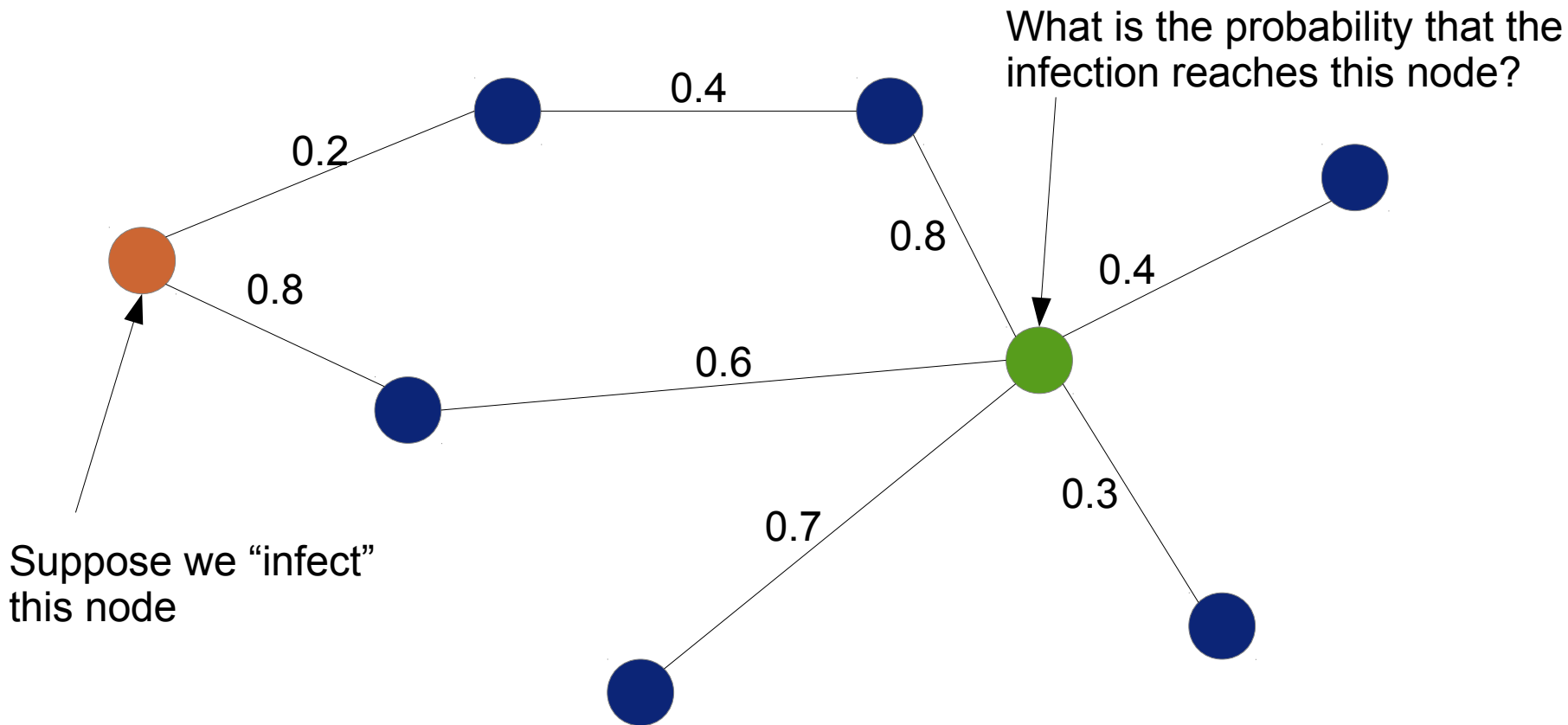
- Can an effective declarative data mining system be built in Python, based on “Numberjack” and “scikit-learn”?
- Can X10 be used to perform declarative data mining?
- Requirements:
 - An interest in programming in developing and learning new languages
 - An interest in algorithms
 - An interest in artificial intelligence

Inference in Graphical Models and Probabilistic Databases



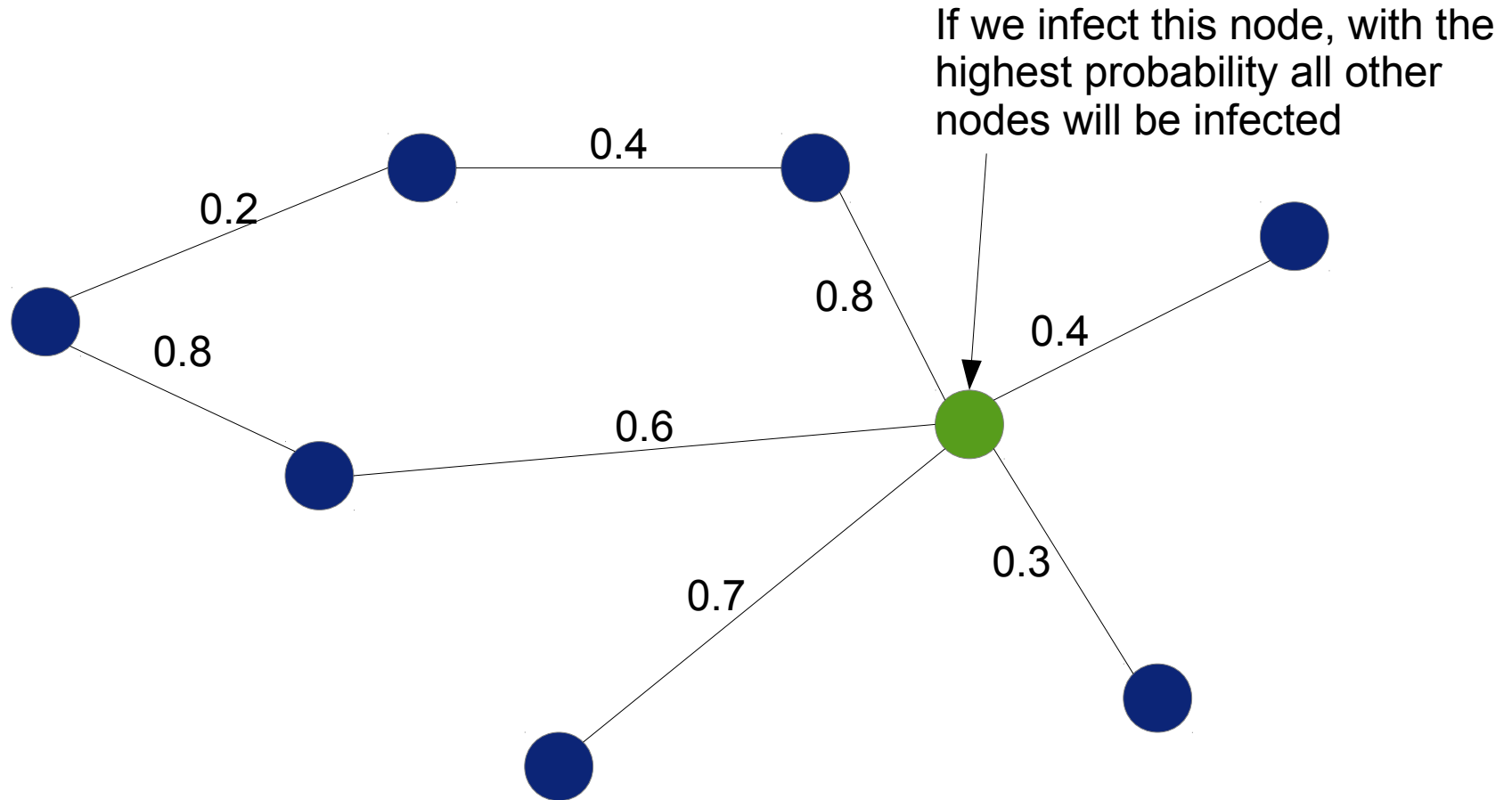
Social network, protein interaction network, ... with uncertain or unreliable links

Inference in Graphical Models and Probabilistic Databases



Inference task: calculating probabilities

Inference in Graphical Models and Probabilistic Databases



Inference task: maximum likelihood

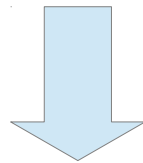
Inference in Graphical Models and Probabilistic Databases

- How to make inference efficient?
- Can we develop or use 4th generation languages for inference?
- Requirements:
 - An interest in algorithms
 - An interest in developing and using new languages
 - An interest in probabilistic reasoning, artificial intelligence

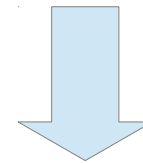
Patterns in Data Visualization

	A	B	C	D
1	Dark	Light	Dark	Light
2	Light	Dark	Light	Dark
3	Dark	Light	Dark	Light
4	Light	Dark	Light	Dark

	A	D	B	C
1	Dark			Light
3	Dark			Light
2	Light			Dark
4	Light			Dark



Reduction to small screen



Patterns in Data Visualization

- What do the data visualizations look like for different types of patterns?
- Can we make a system that “tells the story of the data” in a video?
- Requirements:
 - An interest in making visualizations in Python, C++, ...
 - An interest in running existing data mining programs in C++
 - An interest in algorithms