Knowledge-Based Systems
and Deductive Databases

Christoph Lofi
Institut für Informationssysteme
Technische Universität Braunschweig
http://www.ifis.cs.tu-bs.de
1.0 Organizational Issues

• Lecture
    • NOT on 18.04.12, but instead during Excursion Week
  – 09:45-12:00h (3 lecture hours with a short intermediate break)

• 5 Credits

• Exams
  – Oral Exams
    • Individual appointments with our secretary
  – Most probably:
    • 1 week in early August
    • 1 week in early September
1.0 Organizational Issues

– Weekly exercises
  • You don’t really need to do the exercises, but it will surely be helpful
  • Exercises will be neither collected, nor corrected, nor graded
1.0 Why should you be here?

• The increase of knowledge grows \textbf{exponentially}
  – e.g. in terms of new publications

• Huge amounts of data have to be sifted and analyzed to gain intelligence from data
  – Need \textbf{knowledge-based technology} for this task!
  – Analysts have interesting perspectives, i.e. \textbf{exciting and well-paid work}
  – Find out about the things you always wanted to know…

\textit{Knowledge-Based Systems and Deductive Databases - Christoph Lofi – IfIS – TU Braunschweig}
1.0 Why should you be here?

• But you can also go into the direction of **big bucks**…
  – Write an **automated reasoning system** and license call-center technology
  – Ronald A. Katz, founder of **Ronald A. Katz Technology Licensing, LP**)
    • Licensing the technology earned more than **one billion USD**
    • Customers include AT&T, Bank of America, Citibank, Delta Air Lines, Hewlett Packard, IBM, Microsoft…
1.0 What will you learn?

• What are knowledge-based systems? What can you do with them?

• Many KBS are based on formal logics. You will:
  – …learn about different kinds of formal logic
  – …learn syntactic basics of predicate logic
  – …learn of how to interpret logical expressions
  – …learn how to efficiently evaluate logical expressions in a database setting
  – …how to design a KBS using different flavors of formal logic

\[ \forall x \, P(x) \land \forall x \, Q(x) \iff \forall x \, (P(x) \land Q(x)) \]
1.0 What will you learn?

• We will show you how the vision of knowledge-based systems was born and what became of it
• We will show you how all KBS ideas have been reborn within the semantic web
• We will show you how the semantic web works and what it tries to achieve
1.0 What will you learn?

Introduction

Predicate Logics
  Syntax and Semantics
  Model Theory

Datalog
  Syntax
  Semantics
  Fixpoint Iteration
  Optimization

Uncertain Reasoning

Expert Systems

Semantic Web
  Basics
  RDF & Co

Description Logics

Question Answering
1.0 Recommended Literature

• There is no “standard literature”

• General literature
1.0 Recommended Literature

• German titles
• … and many academic papers…
1.1 Dreams of Artificial Intelligence
1.2 Applications of Knowledge-Based Systems
1.2 Deduction in Databases
1.4 The Semantic Web
• Since ancient times, people dream of intelligent machines
  – Golden robots of Hephaestus
  – Archytas’ wooden pigeon (400 BC)
  – Leonardo da Vinci’s mechanical knight (1495)
  – The Turk of Wolfgang von Kempelen (1770)
  – …
• In computer science, this gave birth to the field of Artificial Intelligence
1.1 AI Dreams

- In the 20th century, the field of **A.I. (Artificial Intelligence)** became popular
  - 1950: Alan Turing
    - “The brain is just like a complex machine.”
    - Turing test
  - 1956: Dartmouth Conference
    - Founding of the A.I. laboratories
  - 1965: Herbert A. Simon
    - "Machines will be capable, within twenty years, of doing any work a man can do"
  - 1967: Marvin Minsky
    - "Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved."
1.1 AI Dreams

• In the initial phase of A.I. research, people were highly motivated and full of visions
  – High amount of research money available, mainly from the military (DARPA)
• In the mid seventies, the great visions died…
  – A long series of failures took its toll
  – The A.I. winter – funding stopped
• Change of research direction
  – Do not imitate the full human brain, but find intelligent algorithms for solving some particular, difficult problems
  – Today the basic ideas are part of the Semantic Web efforts
• Main critique – Hubert Dreyfus (UC Berkeley, USA)
  – Expertise cannot readily be extracted from human experts
  – Much knowledge is not explicit, but somehow embodied
    • The brain is not simply hardware running a program based on discrete symbolic calculations
In the 1980ies, A.I. focused on well-defined problem domains building first commercially successful systems

- **Knowledge-based systems** or ‘expert systems’

**Idea:** Create a system which can draw **conclusions** and thus support people in difficult decisions

- Simulate a **human expert**
- **Main idea:** extract knowledge of experts and just cheaply copy it to all places you might need it
Expert Systems were supposed to be especially useful in

- Medical diagnosis
  - “Great failure up to now”?
- Production and machine failure diagnosis
  - Works quite well
- Financial services
  - Widely used
1.2 Notable Expert Systems

- **MYCIN**
  - Developed 1970 at Stanford University, USA
  - Medical expert system for treating *infections*
    - Diagnosis of infection types and recommended antibiotics (antibiotics names usually end with \(\sim\)mycin)
  - Around 600 rules (also supporting uncertainty)
  - MYCIN was treated as a success by the project team…
    - Experiments showed good results, especially with rare infections
  - … but was never used in practice
    - Too clumsy
    - Technological constraints
1.2 Notable Expert Systems

• **MYCIN** example rule

  If the organism 1) stains grampos 2) has coccus shape 3) grows in chains
  then there is a suggestive evidence of 0.7 that it is *streptococcus*

• MYCIN used backward chaining from goal to facts for reasoning

  – Try to apply rules concluding the reasoning process by asking the user about the rules premises
1.2 Notable Expert Systems

• **MYCIN** example

-----PATIENT-1-----
1) Patient's name: **FRED SMITH**
2) Sex: **MALE**
3) Age: **55**
4) Have you been able to obtain positive cultures from a site at which Fred Smith has an infection? **YES**

-----INFECTION-1-----
5) What is the infection? **PRIMARY-BACTEREMIA**
6) Please give the date when signs of INFECTION-1 appeared. **5/5/75**
The most recent positive culture associated with the primary bacteremia will be referred to as:

-----CULTURE-1-----
7) From what site was the specimen for CULTURE-1 taken? **BLOOD**
8) Please give the date when this culture was obtained. **5/9/75**
The first significant organism from this blood culture will be called:

-----ORGANISM-1-----
9) Enter the identity of ORGANISM-1. **UNKNOWN**
10) Is ORGANISM-1 a rod or coccus (etc.)? **ROD**
11) The gram stain of ORGANISM-1: **GRAMNEG**
1.2 Notable Expert Systems

- **MYCIN** explanations example

** Did you use RULE 163 to find out anything about ORGANISM-1?**
RULE163 was tried in the context of ORGANISM-1, but it failed because it is not true that the patient has had a genito-urinary tract manipulative procedure (clause 3).

** Why didn't you consider streptococcus as a possibility?**
The following rule could have been used to determine that the identity of ORGANISM-1 was streptococcus: RULE033
But clause 2 (“the morphology of the organism is coccus”) was already known to be false for ORGANISM-1, so the rule was never tried.
1.2 Notable Expert Systems

• **Dipmeter Advisor**
  – Developed 1980 by Schlumberger Doll Research
  – Support for oil drilling operations
  – Very simple system with ~90 rules
  – However, system was a huge success
    • Heavily used in field and appreciated by its users
    • Sold as fully functional unit including hardware
    • First great success after A.I. winter!
1.2 Notable Expert Systems

- **Dipmeter Advisor**: functionality
  - Dipmeters analyze ground properties by conductivity measurements
  - Usually end up with a ‘dipmeter log’
    - Hard to interpret, even by experts
1.2 Notable Expert Systems

- **Dipmeter Advisor:** functionality
  - Dipmeter advisor creates log analysis and provides a summary using rules

**Example rule**

**IF**
- there exists a delta-dominated, continental-shelf marine zone, and
- there exists a sand zone intersecting the marine zone, and
- there exists a blue pattern within the intersection

**THEN**
- assert a distributary fan zone
- top ← top of blue pattern
- bottom ← bottom of blue pattern
- flow ← azimuth of blue pattern

**Example summary**

<table>
<thead>
<tr>
<th>Interpretation Summary for Log 15702</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STRUCTURAL DIP</strong></td>
</tr>
<tr>
<td>From 13140 to 13770: 3.9 degrees at azimuth of 327.</td>
</tr>
<tr>
<td>From 13780 to 14444: 5.7 degrees at azimuth of 213.</td>
</tr>
<tr>
<td>From 14444 to 15500: 25.9 degrees at azimuth of 243.</td>
</tr>
<tr>
<td><strong>FAULTS AND MISSING SECTIONS</strong></td>
</tr>
<tr>
<td>From 13762 to 13790: there is a growth fault oriented along the line from 83 to 243 degrees, with the downthrown block at 163 degrees.</td>
</tr>
<tr>
<td>From 14352 to 14444 there is an unconformity or middle age fault.</td>
</tr>
<tr>
<td><strong>STRATIGRAPHY</strong></td>
</tr>
<tr>
<td>From 15114 to 15168 there is a distributary-front with an associated channel. The channel axis is at 163 degrees, flow was at 75 degrees.</td>
</tr>
</tbody>
</table>
1.2 Notable Expert Systems

• NASA Shine
  – **Spacecraft Health Inference Engine**
  – Development started in mid 70s by NASA and JPL (Jet Propulsion Lab) for the Deep Space Network
    • Commercially used by ViaSpace
  – Multi-purpose inference system
  – Detects system failures within complex mission critical machineries
  – Designed to run in real-time in embedded and distributed systems
1.2 Notable Expert Systems

• NASA Shine: currently used by
  – Deep Space Network
  – Lockheed Martin F-35 Lightning 35
  – McDonnel Douglas F/A-18 Hornet
  – NASA CEV (Crew Exploration Vehicle)
  – NASA Ares Rocket Program
  – NASA Voyager spacecrafts
  – Lockheed Martin X-33
  – Galileo Space Probe
  – Extreme Ultraviolet Explorer
  – …
1.3 Deduction in Databases

• System may deduce new **facts** using **rules**
  – Leads to **inference chains**

• Most systems heavily rely on mathematical logics
  – **First-Order Predicate Logics**
1.3 Deduction in Databases

• Usually deriving new knowledge is based on interference rules and specific problem data
  – **Fact:** Hektor is a frog
  – **Rule:** All frogs are green
  – Implies new **fact:** Hektor is green

• Also, **uncertainty** can be supported
  – **Fact:** Tweety is a bird
  – **Rule:** Almost all birds can fly except ostriches, chicken and some others
  – **Query:** Can Tweety fly?
    • Only few species are ostrichs, chicken or penguins
    • Tweety can fly with high probability
1.3 Deduction in Databases

• But sometimes several steps are needed
  – **Fact:** *Hektor* is a frog
  – **Rules:** All *frogs* are *green*
    green things can hide in *undergrowth*
  – **Implies:** *Hektor* can hide in *undergrowth*

• Needs to apply the second rule on the result of the first rule
  – **Recursion!**
• Rules can be used to **derive new knowledge** over data collections (called **facts**)
  – Based on **symbolic calculation** the 70ies saw the development of logic programming languages like LISP or Prolog

• But how do you deal with **large fact collections**?
  – It’s easy… use a database for storage and then reason over the database instance
1.3 Deduction in Databases

- **Common architecture** of an expert system
  - **User Interface**: Usually based on a question-response dialog
  - **Inference Engine**: Tries to deduce an answer based on the knowledge base and the problem data
  - **Explanation System**: Explains to the user why a certain answer was given or question asked
  - **Knowledge Base**: Set of rules and base facts
  - **Problem Data**: Facts provided for a specific problem via user interface
1.3 Deduction in Databases

• **Expert systems** have to keep and manage **valuable data** in their knowledge base
  – Basically expert systems just support another query type, but have the same requirements like a normal database system

• Can we simply build deductive databases on top of a **relational database engine**?
1.3 Recursion in DB-Queries

• Why don’t normal databases do the trick?
  – **SQL queries** can be read as follows
    • “If some tuples exist in the **FROM**-tables satisfying the **WHERE**-conditions, then the **SELECT**-tuples are the answer”
  – **Datalog** is a query language that has the same if-then flavor, but…
    • An intermediate answer table can appear in the **FROM** clause to facilitate **recursion**
1.3 Recursion in DB-Queries

- Example: a public transport information system
- Database stores **connected** stops as **facts**, e.g.,
  - `connection(Maschplatz, Hamburgerstr, 2 minutes)`.
  - **Transitive closure** contains all connections
- Additional **rule**
  - Connections are transitive. If you can go from A to B and from B to C, you can also go from A to C
## 1.3 Recursion in DB-Queries

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Line</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maienstr</td>
<td>Kälberwiese</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>Kälberwiese</td>
<td>Rudolfplatz</td>
<td>19</td>
<td>1.5</td>
</tr>
<tr>
<td>Rudolfplatz</td>
<td>Amalienplatz</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>Rudolfplatz</td>
<td>Petristr.</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>Petristr.</td>
<td>Maschstr.</td>
<td>11</td>
<td>1.5</td>
</tr>
<tr>
<td>Amalienplatz</td>
<td>Maschstr.</td>
<td>16</td>
<td>2</td>
</tr>
</tbody>
</table>
1.3 Recursion in DB-Queries

• How long does it take to go from ‘Maienstr.’ to ‘Maschstr.’ in SQL-92?
  – SQL-92 does not support transitive closures
  – Only solution: Create a view materializing all connections

• Big challenge on storage space and data consistency

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Line</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maienstr.</td>
<td>Kälberwiese</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>Maienstr.</td>
<td>Rudolfplatz</td>
<td>19, 19</td>
<td>2.5</td>
</tr>
<tr>
<td>Maienstr.</td>
<td>Petristr.</td>
<td>19, 19, 11</td>
<td>3.5</td>
</tr>
<tr>
<td>Maienstr.</td>
<td>Maschstr.</td>
<td>19, 19, 11, 11</td>
<td>4.5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
• Deductive queries/programs are often stated in **Prolog or Datalog**
  
  – **Prolog** is a logical programming language created in 1972
  
  – **Datalog** is a subset of Prolog especially designed for deductive databases
    
    • No predicates are allowed as arguments
    • Only fix-point iteration
    • Efficient bottom-up evaluation
1.3 Deduction in Databases

- **In Datalog**: Queries use **recursive** rules
  - Facts given by the single connections
    - ?connection(Maienstr., Maschstr., L, X)
    - Can be answered efficiently by binding the start and goal stop and deducing everything in between
  - connection(X,Y, L, T) :- connection (X, Z, L₁, T₁),
    connection (Z, Y, L₂, T₂),
    T = T₁ + T₂, L = L₁∥∥L₂

- **Big challenge**: How can this query be evaluated efficiently?
1.3 Deduction in Databases

• A **deductive DBS** is a database system with limited support for reasoning
  
  • All the goodies of databases (**transactions, recovery, etc.**)
  • Queries based on (**recursive**) views are possible
  • Efficient (bottom up) **query optimization**
1.3 Deduction in Databases

- **Typical example for the application of deductive databases:**
  - **Facts**
    - parent(bill, mary).
    - parent(mary, john).
  - **Rules**
    - ancestor(X, Y) :- parent(X, Y)
    - ancestor(X, Y) :- ancestor(X, Z), ancestor(Z, Y)
  - **Query**
    - :- ancestor(bill, X)
  - **Answer**
    - ancestor(bill, mary)
    - ancestor(bill, john)
Since becoming popular, numerous deductive database systems have been developed.

- All of them are system-centered with proprietary storage engines.
- No usage of RDBMS or OODBMS.

**Coral**

- Developed since 1988 by University of Wisconsin.
- Provides native interfaces directly into C++.
- Full transactional ACID support due to Exodus storage manager.
1.3 History of Deductive DBs

- **LDL**
  - Developed since 1984 by MCC Research in Austin
  - Proprietary Query language based on Horn clauses
  - Initially developed for a special parallel logic computer hardware developed by MCC Research in Austin (which never went into production)
  - After the failure of the 5th generation computer project, also funds for LDL ceased
  - Project adopted by University of California, LA
  - Nowadays, it is part of the InfoSleuth Agent System Project

- **Lola/Butterfly**
  - Developed by University of Passau since 1995
  - Fully implemented in Common Lisp
  - Works Bottom-Up as well as Top-Down, focus on optimization

- **Declare, Nail!, …**

- **Unfortunately, deductive DBs were a commercial failure...**
1.3 History of Deductive DBs

• …but still, the spirit of deductive databases lives on!

  – Relational Databases adopted many deductive concepts in form of common table expressions
    • Standardized in SQL-99
    • Allow recursive querying

  – The currently very popular Semantic Web reuses many ideas and techniques developed for deductive databases
• The expression was notably coined in Tim Berners Lee’s article in Scientific American, May 2001
  – Describes his vision of a future Internet
  – Describes a story of a guy named Pete looking for medical care for his mom
    • He uses a Semantic Web Agent
    • The agent is able to plan complex tasks by just accessing information in the internet
    • E.g. Finding a specialist doctor within direct vicinity of Peters home, offering the exact treatment necessary, available for a meeting at times fitting into Pete’s schedule.
The semantic web agent is able to perform those tasks by accessing and understanding the Web pages:

- e.g. recognize the opening times of the clinic, finding out, if it is close or if it is covered by the health plan.
- Also, it is able to deduce new information:
  - “Asthma is a chronic lung disease.”
  - “A pulmonologist is a doctor for diseases of the lung and the respiratory tract.”
  - “A doctor can offer treatments in his/her special field.”
- Dr. Paul is a pulmonologist $\Rightarrow$ Dr. Paul can offer asthma treatments.
1.4 The Semantic Web

- The goal of the semantic web is not to understand natural language, but provide web pages in an computer readable form.

The Semantic Web is a web of data. There is lots of data we all use every day, and it is not part of the web. I can see my bank statements on the web, and my photographs, and I can see my appointments in a calendar. But can I see my photos in a calendar to see what I was doing when I took them? Can I see bank statement lines in a calendar? Why not? Because we don't have a web of data. Because data is controlled by applications, and each application keeps it to itself.

The Semantic Web is about two things. It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing.
1.4 The Semantic Web

• This is how a website looks to a machine
  – Example: A personal website of an faculty member...

林克昌 根留台灣 可能增高

在受騙者熱心奔走之下，華裔名指揮家林克昌根留台灣的可行性又提升了幾分。兩廳院主任李炎、國家音樂廳樂團副團長黃奕明日前親赴林克昌、石聖芳寓所拜會，並提出多場客席邀約。此外，台灣省立交響樂團團長陳溢雄也早早「下訂」，邀請林克昌赴台中霧峰，從八月十日起訓練省交，為期長達一個月。

在台灣諸多公家樂團中，陳溢雄是以實際行動表達對林克昌肯定的樂界人士之一，曾多次公開表示對林克昌指揮才華的欣佩，而且幾乎每個樂季都邀請林克昌客席演出。

此外，林克昌上個月赴俄羅斯與頂尖的「俄羅斯國家管絃樂團」灌錄了柴可夫斯基晚期三大交響曲以及「羅密歐與茱麗葉」、「斯拉夫進行曲」、「義大利隨想曲」，最後的DAT母帶也在前兩天寄回台灣。製作人楊忠衡與林克昌試聽之後，都對錄音效果、音質表現感到相當滿意，楊忠衡估計呈現了七分林克昌指揮神韻。

俄羅斯國家管絃樂團首席布魯尼日前也讚譽林克昌的指揮藝術有三大特點：一是控制自如的彈性速度；二是強烈的動態對比；三是宛如呼吸歌唱的旋律處理。這些對錄音師而言都構成很大挑戰。俄國錄音師雖然採用多軌混音，但定位、場面都有可觀之處。。
Now, you could start adding some tags describing the semantic nature of some parts of the text.
1.4 The Semantic Web

- Unfortunately, the machine also does not understand the tags...

Leftovers from a previous project:

In the project, we have found it challenging to manage the data tags automatically. The machine learning algorithms struggled to interpret the complex relationships within the metadata. As a result, we have decided to manually override these tags and manually map them to a consistent schema. This approach has been time-consuming but has yielded better results in terms of data accuracy.
A first step in helping the machine is providing a schema of valid documents

— Still, this does not really help to understand the text
...especially if somebody uses a different schema

- At least, schemas can be matched:
  
  name [education] [work] [private]

- In the Semantic Web, names can be matched:
  
  name [education] [work] [private]
The semantic web goes one step further by providing also **ontologies** describing all possible tags, what they mean and how they are related

– What is a CV? How can it be read?

– If there a listing of recent lectures, then is also contributes to the education history

– etc.
During the **semantic web** section of this lecture, we will show you:

- How to model knowledge as **ontologies or taxonomies**
- How to perform **reasoning** on ontologies
- Existing standards and languages
  - RDF, RDF/S, OWL, DAML, OIL, SPARQL, etc.
- Other cool stuff!
The Semantic Web is based on ontologies

What are these? Where do they come from?

Science and philosophy always strived to explain the world and the nature of being

First formal school of studies: Aristotle’s metaphysics (‘beyond the physical’, ca. 360 BC)

Traditional branches of metaphysics

• Ontology
  – Study of being and existence

• Natural theology
  – Study of God, nature and creation

• Universal science
  – “First Principles”, logics
1.4 Taxonomies & Ontologies

- **Ontology** tries to describe everything which is (exists), and it’s relation and categorization with respect to other things in existence
  - What is existence? Which things exist? Which are entities?
  - Is existence a property?
  - Which entities are fundamental?
  - What is a physical object?
  - How do the properties of an object relate to the object itself? What features are the essence?
  - What does it means when a physical object exists?
  - What constitutes the identity of an object?
  - When does an object go out of existence, as opposed to merely change?
  - Why does anything exist rather than nothing?
• Parts of metaphysics evolved into natural philosophy
  – Study of nature and the physical universe
  – In the late 18th century, it became just ‘science’
  – Ontology is still a dominant concept in science
    • Representation of all knowledge about things
• **Taxonomies** (Τάξις: arrangement) are part of ontology

  – Groups things with similar properties into **taxa**

  – **Taxa** are put into an **hierarchical structure**
    - Hierarchy represents supertype-subtype relationships
    - Represents a **specialization** of taxa, starting with the most general one

  – Taxonomies thus **model** information into a **data structure**
    - Most notable: Linnaean Taxonomy of Life
1.4 Taxonomies & Ontologies

• **Ars Generalis Ultima**
  
  – Created in 1305 by Ramon Llull
  
  – “Ultimate” solution for the **Ars Magna** (Great Art)
    • Mechanical combination of terms to create knowledge
    • Base hope: all facts and truths can be created in such a way
  
  – Heavy use of Arbor Scientiae (**Tree of Knowledge**)
    • Tree structure showing an hierarchy of philosophical concepts
    • Together with various “machines” (paper circles, charts, etc.) reasoning was possible
Recently, creating **ontological models** became fashionable in CS

- So called **ontologies**
- Widely used in medical informatics, bio-informatics, Semantic Web, etc.

In addition to ‘normal’ data models, ontologies offer **reasoning capabilities**

- Allow to classify instances automatically
- Allow to extract additional facts from the model

Computer ontologies are usually modeled using **special languages**

- OWL, DAML+OIL, IDEF, ....

Most notably, ontologies can be found within the semantic web
Expert systems are cool!
- They are born in early 60ties as a major AI application
- Mostly based on formal logics
- Early expert systems failed their high expectations, topic became infamous
- …but: every good thing returns, **expert systems are back!**

The semantic web are also based on ideas from expert systems
- Ontologies represent knowledge
  - Idea of Ontologies are as old as science itself…
  - Technology is “new”
• **Next Lecture**

  – **Basics of Knowledge Based Systems:**
    **The 1x1 of Boolean and First Order Logics**