

From Constraint Programming to Probabilistic Programming to Approximate Programming: Observations and Thoughts

Rina Dechter

Bren School of Information and Computer Sciences

Workshop goals

- From their early days, Constraint Programming has played an important role in Artificial Intelligence, and Artificial Intelligence has played an important role in Constraint Programming.
- Artificial Intelligence is currently experiencing a great surge of interest, attention and funding. This workshop is intended to encourage efforts to increase the participation and visibility at this time of Constraint Programming in the wider AI community, to the mutual benefit of both.
- In particular, the Workshop will seek to formulate specific recommendations as to actions that members of the community might undertake, individually, or collectively through the ACP, to further identified objectives.
- As a start, it is hoped that participants at the Workshop will volunteer to take on specific roles that are deemed useful for achieving these objectives.

“From their early days, Constraint Programming has played an important role in Artificial Intelligence, and Artificial Intelligence has played an important role in Constraint Programming”

Indeed: CSP solvers and languages made a significant impact:

- **CSP** --> SAT --> Sat technology, revolutionized reasoning in AI and CS through sat-based algorithms:
 - **Logic programming**, answer-set programming, verification,
 - **Probabilistic reasoning**: max-sat, max-weighted-sat, weighted model counting.
 - **Sat-base and constraint-based planning** algorithms.
- **CSP + Optimization, soft constraints**: combined with OR, examples (AOBB, Toolbar, CPLEX)
- **Constraint Programming Languages**: CP emerged from the desire to have more efficient languages than prolog, leading to many languages. (should extend into modern probabilistic languages).
- **Global Constraints**: macros for constraint propagation, for languages.
- **AI, CP, OR**: combined with OR

“Artificial Intelligence is currently experiencing a great surge of interest, attention and funding. This workshop is intended to encourage efforts **to increase the participation and visibility at this time of Constraint Programming in the wider AI community, to the mutual benefit of both. The workshop seeks recommendation from us....”**

The excitements in AI is Learning, learning learning....Because of neural networks, lots of data, hardware

The visibility comes from success in applications: Vision, NLP, Speech, Robotics

The visibility is due, in part, also to showcase: Deep Blue, AlphaGo, Watson

CP is not about learning... currently... but so are: Planning, KR, logic programming

Should we learn from other communities:

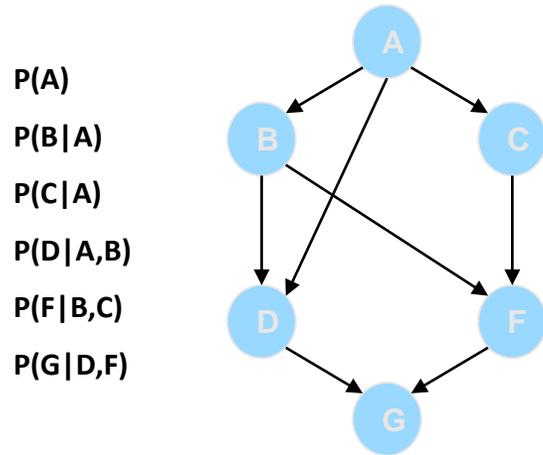
- Planning (reinforcement learning)
- Logic programming (answer set programming)
- Inductive logic programming

How to join the excitement?

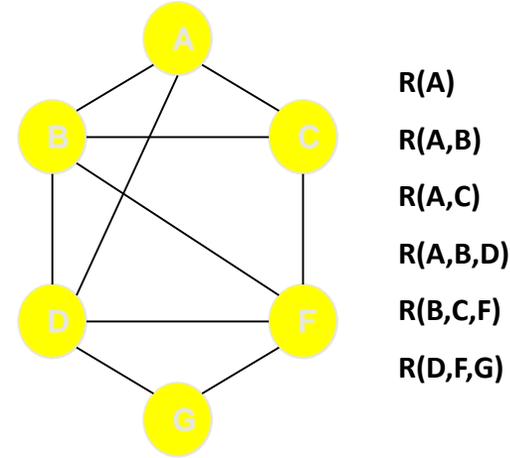
How to be visible?

- One way to join in the current excitement in AI is by going back to the roots and focus on **languages** and their accompanied algorithms for a variety of tasks. Some ways are:
 - From constraint programming to probabilistic programming,
 - From CSP to **First Order** algorithms,
 - From satisfiability and optimality to **weighted counting**,
 - Combine: uncertainty, temporal information, first order, global constraints, **continuous variables**.
 - CP should develop **algorithms improving Learning** (EM like?).
 - **Data-driven algorithms ... for what? ...to predict solver's performance and to choose solver's configurations**

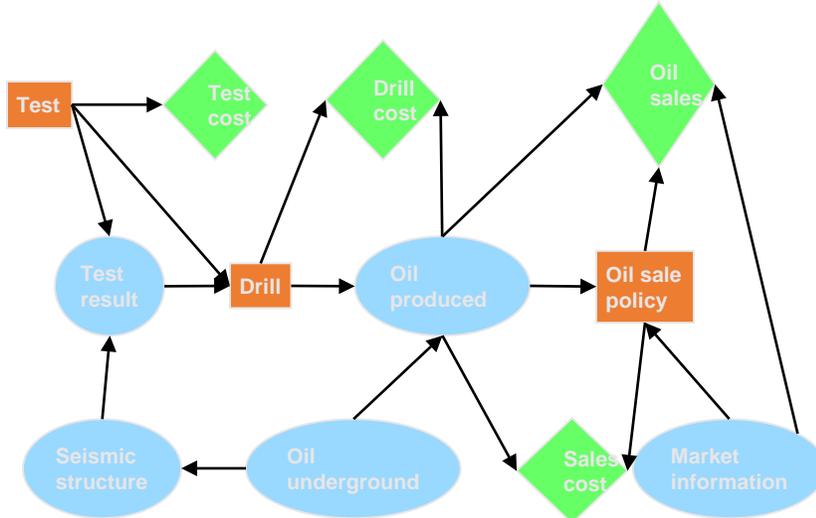
Graphical Models



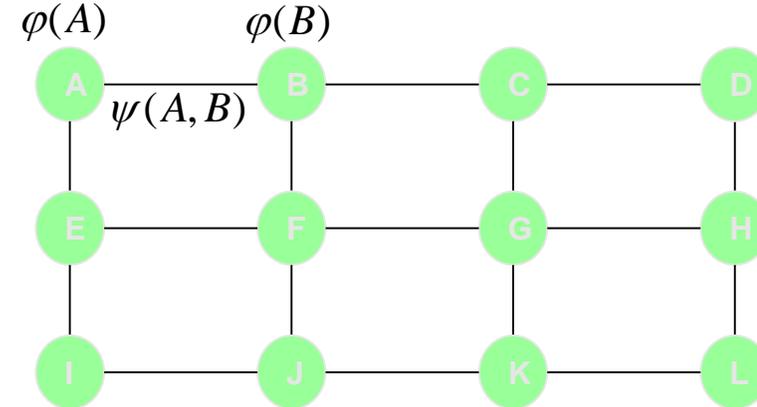
a) Belief network (directed)



b) Constraint network (undirected)



c) Influence diagram



d) Markov network

From Constraint Programming to Probabilistic Programming

- Numberjack
- Minizinc
- CLP(R)
- CHIP
- ECLIPSE
- CPLEX
- OPL
- https://en.wikipedia.org/wiki/Constraint_programming
- Markov logic networks
- Church
- Blog
- Figaro
- Problog
- BUGS
- IBAL
- PRISM
- <http://probabilistic-programming.org/wiki/Home>

Constraint programming languages are mostly based on search and is exact

Most probabilistic programming languages are sampling based and approximate

The probabilistic programming approach

Probabilistic graphical models provide a formal lingua franca for modeling and a common target for efficient inference algorithms. Their introduction gave rise to an extensive body of work in machine learning, statistics, robotics, vision, biology, neuroscience, artificial intelligence (AI) and cognitive science. However, many of the most innovative and useful probabilistic models published by the AI, machine learning, and statistics community far outstrip the representational capacity of graphical models and associated inference techniques. Models are communicated using a mix of natural language, pseudo code, and mathematical formulae and solved using special purpose, one-off inference methods. Rather than precise specifications suitable for automatic inference, graphical models typically serve as coarse, high-level descriptions, eliding critical aspects such as fine-grained independence, abstraction and recursion.

PROBABILISTIC PROGRAMMING LANGUAGES aim to close this representational gap, unifying general purpose programming with probabilistic modeling; literally, users specify a probabilistic model in its entirety (e.g., by writing code that generates a sample from the joint distribution) and inference follows automatically given the specification. These languages provide the full power of modern programming languages for describing complex distributions, and can enable reuse of libraries of models, support interactive modeling and formal verification, and provide a much-needed abstraction barrier to foster generic, efficient inference in universal model classes.

We believe that the probabilistic programming language approach within AI has the potential to fundamentally change the way we understand, design, build, test and deploy probabilistic systems. This approach has seen growing interest within AI over the last 10 years, yet the endeavor builds on over 40 years of work in range of diverse fields including mathematical logic, theoretical computer science, formal methods, programming languages, as well as machine learning, computational statistics, systems biology, probabilistic AI.

Existing probabilistic programming systems

Below we have compiled a list of probabilistic programming systems including languages, implementations/compilers, as well as software libraries for constructing probabilistic models and toolkits for building probabilistic inference algorithms.

- [Alchemy](#) is a software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation.
- [Anglican](#) is a portable Turing-complete research probabilistic programming language that includes particle MCMC inference.
- [BLOG](#), or Bayesian logic, is a probabilistic programming language with elements of first-order logic, as well as an MCMC-based inference algorithm. BLOG makes it relatively easy to represent uncertainty about the number of underlying objects explaining observed data.
- [BUGS](#) is a language for specifying finite graphical models and accompanying software for performing B(ayesian) I(nference) U(sing) G(ibbs) S(ampling), although modern implementations (such as [WinBUGS](#), [JAGS](#), and [OpenBUGS](#)) are based on Metropolis-Hastings. [BiIPS](#) is an implementation based on interacting particle systems methods like Sequential Monte Carlo.
- [Church](#) is a universal probabilistic programming language, extending Scheme with probabilistic semantics, and is well suited for describing infinite-dimensional stochastic processes and other recursively-defined generative processes (Goodman, Mansinghka, Roy, Bonawitz and Tenenbaum, 2008). The active implementation of Church is [webchurch](#). Older implementations include [MIT-Church](#), [Cosh](#), [Bher](#), and [JSChurch](#). See also [Venture](#) below.
- [Dimple](#) is a software tool that performs inference and learning on probabilistic graphical models via belief propagation algorithms or sampling based algorithms.
- [FACTORIE](#) is a Scala library for creating relational factor graphs, estimating parameters and performing inference.
- [Figaro](#) is a Scala library for constructing probabilistic models that also provides a number of built-in reasoning algorithms that can be applied automatically to any constructed models.
- [HANSEI](#) is a domain-specific language embedded in OCaml, which allows one to express discrete-distribution models



Probabilistic Inference Modulo Theories

Rodrigo de Salvo Braz
Ciaran O'Reilly
Artificial Intelligence Center - SRI International

Vibhav Gogate
University of Texas at Dallas

Rina Dechter
University of California, Irvine

IJCAI-16, July 2016



Motivation

- Consider a probabilistic model on string-valued variables:

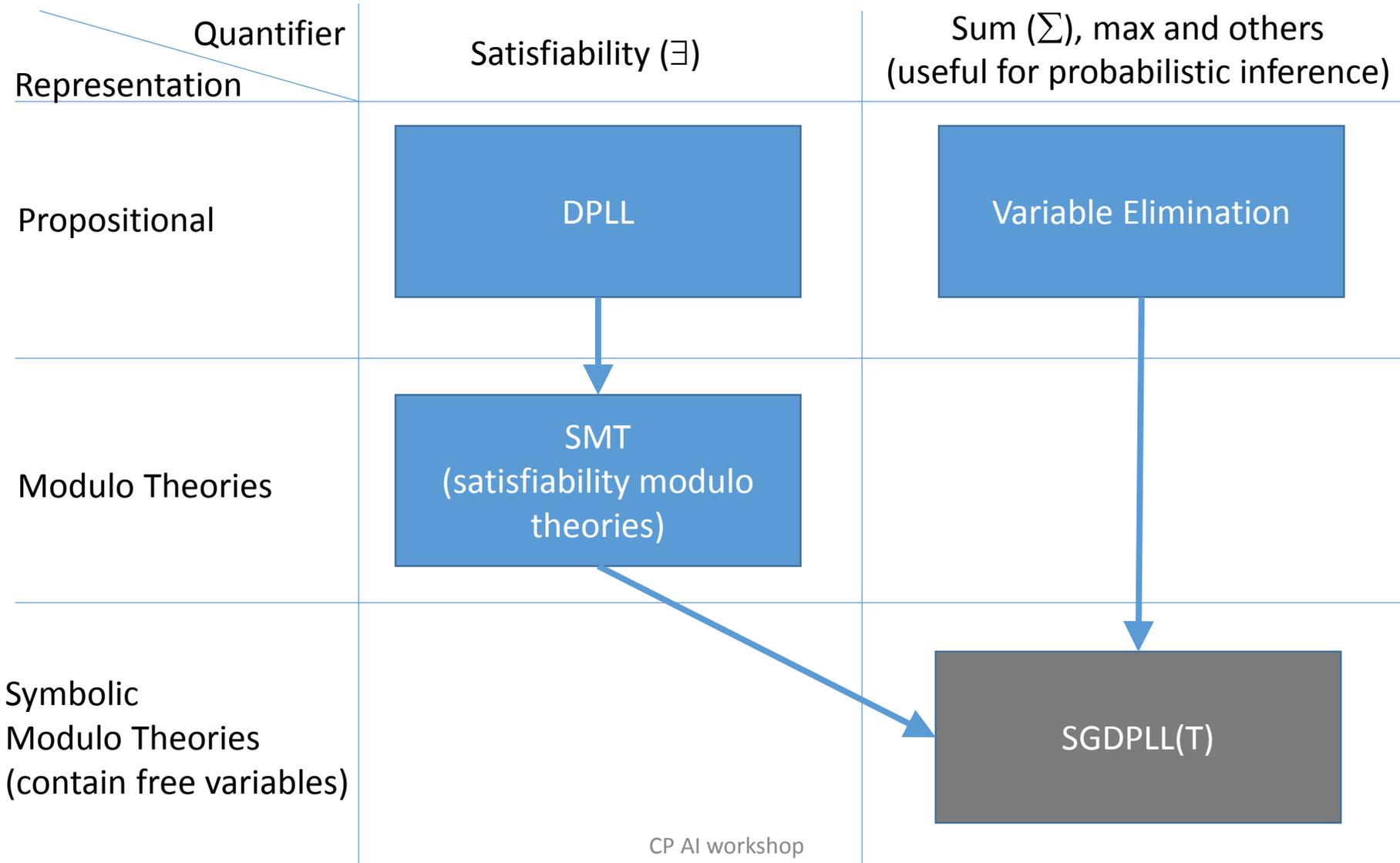
```
P(announcement | title, speaker, venue, abstract) =  
  if announcement = title + speaker + venue + abstract  
    then 0.7 else  
  if announcement = title + venue + speaker + abstract  
    then 0.3 else 0
```

```
P(speaker | name) =  
  if speaker = "Prof." + name  
    then 0.1 else  
  if speaker = name then 0.9 else 0
```

```
... // more statements, defining knowledge about  
    // names, titles etc.
```

- Exact graphical models algorithms typically iterate over values of each variable, but here they are infinite
- Sampling has its own set of disadvantages

Overview



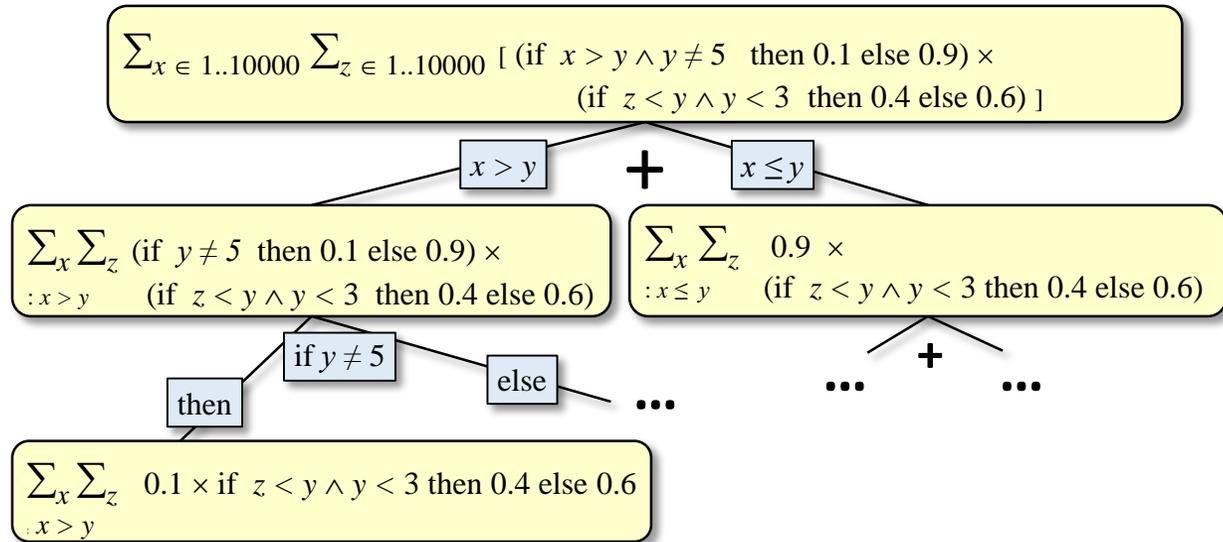
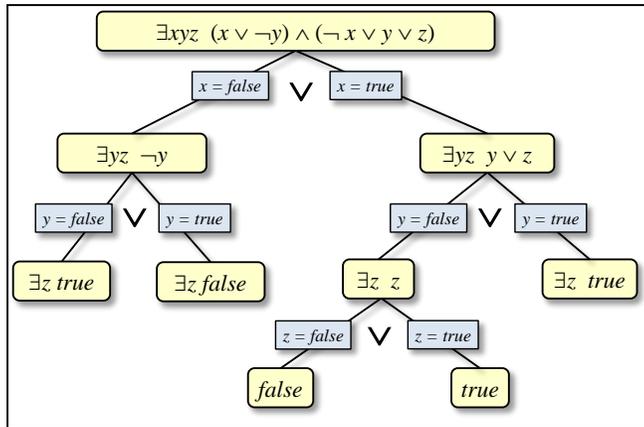
Symbolic Generalized DPLL(T)

- Similar to SMT, but based on
 - Summation (or other quantifiers), besides \exists
 - Partial quantification (free variables)

$$\sum_{x \in 1..10000} \sum_{z \in 1..10000} \quad (\text{if } x > y \wedge y \neq 5 \text{ then } 0.1 \text{ else } 0.9) \\ \times (\text{if } z < y \wedge y < 3 \text{ then } 0.4 \text{ else } 0.6)$$

- Note that y is a free variable
- Summed expression is not Boolean
- Language is not propositional (\neq , $<$, ...)

Symbolic Generalized DPLL(T) – SGDPLL(T)

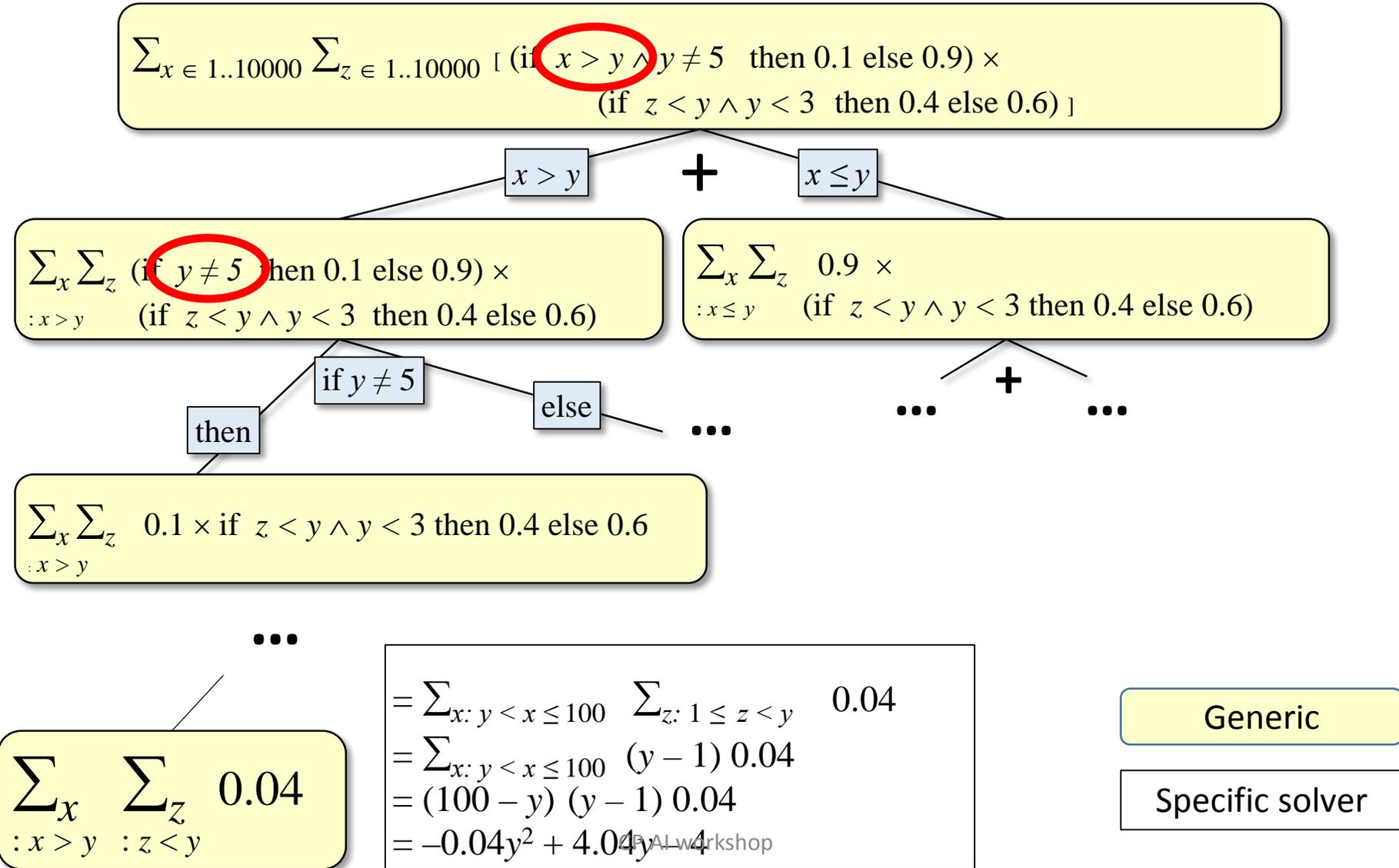


Condition on literals until base case with no literals in main expression:

$$\sum_{x > y} \sum_{z < y} 0.04$$

$$\begin{aligned} &= \sum_{x: y < x \leq 100} \sum_{z: 1 \leq z < y} 0.04 \\ &= \sum_{x: y < x \leq 100} (y - 1) 0.04 \\ &= (100 - y) (y - 1) 0.04 \\ &= -0.04y^2 + 4.04y - 4 \end{aligned}$$

Symbolic Generalized DPLL(T)

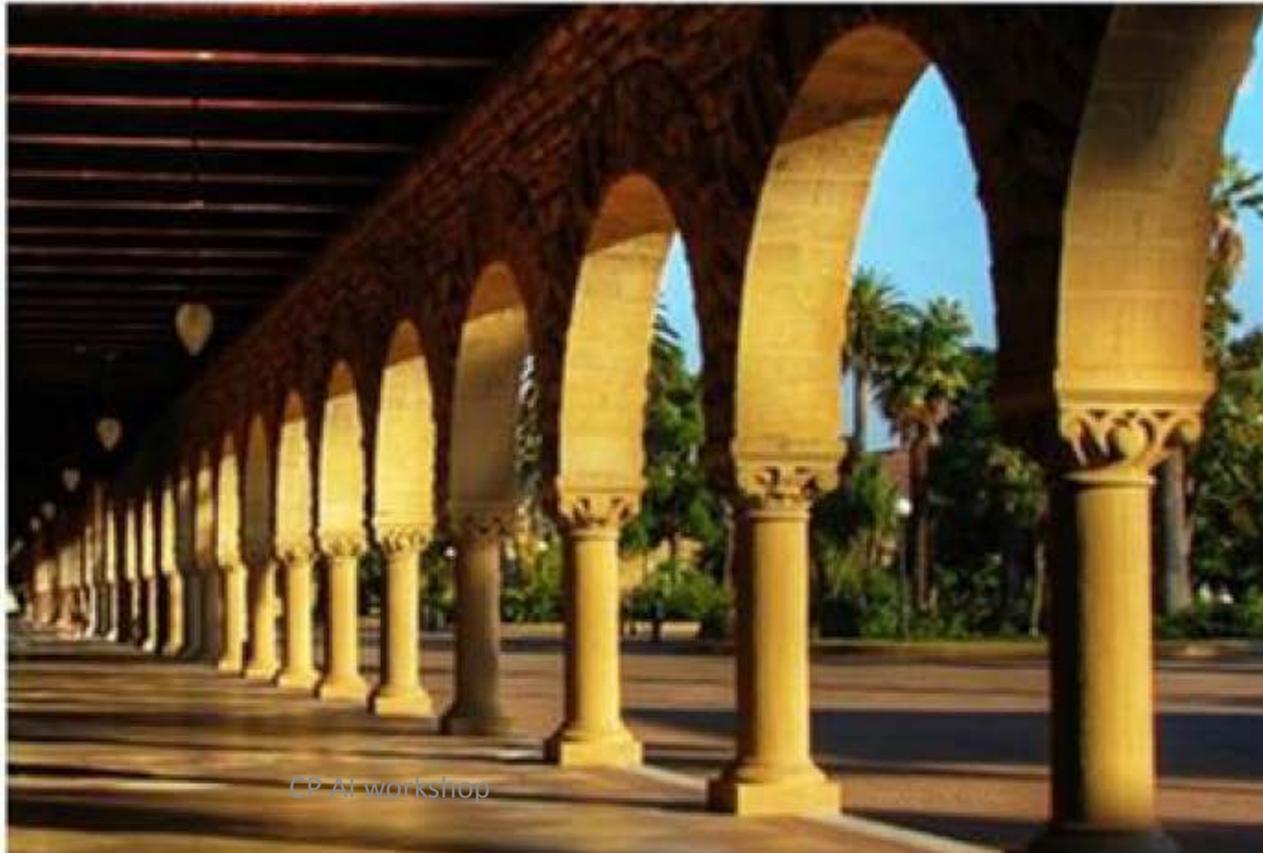


ARTIFICIAL INTELLIGENCE AND LIFE IN 2030

ONE HUNDRED YEAR STUDY ON ARTIFICIAL INTELLIGENCE | REPORT OF THE 2015 STUDY PANEL | SEPTEMBER 2016

PREFACE

The One Hundred Year Study on Artificial Intelligence, launched in the fall of 2014, is a long-term investigation of the field of Artificial Intelligence (AI) and its influences on people, their communities, and society. It considers the science, engineering, and deployment of AI-enabled computing systems. As its core activity, the Standing Committee that oversees the One Hundred Year Study forms a Study Panel every five years to assess the current state of AI. The Study Panel reviews AI's progress in the years following the immediately prior report, envisions the potential advances that lie ahead, and describes the technical and



AI and Life in 2030

ONE HUNDRED YEAR STUDY ON ARTIFICIAL
INTELLIGENCE | REPORT OF THE 2015 STUDY PANEL |
SEPTEMBER 2016

https://ai100.stanford.edu/sites/default/files/ai_100_report_0831fnl.pdf

- **STUDY PANEL**
- Peter Stone, University of Texas at Austin, Chair
- Rodney Brooks, Rethink Robotics

The committee ultimately focused on “AI and Life in 2030” that AI’s various uses and occur independently of one multitude of other societal developments.

CP and Life in 2030

Let’s see what questions/ideas in the report can advise us in CP

Standing Committee of the One Hundred Year Study of Artificial Intelligence

Barbara J. Grosz, Chair
Russ Altman
Eric Horvitz
Alan Mackworth
Tom Mitchell
Deidre Mulligan
Yoav Shoham

- David Parkes, Harvard University
- William Press, University of Texas at Austin
- AnnaLee (Anno) Saxenian, University of California, Berkeley
- Julie Shah, Massachusetts Institute of Technology
- Milind Tambe, University of Southern California
- Astro Teller, X

What's happening currently (in AI)

“Computer vision and AI planning, for example, drive the video games that are now a bigger entertainment industry than Hollywood. **Deep learning, a form of machine learning based on layered representations of variables referred to as neural networks, has made speech-understanding practical** on our phones and in our kitchens, and its algorithms can be applied widely to an array of applications that rely on pattern recognition. **Natural Language Processing (NLP) and knowledge representation and reasoning have enabled a machine to beat the Jeopardy champion** and are bringing new power to Web searches.”

So, let's ask: What's happening currently in CP?

What applications will have the greatest impact

“The Study Panel further narrowed its inquiry to eight domains where AI is already having or is projected to have the greatest impact: **transportation, healthcare, education, low-resource communities, public safety and security, employment and workplace, home/service robots, and entertainment**”

We ask: How CP can contribute to transportation, healthcare, education, human-aware programming?

What's next for AI?

What's next for AI research?

The research that fuels the AI revolution has also seen rapid changes. Foremost among them is the maturation of machine learning, stimulated in part by the rise of the digital economy, which both provides and leverages large amounts of data. Other factors include the rise of cloud computing resources and consumer demand for widespread access to services such as speech recognition and navigation support.

Machine learning has been propelled dramatically forward by impressive empirical successes of artificial neural networks, which can now be trained with huge data sets and large-scale computing. This approach has been come to be known as “deep learning.” The leap in the performance of information processing algorithms has been accompanied by significant progress in hardware technology for basic operations such as sensing, perception, and object recognition. New platforms and markets for data-driven products, and the economic incentives to find new products and markets, have also stimulated research advances. **Now, as it becomes a central force in society, the field of AI is shifting toward building intelligent systems that can collaborate effectively with people, and that are more generally human-aware, including creative ways to develop interactive and scalable ways for people to teach robots.**

Here we ask what's next for CP...

Overall trends and the future of AI research

The resounding success of the data-driven paradigm has displaced the traditional paradigms of AI. Procedures such as theorem proving and logic-based knowledge representation and reasoning are receiving reduced attention, in part because of the ongoing challenge of connecting with real-world groundings. Planning, which was a mainstay of AI research in the seventies and eighties, has also received less attention of late due in part to its strong reliance on modeling assumptions that are hard to satisfy in realistic applications. Model-based approaches—such as physics-based approaches to vision and traditional control and mapping in robotics—have by and large given way to data-driven approaches that close the loop with sensing the results of actions in the task at hand. Bayesian reasoning and graphical models, which were very popular even quite recently, also appear to be going out of favor, having been drowned by the deluge of data and the remarkable success of deep learning.

Over the next fifteen years, the Study Panel expects **an increasing focus on developing systems that are human-aware**, meaning that they specifically model, and are specifically designed for, the characteristics of the people with whom they are meant to interact. There is a lot of interest in trying to find new, creative ways to develop interactive and scalable ways to teach robots. ...

The Study Panel also expects a reemergence of some of the traditional forms of AI as practitioners come to realize the inevitable limitations of purely end-to-end deep learning approaches. We encourage young researchers not to reinvent the wheel, but rather to maintain an awareness of the significant progress in many areas of AI during the first fifty years of the field, and in related fields such as control theory, cognitive science, and psychology.

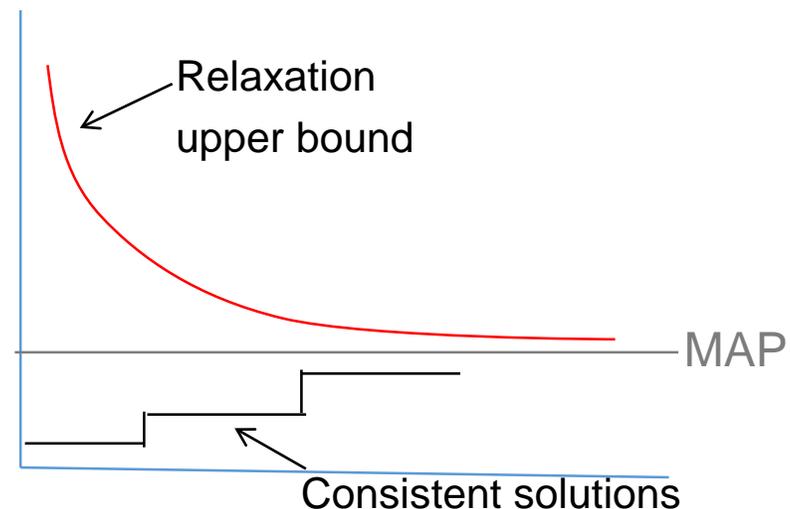
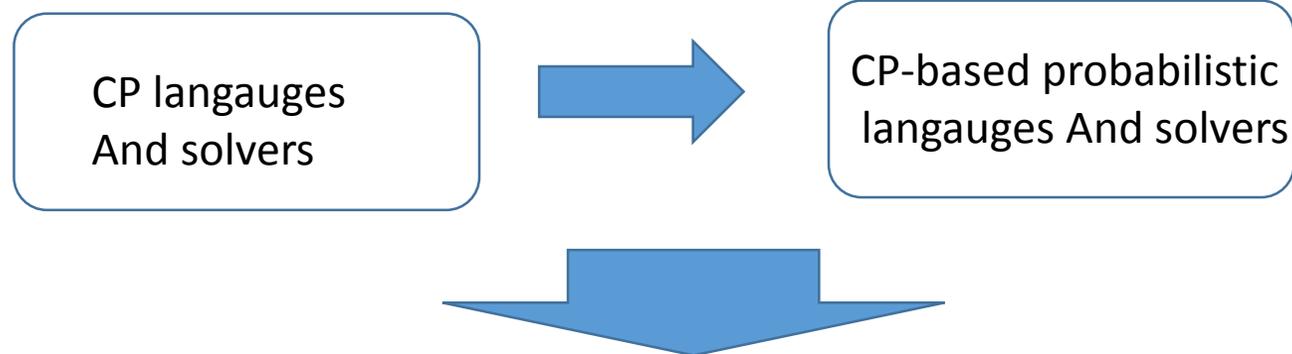
Trends for AI

These trends drive the currently “hot” areas of AI research into both fundamental methods and application areas:

- **Large-scale machine learning** concerns the design of learning algorithms, and scaling existing algorithms, to work with extremely large data.
- **Deep learning**, a class of learning procedures, has facilitated object recognition in images, video labeling, and activity recognition, and is making significant inroads into other areas of perception, such as audio, speech, and natural language processing.
- **Reinforcement learning** is a framework that shifts the focus of machine learning from pattern recognition to experience-driven sequential decision-making. It promises to carry AI applications forward toward taking actions in the real world. While largely confined to academia over the past several decades, it is now seeing some practical, real-world successes.
- **Robotics** is currently concerned with the state manipulation of objects in interactive environments, the reliability and generality of computer vision and other perception, and the deep learning. For the first time, computers are able to learn from experience.
- **Computer vision** is currently concerned with the state manipulation of objects in interactive environments, the reliability and generality of computer vision and other perception, and the deep learning. For the first time, computers are able to learn from experience.
- **Natural Language Processing** research is now shifting from rule-based systems to deep learning. For the first time, computers are able to learn from experience.
- **Collaborative systems** research is now shifting from rule-based systems to deep learning. For the first time, computers are able to learn from experience.
- **Crowdsourcing and human computation** research investigates methods to augment computer systems by making automated calls to human expertise to solve problems that computers alone cannot solve well.
- **Algorithmic game theory and computational social choice** draw attention to the economic and social computing dimensions of AI, such as how systems can handle potentially misaligned incentives, including self-interested human participants or firms and the automated AI-based agents representing them.
- **Internet of Things (IoT)** research is devoted to the idea that a wide array of devices, including appliances, vehicles, buildings, and cameras, can be interconnected to collect and share their abundant sensory information to use for intelligent purposes.
- **Neuromorphic computing** is a set of technologies that seek to mimic biological neural networks to improve the hardware efficiency and robustness of computing systems, often replacing an older emphasis on separate modules for input

What is CP relevance to any of these trends?

Future trends for CP?



Approximate Programming:

1. Anytime bounds on accuracy
2. likelihood of solving with current assumptions
3. What if programming tools for user
4. how hard is the problem?
5. Learn to predict from simulations
6. 6. what parameters should I choose for the solver?
7. Software, tools, open source. Their availability makes it or breaks it

Discussion?

- Thanks you!