AND/OR Search Spaces: for Anytime Probabilistic Reasoning

Rina Dechter
UCI

Main Collaborators:
Alexander Ihler
Kalev Kask
Radu Marinescu
Bobak Pezeshki
Junky Lee
Outline

• AND/OR search spaces vs. Probabilistic circuits
• Review AND/OR search spaces for PGM
• AND/OR Multi-valued Decision Diagrams (AOMDD)
• Anytime algorithms over AND/OR search spaces
• AND/OR Abstraction sampling.
• Moving forward: Neurosymbolic, causality
Outline

• AND/OR search spaces vs. Probabilistic circuits
• Review of AND/OR search spaces for PGM
• AND/OR Abstraction sampling, balancing exact vs approximate, time vs memory vs accuracy.
• Moving forward: Reasoning under partial models and data.
AND/OR vs Arithmetic Circuit Example

\[ \begin{array}{c|c} \hline A & \theta_A \\
\hline true & .5 \\
false & .5 \\
\hline \end{array} \quad \begin{array}{c|c|c} \hline A & B & \theta_{D|A} \\
\hline true & true & 1 \\
true & false & 0 \\
false & true & 0 \\
false & false & 1 \\
\hline \end{array} \quad \begin{array}{c|c|c} \hline A & C & \theta_{C|A} \\
\hline true & true & .8 \\
true & false & .2 \\
false & true & .2 \\
fakefalse & false & .8 \\
\hline \end{array} \]

\[ \begin{array}{c} \lambda_a \\
\theta_a \\
\theta_{b|\bar{a}} \\
\theta_{b|a} \\
\theta_{c|\bar{a}} \\
\theta_{c|a} \\
\hline 0 \\
1 \\
0 \\
1 \\
0 \\
1 \\
\hline \end{array} \quad \begin{array}{c} + \\
* \\
\lambda_b \\
\theta_{b|\bar{a}} \\
\theta_{b|a} \\
\theta_{c|\bar{a}} \\
\theta_{c|a} \\
\hline \lambda_c \\
\lambda_{c|\bar{a}} \\
\lambda_{c|a} \\
\lambda_{\bar{c}|a} \\
\lambda_{\bar{c}|\bar{a}} \\
\lambda_{\bar{c}|a} \\
\hline \end{array} \]
AND/OR Spaces and Circuits

**AND/OR space**
- Isomorphic in practice
- Pseudo trees
- *Used, anytime algorithms*
- Input: a full graphical model
- Input is a graph + data
- Can exploit local structure
- Multi-valued variables and tabular representation

**Probabilistic Circuits**
- Can be more expressive
- Dtrees
- *Used for compilation*
- Input: a full graphical model
- Input is a graph/circuit + data.
- Exploit logical structure.
- Bi-valued variables, logical functions.
Graphical Models – Overview

Bayesian Networks

Markov Logic

Protein Folding and Design

Deep Boltzmann Machines

Influence Diagrams

[ Yanover & Weiss 2002 ]
Probabilistic Reasoning Problems

- Exact Algorithm by BE or AND/OR search, Complexity

| Max-Inference: | \( f(x^*) = \max_x \prod_{\alpha} f_{\alpha}(x_{\alpha}) \) |
| Sum-Inference: | \( Z = \sum_x \prod_{\alpha} f_{\alpha}(x_{\alpha}) \) |
| Mixed-Inference (MMAP): | \( f_M(x^*_M) = \max_{x_M} \sum_{x_S} \prod_{\alpha} f_{\alpha}(x_{\alpha}) \) |
| Mixed-Inference (MEU): | \( \text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots,X_n} (\prod_{i} P_i) \times (\sum r_i) \) |

Influence diagrams For planning
Anytime vs Compilation Methodology

- We want a unifying methodology that is anytime and provide bounds that improve with time regardless of memory.
- Winning frameworks: search, or sampling guided by heuristics generated via compilation.
Outline

• AND/OR search spaces and Probabilistic circuits
• AND/OR search spaces for PGM
• Anytime algorithms over AND/OR space
• AND/OR Abstraction sampling, balancing exact vs approximate, time vs memory vs accuracy.
• Moving forward: Reasoning under partial models and data.
AND/OR vs. OR

AND/OR size: \(\exp(4)\), OR size \(\exp(6)\)
From AND/OR Tree
To an AND/OR Graph
### Potential Search Spaces

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>(f_1)</th>
<th>(f_2)</th>
<th>(f_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Full OR search tree
- 126 nodes

#### Context minimal OR search graph
- 28 nodes

#### Full AND/OR search tree
- 54 AND nodes

#### Context minimal AND/OR search graph
- 18 AND nodes

---

**Computes any query:**
- Constraint satisfaction
- Optimization (MAP)
- Marginal \(P(e)\)
- **Marginal map**

---

Any query is best computed
Over the c-minimal AO search space
Cost of a Solution Tree

Cost of the solution tree: the product of weights on its arcs

Cost of \((A=0,B=1,C=1,D=1,E=0)\) = \(0.6 \cdot 0.6 \cdot 0.5 \cdot 0.8 \cdot 0.5 = 0.0720\)
Value of a Node (e.g., Probability of Evidence)

\[
P(E | A, B)\quad P(B | A)\quad P(C | A)\quad P(A)
\]

\[\begin{array}{ccc}
A & B & E=0 \quad E=1 \\
0 & 0 & .4 \quad .6 \\
0 & 1 & .5 \quad .5 \\
1 & 0 & .7 \quad .3 \\
1 & 1 & .2 \quad .8
\end{array}\]

Evidence: E=0

\[
\begin{array}{ccc}
P(D=1, E=0) = ?
\end{array}
\]

\[
\begin{array}{c}
P(D | B, C) \\
\end{array}
\]

\[\begin{array}{ccc}
B & C & D=0 \quad D=1 \\
0 & 0 & .2 \quad .8 \\
0 & 1 & .1 \quad .9 \\
1 & 0 & .3 \quad .7 \\
1 & 1 & .5 \quad .5
\end{array}\]

Evidence: D=1

**Value of node = updated belief for sub-problem below**

**AND node: product**

**OR node: Marginalization by summation**

\[\prod_{n' \in \text{children}(n)} v(n')\]

\[\sum_{n \in \text{children}(n)} w(n, n') v(n')\]
### Answering Queries: Sum-Product (Belief Updating)

#### Context

- **Context:** \([\text{A}]\), \([\text{B}]\), \([\text{C}]\), \([\text{E}]\)
- **Evidence:** \(\text{E}=0\)

#### Cache table for D

<table>
<thead>
<tr>
<th>(\text{B} \text{C})</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>.8</td>
</tr>
<tr>
<td>0 1</td>
<td>.9</td>
</tr>
<tr>
<td>1 0</td>
<td>.7</td>
</tr>
<tr>
<td>1 1</td>
<td>.1</td>
</tr>
</tbody>
</table>

#### Result: \(P(D=1, E=0)\)

- **Cache table for D:**
  - \(\text{B} \text{C} \text{D}=0 \text{D}=1\)
  - \(0 0 .2 .8\)
  - \(0 1 .1 .9\)
  - \(1 0 .3 .7\)
  - \(1 1 .5 .5\)

**Graphical Representation:**

1. **Node A**
2. **Node B**
3. **Node C**
4. **Node D**
5. **Node E**

**Edges and Connections:**

- **A** to **B**, **C**, **D**, **E**
- **B** to **C**
- **C** to **D**, **E**
- **D** to **E**
- **E** to **D**

**Values:**

- **A:**
  - \(P(A)\)
  - \(P(A|E=0)\) = 0.6
  - \(P(A|E=1)\) = 0.4

- **B:**
  - \(P(B)\)
  - \(P(B|A)\)
  - \(P(B|E=0)\) = 0.6
  - \(P(B|E=1)\) = 0.5

- **C:**
  - \(P(C)\)
  - \(P(C|A)\)
  - \(P(C|E=0)\) = 0.8
  - \(P(C|E=1)\) = 0.7

- **D:**
  - \(P(D|B,C)\)
  - \(P(D|E)\)
  - \(P(D|\text{Evidence})\)

- **E:**
  - \(P(E|A,B)\)
  - \(P(E|A)\)
  - \(P(E|B)\)
  - \(P(E|C)\)
  - \(P(E|D)\)
  - \(P(E|\text{Evidence})\)
The Impact of the Pseudo Tree

w=4, h=8

What is a good pseudo tree?

w=5, h=6

How to find a good one?

Min-fill [Kjaerulff, 1990]

Hypergraph Partitioning (hMetis)
Outline

• AND/OR search spaces vs. Probabilistic circuits
• Review AND/OR search spaces for PGM
• AND/OR Multi-valued Decision Diagrams (AOMDD)
• Anytime algorithms over AND/OR search spaces
• AND/OR Abstraction sampling, balancing exact vs approximate, time vs memory vs accuracy.
• Moving forward: Neurosymbolic, causality
From Context Minimal AND/OR Graphs to AND/OR MDDs

[Mateescu, Marinescu, Lam, Dechter, 2007, 2013]

Figure 20: AND/OR search tree and context minimal graph

Figure 22: AOMDD for the weighted graph
Outline

- AND/OR search spaces vs. Probabilistic circuits
- Review AND/OR search spaces for PGM
- AND/OR Multi-valued Decision Diagrams (AOMDD)
- Anytime algorithms via AND/OR heuristic search
- AND/OR Abstraction sampling, balancing exact vs approximate, time vs memory vs accuracy.
- Moving forward: Neurosymbolic, causality
Anytime Algorithms via Heuristic Search

Heuristic $h(n)$: Estimate of the mass/value of the subtree rooted at node $n$.

Applicable to any task.
Mini-Bucket Elimination

For optimization

[Dechter & Rish 2003]

Tighten by cost-shifting

```
\begin{align*}
\lambda_{B\to C}(a, c) &= \max_b f(a, b) f(b, c) \\
\lambda_{B\to D}(d, e) &= \max_b f(b, d) f(b, e) \\
\lambda_{C\to E}(a, e) &= \max_f \ldots
\end{align*}
```

Can tighten heuristics using cost-shifting, Power summation and increased i-bound
Weighted Mini-Bucket
(for summation bounds)

Exact bucket elimination:
\[
\lambda_B(a, c, d, e) = \sum_b \left[ f(a, b) \cdot f(b, c) \cdot f(b, d) \cdot f(b, e) \right]
\leq \left[ \sum_b f(a, b) f(b, c) \right] \cdot \left[ \sum_b f(b, d) f(b, e) \right]
= \lambda_{B \rightarrow C}(a, c) \cdot \lambda_{B \rightarrow D}(d, e)
\]

where \( \sum_x f(x) = \left[ \sum_x f(x)^{1/w} \right]^w \)
is the weighted or "power" sum operator

By Holder's inequality,
\[
\sum_x f_1(x) f_2(x) \leq \left[ \sum_x f_1(x) \right] \left[ \sum_x f_2(x) \right]
\]
where \( w_1 + w_2 = w \) and \( w_1 > 0, w_2 > 0 \)
(lower bound if \( w_1 > 0, w_2 < 0 \))

mathbf{U} = \text{upper bound}

IJCAI 2015

[Liu & Ihler 2011]
Anytime Algorithms via Heuristic Search

- We used a wide spectrum of heuristic search ideas to yield anytime algorithms with anytime bounds.
- **Tasks**: MAP, m-best, Partition function, Summation, Marginal Maps, Influence diagrams
- **Search methods**: Best-first, BnB, recursive BFs, Breadth-rotating for anytime AND/OR, Weighted heuristic, Dynamic vs static heuristic, look-ahead, parallel and distributed processing
MBE Heuristic for AO Search (MAP)

\[ f(T') = w(A,0) + w(B,1) + w(C,0) + w(D,0) + h(D,0) + h(F) = 12 \leq f^*(T') \]

\[ h(n) \leq v(n) \]

**Bucket A:**
\[ f(a) \lambda_{E \to A}(a) \]

**Bucket E:**
\[ \lambda_{C \to E}(a,e) \lambda_{D \to E}(a,e) \]

**Bucket D:**
\[ f(a,d) \lambda_{B \to D}(d,e) \]

**Bucket C:**
\[ \lambda_{B \to C}(a,c) f(c,a) f(c,e) \]

**Bucket B:**
\[ f(a,b) f(b,c) f(b,d) f(b,e) \]

**Lower Bound:**
\[ L = \min_B \sum f(\cdot) \]
AND/OR Search for Marginal MAP

MAP variables

SUM variables

constrained pseudo tree

primal graph

\[ f(a, b) f(b, c) f(b, d) \]

\[ \lambda_{B \rightarrow C}(a, c) f(a, c) f(c, e) \]

\[ f(a, d) \lambda_{B \rightarrow D}(d, e) \]

\[ \lambda_{C \rightarrow E}(a, e) \lambda_{D \rightarrow E}(a, e) \]

\[ f(a) \lambda_{E \rightarrow A}(a) \]

[Marinescu, Dechter and Ihler, 2014]
Anytime Solvers for Marginal MAP


- **Weighted Best-First search:**
  - Weighted Restarting AOBF (WAOBF)
  - Weighted Restarting RBFAO0 (WRBFAO0)
  - Weighted Repairing AOBF (WRAOBF)

- **Interleaving Best-first and depth-first search:**
  - Look-ahead (LAOBF),
  - alternating (AAOBF)

**Weighted A* search** [Pohl 1970]
- non-admissible heuristic
- Evaluation function:
  \[ f(n) = g(n) + w \cdot h(n) \]
- Guaranteed w-optimal solution, cost \( C \leq w \cdot C^* \)

- Better guidance for depth-first dives using improved heuristics
- Memory robust best-first search using improved lower bounds
Anytime Bounding of Marginal MAP

(UAI’14, IJCAI’15, AAAI’16, AAAI’17, (Marinescu, Lee, Ihler, Dechter)

• Search: LAOBF, AAOBF, BRAOBB, WAOBF, WAOBF-rep
• heuristic: WMB-MM (20)
• memory: 24 GB

• Anytime lower and upper bounds from hard problem instances with i-bound 12 (left) and 18 (right).

• The horizontal axis is the CPU time in log scale and the vertical axis is the value of marginal MAP in log scale.
Students’ Theses

- **Bozhena Bidyuk.** "Exploiting Graph Cutsets for Sampling-Based Approximations in Bayesian Networks“, **2006**
- **Robert Mateescu.** "AND/OR Search Spaces for Graphical Models", **2007**.
- **Radu Marinescu.** "AND/OR Search Strategies for Combinatorial Optimization in Graphical Models."
- **Vibhav Gogate.** "Sampling Algorithms for Probabilistic Graphical Models with Determinism." , **2009**.
- **Andrew Gelfand.** "Bottom-Up Approaches to Approximate Inference and Learning in Discrete Graphical Models."
- **Natalia Flerova.** "Methods for advancing combinatorial optimization over graphical models", **2015**.
- **William Lam.** "Advancing Heuristics for Search over Graphical Models" **2017**.
- **Qi Lou.** "Anytime Approximate Inference in Graphical Models" *Ph. D Thesis **2018***.
- **Junkyu Lee.** "Decomposition Bounds for Influence Diagrams" *Ph.D Thesis, **2020***.
AO search for MAP winning
UAI Probabilistic Inference Competitions

- **2006** (aolib)
- **2008** (aolib)
- **2011** (daoopt)
- **2014** (daoopt) (daoopt) (merlin)

MPE/MAP

MMAP
Software

- **My software page**
- **daoopt**
  - [https://github.com/lotten/daoopt](https://github.com/lotten/daoopt)
    (distributed and standalone AOBB solver)
- **merlin**
    (standalone WMB, AOBB, AOBF, RBFAOO solvers)
  - Open source, BSD license

**pyGMs**: Python Toolbox for Graphical Models by Alexander Ihler.
Outline

• AND/OR search spaces vs. Probabilistic circuits
• Review AND/OR search spaces for PGM
• AND/OR Multi-valued Decision Diagrams (AOMDD)
• Anytime algorithms via AND/OR heuristic search
• AND/OR Abstraction sampling
• Moving forward: Neurosymbolic, causality
Between Sampling to Searching

Summation queries, partition function

Importance sampling

2-config-subtree sampling

4-config-subtree sampling

More searching less sampling

Simon Institute 10/17/2023
Stratified sampling

- Knuth 1975, Chen 1992 estimate search space size
- Partially enumerate, partially sample
  - Subdivide space into parts
  - Enumerate over parts, sample within parts
  - “Probe”: random draw corresponding to multiple states
- Theorem (Rizzo 2007): The variance reduction moving from Importance Sampling (IS) to Stratified IS with $k$ strata’s (under some conditions) is
  \[ k \cdot \text{var}(Z_j) \]
An abstraction function, $a : T \to I^+$ partitions the nodes in $T$.

It is layer-based: Only nodes at the same level have the same abstract state.

Examples: a heuristic function, Context-based abstraction.
Full OR Tree

\[ Z(A=0, B=1, C=1) = 0.6 \times 0.7 \times 0.8 \]
Method 1 – OR Tree

\[ Z_{est} = 4 \times (0.6 \times 0.7 \times 0.8) + 4 \times (0.4 \times 0.1 \times 0.6) = 1.44 \]
Abstraction Sampling - AND/OR

- Input: Abstraction function \( a \), (partition the states at each level), a sampling proposal \( p \).
- Traverse AND/OR search tree breadth-first
- Compute estimate \( \hat{Z} \)

(d) Full AND/OR search tree
(e) Probe - AND/OR

\[ Z = 4070 \]

\[ \bar{Z} = 3784 \]
AND/OR Abstraction Sampling

Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$, pseudo-tree

Key Points:
- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
New Scalable AOAS

New AND/OR abstraction sampling scheme that allows for non-proper abstractions while still ensuring formation of valid probes.

Key Points:

- Performs non-proper abstractions
- Expands along a depth first traversal of the guiding pseudo tree
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
New Scalable AOAS

New AND/OR abstraction sampling scheme that allows for non-proper abstractions while still ensuring formation of valid probes.

Key Points:
- Performs non-proper abstractions
- Expands along a depth first traversal of the guiding pseudo tree
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$

Key Points:
- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$

Key Points:
- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
Input: Abstraction function \( a \), (partition the states at each level). Sampling proposal \( p \)

Key Points:
- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$.

Key Points:

- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$

Key Points:

- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
AND/OR Abstraction Sampling

Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$

Key Points:

- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
AND/OR Abstraction Sampling

Input: Abstraction function $a$, (partition the states at each level). Sampling proposal $p$

Key Points:
- Expands along a depth first traversal of the guiding pseudo tree
- Perform abstraction at each level
- Immediately performs recursive pruning of branches that cannot be part of valid configurations
AND/OR Abstraction Sampling

Input: Abstraction function $\alpha$, (partition the states at each level). Sampling proposal $p$

\[ \hat{Z} = \frac{1}{K} \sum_{k=1}^{K} Z'_k \]

\[ p(n) \leftarrow \frac{w(n) \cdot g(n) \cdot h(n) \cdot r(n)}{\sum_{m \in A_i} w(m) \cdot g(m) \cdot h(m) \cdot r(m)} \]
Properties

**Complexity**

$O(n \cdot m)$ where $n$ is the number of variables, and $m$ is the number of abstract states per variable.

**AOAS is and Unbiased Estimator of the Partition Function**

**Theorem 2** (unbiasedness). Given a graphical model $\mathcal{M} = (\mathbf{X}, \mathbf{D}, \Phi)$, algorithm AOAS provides an unbiased estimate for the partition function of $\mathcal{M}$.

**Accuracy/Variance reduction:** Stratified Importance Sampling reduce the variance linearly in number of abstract states and the variance between abstract states.
## Abstraction Function Comparison

<table>
<thead>
<tr>
<th>Abstraction Function</th>
<th>Description</th>
<th>Randomized</th>
<th>Refinement Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>randCB</td>
<td>nodes partitioned into abstract states based on assignments to a random subset of their context variables</td>
<td>yes</td>
<td>number of abstract states</td>
</tr>
<tr>
<td>relCB</td>
<td>nodes partitioned into abstract states based on equivalent assignments to their most recent context variables</td>
<td>no</td>
<td>number of immediate context variables to consider</td>
</tr>
<tr>
<td>simpleHB</td>
<td>nodes partitioned into equal cardinality abstract states after being ordered by their sub-problem heuristic estimates</td>
<td>no</td>
<td>number of abstract states</td>
</tr>
<tr>
<td>minVarHB</td>
<td>nodes partitioned into abstract states to minimize the total internal variance of each abstract state w.r.t. node sub-problem heuristic estimates</td>
<td>no</td>
<td>number of abstract states</td>
</tr>
</tbody>
</table>

How can we determine which abstraction and what granularity to use?
Results

grid80x80.f10.wrap


AOAS
i-bound: 10. w: 29. h: 374. upB: 23580.7
Current Status of AOAS

• AOAS is highly promising
• Trading off sampling and searching is better over AND/OR space
• Using abstractions yield often superior performance
• A lot more to explore (what abstraction function and what granularity, can we learn the abstraction function)
• But no bounds. Only unbiasedness.
## New UAI Competition

- **UAI Competition 2022**

<table>
<thead>
<tr>
<th>Solver</th>
<th>20sec</th>
<th>1200sec</th>
<th>3600sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>uai14-pr</td>
<td>61.7</td>
<td>96.8</td>
<td>96.7</td>
</tr>
<tr>
<td>ibia-pr</td>
<td>53.6</td>
<td>96.6</td>
<td>97.1</td>
</tr>
<tr>
<td>AbstractionSampling</td>
<td>78.9</td>
<td>91.7</td>
<td>93.9</td>
</tr>
<tr>
<td>lbp-pr</td>
<td>90.3</td>
<td>89.9</td>
<td>90.2</td>
</tr>
</tbody>
</table>
Summary

• AND/OR search spaces vs. Probabilistic circuits
• Review AND/OR search spaces for PGM
• AND/OR Multi-valued Decision Diagrams (AOMDD)
• Anytime algorithms over AND/OR search spaces
• AND/OR Abstraction sampling.
• Moving forward: Neuro-symbolic, causality
Thank You!

For publication see:
http://www.ics.uci.edu/~dechter/publications.html