ICAPS 2018 Tutorial

Decision Diagrams in Automated Planning and Scheduling, Part II

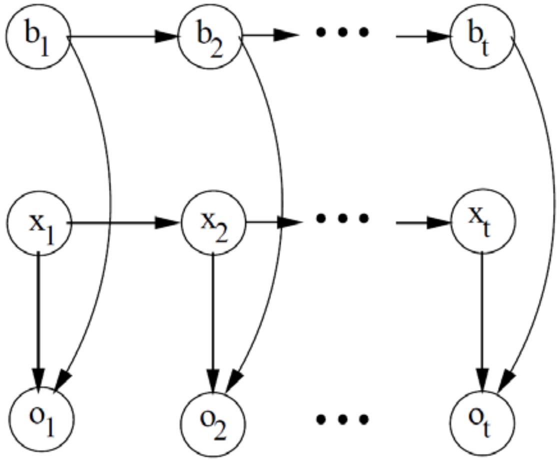
Scott Sanner

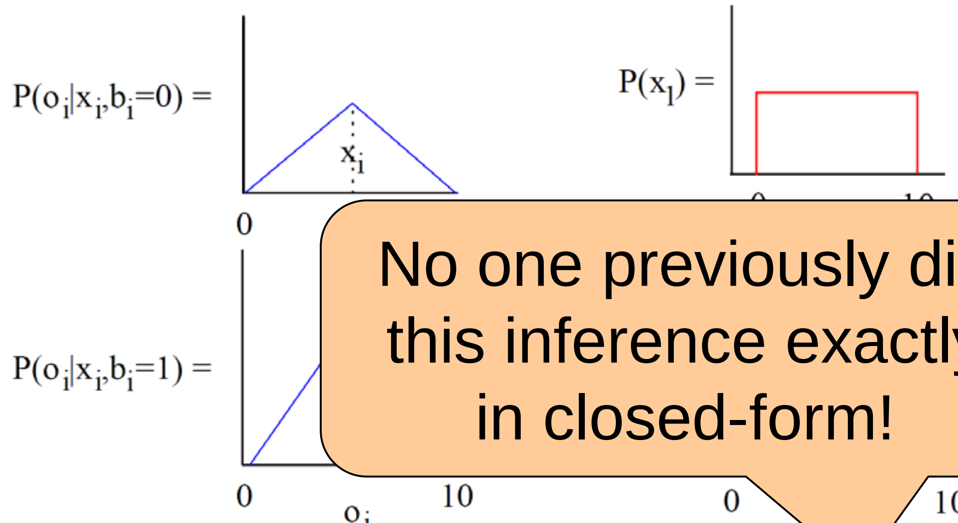
UNIVERSITY OF TORONTO

Part II:

Extensions to Continuous Inference

Inference for Continuous HMMs

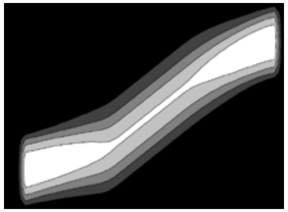




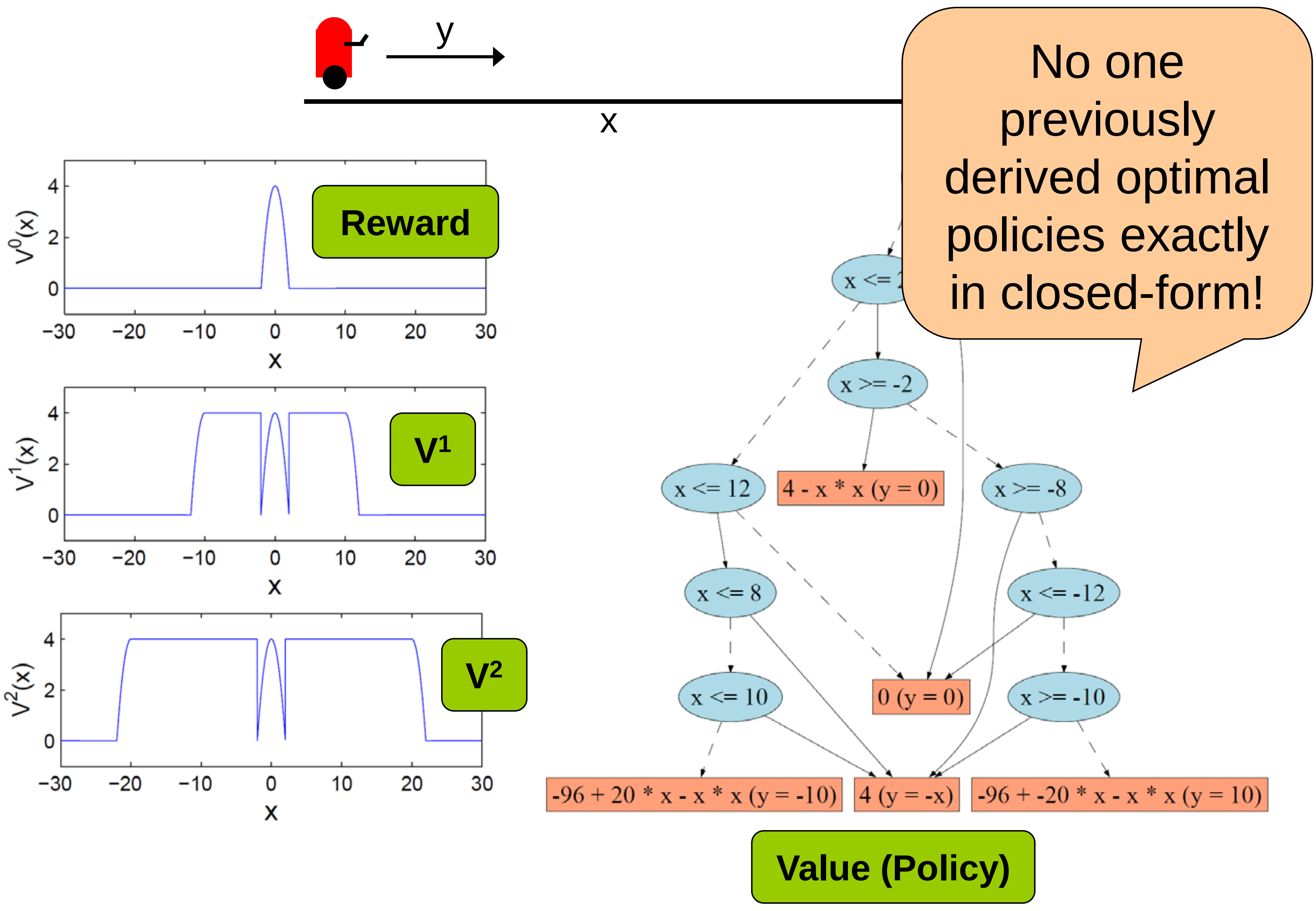
No one previously did this inference exactly in closed-form!

What's that function?

WTF?



Optimal Policies in Hybrid MDPs



How to obtain closed-form *exact* solutions?

Symbolic representations
and operations on
piecewise functions

Think Mathematica, not Matlab

Piecewise Functions (Cases)

$$z = f(x,y) = \begin{cases} (x > 3) \wedge (y \square x) : & x + y \\ (x \square 3) \vee (y > x) : & x^2 + xy^3 \end{cases}$$

Partition

Constraint

Value

Linear constraints and value

Linear constraints, constant value

Quadratic constraints and value

Formal Problem Statement

- General continuous graphical models represented by piecewise functions (cases)

$$f = \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases}$$

- Exact closed-form solution inferred via the following piecewise calculus:
  - $f_1 \oplus f_2$, $f_1 \otimes f_2$
  - $\max(f_1, f_2)$, $\min(f_1, f_2)$
  - $\int_x f(x)$
  - $\max_x f(x)$, $\min_x f(x)$

Question: how do we perform these operations in closed-form?

Polynomial Case Operations: ⊕, ⊗

$$\begin{cases} \phi_1 : & f_1 \\ \phi_2 : & f_2 \end{cases} \oplus \begin{cases} \psi_1 : & g_1 \\ \psi_2 : & g_2 \end{cases} = \quad ?$$

Polynomial Case Operations: ⊕, ⊗

$$\begin{cases} \phi_1 : & f_1 \\ \phi_2 : & f_2 \end{cases} \oplus \begin{cases} \psi_1 : & g_1 \\ \psi_2 : & g_2 \end{cases} = \begin{cases} \phi_1 \wedge \psi_1 : & f_1 + g_1 \\ \phi_1 \wedge \psi_2 : & f_1 + g_2 \\ \phi_2 \wedge \psi_1 : & f_2 + g_1 \\ \phi_2 \wedge \psi_2 : & f_2 + g_2 \end{cases}$$

- Similarly for ⊗
  - Polynomials closed under +, *

- What about max?
  - Max of polynomials is not a polynomial ☹

Polynomial Case Operations: max

$$\max\left(\begin{cases}\phi_1: & f_1\\ \phi_2: & f_2\end{cases}, \begin{cases}\psi_1: & g_1\\ \psi_2: & g_2\end{cases}\right) =$$

?

Polynomial Case Operations: max

$$\max\left(\begin{cases}\phi_1 : & f_1 \\ \phi_2 : & f_2\end{cases}, \begin{cases}\psi_1 : & g_1 \\ \psi_2 : & g_2\end{cases}\right) = \begin{cases}\phi_1 \wedge \psi_1 \wedge f_1 > g_1 : & f_1 \\ \phi_1 \wedge \psi_1 \wedge f_1 \square\, g_1 : & g_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 > g_2 : & f_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 \square\, g_2 : & g_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 > g_1 : & f_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 \square\, g_1 : & g_1 \\ \phi_2 \wedge \psi_2 \wedge f_2 > g_2 : & f_2 \\ \phi_2 \wedge \psi_2 \wedge f_2 \square\, g_2 : & g_2\end{cases}$$

- Still a piecewise polynomial!

Size blowup?
We'll get to that…

Definite Integration: $\int_{x=-\infty}^{\infty}$

- Closed for polynomials
  - But how to compute for case?

$$\int_{x=-\infty}^{\infty} \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases} dx$$

  - Just integrate case partitions, $\oplus$ results!

Partition Integral

1. Determine integration bounds

$$\int_{x=-\infty}^{\infty} [\phi_1] \cdot f_1 \, dx$$

$$\phi_1 := [x > -1] \wedge [x > y - 1] \wedge [x \square z] \wedge [x \square y + 1] \wedge [y > 0]$$

$$f_1 := x^2 - xy$$

UB and LB are symbolic!

What constraints here?

- independent of x
- pairwise UB > LB

How to evaluate?

Definite Integral Evaluation

- How to evaluate integral bounds?

$$\int_{x=LB}^{UB} x^2 - xy \;=\; \frac{1}{3}x^3 - \frac{1}{2}x^2 y \Big|_{LB}^{UB}$$

- Can do polynomial operations on cases!

Symbolically, exactly evaluated!

Exact Graphical Model Inference!

(directed and undirected)

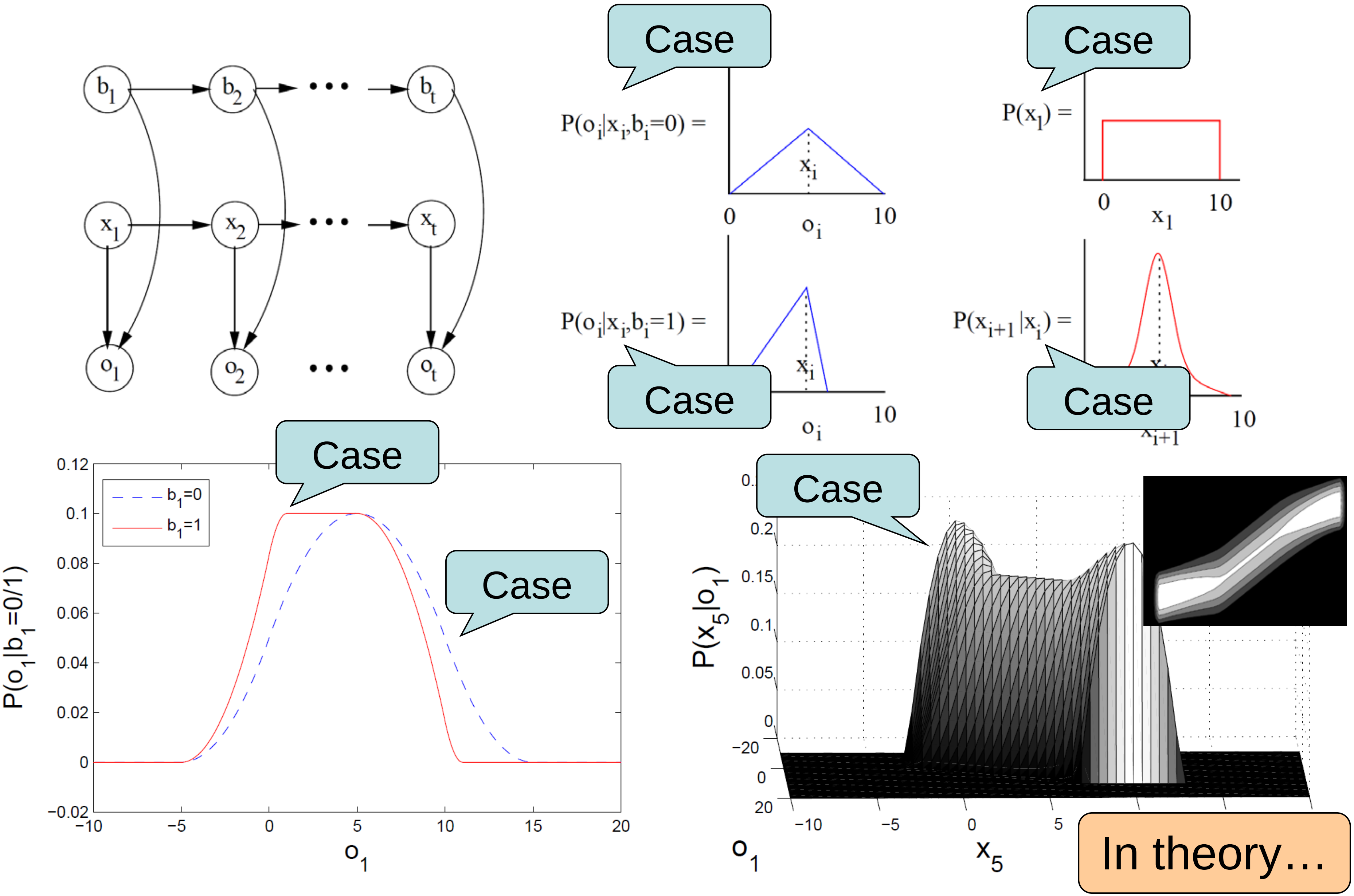
- Can do general probabilistic inference

$$p(x_2|x_1) = \frac{\int_{x_3} \cdots \int_{x_n} \bigotimes_{i=1}^{k} case_i \; dx_n \cdots dx_3}{\int_{x_2} \cdots \int_{x_n} \bigotimes_{i=1}^{k} case_i \; dx_n \cdots dx_2}$$

- Or an exact expectation of *any* polynomial

  – *poly* = Mean, variance, skew, curtosis, …, x^2+y^2+xy

Voila: Closed-form Exact Inference via SVE!



Computational Complexity?

- In theory for SVE on graphical models
  - Worst-case complexity is O(exp(#operations))
    - Not explicitly tree-width dependent!
    - **But worse:** integral may invoke 100's of operations!

Fortunately decision diagrams can mitigate worst-case complexity

BDD / ADDs

Quick Introduction

Function Representation (Tables)

- How to represent functions: $B^n \rightarrow R$?
- How about a fully enumerated table…
- …OK, but can we be more compact?

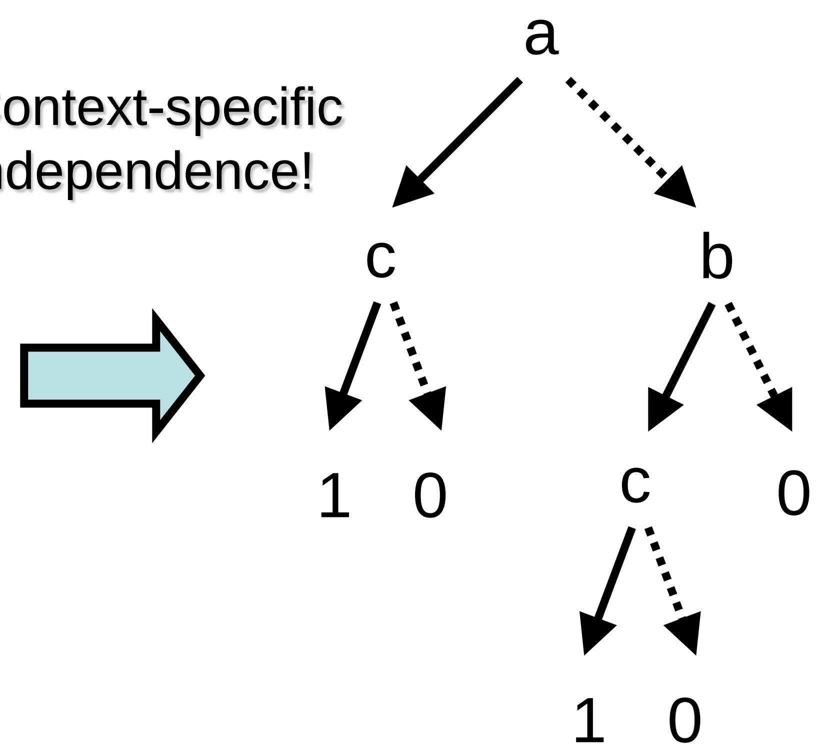
a	b	c	F(a,b,c)
0	0	0	0.00
0	0	1	0.00
0	1	0	0.00
0	1	1	1.00
1	0	0	0.00
1	0	1	1.00
1	1	0	0.00
1	1	1	1.00

Function Representation (Trees)

- How about a tree? Sure, can simplify.

a	b	c	F(a,b,c)
0	0	0	0.00
0	0	1	0.00
0	1	0	0.00
0	1	1	1.00
1	0	0	0.00
1	0	1	1.00
1	1	0	0.00
1	1	1	1.00

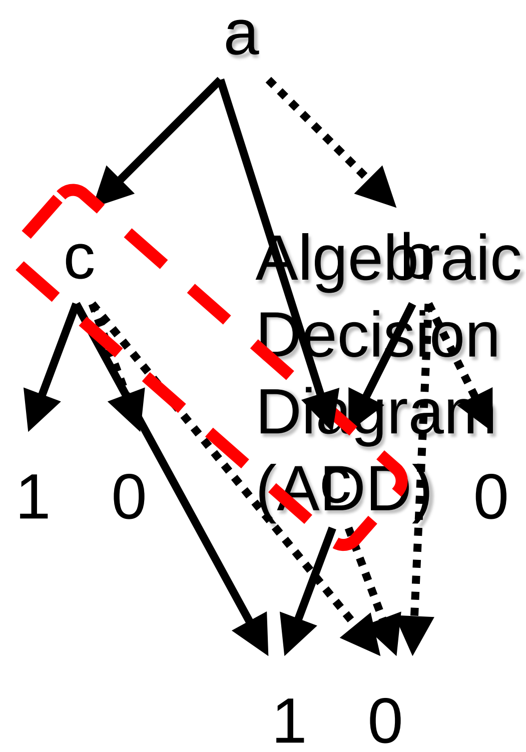
Context-specific independence!



Function Representation (ADDs)

- Why not a directed acyclic graph (DAG)?

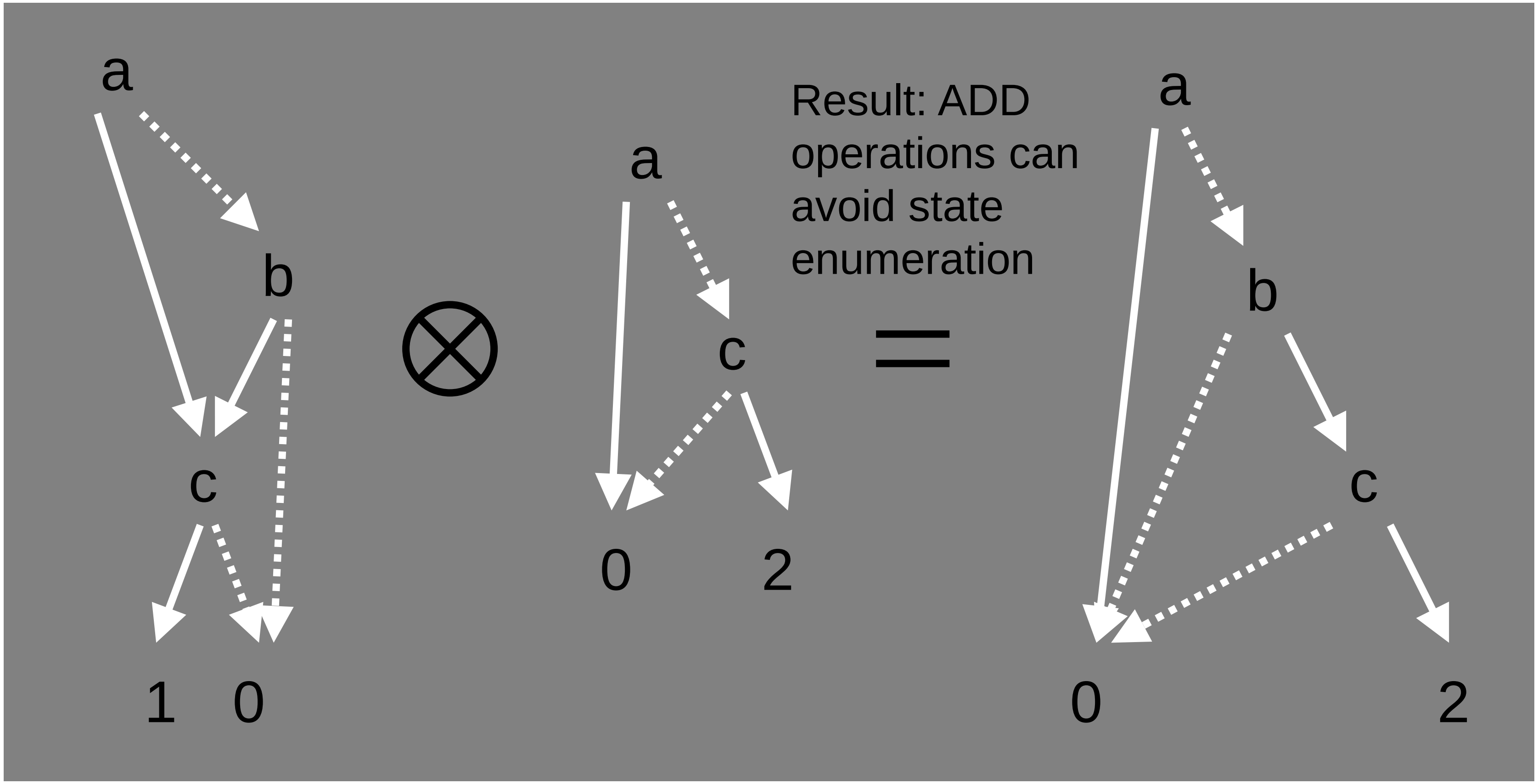
a	b	c	F(a,b,c)
0	0	0	0.00
0	0	1	0.00
0	1	0	0.00
0	1	1	1.00
1	0	0	0.00
1	0	1	1.00
1	1	0	0.00
1	1	1	1.00



Exploits context-specific independence (CSI) and shared substructure.

Binary Operations (ADDs)

- Why do we order variable tests?
- Enables us to do efficient binary operations…

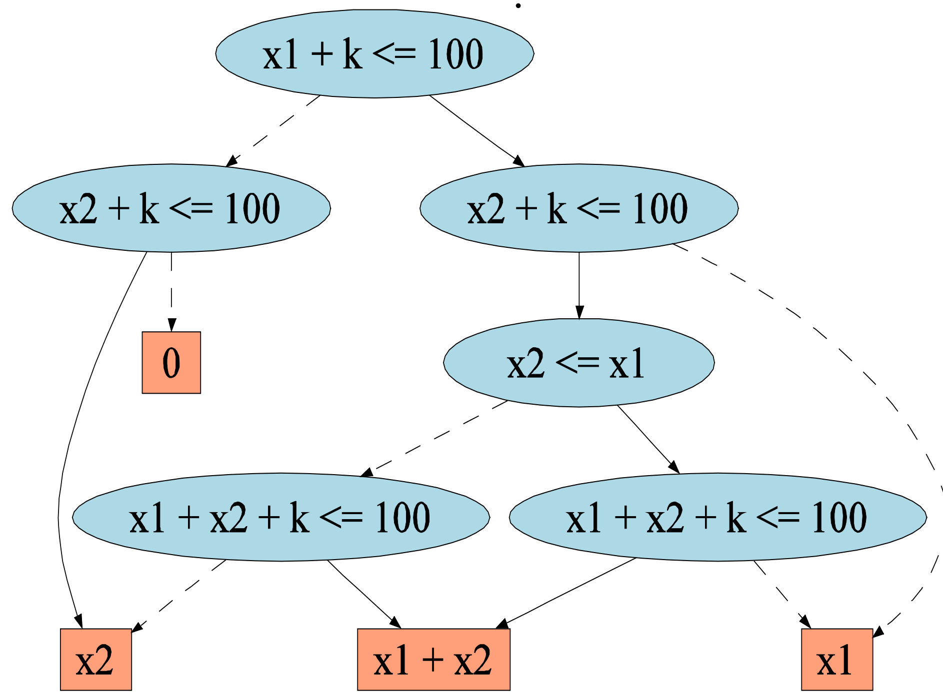


Case $\rightarrow$ XADD

XADD = continuous variable e**x**tension
of **a**lgebraic **d**ecision **d**iagram

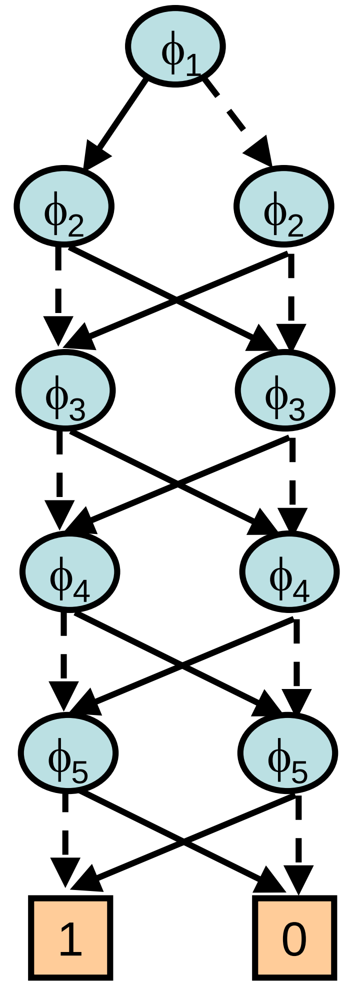
Case → XADD

$$V = \begin{cases} x_1 + k > 100 \wedge x_2 + k > 100 : & 0 \\ x_1 + k > 100 \wedge x_2 + k \leq 100 : & x_2 \\ x_1 + k \leq 100 \wedge x_2 + k > 100 : & x_1 \\ x_1 + x_2 + k > 100 \wedge x_1 + k \leq 100 \wedge x_2 + k \leq 100 \wedge x_2 > x_1 : & x_2 \\ x_1 + x_2 + k > 100 \wedge x_1 + k \leq 100 \wedge x_2 + k \leq 100 \wedge x_2 \leq x_1 : & x_1 \\ x_1 + x_2 + k \leq 100 : & x_1 + x_2 \\ \vdots & \vdots \end{cases}$$

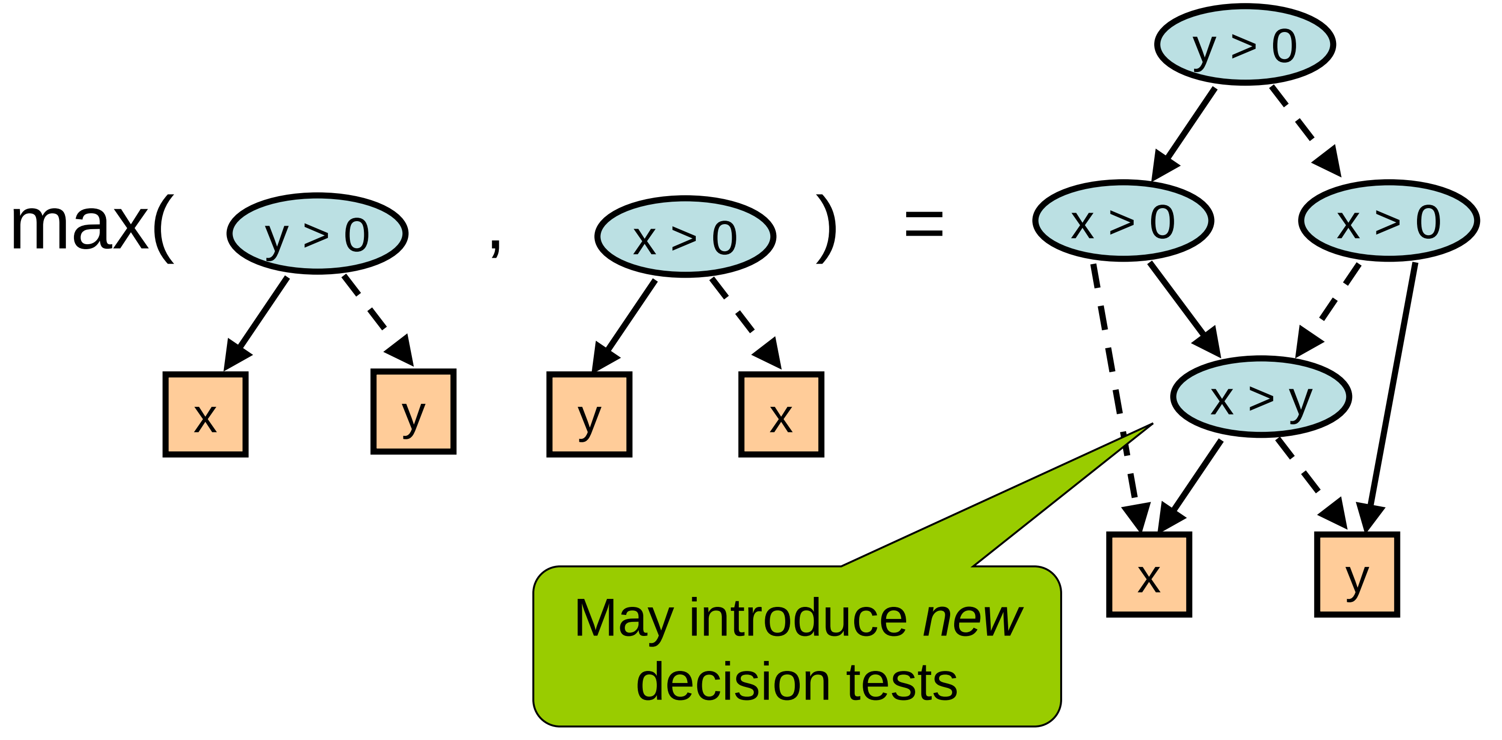


Compactness of (X)ADDs

- XDD is linear in # of decisions ϕ_i
- Case version has exponential number of partitions!



XADD Maximization



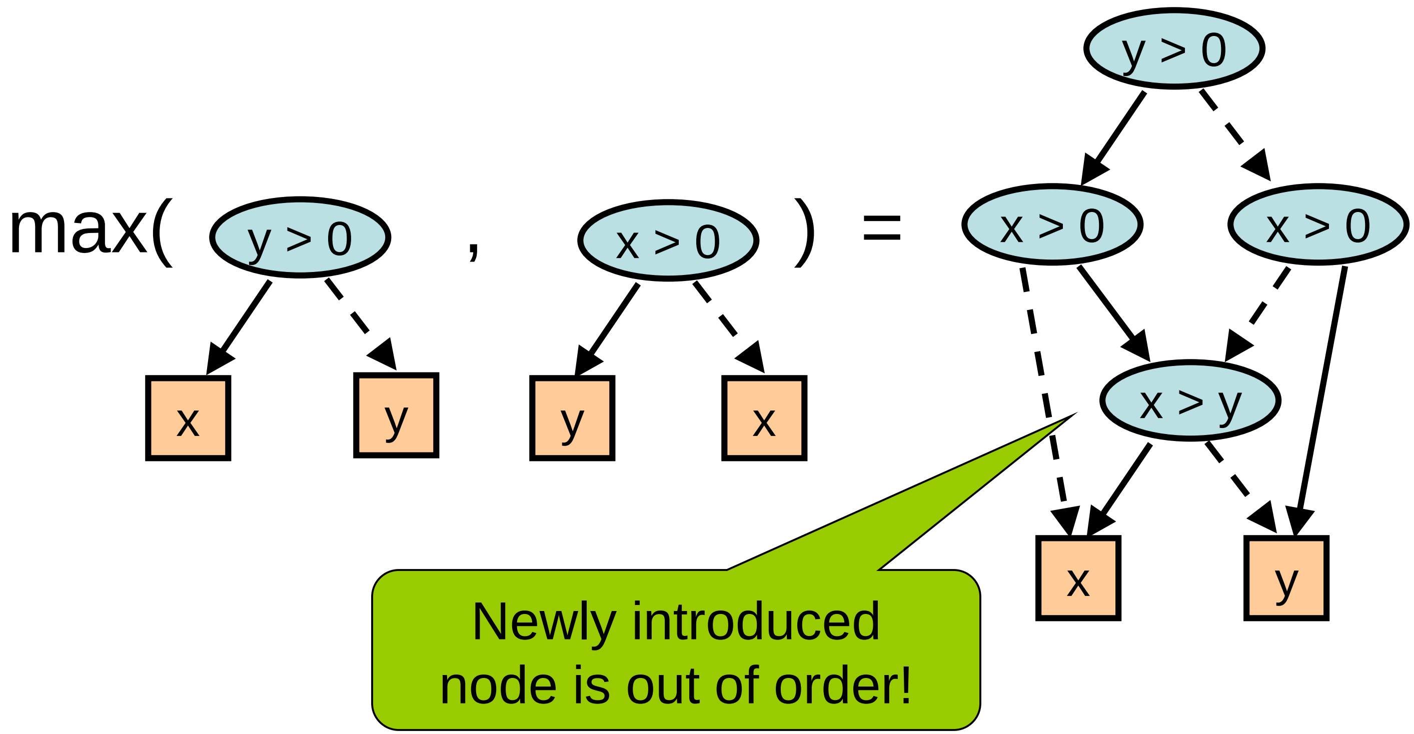
Operations exploit structure: O(|f||g|)

Maintaining XADD Orderings

- Max may get decisions out of order

Decision ordering (root→leaf)

- x > y
- y > 0
- x > 0

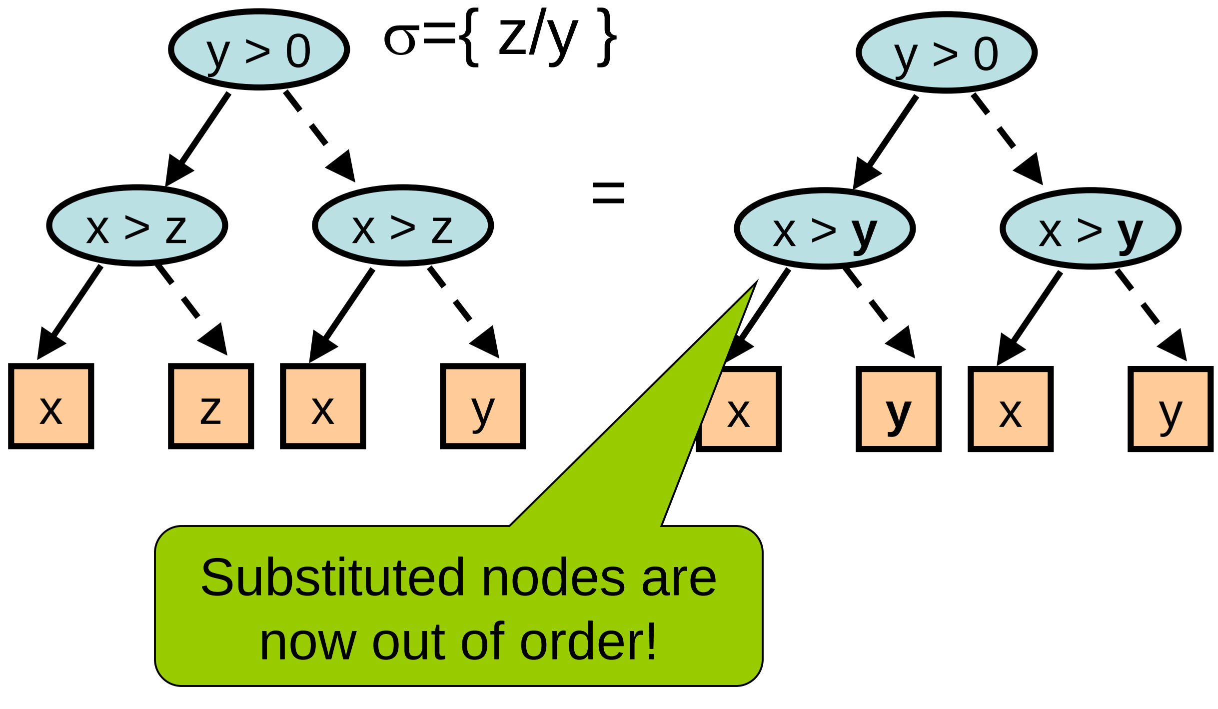


Maintaining XADD Orderings

- Substitution may get decisions out of order

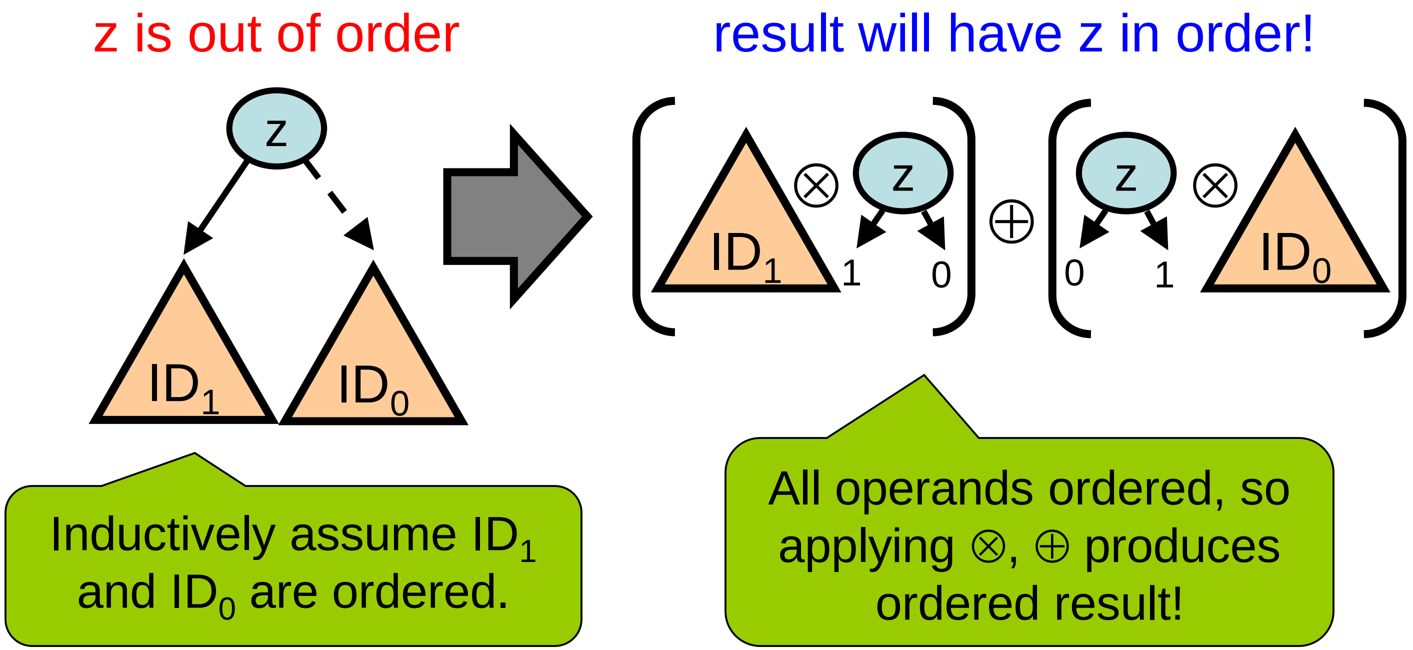
Decision ordering (root→leaf):

- x > y
- y > 0
- x > z

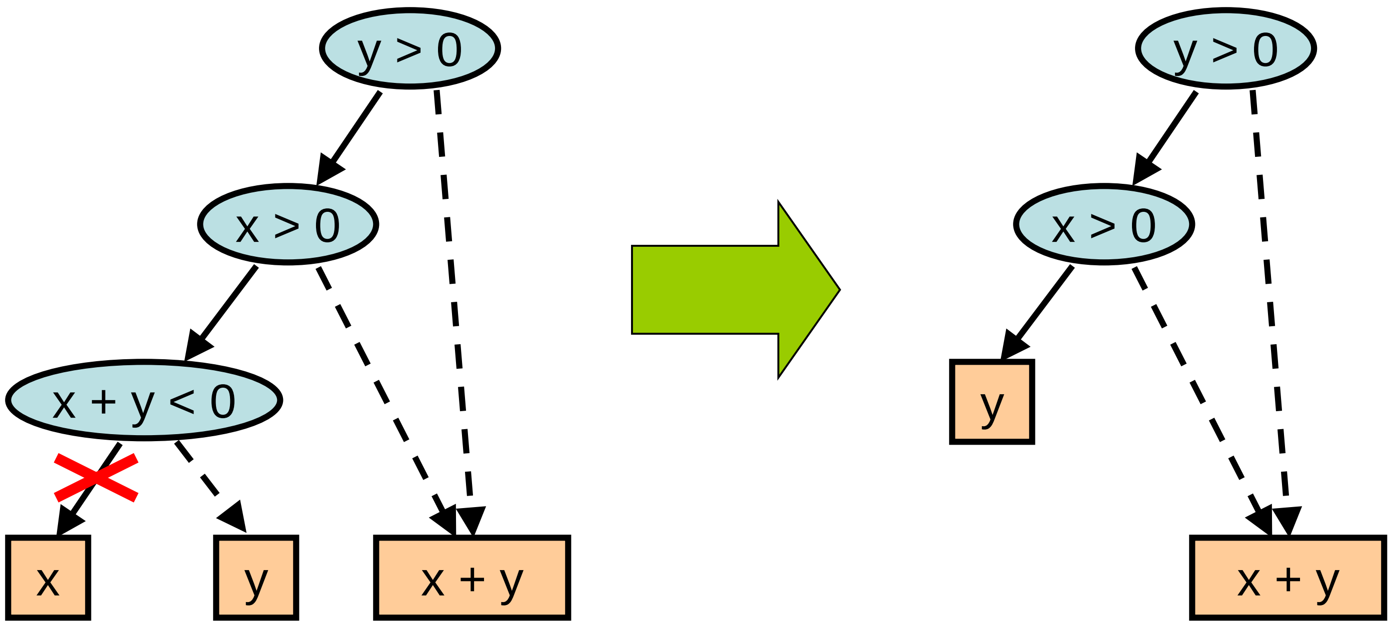


Correcting XADD Ordering

- Obtain *ordered* XADD from *unordered* XADD
  - key idea: binary operations maintain orderings



Maintaining Minimality



Node unreachable – x + y < 0 always false if x > 0 & y > 0

If **linear**, can detect with feasibility checker of LP solver & prune

More subtle prunings as well.

XADD enables all previous inference!

Solution is inherently piecewise, need a data structure to maintain compact form

Beyond Inference

Optimal Sequential Decision-making

Integration with a δ: substitution

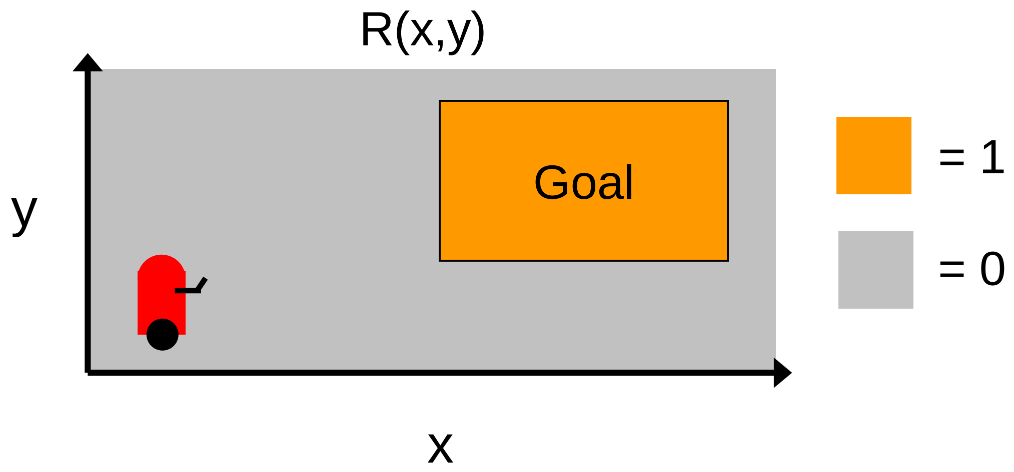
- Special case for integrals with δ-functions
  - $\int_x \delta[x - y] f(x) dx = f(y)$ triggers symbolic *substitution*
  - More generally: $\int_{x'_j} \delta[x'_j - g(\vec{x})] V' dx'_j \;=\; V'\{x'_j / g(\vec{x})\}$
  - E.g.,

$$\int_{x'_1} \delta[x'_1 - (x_1^2 + 1)] \left(\begin{cases} \underline{x'_1 < 2:} & \underline{x'_1} \\ \underline{x'_1 \geq 2:} & \underline{x'^2_1} \end{cases} \right) dx'_1 \;=\; \begin{cases} \underline{x_1^2 + 1 < 2:} & \underline{x_1^2 + 1} \\ \underline{x_1^2 + 1 \geq 2:} & \underline{(x_1^2 + 1)^2} \end{cases}$$

  - If *g* is *case*: need *conditional substitution*
    - see Sanner, Delgado, Barros (UAI 2011)

Continuous State MDPs

- 2-D Navigation
- State: (x,y) $\in \mathbb{R}^2$
- Actions:
  - move-x-2
    - x' = x + 2
    - y' = y
  - move-y-2
    - x' = x
    - y' = y + 2
- Reward:
  - R(x,y) = I[(x > 5) ^ (x < 10) ^ (y > 2) ^ (y < 5)]

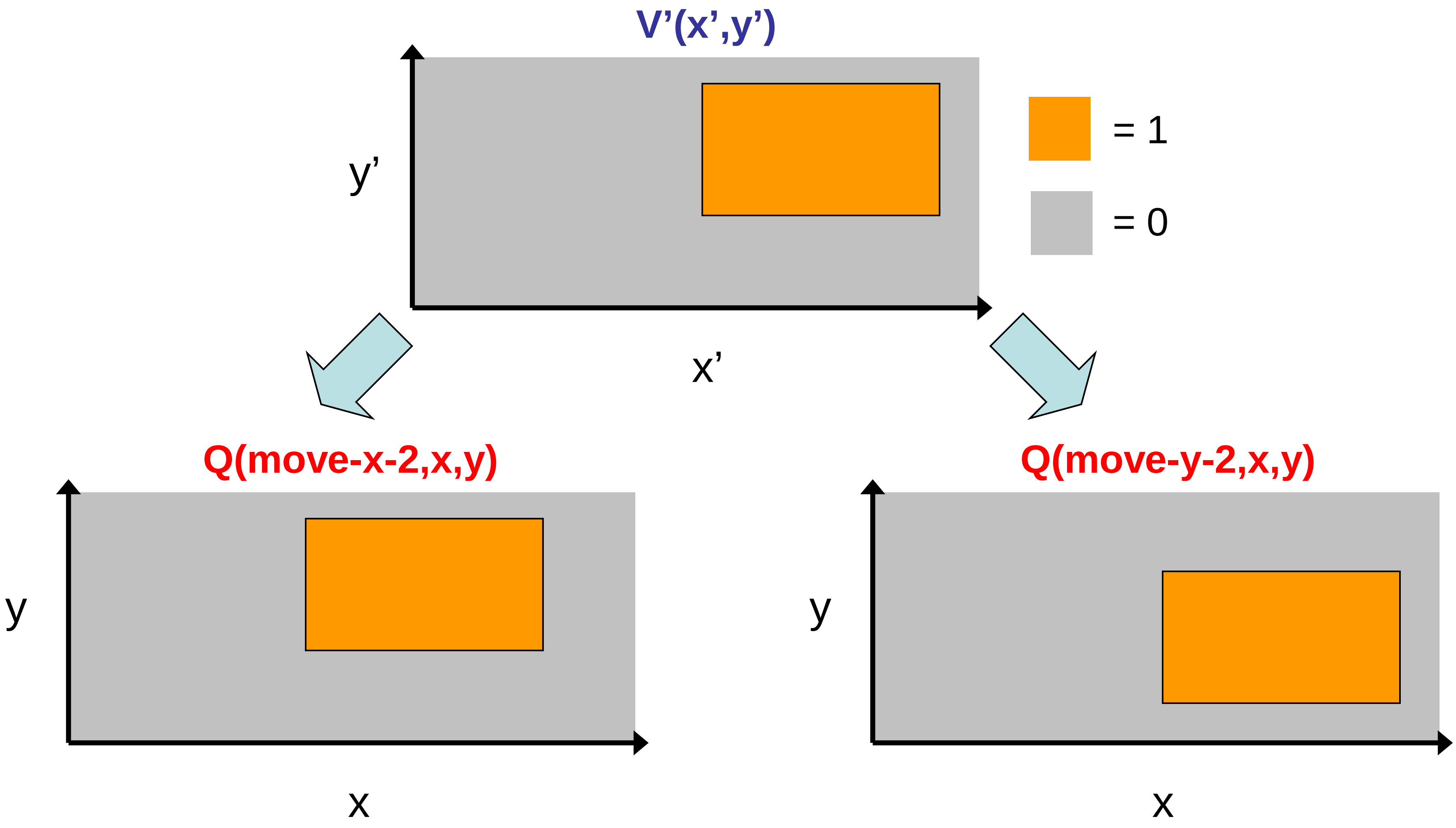


Feng *et al* (UAI-04) Assumptions:

1. Continuous transitions are deterministic and linear functions of **single variable**
2. Discrete transitions can be stochastic
3. Reward is piecewise rectilinear convex

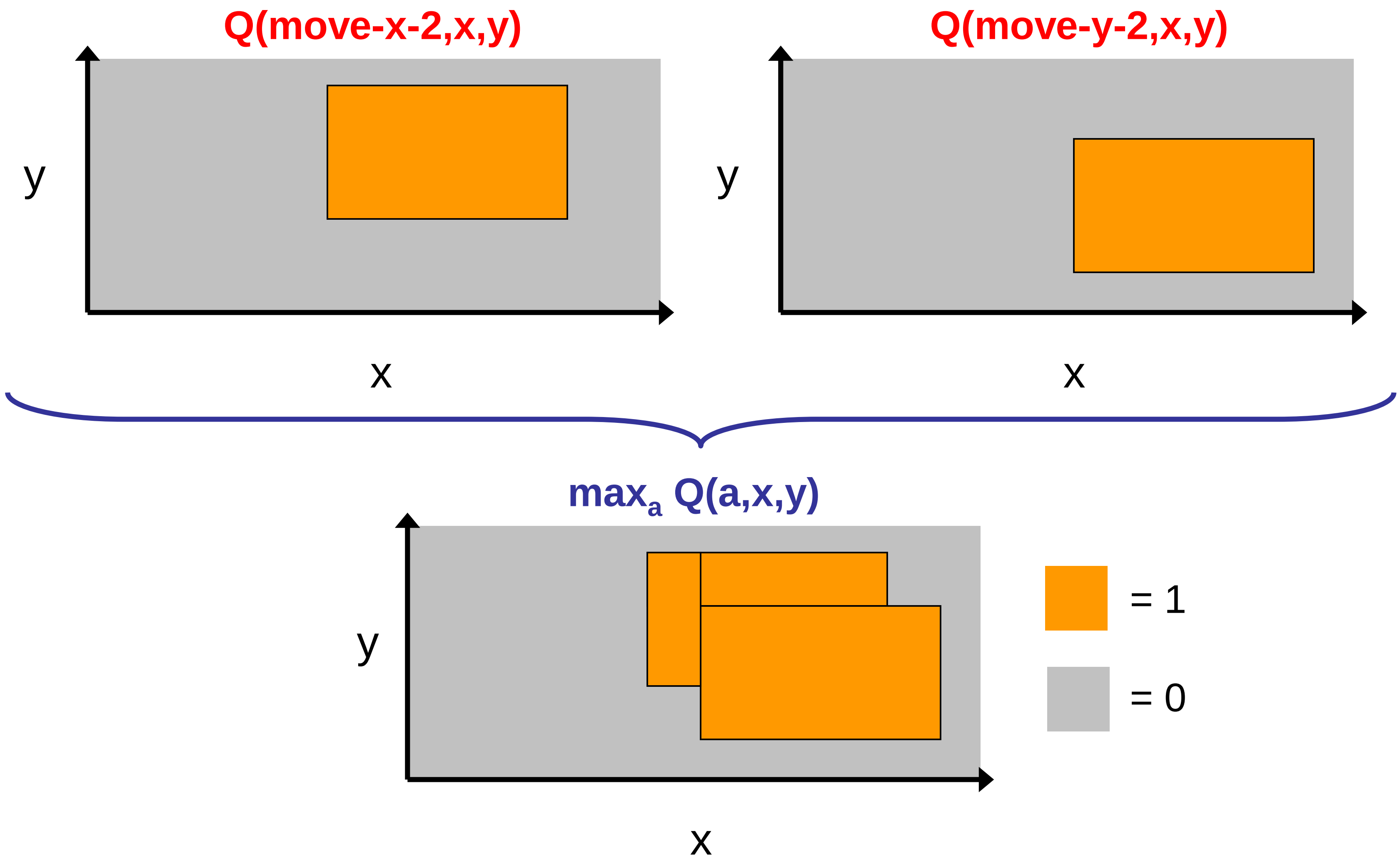
Exact Solutions to DC-MDPs: Regression

- Continuous regression is just translation of “pieces”



Exact Solutions to DC-MDPs: Maximization

- Q-value maximization yields piecewise rectilinear solution



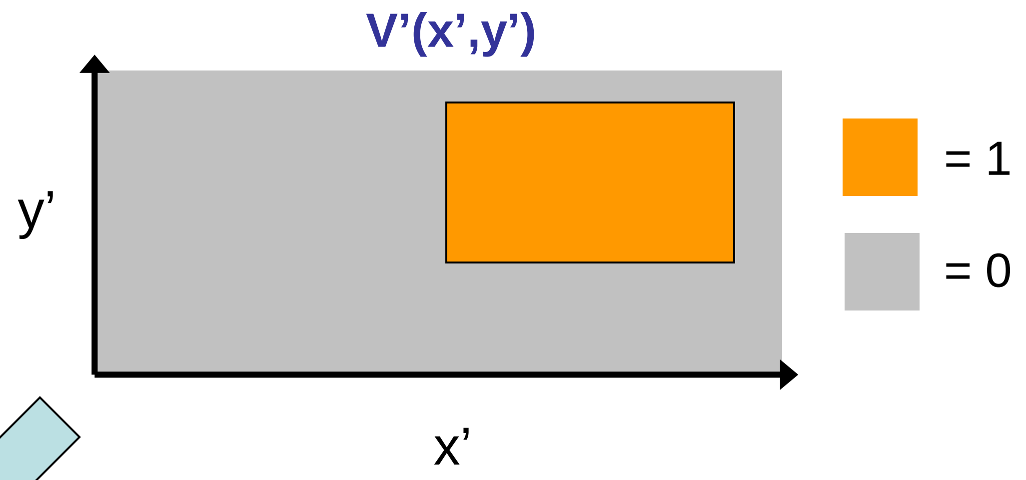
Previous Work Limitations I

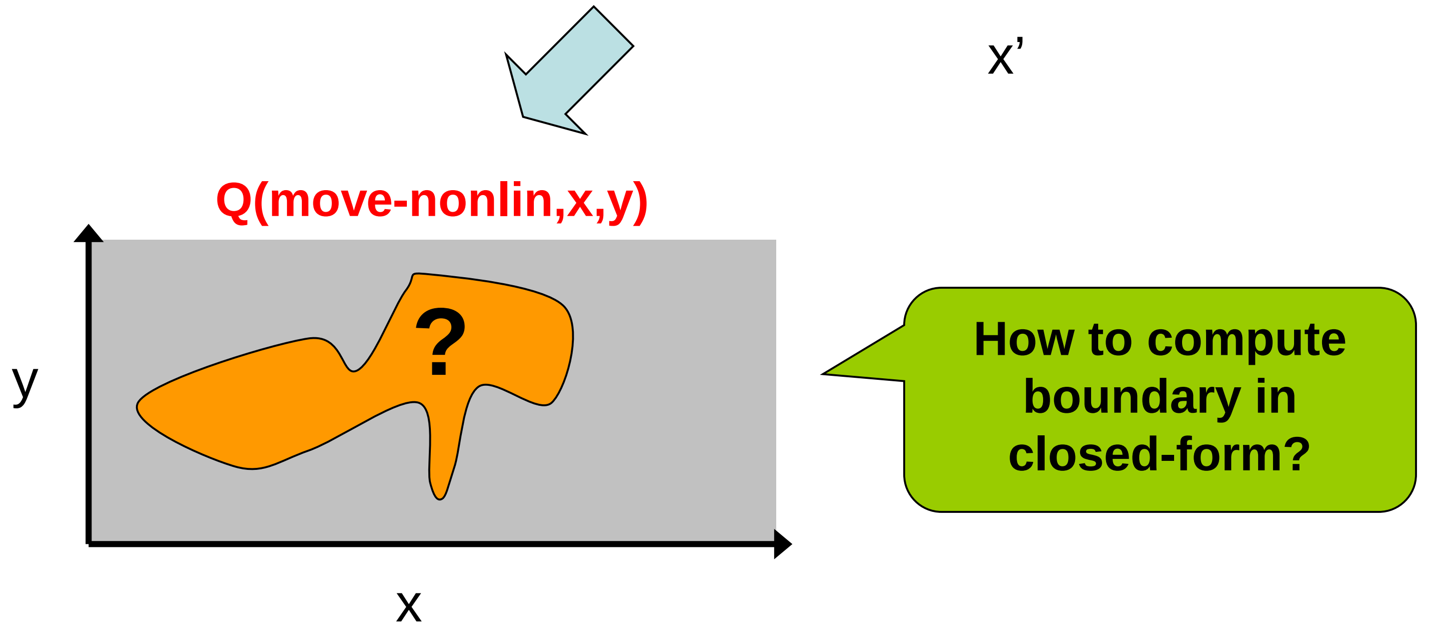
- Exact regression when transitions multivariate nonlinear?

Action move-nonlin:

– $x' = x^3y + y^2$

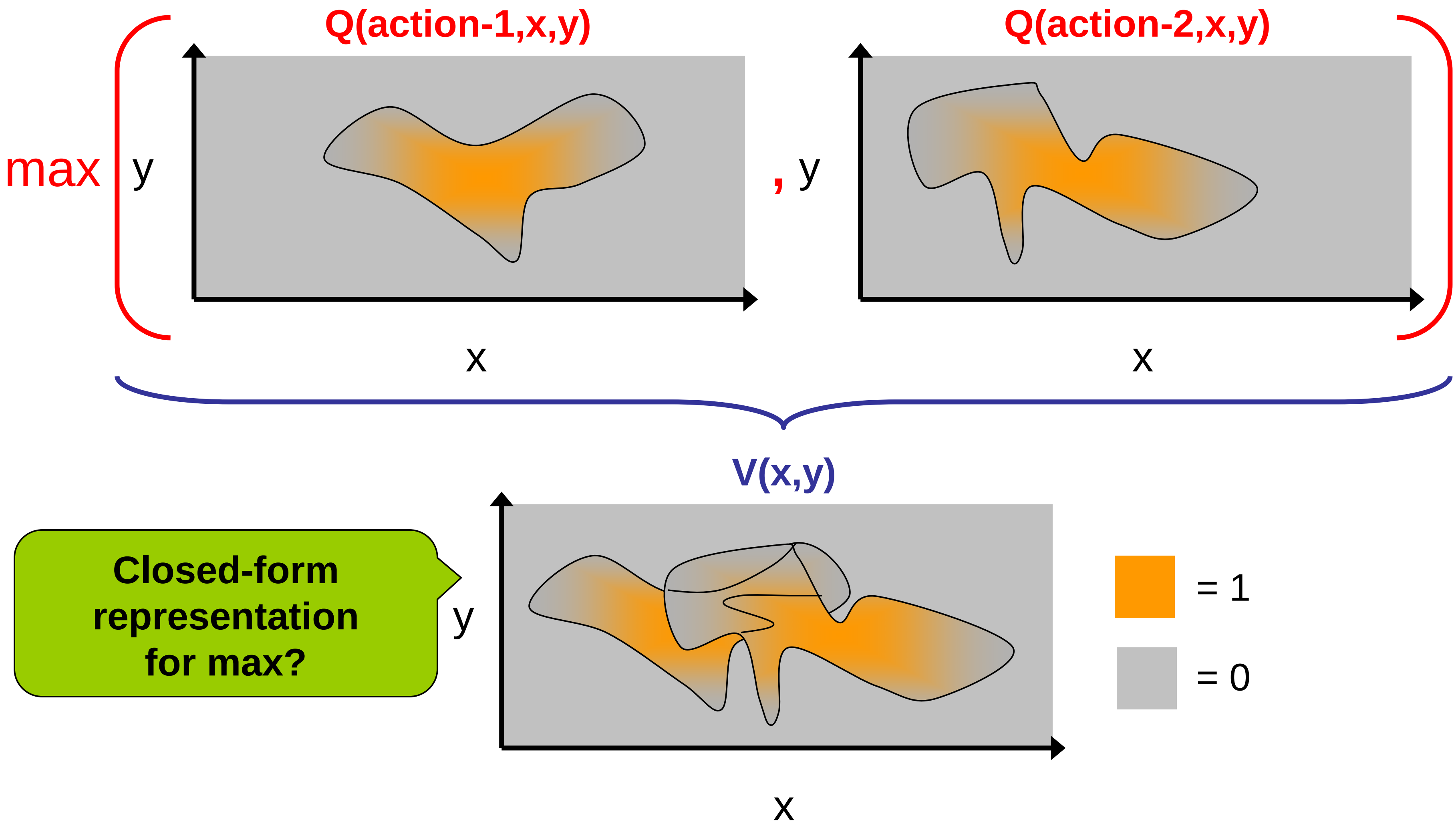
– $y' = y * \log(x^2y)$





Previous Work Limitations II

- max(.,.) when reward/value arbitrary piecewise?



A solution to previous limitations:

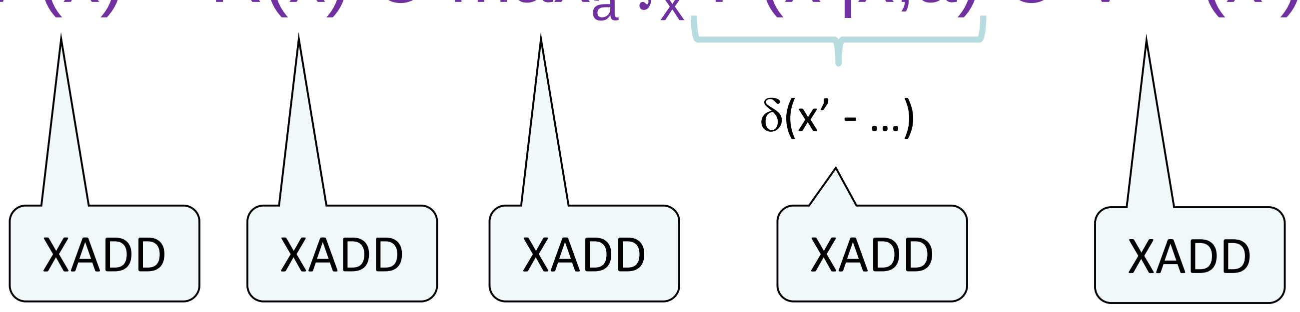
Symbolic Dynamic Programming (SDP)

n.b., motivated by SDP from Boutilier *et al* (IJCAI-01) but here continuous instead of relational

Using SDP, we can compute the MDP solution symbolically!

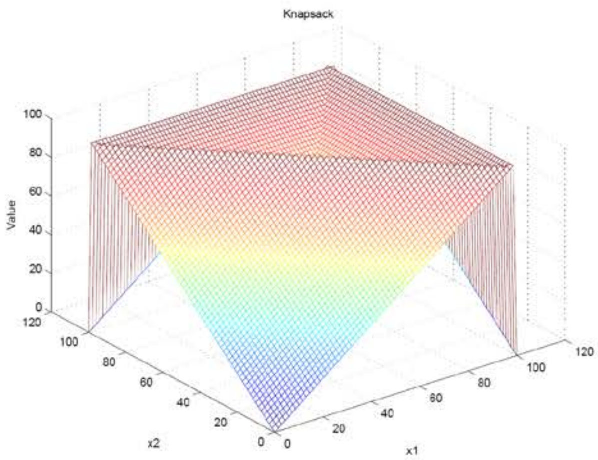
Compute the following in symbolic closed-form:

$$V^t(x) = R(x) \oplus \max_a \int_{x'} P(x'|x,a) \otimes V^{t+1}(x')$$

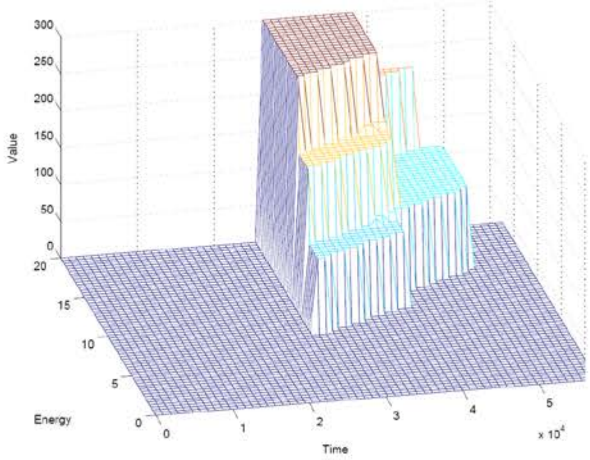


That's SDP! 3D Value Function Gallery

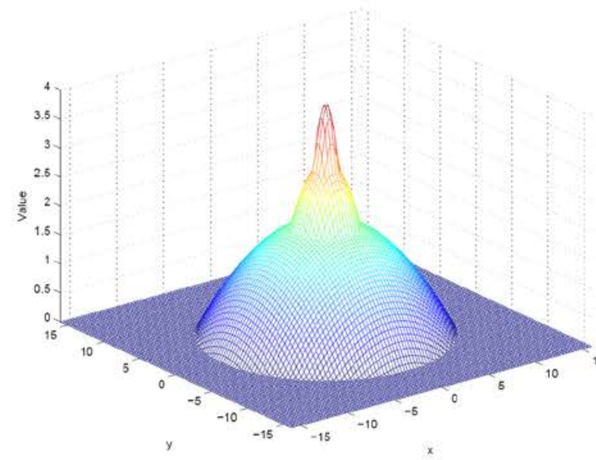
Knapsack



Mars Rover Linear



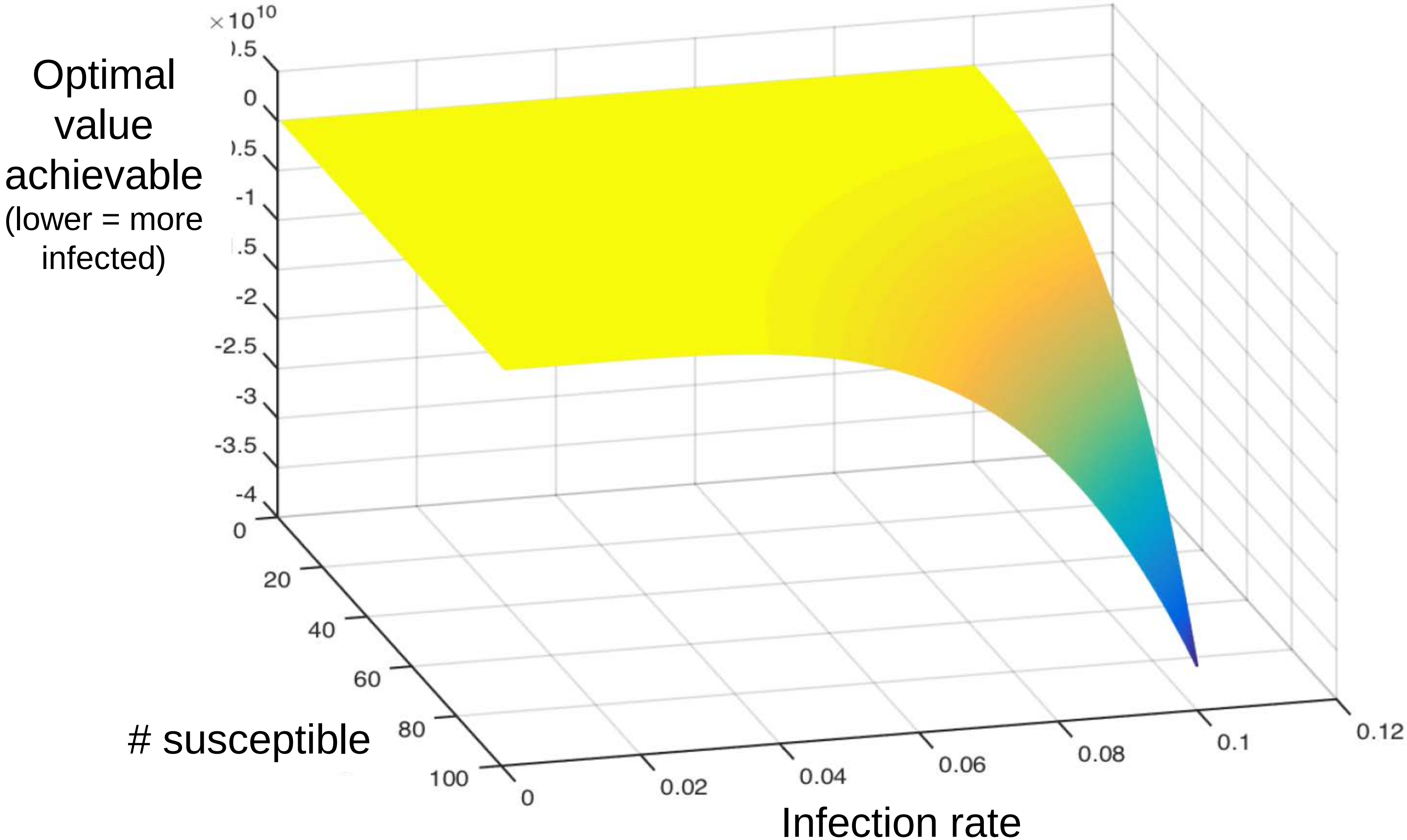
Mars Rover Nonlinear



Exact value functions for **discrete action** hybrid MDPs:

- Arbitrary reward, transitions – cos(xy)
- (non)linear piece boundaries and function surfaces!

Example: Optimal Controllability of Epidemics in Nonlinear SIR Models



Continuous Actions?

If we can solve this, can solve **multivariate inventory control** – closed-form policy unknown for 50+ years!

Continuous Actions

- Inventory control
  - Reorder based on stock, future demand
  - Action: $a(\vec{\Delta}); \vec{\Delta} \in \mathbb{R}^{|a|}$

- Need $\max_{\Delta}$ in Bellman backup

$$V_{h+1} = \max_{a \in A} \max_{\vec{\Delta}} Q_a^{h+1}(\vec{b}, \vec{x}, \vec{\Delta})$$

- How to do $\max_{\Delta}$?
  - And track maximizing Δ substitutions to recover π?

Symbolic MILP Solutions

- How to max out variable x from a case?

$$\max_{x \in (-\infty,\infty)} \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases} = $$

Actually defining a MILP. Maxing out all vars leads to solution!

$$= \max_{i=1..k} \boxed{\max_{x \in (-\infty,\infty)} \begin{cases} \phi_i : & f_i \\ \neg\phi_i : & -\infty \end{cases}}$$

– Just max x in case partitions, then max results!

Partition Max

1. Determine integration bounds

$$\max_{x \in (-\infty, \infty)} \{\phi_1 \cdot : f_1$$

$$\phi_1 := [x > -1] \wedge [x > y - 1] \wedge [x \square z] \wedge [x \square y + 1] \wedge [y > 0]$$

$$f_1 := x - 2y$$

UB and LB are symbolic!

What constraints here?

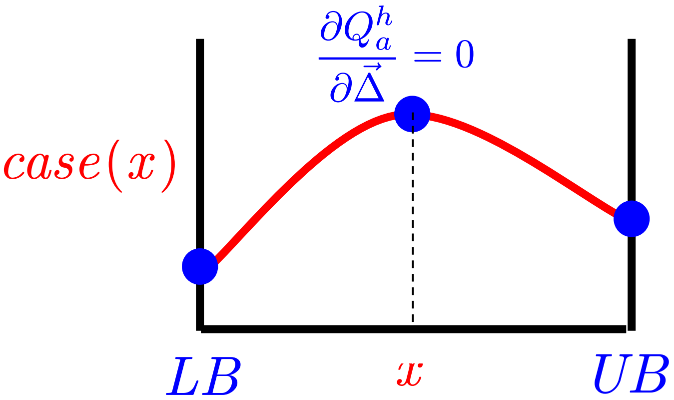
- independent of x
- pairwise UB > LB

$$\{\phi_{cons} : \quad \max_{x \in [LB, UB]} \; f_1$$

How to evaluate?

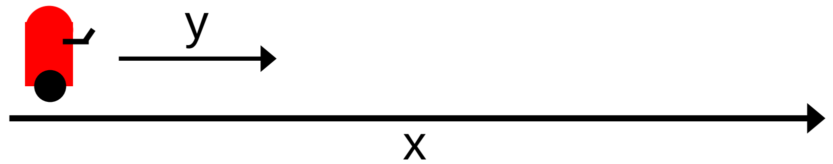
Max-out Case Operation

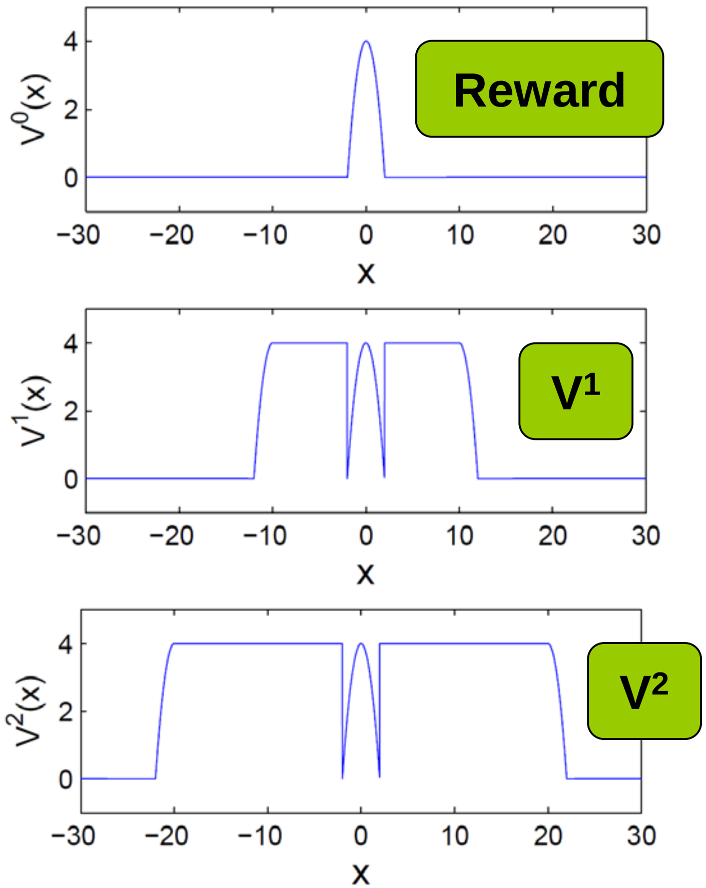
- $\max_x$ case(x) reduced to a "casemax" of single partition $\max_x$'s:
  - In a *single* case partition …*max* w.r.t. critical points
    - LB, UB
    - Derivative is zero (Der0)
    - max(case(x/LB),
      case(x/UB),
      case(x/Der0))
  - Can even track substitutions through max
    - Recovers function of maximizing assignments!

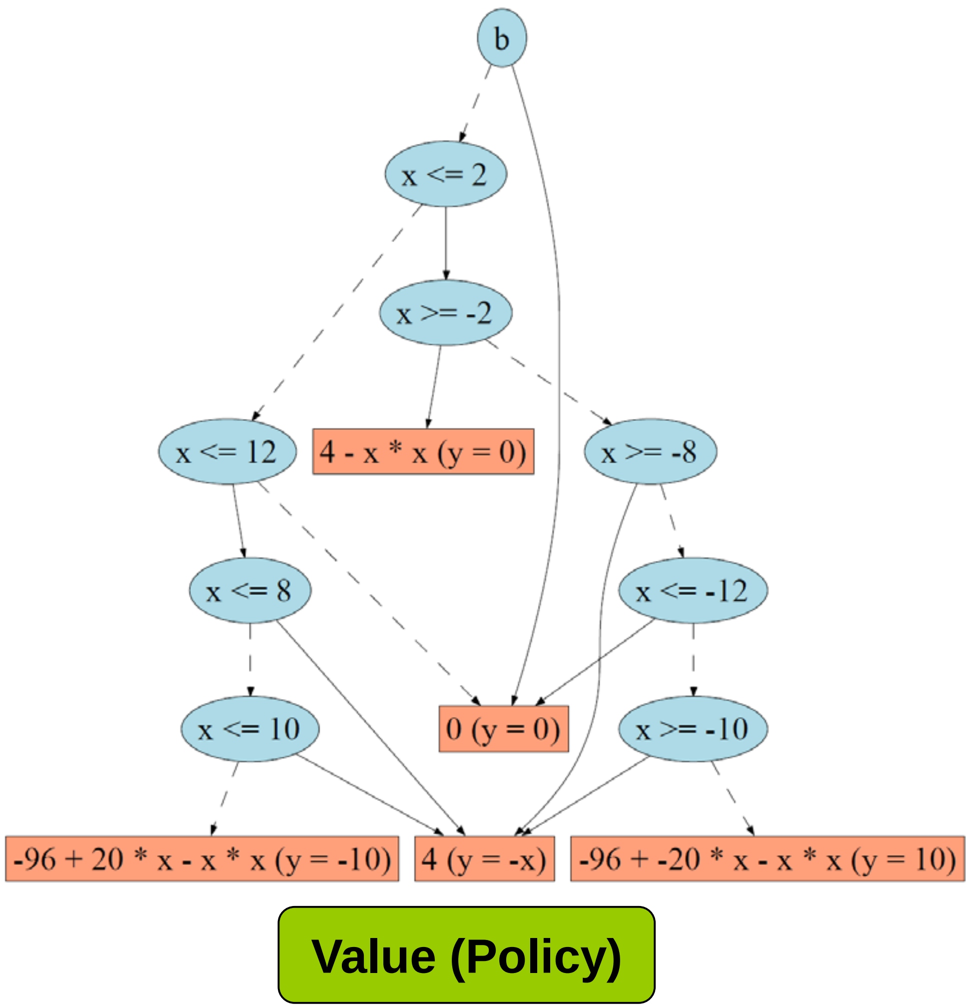


See AAAI 2012 for more details

Illustrative Value and Policy



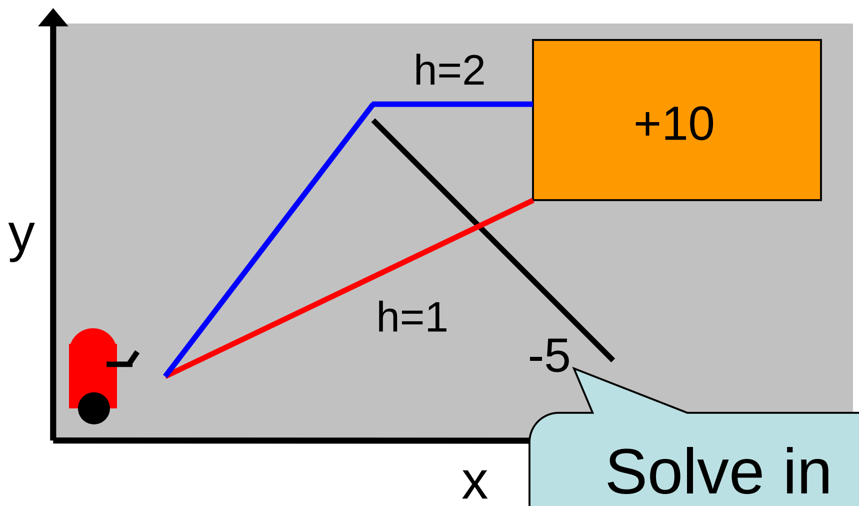




Continuous Actions, Nonlinear

- **Robotics**
  - Continuous position, joint angles
  - Represent exactly with polynomials
    - Radius constraints

- **Obstacle Navigation**
  - 2D, 3D, 4D (time)
  - Don’t discretize!
    - ~~Grid worlds~~
  - But nonlinear ☹



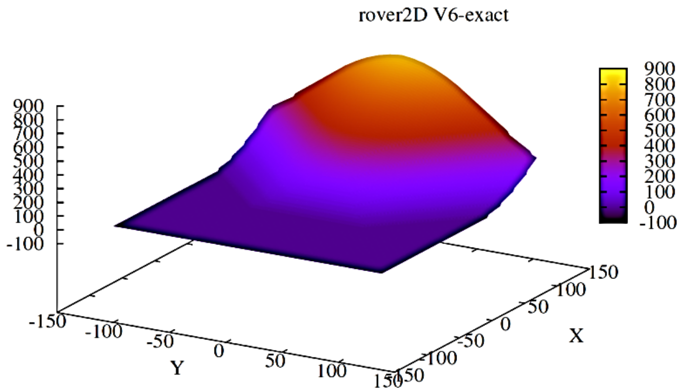
Solve in 2 steps!

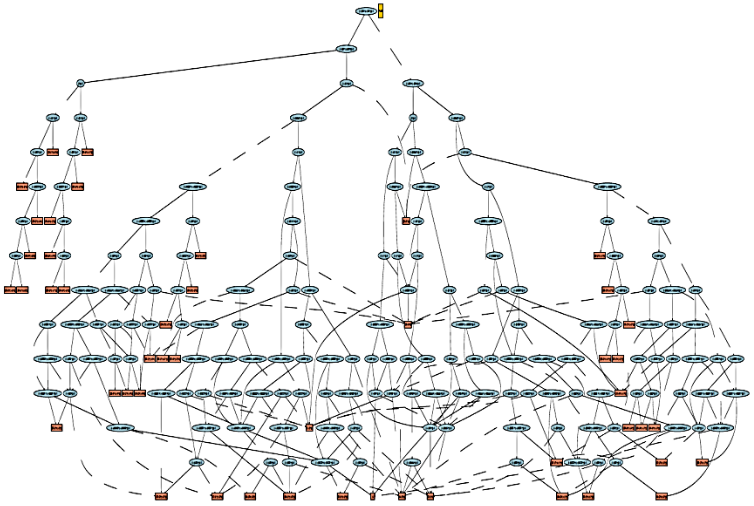
Multilinear, quadratic extensions.

In general: algebraic geometry.

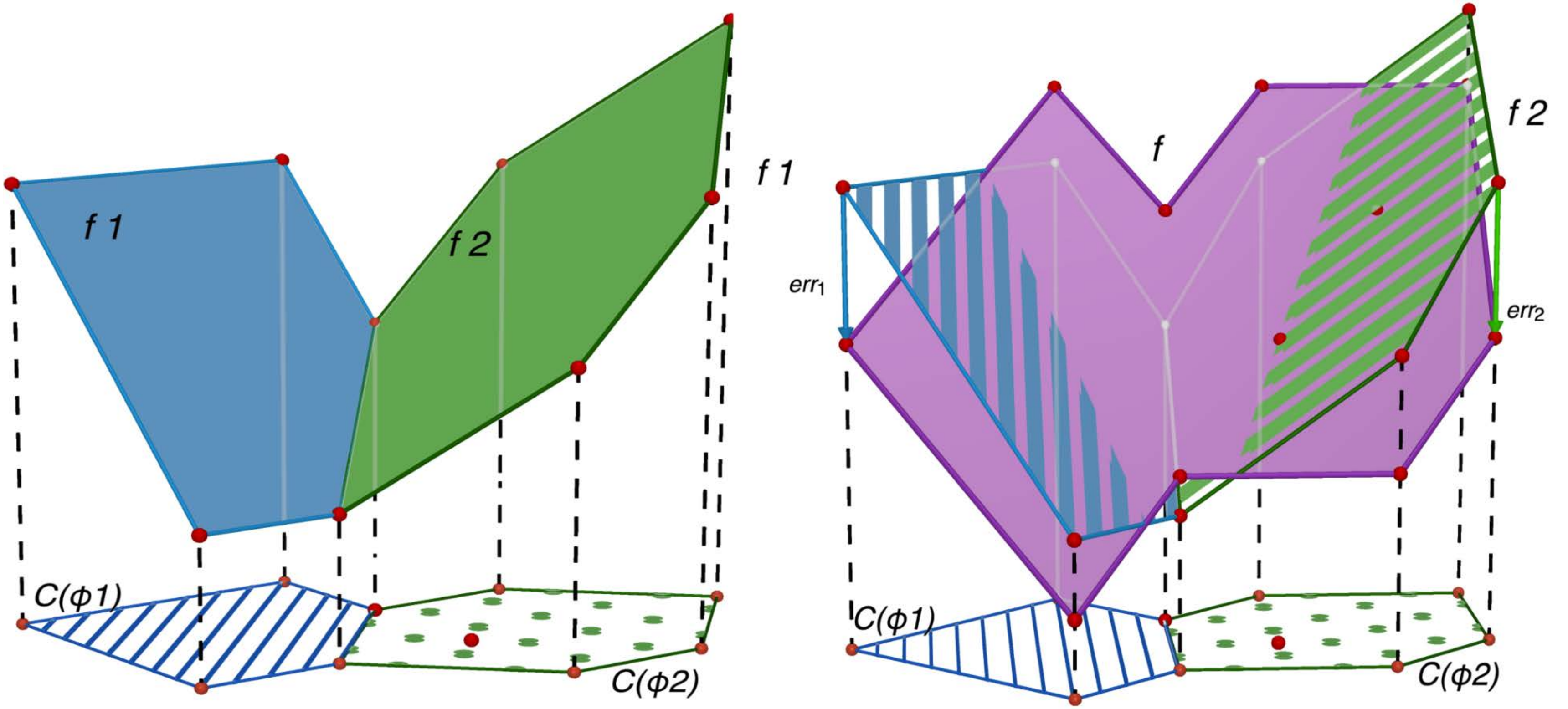
Need bounded / targeted approximations to scale…

Bounded Error Compression





Linear XADD Leaf Merging



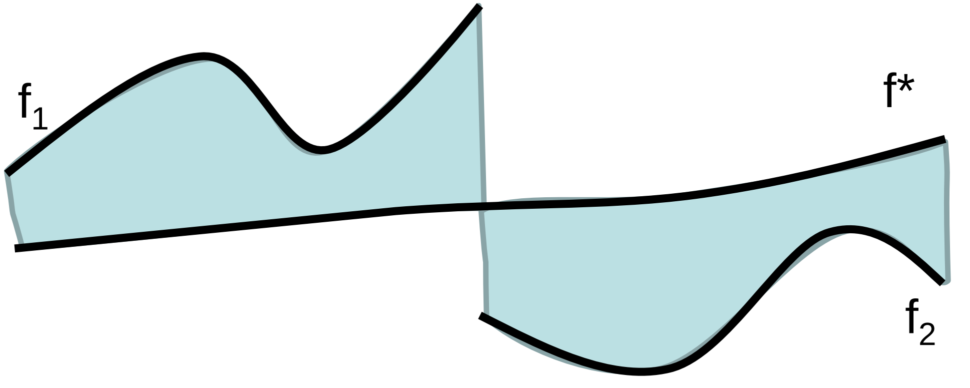
Best f*

Max error

$$\min_{\vec{c}^*} \max_{i\in\{1,2\}} \max_{\vec{x}\in S_{\phi_i}} \left| \underbrace{\vec{c}_i^{\,T} \begin{bmatrix}\vec{x}\\1\end{bmatrix}}_{f_i} - \underbrace{\vec{c}^{*\,T}\begin{bmatrix}\vec{x}\\1\end{bmatrix}}_{f^*} \right|$$

Nonlinear XADD Approximation?

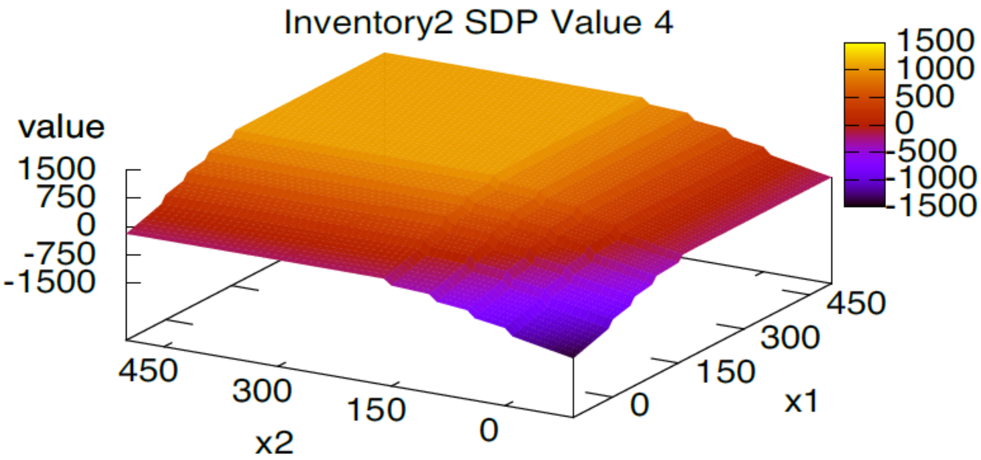
- 1D Example

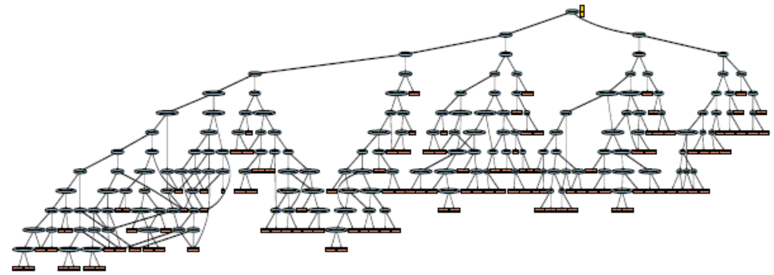


- Questions
  - What approximating class?
  - What error function?
    - Max not feasible
    - Volume of squared error? Integral is exact.

Vianna, Sanner, Barros (AAAI-15)

Real-time Symbolic Dynamic Programming





Sequential Control Summary

- Continuous state, action, observation (PO)MDPs
  - **Discrete action, nonlinear MDPs** **UAI-11**
  - **Continuous action MDPs (incl. exact policy)** **AAAI-12b**
  - Continuous observation POMDPs **NIPS-12**
  - **Chance-constrained solutions with continuous noise** **IJCAI-13**
  - **XADD Compression** **UAI-13**
  - Game-theoretic / adversarial setting **UAI-14**
  - RTDP: scaling nonlinear and continuous action solutions **AAAI-15**
  - First optimal solutions to market-making with inventory **IJCAI-16**
  - Analytic Policy Gradient, Sensitivity Analysis, Inverse Learning **ICAPS-17**

More Applications

Constrained Optimization

$\max_x$ case(x) = Constrained Optimization!

- Conditional constraints
  - E.g., if $(x > y)$ then $(y < z)$
  - MILP, MIQP equivalent

- Factored / sparse constraints
  - Constraints may be sparse!

    $x_1 > x_2, \quad x_2 > x_3, \quad \ldots, \quad x_{n-1} > x_n$
  - Dynamic programming for continuous optimization!

Symbolic Bucket Elimination, Student Paper Award, CPAIOR-18

- Parameterized optimization
  - $f(y) = \max_x f(x,y)$
  - Maximum value, substitution as a **function of y**

Recap

- Defined a calculus for piecewise functions
  - $f_1 \oplus f_2$, $f_1 \otimes f_2$
  - $\max(f_1, f_2)$, $\min(f_1, f_2)$
  - $\int_x f(x)$
  - $\max_x f(x)$, $\min_x f(x)$

- **Defined XADD to efficiently compute with cases**

- **Makes possible**
  - **Exact inference in continuous graphical models**
  - **First exact solutions to planning, control, and OR problems**
  - **New paradigms for optimization**

Symbolic Piecewise Calculus + XADD

= Expressive Hybrid Inference, Optimization, & Control

Thank you!

Questions?