On the Representation of Probabilities over Structured Domains

Marius Bozga and Oded Maler

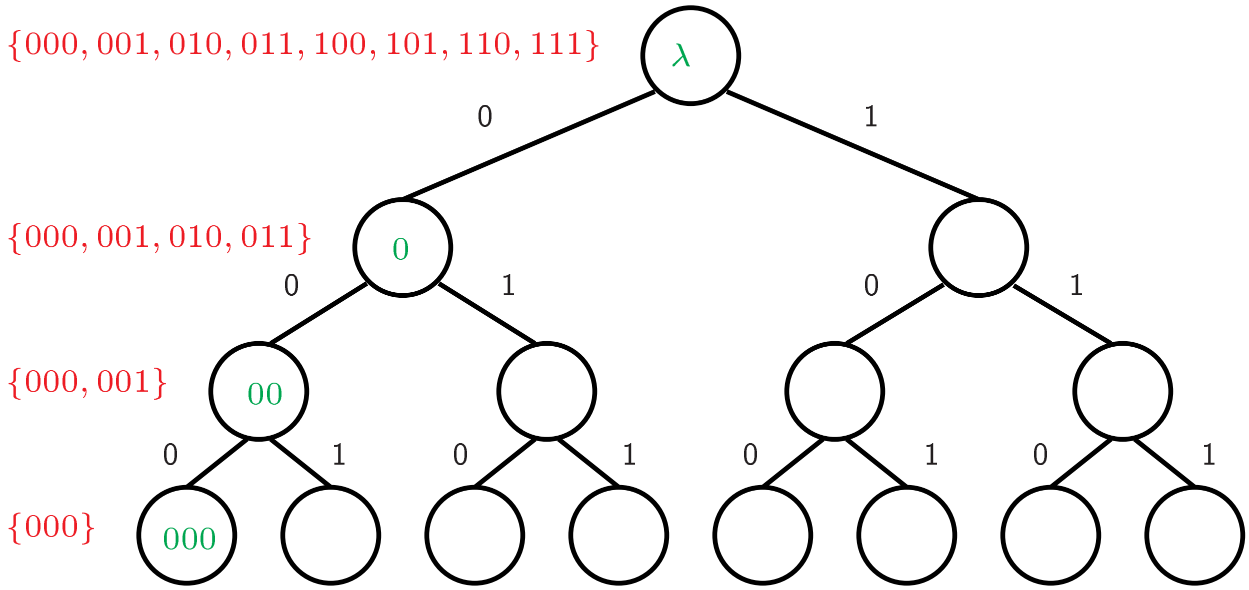
VERIMAG

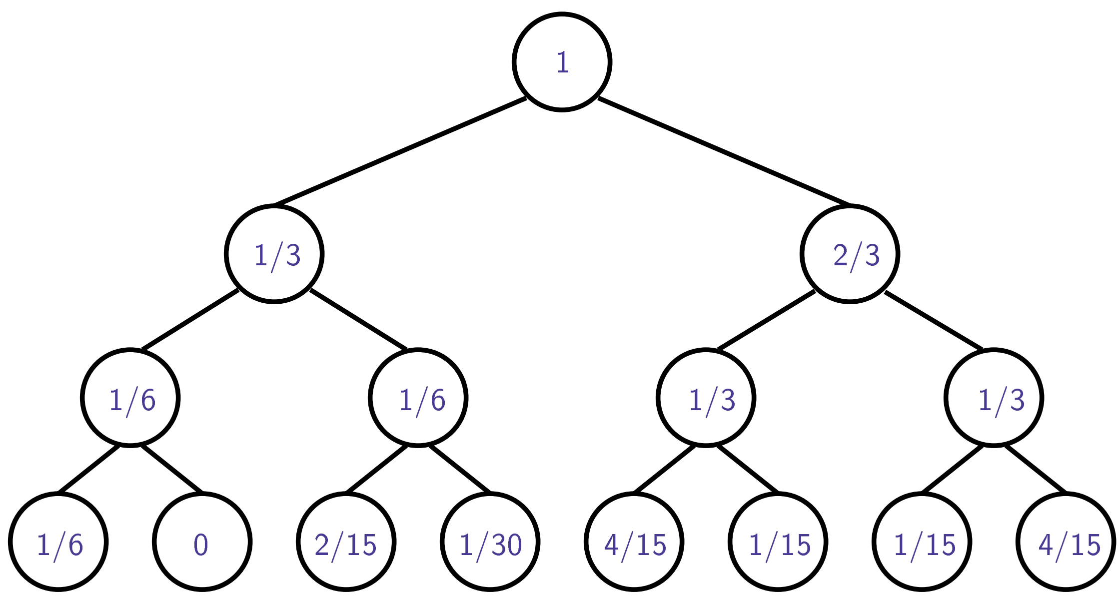
Grenoble, France

Summary

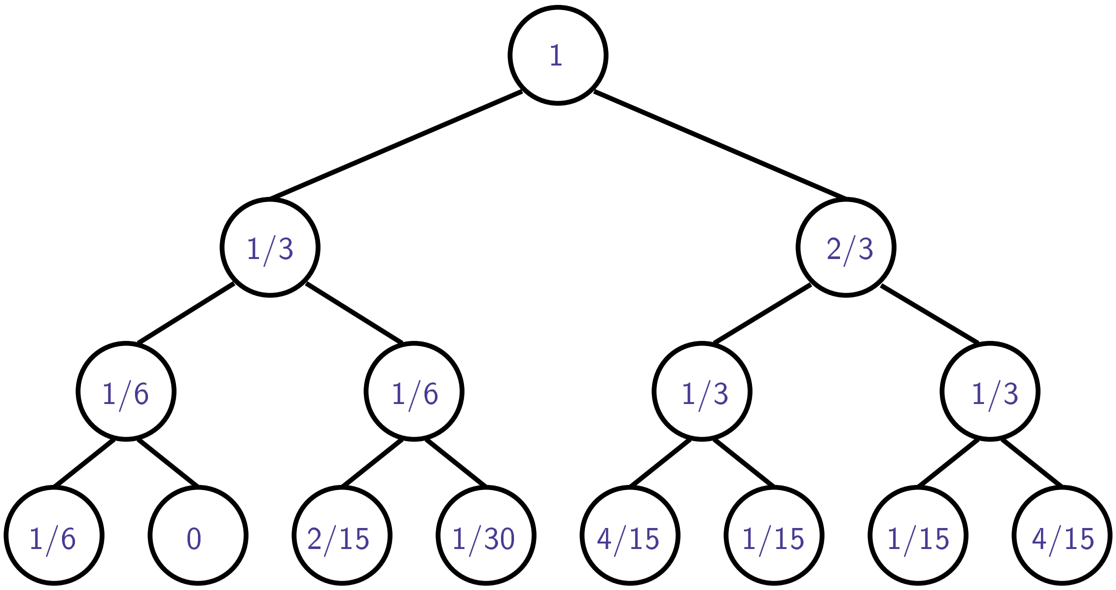
	Qualitative Non-Determinism	Quantitative Non-Determinism
State	Set of states $p : Q \to \{0, 1\}$	Prob on states $p : Q \to [0, 1]$
Next State	$\delta : Q \to (Q \to \{0, 1\})$	$\delta : Q \to (Q \to [0, 1])$
Forward Comp.	$p' = p \cdot A_\delta$ $(\mathbb{B}, \cup, \cap)$	$p' = p \cdot A_\delta$ $(\mathbb{R}, +, \cdot)$
Struct. Systems	BDD	PDG

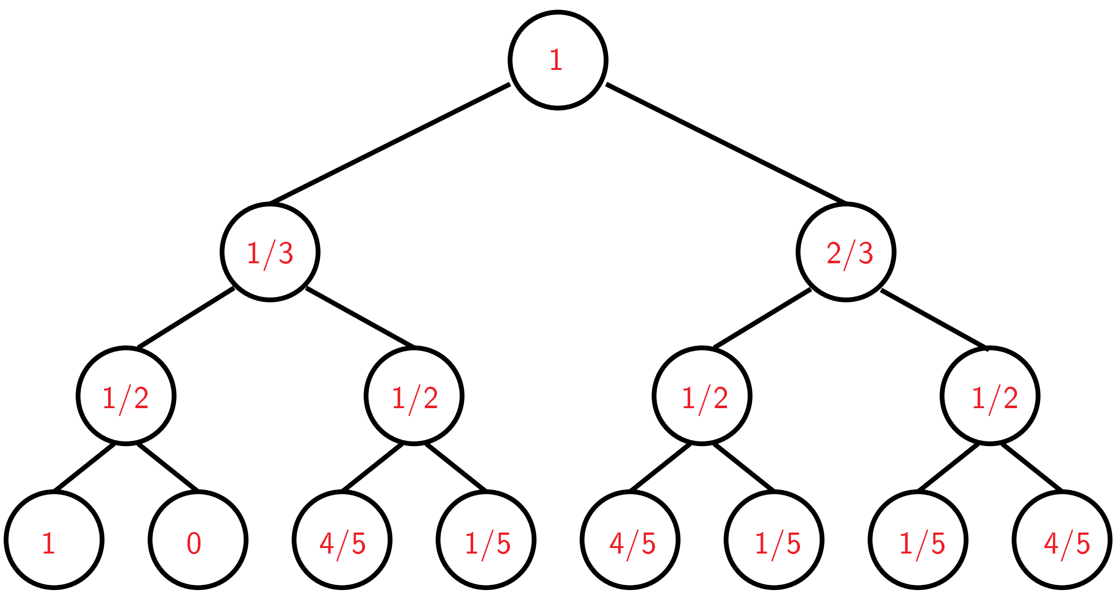
Subsets and Probabilities on $\mathbb{B}^n$





Probabilistic Decision Trees





In other Words

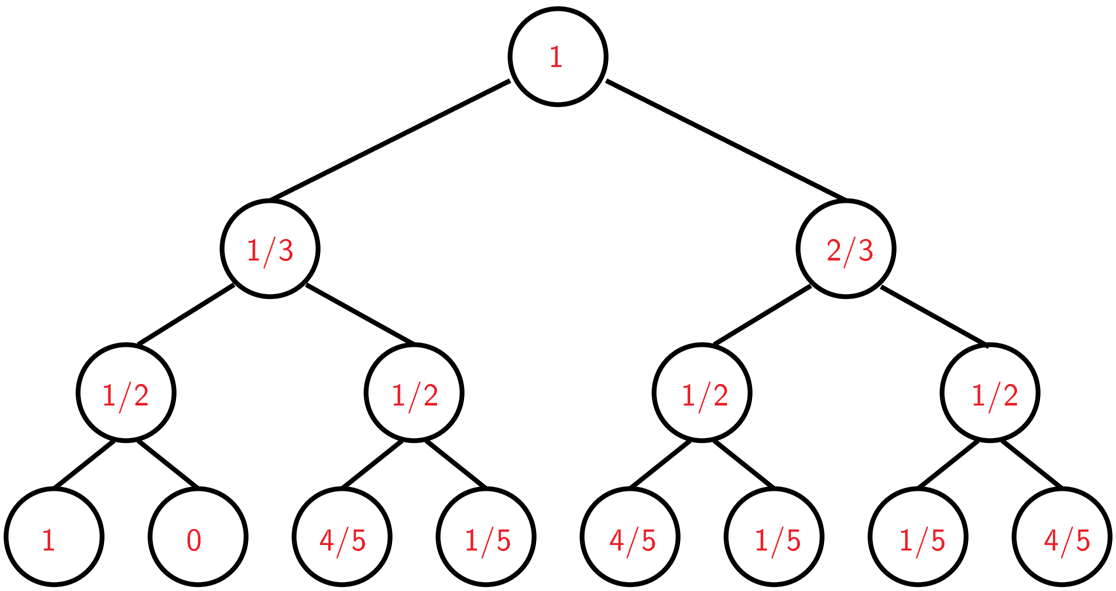
Boolean Functions	Probability Functions
Shannon Factorization	Chain Rule

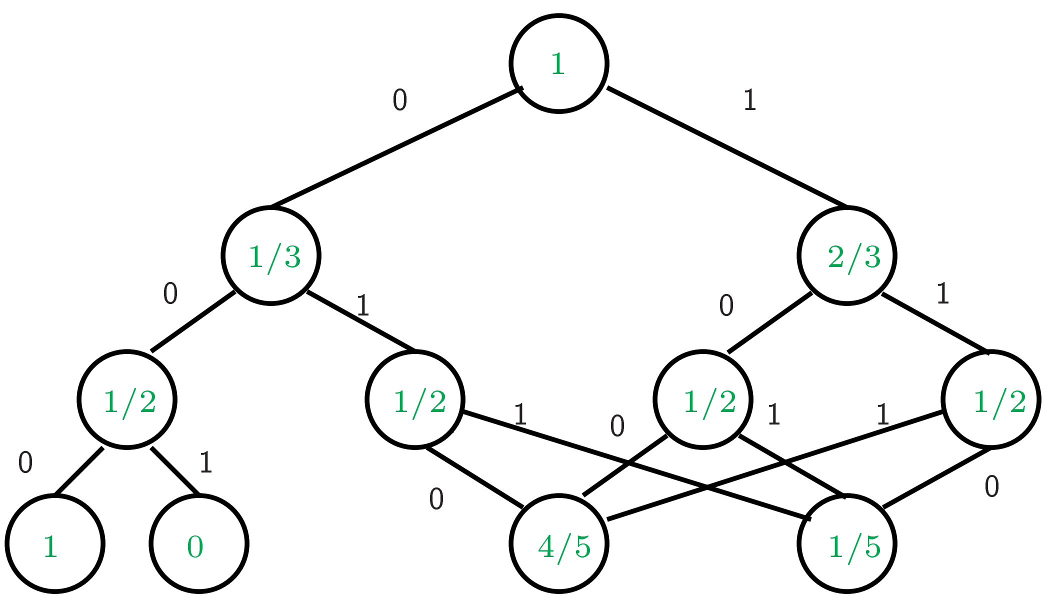
$$p(x_1 x_2 \cdots x_n)$$
$$=$$
$$p(x_1) \cdot p(x_1 x_2 | x_1) \cdots p(x_1 x_2 \cdots x_n | x_1 \cdots x_{n-1})$$
$$=$$
$$p(x_1) \cdot p(x_2 | x_1) \cdots p(x_n | x_1 \cdots x_{n-1})$$

$$p(x_1 x_2 \cdots x_n)$$
$$=$$
$$p(x_1) \cdot p_{x_1}(x_1 x_2) \cdots p_{x_1 \cdots x_{n-1}}(x_1 \cdots x_n)$$
$$=$$
$$p(x_1) \cdot p_{x_1}(x_2) \cdots p_{x_1 \cdots x_{n-1}}(x_n)$$

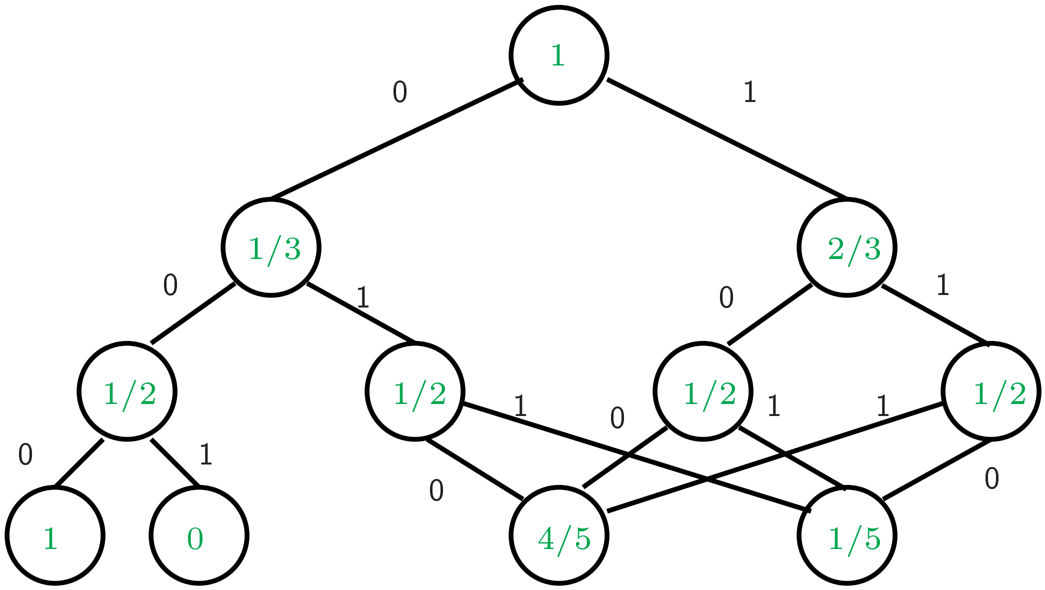
Probablistic Decision Graphs (PDG)

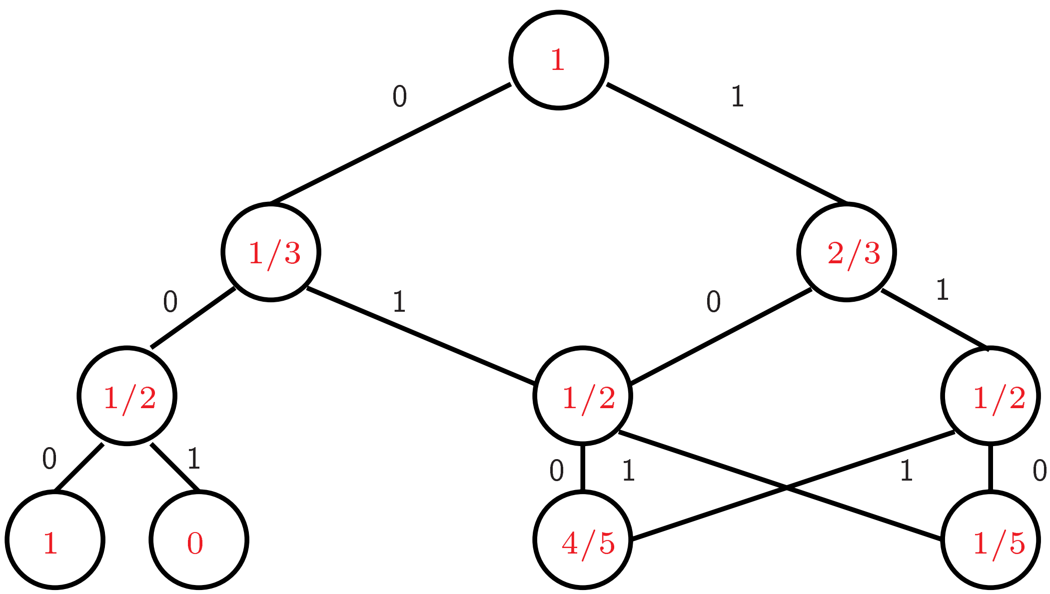
From full trees to DAGs:



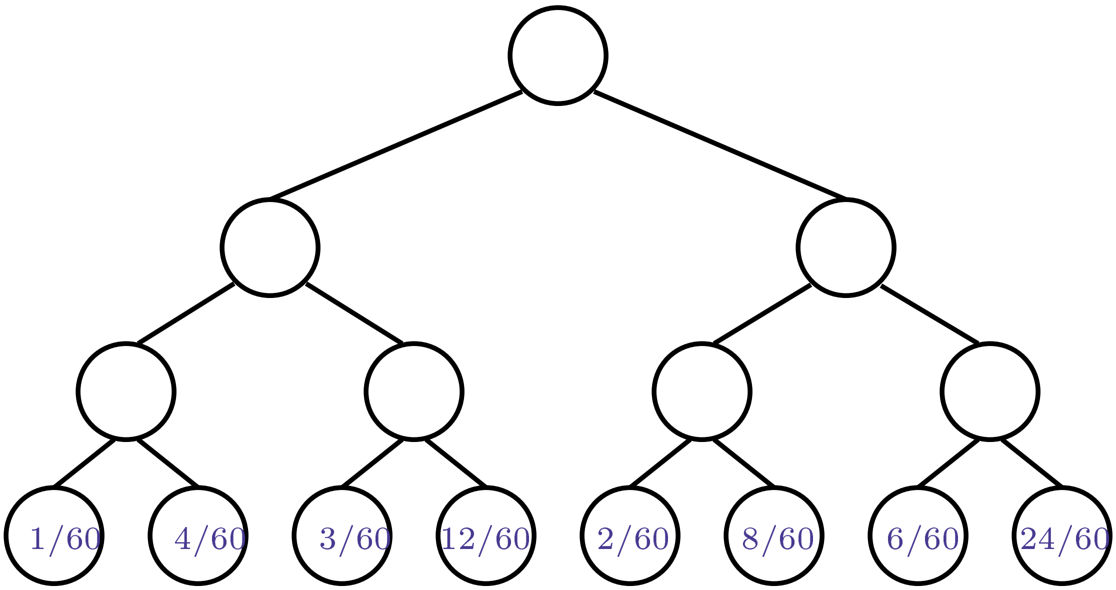


PDG - continued

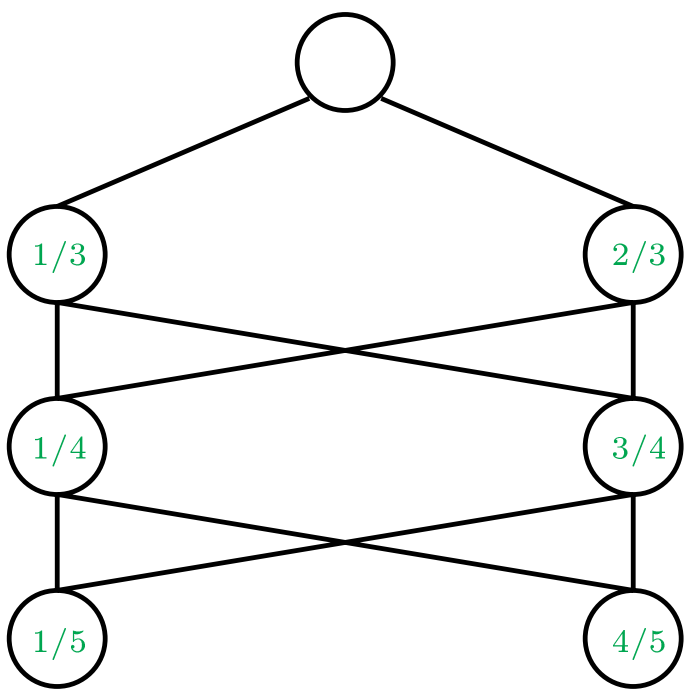




Example: Independent Variables



Exponential MTBDD/ADD vs. Linear PDG



Dynamics: Markov Transition Functions

$$\delta : Q \to (Q \to [0,1])$$

$\forall q \in Q$: $\delta_q : Q \to [0,1]$ is a probability function

Traditional representation as $|Q| \times |Q|$ matrix:

$$\begin{array}{llll} \delta_1(1) & \delta_1(2) & \ldots & \delta_1(n) \\ \delta_2(1) & \delta_2(2) & \ldots & \delta_2(n) \\ \ldots & \ldots & \ldots & \ldots \\ \delta_n(1) & \delta_n(2) & \ldots & \delta_n(n) \end{array}$$

Current state probability $p : Q \to [0,1]$

$$[p_1, \ldots, p_n]$$

Probability of a transition:

$$\begin{array}{llll} p_1 \cdot \delta_1(1) & p_1 \cdot \delta_1(2) & \ldots & p_1 \cdot \delta_1(n) \\ p_2 \cdot \delta_2(1) & p_2 \cdot \delta_2(2) & \ldots & p_2 \cdot \delta_2(n) \\ \ldots & \ldots & \ldots & \ldots \\ p_n \cdot \delta_n(1) & p_n \cdot \delta_n(2) & \ldots & p_n \cdot \delta_n(n) \end{array}$$

Next-state probability:

$$p' = [\sum_i p_i \cdot \delta_i(1), \ldots, \sum_i p_i \cdot \delta_i(n)]$$

Structured Markov Transition Functions

$$\delta : \mathbb{B}^n \to (\mathbb{B}^n \to [0,1])$$

$$\forall x_1 \cdots x_n : \delta_{x_1 \cdots x_n} : \mathbb{B}^n \to [0,1]$$

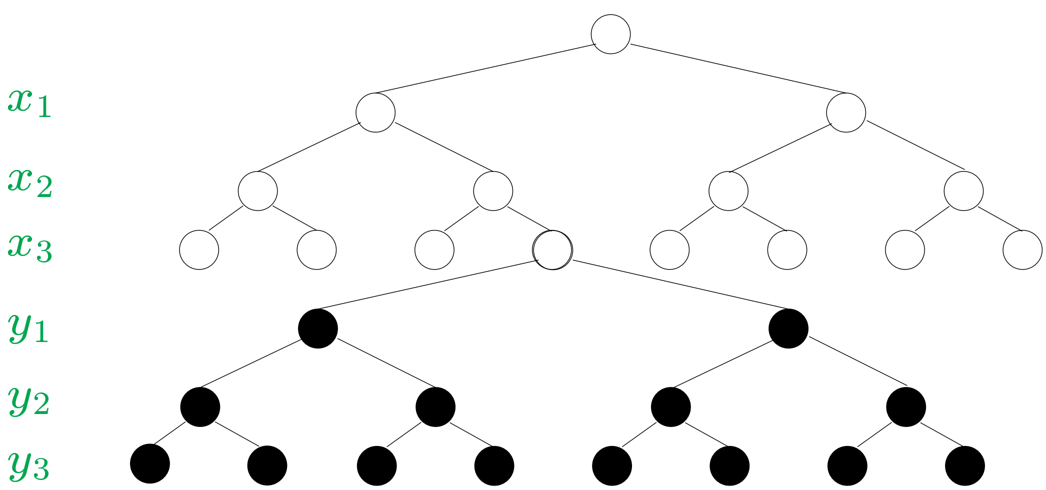
is a probability function

Notation: $\delta_{x_1 \cdots x_n}(y_1 \cdots y_n)$

$\delta_{x_1 \cdots x_n}$ can be decomposed:

$$\delta_{x_1 \cdots x_n}(y_1 \cdots y_n) =$$
$$\delta_{x_1 \cdots x_n}(y_1) \cdots \delta_{x_1 \cdots x_n y_1 \cdots y_{n-1}}(y_1 \cdots y_n).$$

Conditional PDG (two types of nodes):

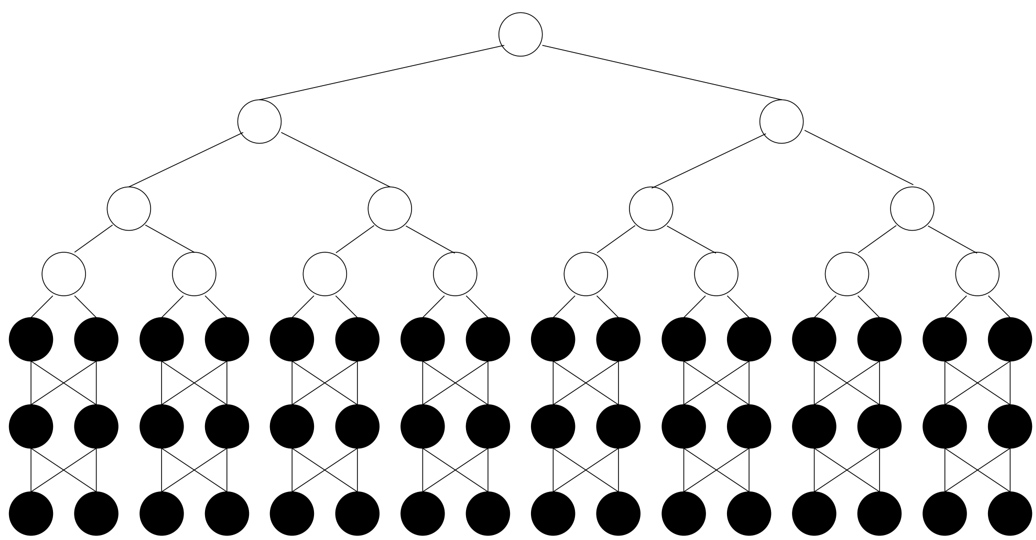


Causal Transition Functions

All the next-state (y) variables depend only on the current state variables (x):

$$\delta_{x_1 \cdots x_n}(y_1 \cdots y_n) =$$

$$\delta_{x_1 \cdots x_n}(y_1) \cdot \delta_{x_1 \cdots x_n}(y_2) \cdots \delta_{x_1 \cdots x_n}(y_n)$$

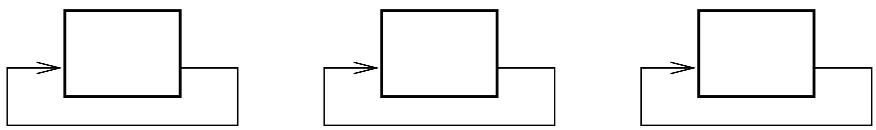


$O(|Q| \log |Q|)$ instead of $O(|Q|^2)$!!

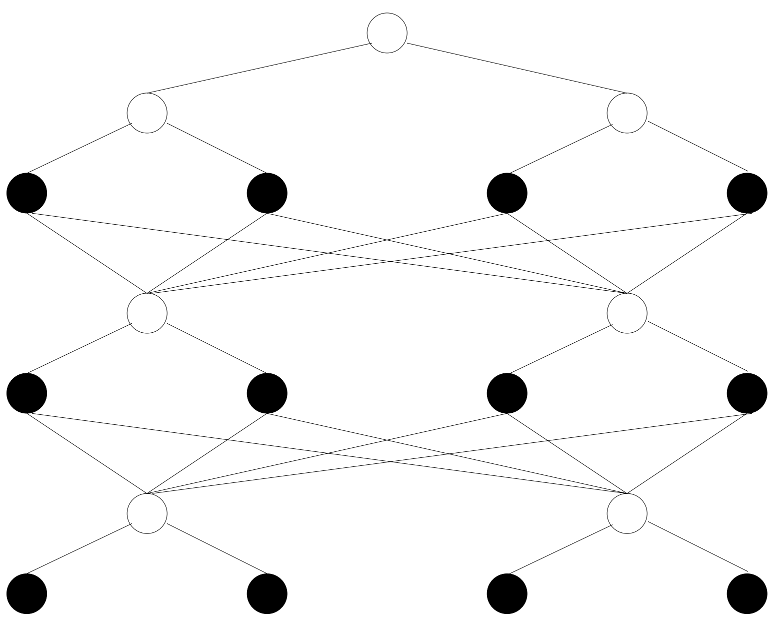
Exploiting Independence (I):

Each y variable **must** appear after all the x variables on which it depends

Independent Markov chains:

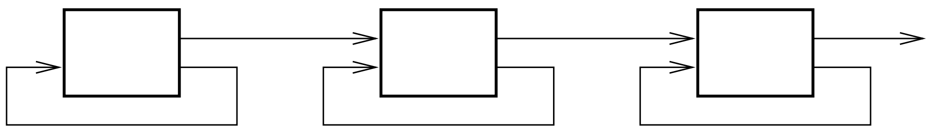


Size $O(n)$:

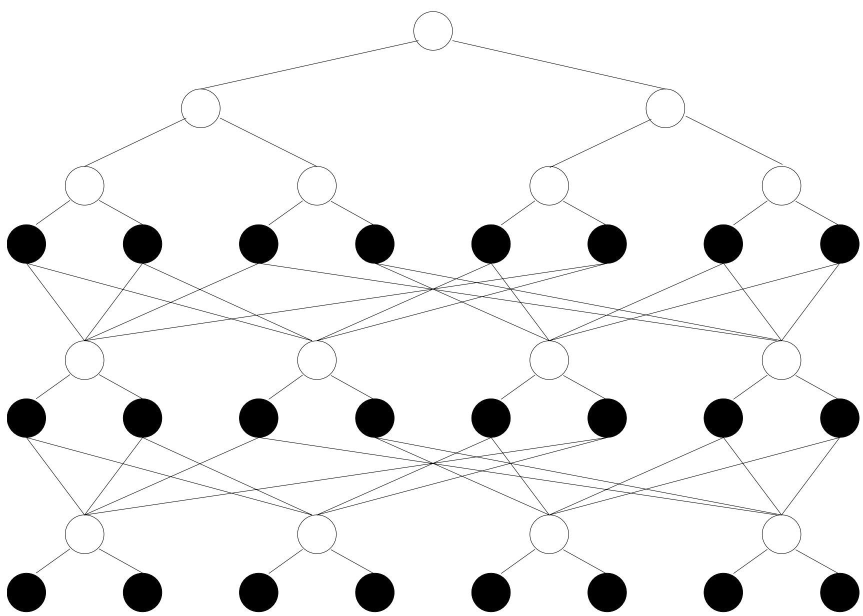


Exploiting Independence (II):

A cascade of probabilistic automata of depth 2:



Size $O(2n)$:

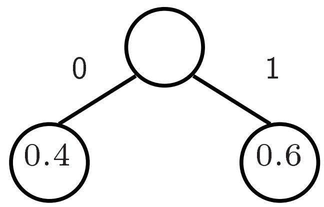


A cascade of depth k: $O(kn)$

Calculating Next-State Probabilities

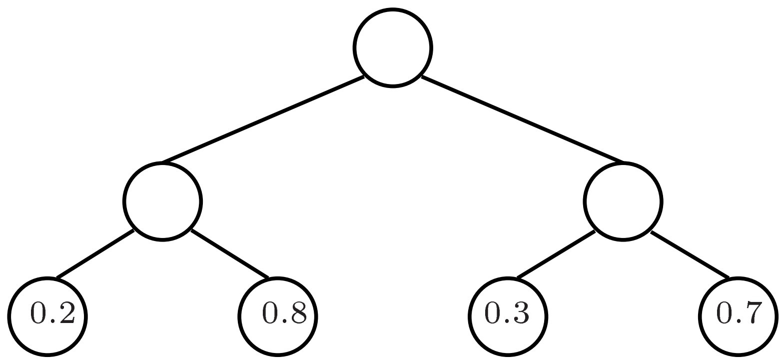
Current probability:

$[0.4, 0.6]$



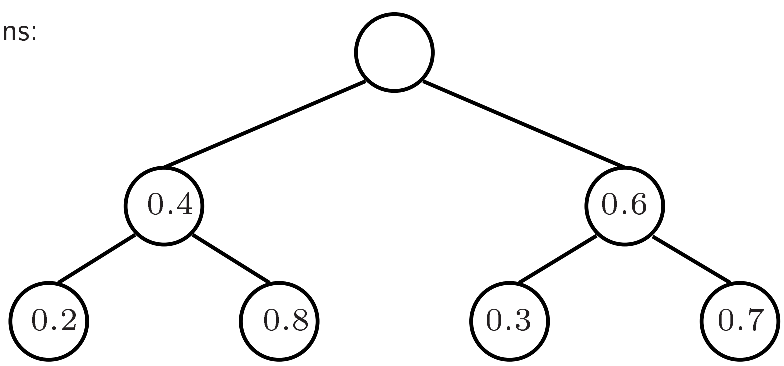
Transition function:

$$\begin{matrix} 0.2 & 0.8 \\ 0.3 & 0.7 \end{matrix}$$



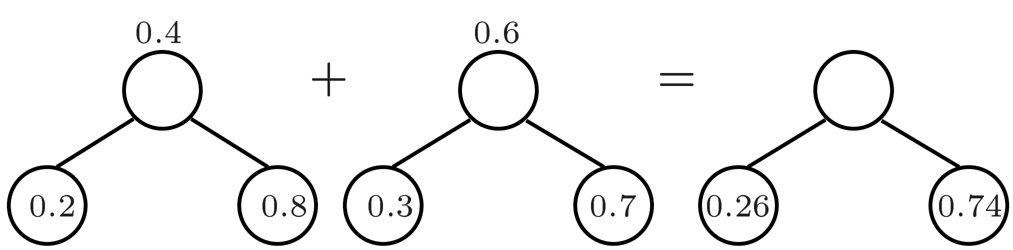
Probability of transitions:

$$\begin{matrix} 0.08 & 0.32 \\ 0.18 & 0.42 \end{matrix}$$



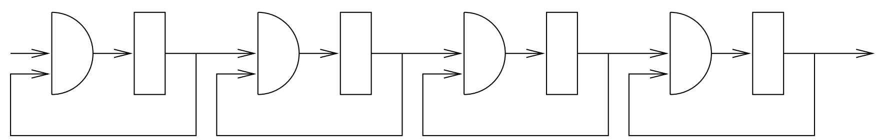
Next probability:

$[0.26, 0.74]$

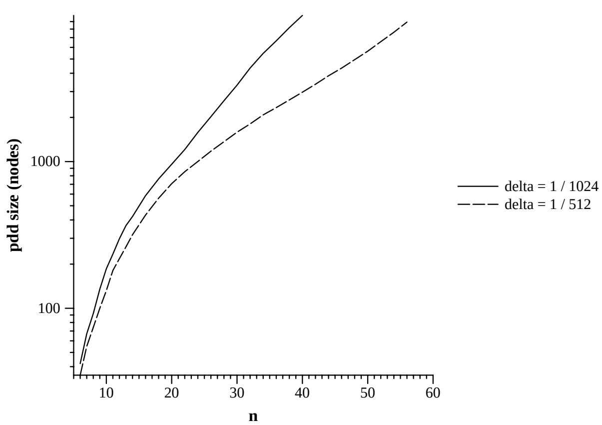


Experimental Results

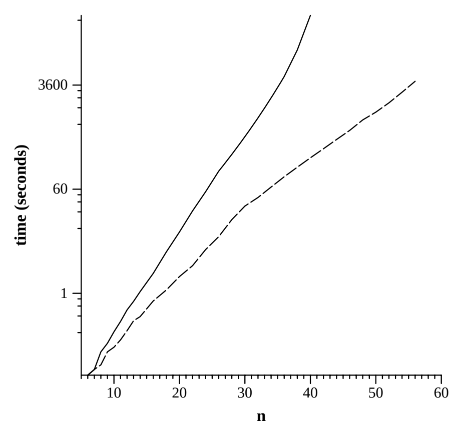
Cascades of noisy AND gates:



Space



Time



Related Work

MTBDD

(Clarke, Fujita, McGeer, McMillan, Yang)

ADD

(Bahar, Frohm, Ganoa, Hachtel, Macii, Pardo, Somenzi)

Edge-valued BDD

(Vrudhula, Pedram, Lai, Tafertshofer)

Bayesian Networks

(AI literature)

Future Work

- Complete the input language for the tool
- Case-studies (noisy protocols and circuits, queues, probabilistic timed automata, etc.)
- Solving large MDPs (controller synthesis, structured utility functions)
- Other algorithms on PDG (eigenvectors?)