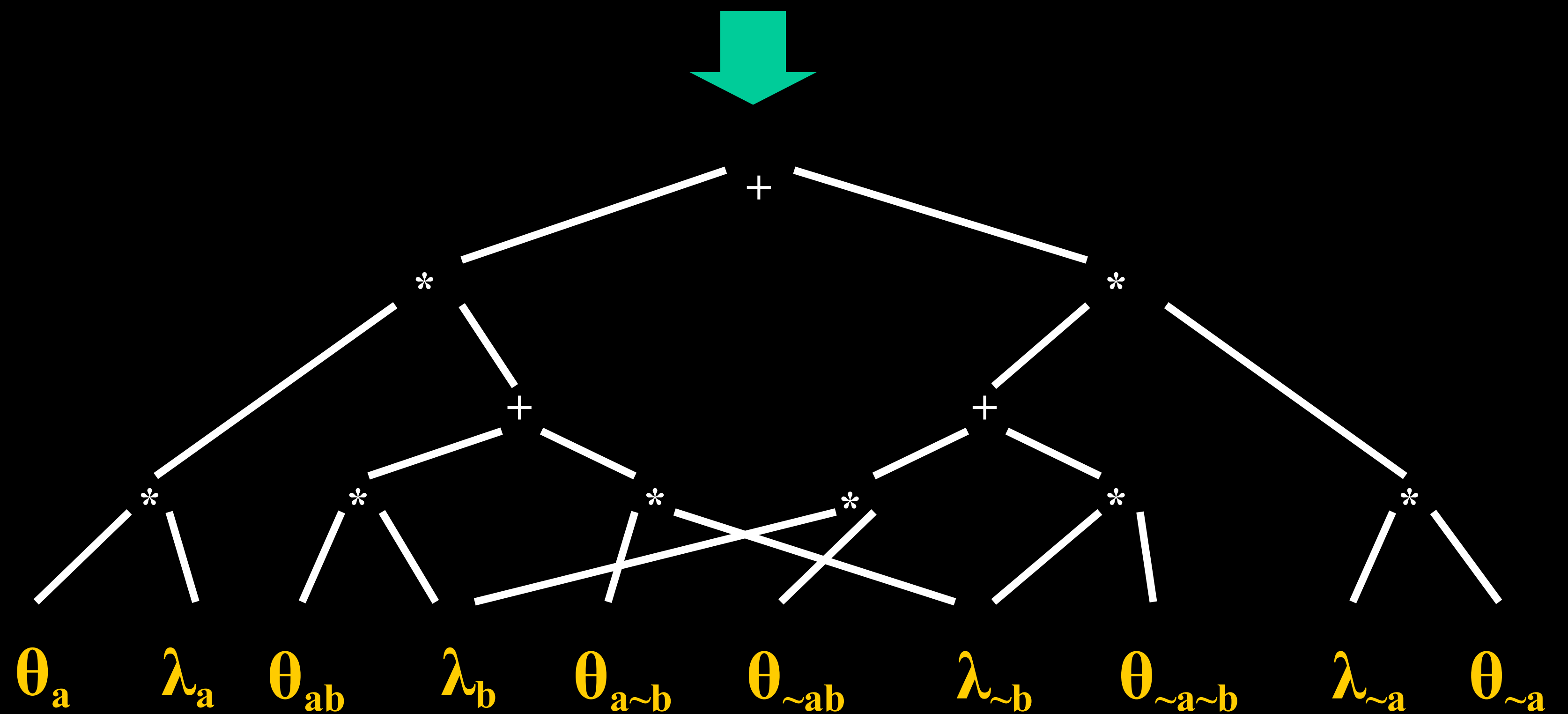
Compiling Bayesian Networks into Arithmetic Circuits (ACs)

Adnan Darwiche

Chapter 12

Polynomials ➔ Arithmetic Circuits

$$F = \lambda_a \lambda_b \theta_a \theta_{b|a} + \lambda_a \lambda_{\sim b} \theta_a \theta_{\sim b|a} + \lambda_{\sim a} \lambda_b \theta_{\sim a} \theta_{b|\sim a} + \lambda_{\sim a} \lambda_{\sim b} \theta_{\sim a} \theta_{\sim b|\sim a}$$



Arithmetic Circuits

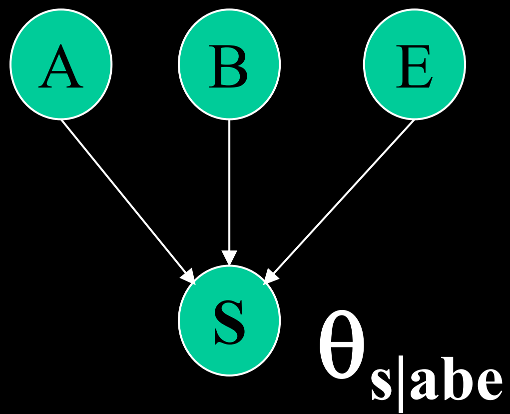
- Exploits

  functional constraints and context-specific independence

- Real-time inference

- Can be learned from data or even handcrafted

Local CPT Structure

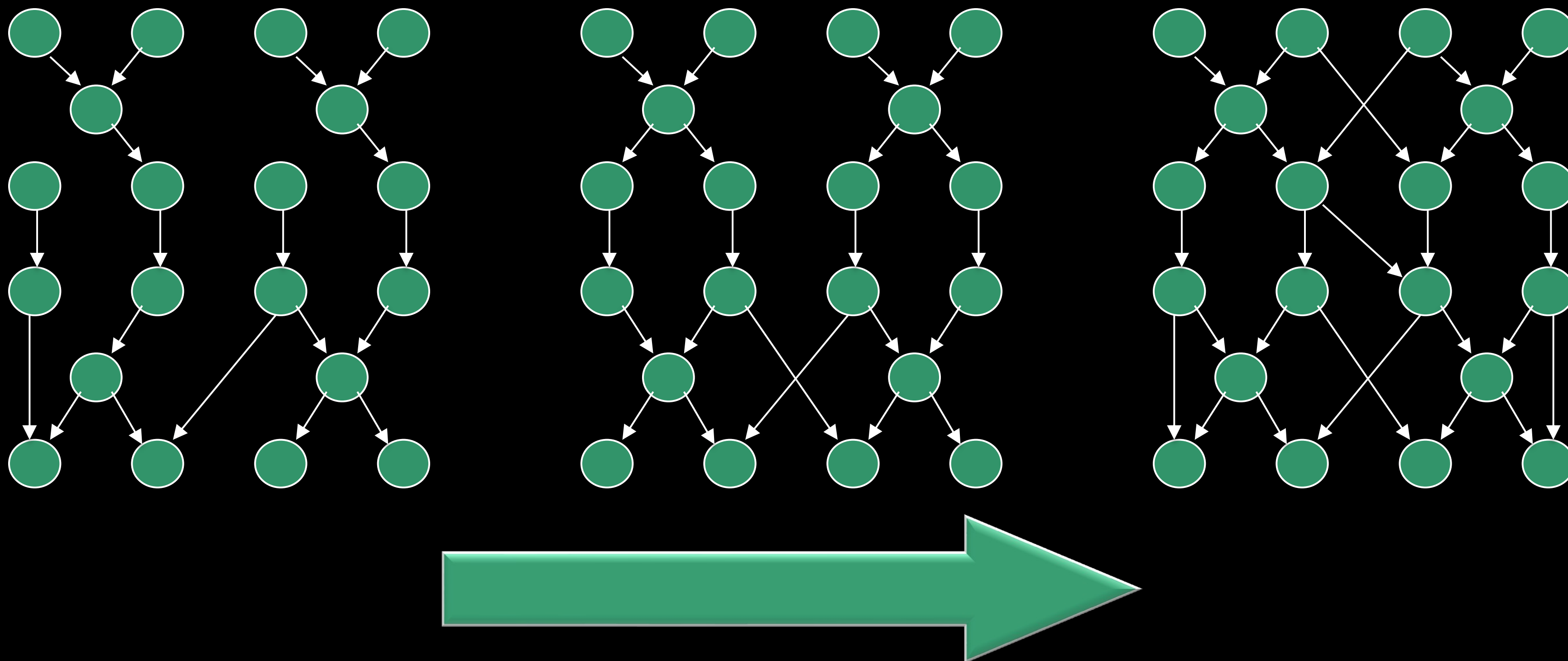


A	B	E	Pr(S|A,B,E)
a	b	e	0.95
a	b	$\bar{e}$	0.95
a	$\bar{b}$	e	0.20
a	$\bar{b}$	$\bar{e}$	0.05
$\bar{a}$	b	e	0.00
$\bar{a}$	b	$\bar{e}$	0.00
$\bar{a}$	$\bar{b}$	e	0.00
$\bar{a}$	$\bar{b}$	$\bar{e}$	0.00

Tabular CPT

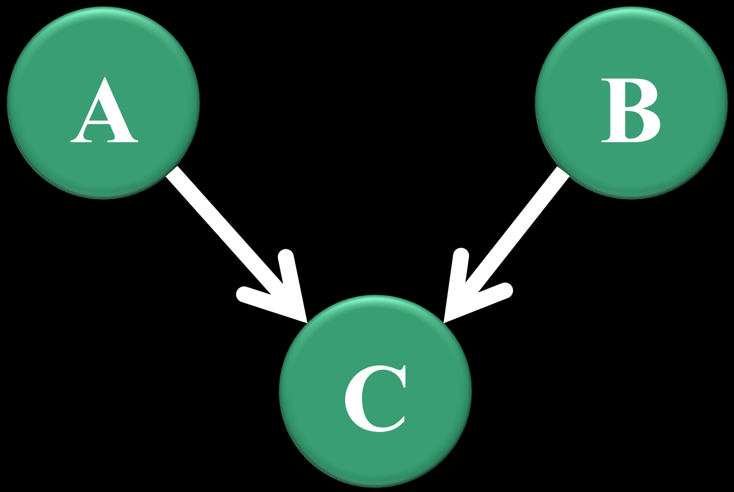
-Functional constraints
-Context-specific independence

Perceived Barrier: Treewidth

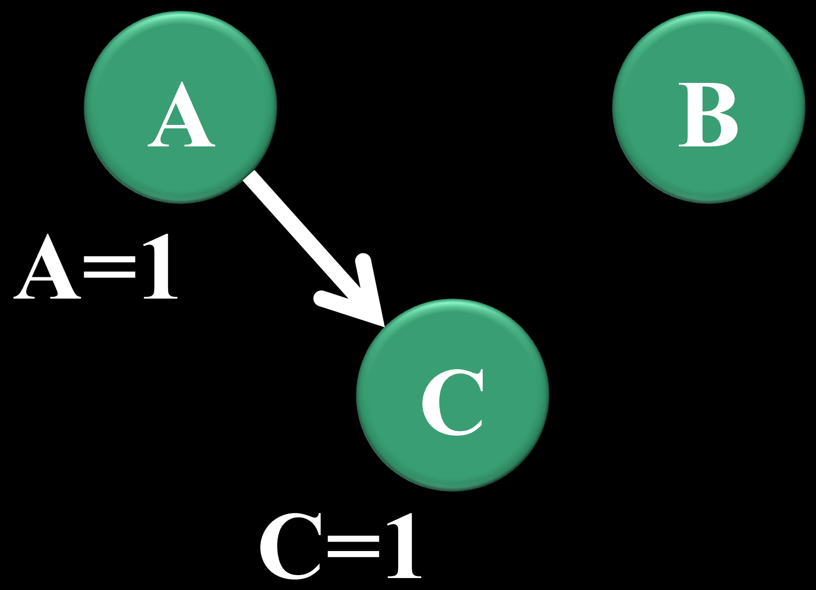


$$O(n \exp\{w\})$$

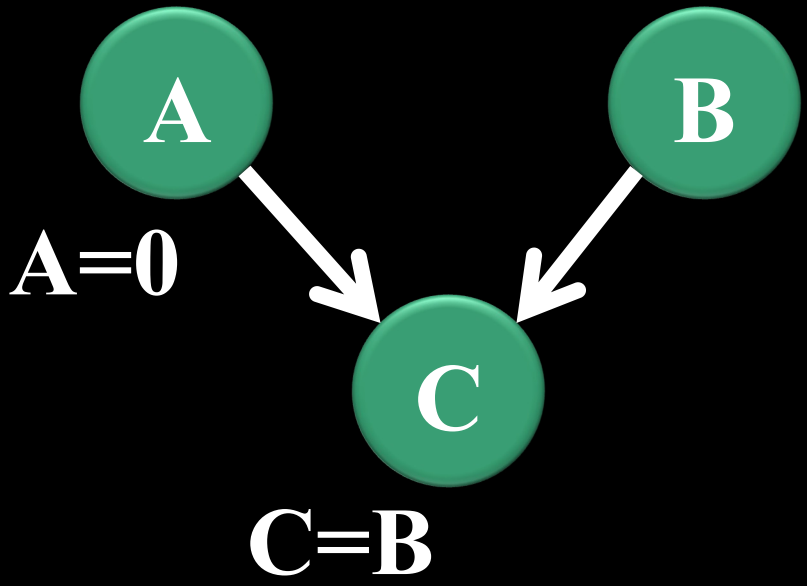
Context-Specific Independence



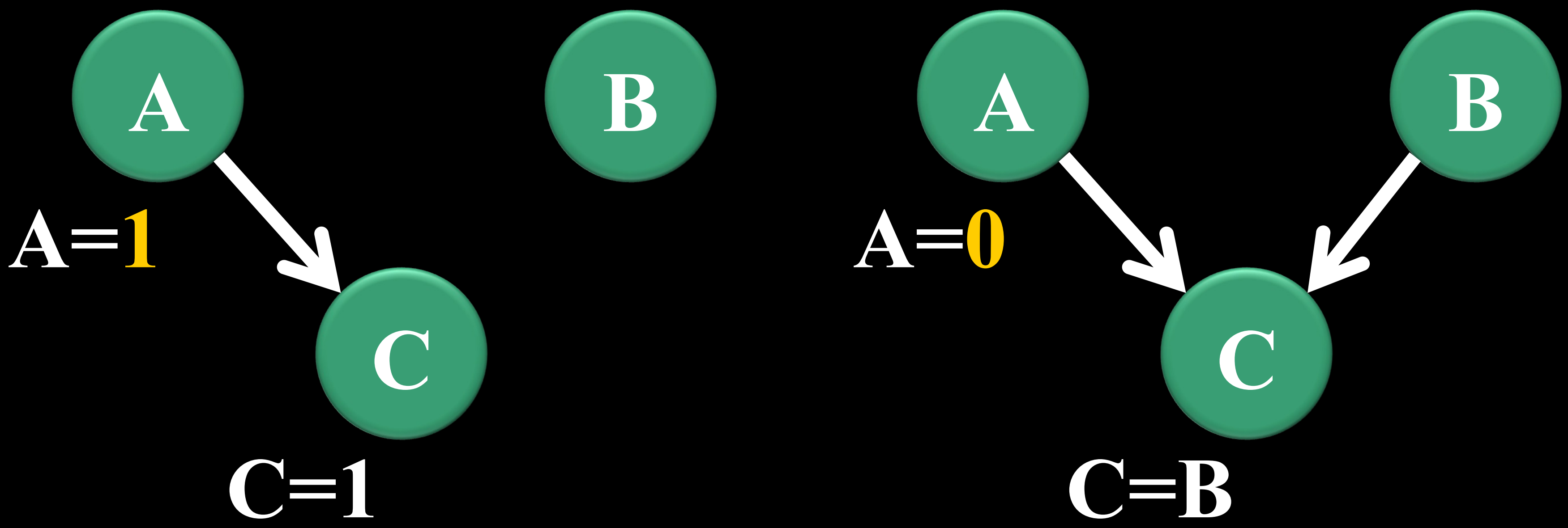
Context-Specific Independence



Context-Specific Independence

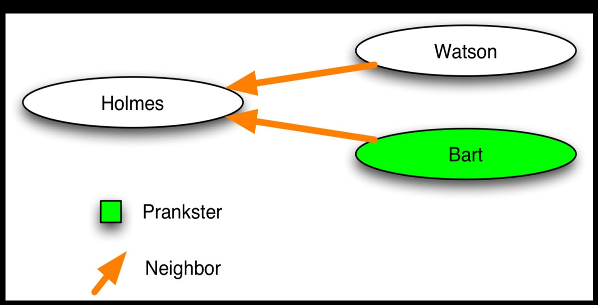


Context-Specific Independence

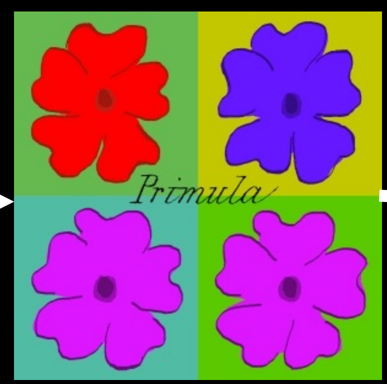


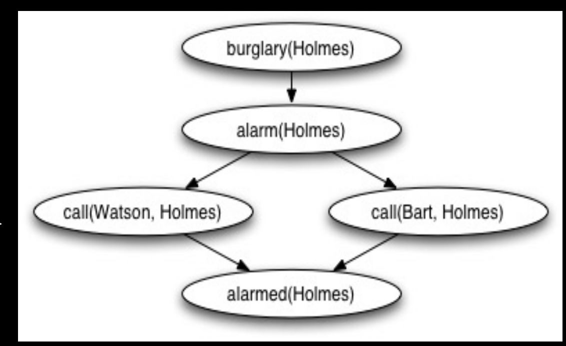
Relational Probabilistic Models

```
burglary(v)=0.005;
alarm(v)=(burglary(v):0.95,0.01);
calls(v,w)=
(neighbor(v,w):
(prankster(v)):
(alarm(w):0.9,0.05),
(alarm(w):0.9,0)),0);
alarmed(v)=
n-or{calls(w,v)|w:neighbor(w,v)}
```

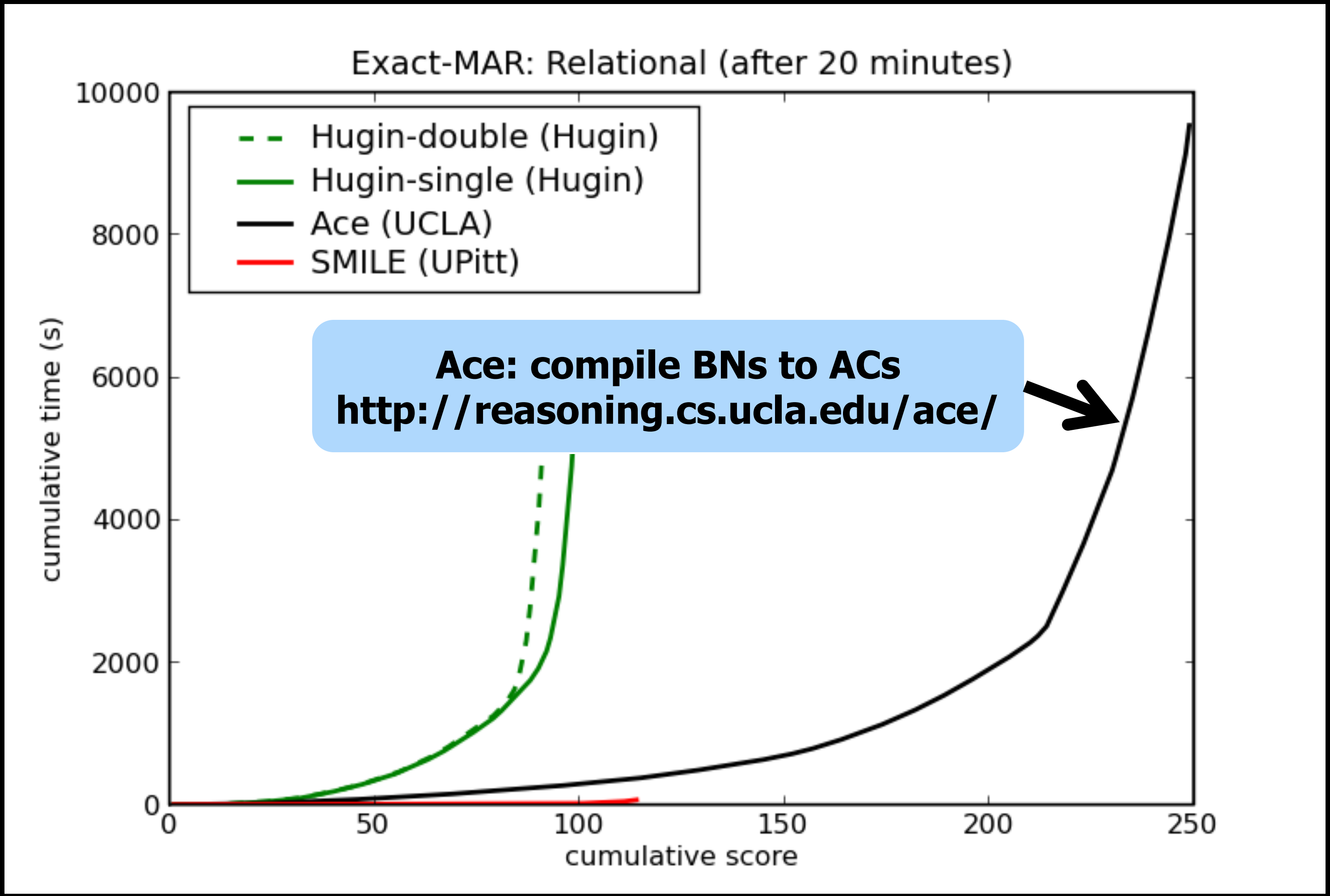


Primula





UAI'08 Competition: Exact Inference using ACs



2021 Evaluation:
Exact Inference using ACs

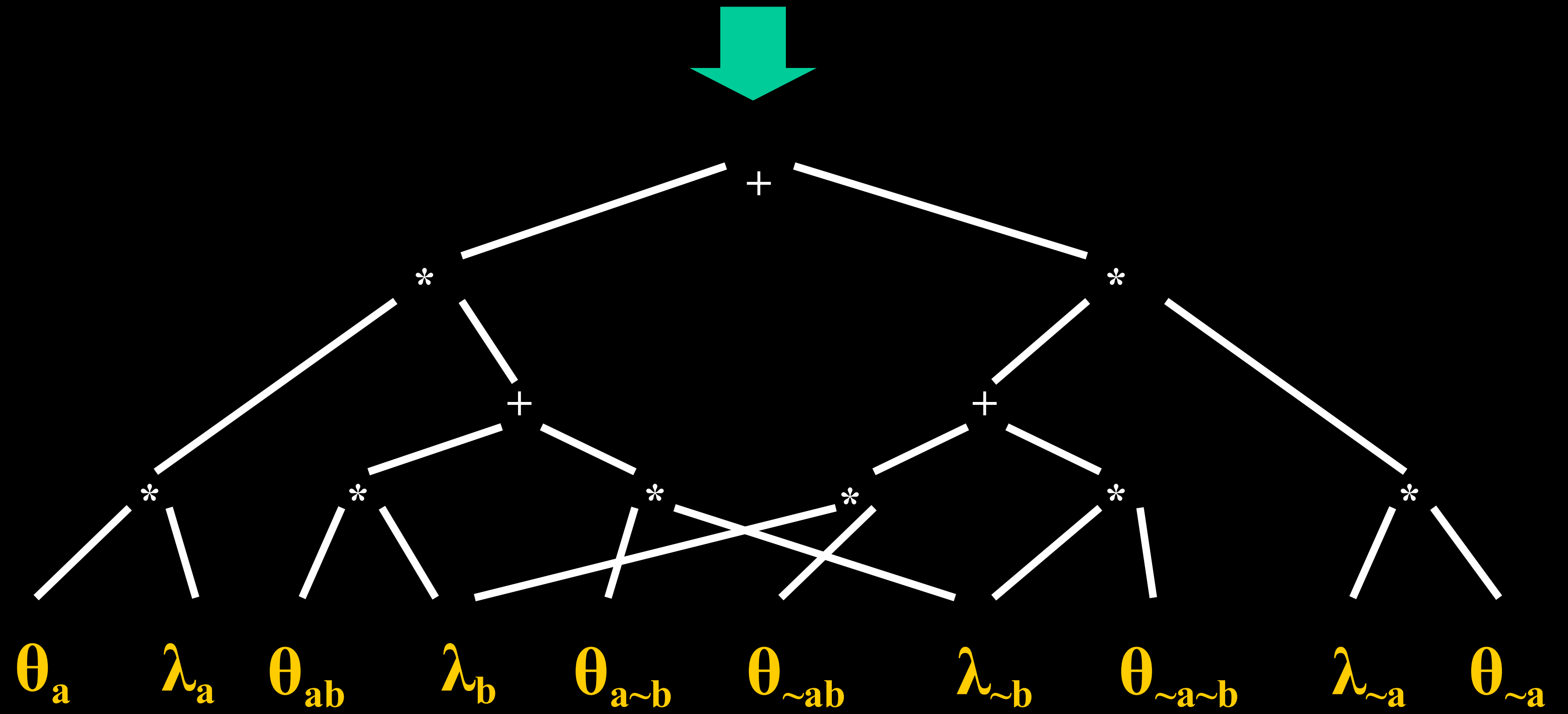
Durgesh Agrawal, Yash Pote, Kuldeep S. Meel:
Partition Function Estimation: A Quantitative Study.

Method Name	Problem Classes								
	Relation-al (354)	Prome-das (65)	BN (60)	Ising (52)	Segment (50)	ObjDetect (35)	Protein (29)	Misc (27)	Total (672)
Ace	354	65	60	51	50	0	16	15	611
Fractional Belief Propagation (FBP)	293	65	58	41	48	32	29	9	575
Loopy Belief Propagation (BP)	292	65	58	41	46	32	29	10	573
Generalized Belief Propagation (GBP)	281	65	36	47	40	34	29	9	541
Edge Deletion Belief Propagation (EDBP)	245	42	56	50	49	35	28	23	528
GANAK	353	58	53	4	0	0	7	14	489
Double Loop Generalised BP (HAK)	199	65	58	43	43	35	29	14	486
Tree Expectation Propagation (TREEEP)	101	65	58	50	48	35	29	15	401
SampleSearch	89	56	33	52	37	35	29	25	356
Bucket Elimination (BE)	98	32	15	52	50	35	29	22	333
Conditioned Belief Propagation (CBP)	109	32	21	41	50	35	29	8	325
Join Tree (JT)	98	32	15	52	50	19	26	21	313
Dynamic Importance Sampling (DIS)	24	65	25	52	50	35	29	27	307
Weighted Mini Bucket Elimination (WMB)	68	13	17	50	50	20	28	12	258
miniC2D	187	1	30	31	0	0	0	1	250
WeightCount	93	0	27	0	0	0	0	0	120
WISH	0	0	0	9	0	0	0	0	9
FocusedFlatSAT	6	0	0	0	0	0	0	0	6

reasoning.cs.ucla.edu/ace

Polynomials ➔ Arithmetic Circuits

$$F = \lambda_a \lambda_b \theta_a \theta_{b|a} + \lambda_a \lambda_{\sim b} \theta_a \theta_{\sim b|a} + \lambda_{\sim a} \lambda_b \theta_{\sim a} \theta_{b|\sim a} + \lambda_{\sim a} \lambda_{\sim b} \theta_{\sim a} \theta_{\sim b|\sim a}$$



A	B	Pr(.)
true	true	.03
true	false	.27
false	true	.56
false	false	.14

$$Pr(a) = .03 + .27 = .3$$

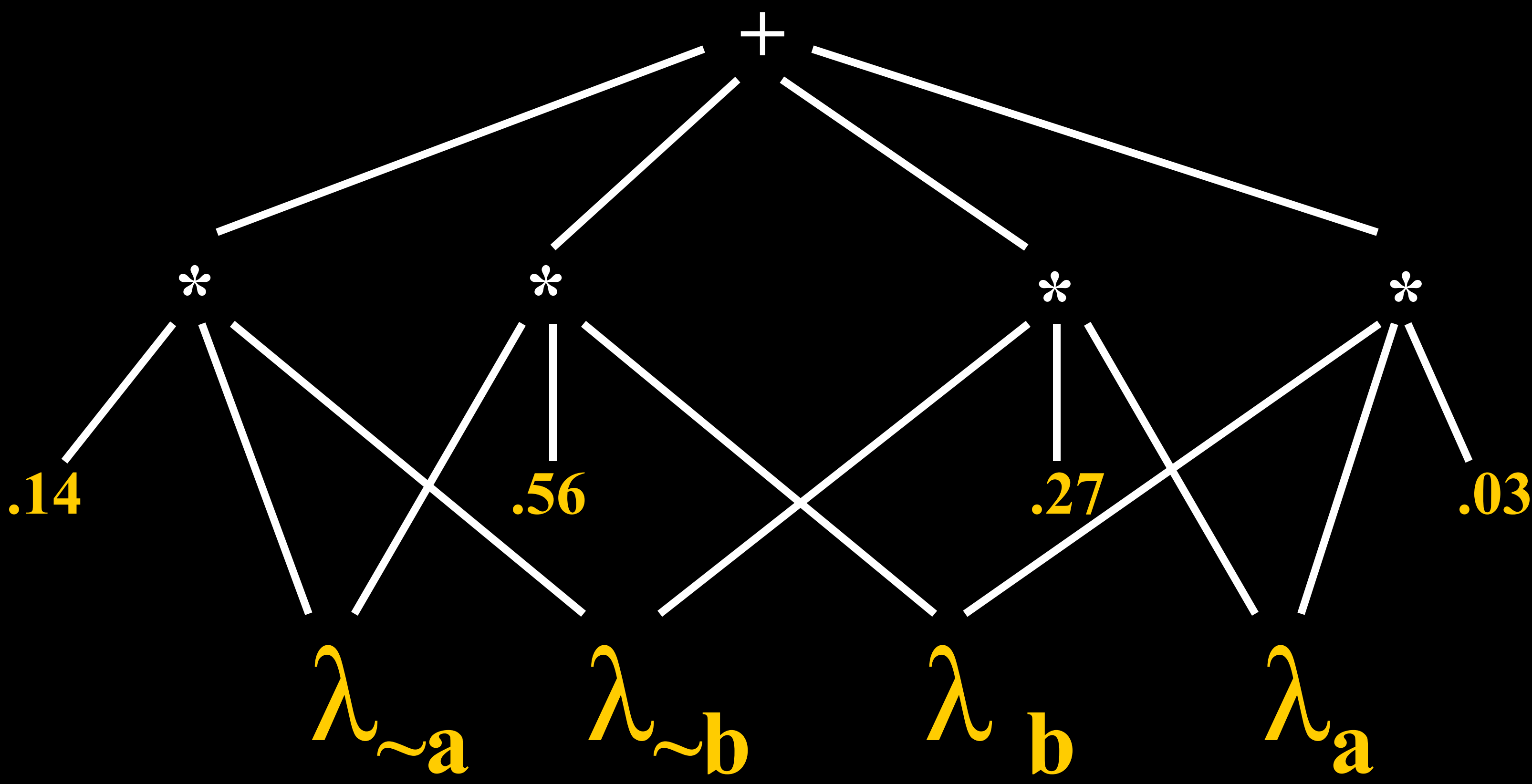
A	B	Pr(.)
true	true	.03
true	false	.27
false	true	.56
false	false	.14

$$Pr(\sim b) = .27 + .14 = .41$$

A	B	Pr(.)
true	true	$\lambda_a\lambda_b$ .03
true	false	$\lambda_a\lambda_{\sim b}$ .27
false	true	$\lambda_{\sim a}\lambda_b$ .56
false	false	$\lambda_{\sim a}\lambda_{\sim b}$ .14

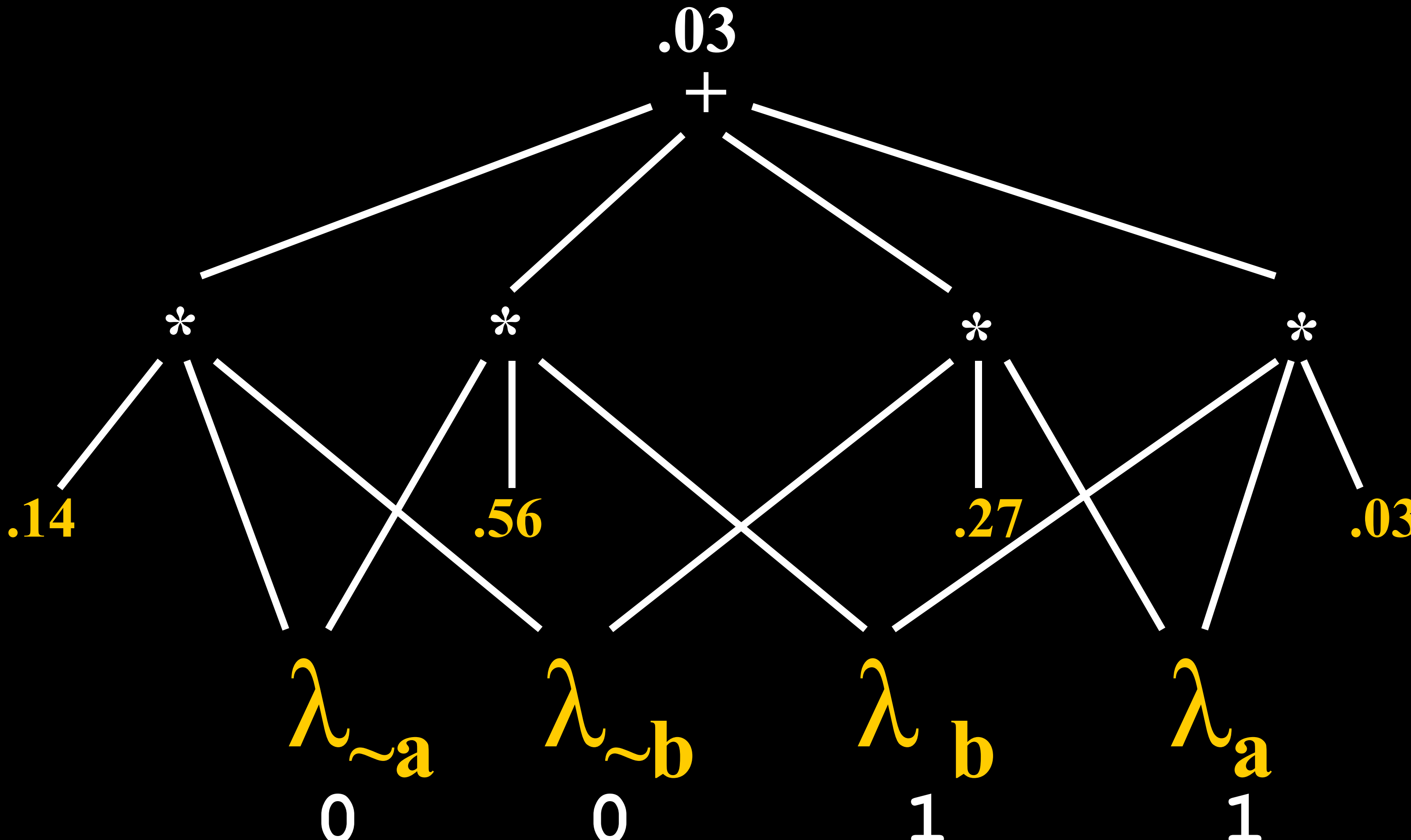
$$F(\lambda_{\sim a}, \lambda_{\sim b}, \lambda_a, \lambda_b) =$$

$$.03\lambda_a\lambda_b + .27\lambda_a\lambda_{\sim b} + .56\lambda_{\sim a}\lambda_b + .14\lambda_{\sim a}\, \lambda_{\sim b}$$

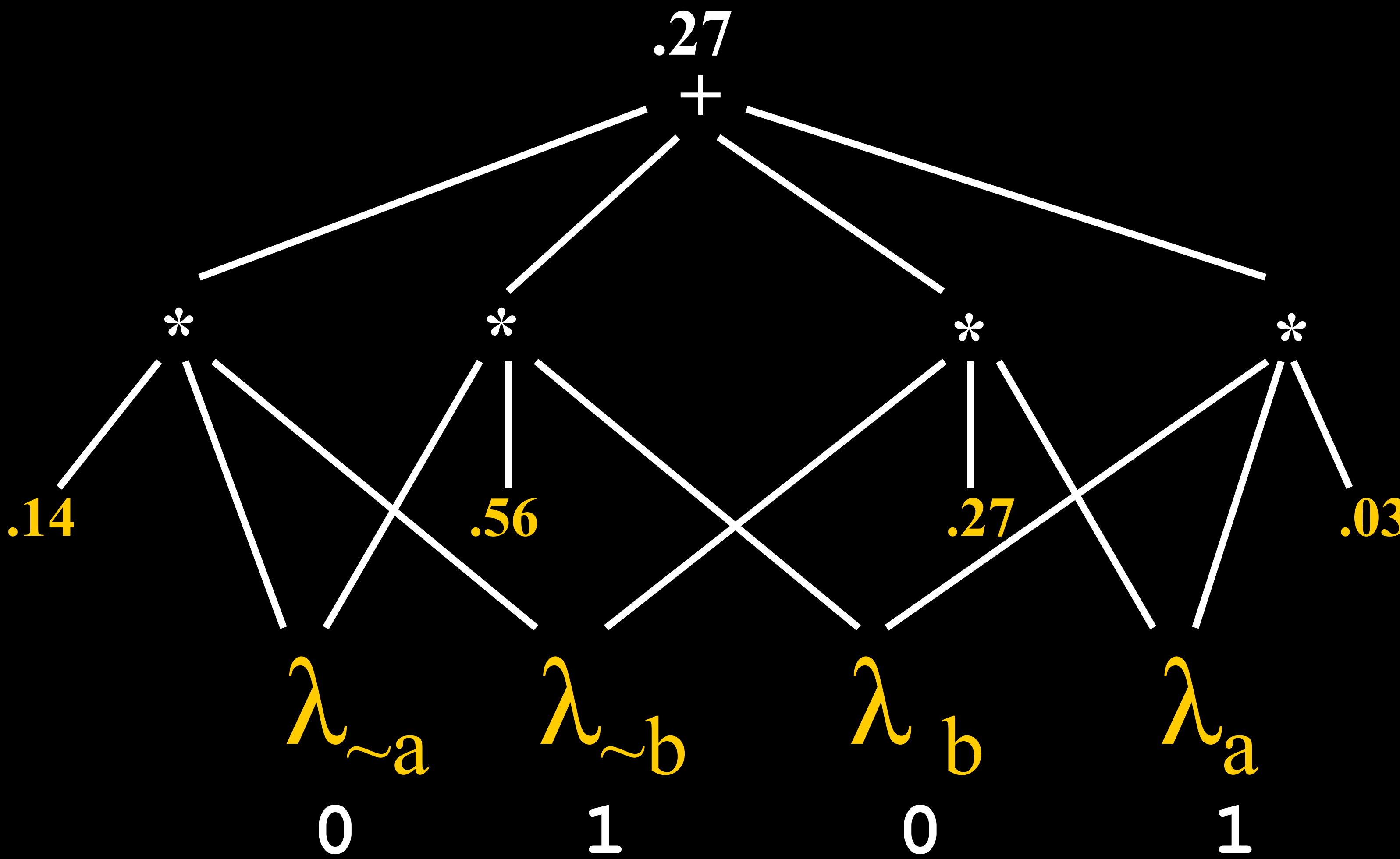


$$\mathbf{F}(\lambda_{\sim a}, \lambda_{\sim b}, \lambda_{b}, \lambda_{a})$$
$$=$$
$$.03\lambda_a\lambda_b + .27\lambda_a\lambda_{\sim b} + .56\lambda_{\sim a}\lambda_b + .14\lambda_{\sim a}\,\lambda_{\sim b}$$

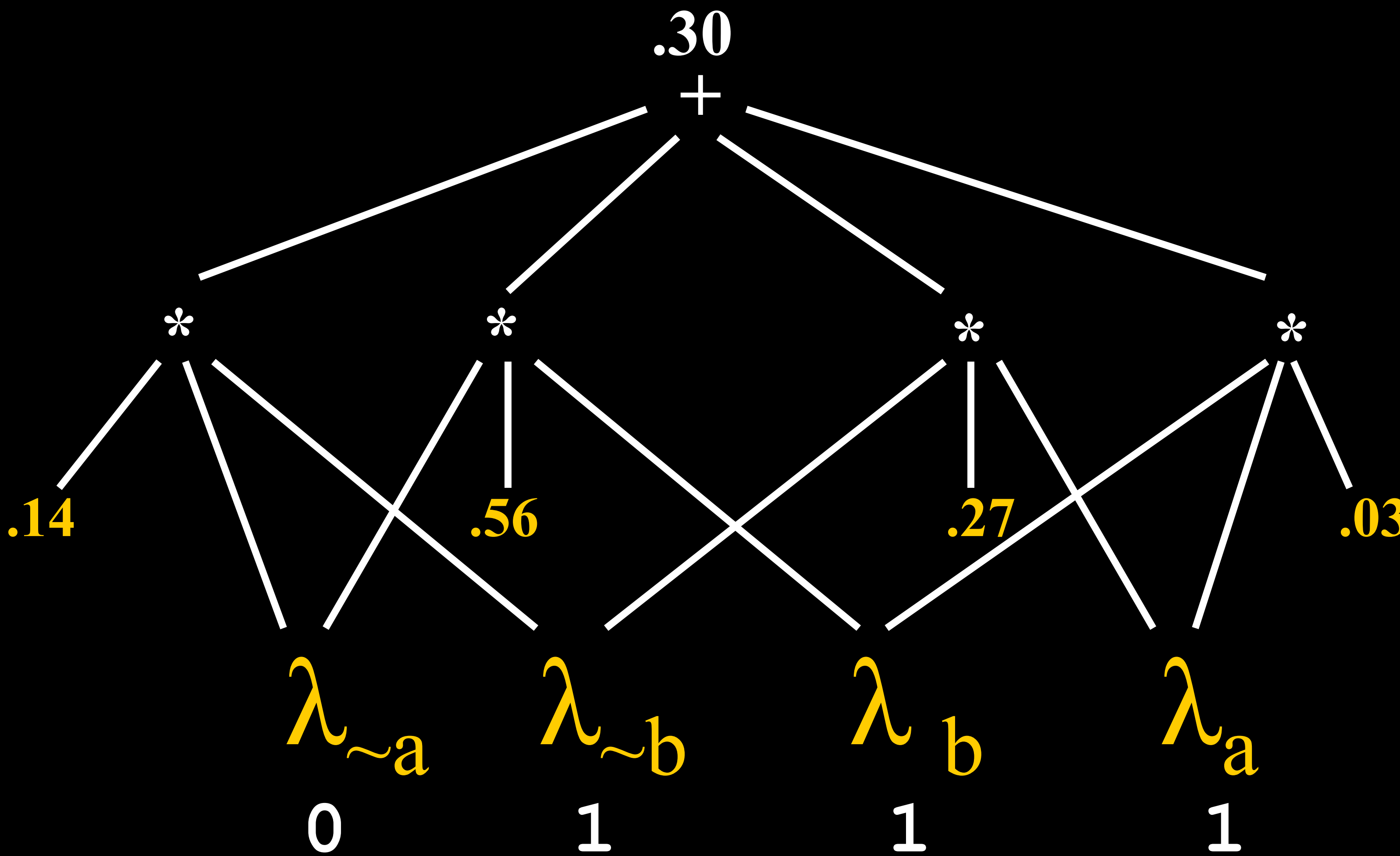
Effect of Evidence on Values of Evidence Indicators



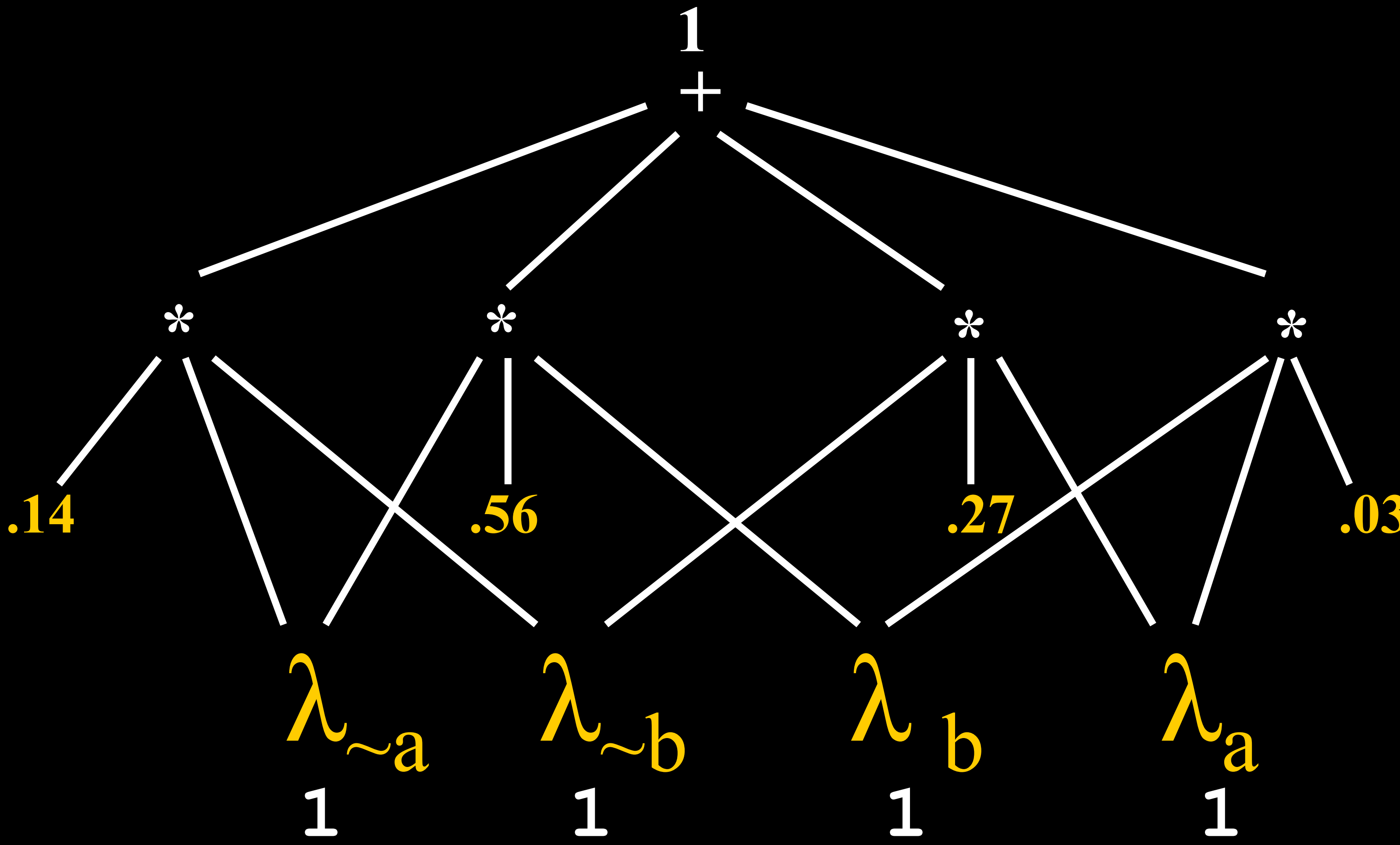
Evidence: a & b



Evidence: a & ~b



Evidence: a



Evidence: true

$F(\lambda_{\sim a}, \lambda_{\sim b}, \lambda_a, \lambda_b)$

$= .03\lambda_a\lambda_b + .27\lambda_a\lambda_{\sim b} + .56\lambda_{\sim a}\lambda_b + .14\lambda_{\sim a}\lambda_{\sim b}$

Pr(a,~b)

$= F(\lambda_{\sim a}{:}0, \lambda_{\sim b}{:}1, \lambda_a{:}1, \lambda_b{:}0)$

$= .27$

Pr(a)

$= F(\lambda_{\sim a}{:}0, \lambda_{\sim b}{:}1, \lambda_a{:}1, \lambda_b{:}1)$

$= .03+.27$

$$F(\lambda_{\sim a}, \lambda_{\sim b}, \lambda_a, \lambda_b)$$

$$= .03\lambda_a\lambda_b + .27\lambda_a\lambda_{\sim b} + .56\lambda_{\sim a}\lambda_b + .14\lambda_{\sim a}\lambda_{\sim b}$$

$Pr(a,\sim b)$

$= F(\lambda_{\sim a}:0, \lambda_{\sim b}:1, \lambda_a:1, \lambda_b:0)$

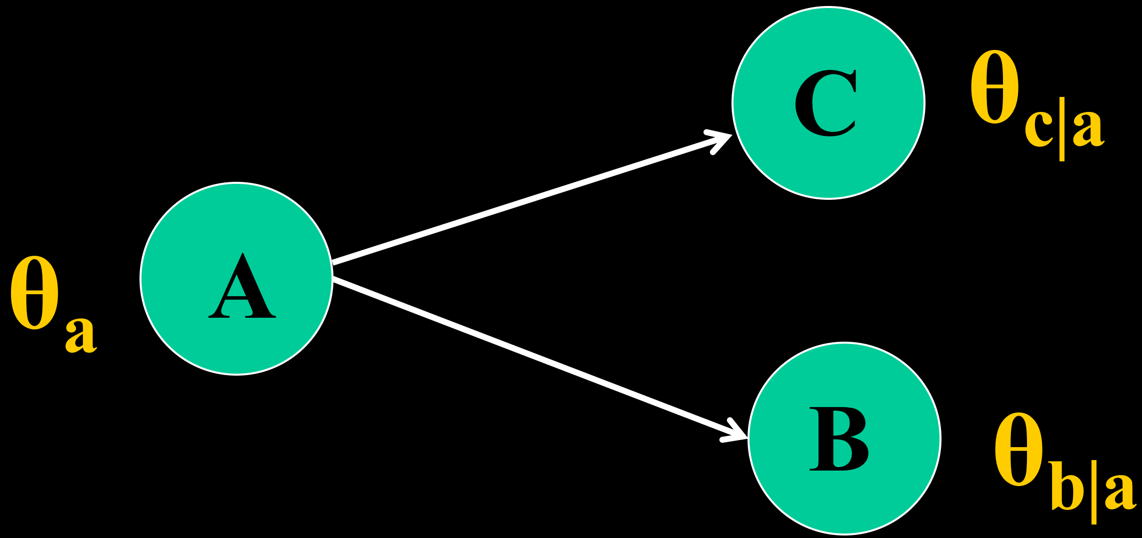
$= F(a,\sim b) = .27$

$Pr(a)$

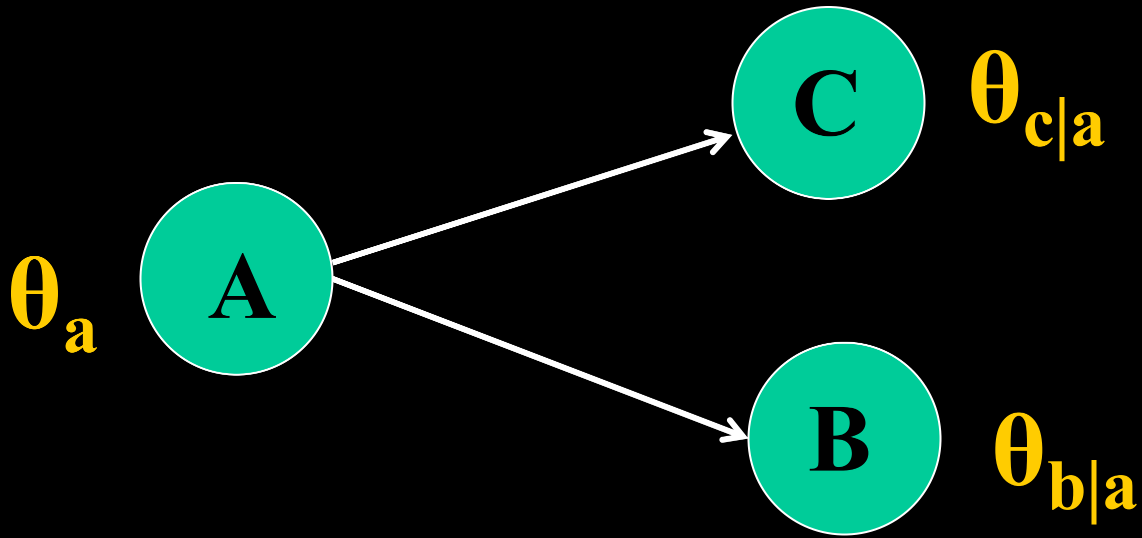
$= F(\lambda_{\sim a}:0, \lambda_{\sim b}:1, \lambda_a:1, \lambda_b:1)$

$= F(a) = .03+.27$

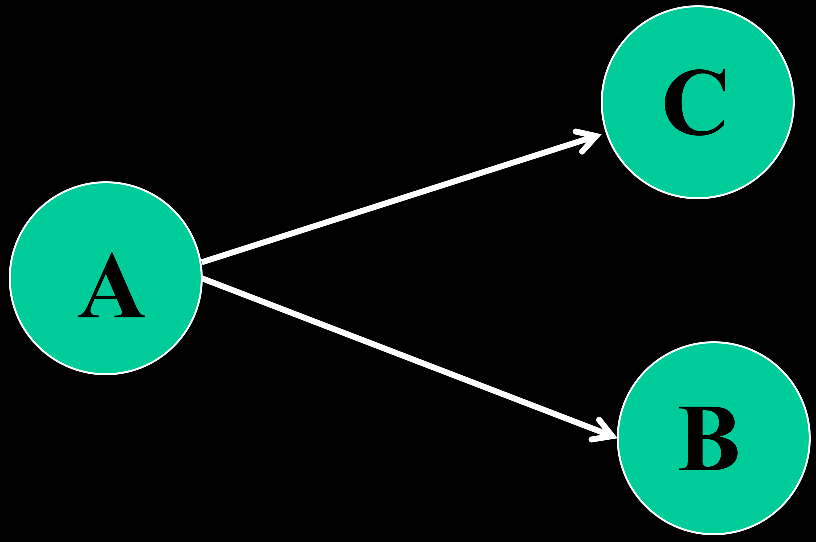
The Network Polynomial



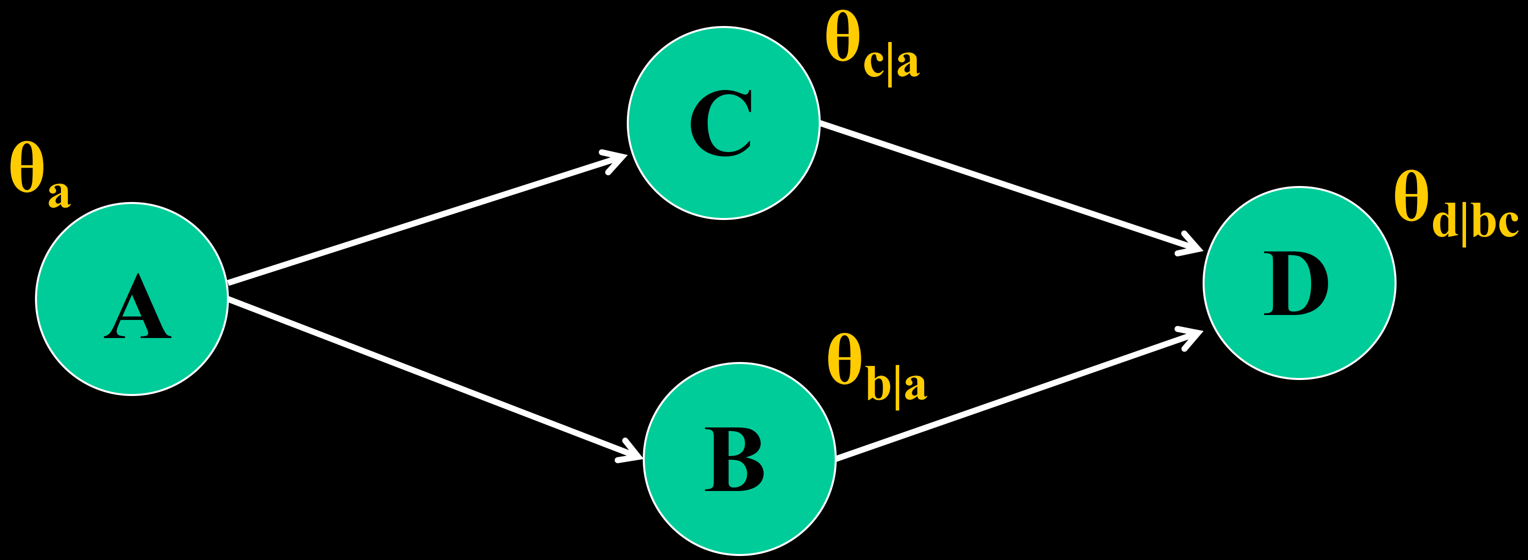
A	B	C	Pr(.)
a	b	c	$\theta_a\,\theta_{b|a}\,\theta_{c|a}$
a	b	~c	$\theta_a\,\theta_{b|a}\,\theta_{\sim c|a}$
a	~b	c	$\theta_a\,\theta_{\sim b|a}\,\theta_{c|a}$
a	~b	~c	$\theta_a\,\theta_{\sim b|a}\,\theta_{\sim c|a}$
.	.	.	…



A	B	C	Pr(.)
a	b	c	$\lambda_a \lambda_b \lambda_c \quad \theta_a \theta_{b\|a} \theta_{c\|a}$
a	b	~c	$\lambda_a \lambda_b \lambda_{\sim c} \quad \theta_a \theta_{b\|a} \theta_{\sim c\|a}$
a	~b	c	$\lambda_a \lambda_{\sim b} \lambda_c \quad \theta_a \theta_{\sim b\|a} \theta_{c\|a}$
a	~b	~c	$\lambda_a \lambda_{\sim b} \lambda_{\sim c} \quad \theta_a \theta_{\sim b\|a} \theta_{\sim c\|a}$
.	.	.	…



$$F = \lambda_a \lambda_b \lambda_c \quad \theta_a \theta_{b|a} \theta_{c|a} +$$
$$\lambda_a \lambda_b \lambda_{\sim c} \quad \theta_a \theta_{b|a} \theta_{\sim c|a} +$$
$$\lambda_a \lambda_{\sim b} \lambda_c \quad \theta_a \theta_{\sim b|a} \theta_{c|a} +$$
$$\lambda_a \lambda_{\sim b} \lambda_{\sim c} \quad \theta_a \theta_{\sim b|a} \theta_{\sim c|a}$$
$$\ldots.$$



$$F = \lambda_a \lambda_b \lambda_c \lambda_d \theta_a \theta_{b|a} \theta_{c|a} \theta_{d|bc} + \lambda_a \lambda_b \lambda_c \lambda_{\sim d} \theta_a \theta_{b|a} \theta_{c|a} \theta_{\sim d|bc} + \ldots$$

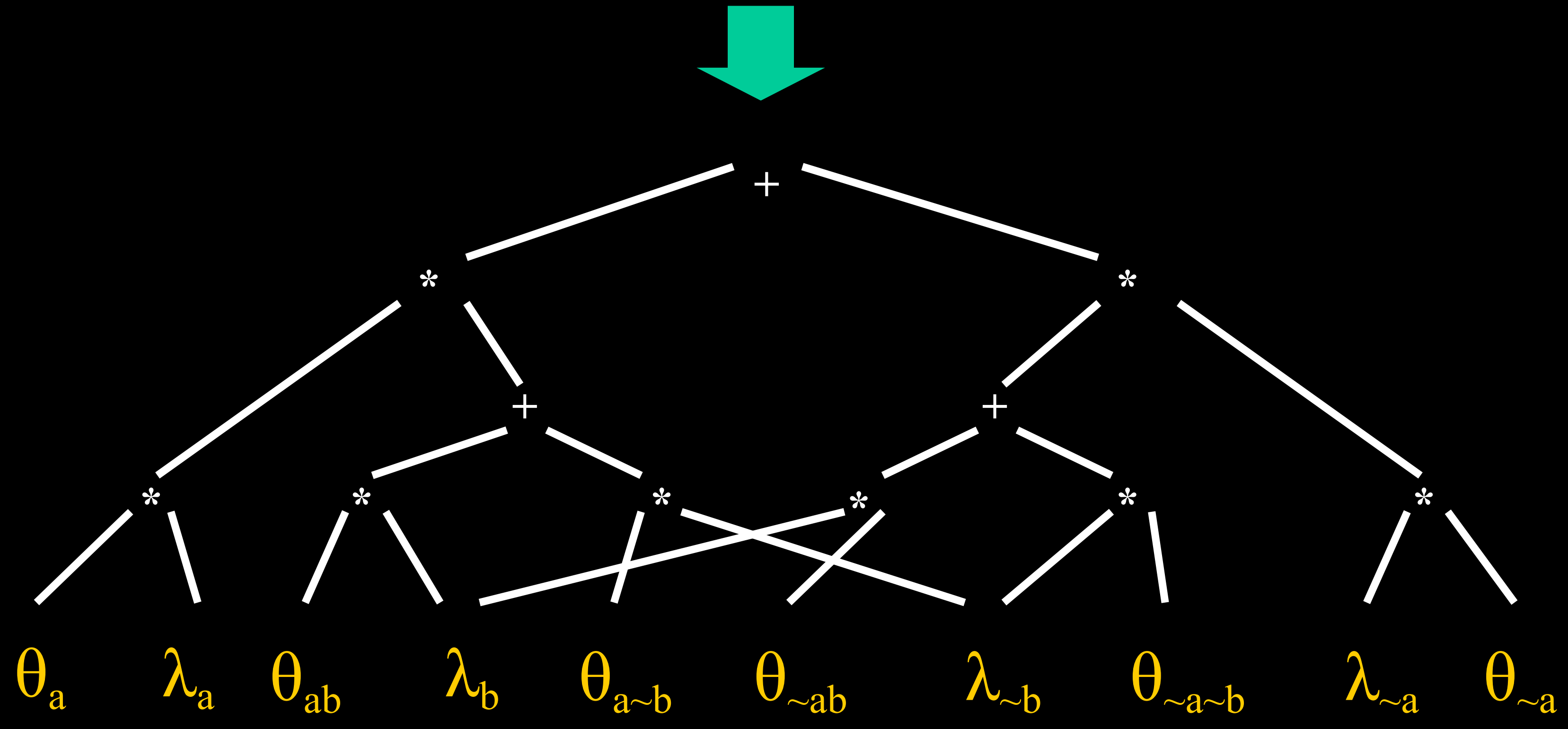
Each term has 2n variables (n indicators, n parameters)

Each variable has degree one (multi-linear function)

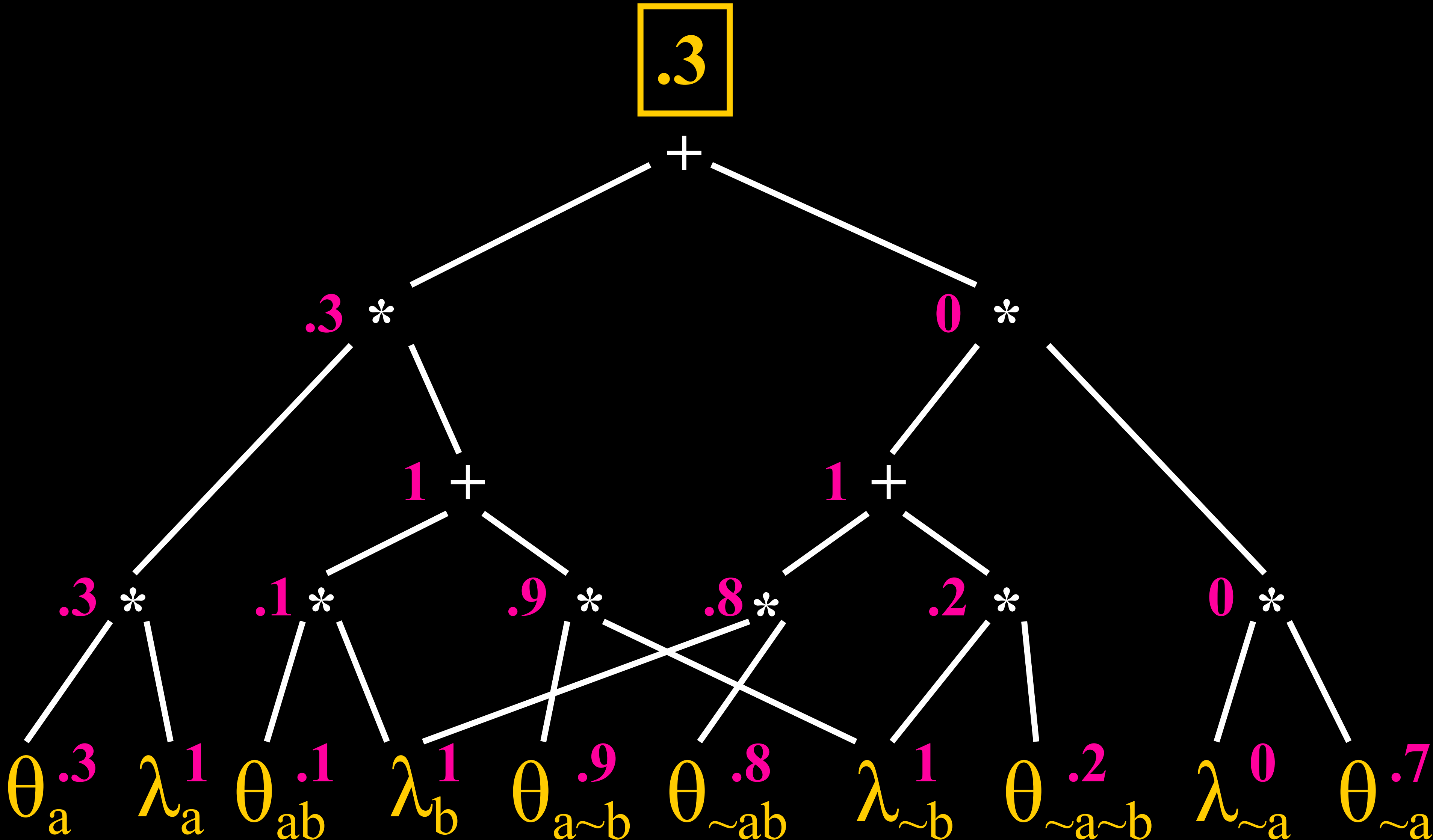
Factoring the Network Polynomial into an Arithmetic Circuit (AC)

Arithmetic Circuits

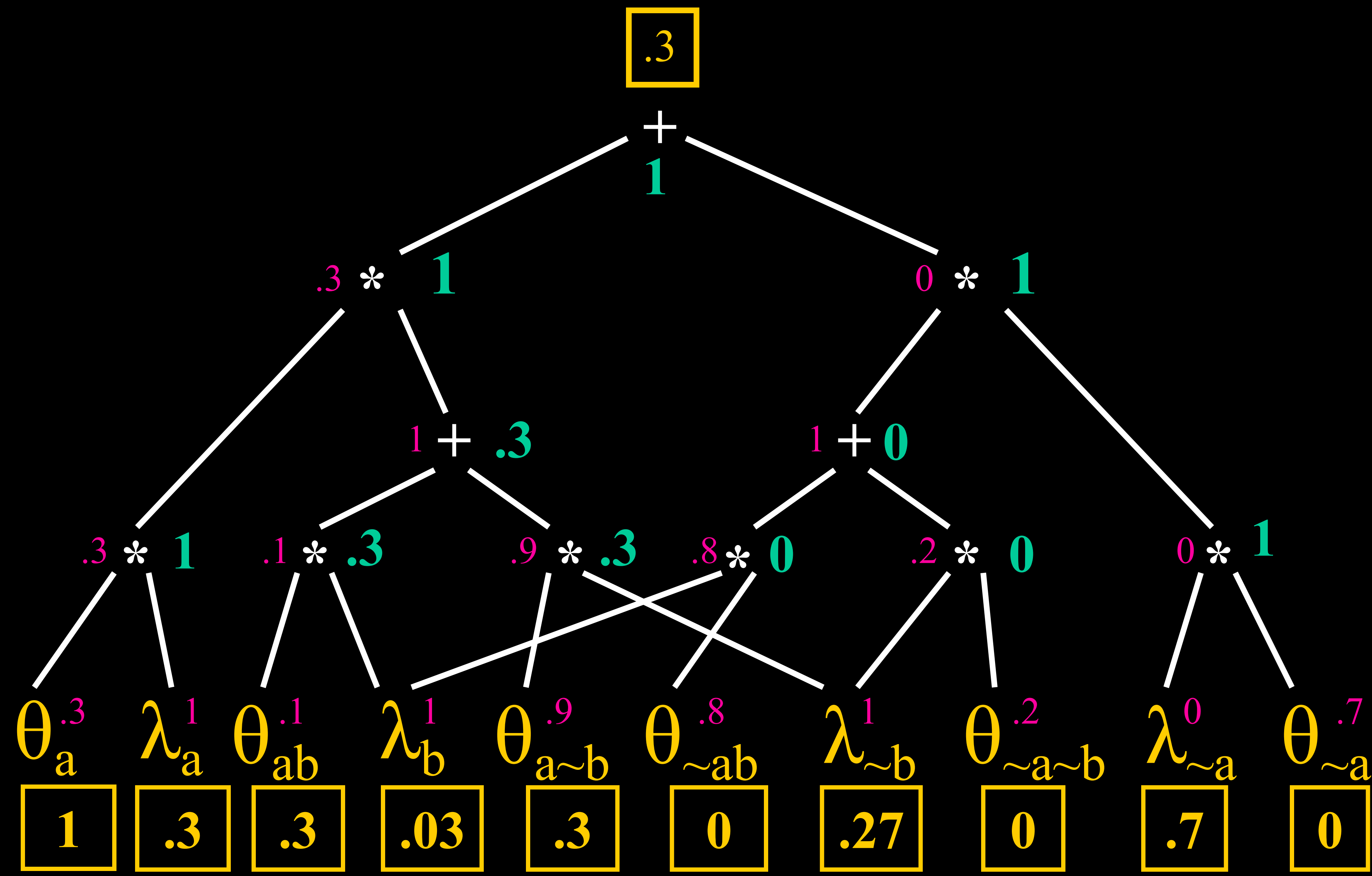
$$F = \lambda_a \lambda_b \theta_a \theta_{b|a} + \lambda_a \lambda_{\sim b} \theta_a \theta_{\sim b|a} + \lambda_{\sim a} \lambda_b \theta_{\sim a} \theta_{b|\sim a} + \lambda_{\sim a} \lambda_{\sim b} \theta_{\sim a} \theta_{\sim b|\sim a}$$



Evaluating and Differentiating Circuits



.3
+
.3 *
0 *
1 +
1 +
.3 *
.1 *
.9 *
.8 *
.2 *
0 *
θa .3
λa 1
θab .1
λb 1
θa~b .9
θ~ab .8
λ~b 1
θ~a~b .2
λ~a 0
θ~a .7



.3
+
1
.3 * 1
0 * 1
1 + .3
1 + 0
.3 * 1
.1 * .3
.9 * .3
.8 * 0
.2 * 0
0 * 1
$\theta_a^{.3}$
λ_a^{1}
$\theta_{ab}^{.1}$
λ_b^{1}
$\theta_{a\sim b}^{.9}$
$\theta_{\sim ab}^{.8}$
$\lambda_{\sim b}^{1}$
$\theta_{\sim a\sim b}^{.2}$
$\lambda_{\sim a}^{0}$
$\theta_{\sim a}^{.7}$
1
.3
.3
.03
.3
0
.27
0
.7
0

Differentiation Schemes

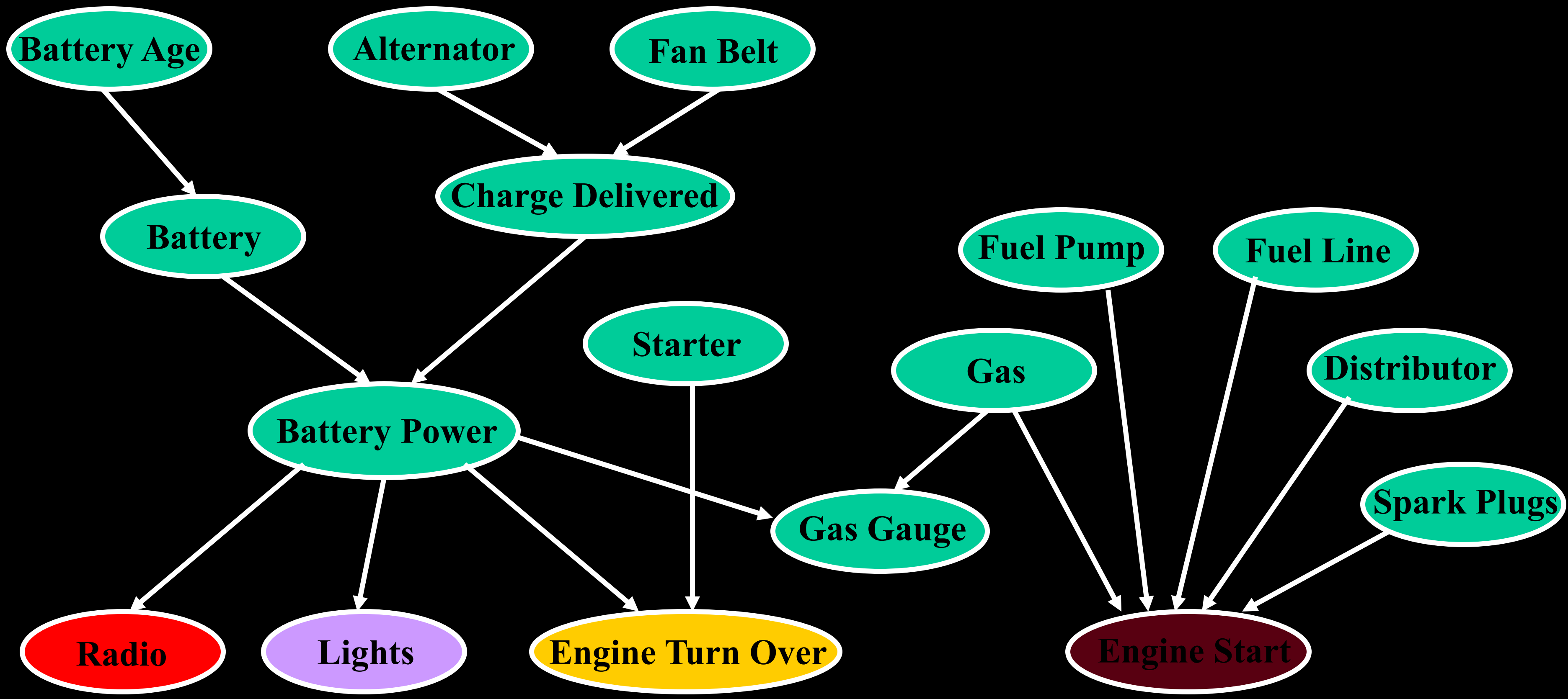
Assume alternating levels of +/* nodes, with one parent per *node

- <u>Method A:</u>
  Two registers per +node
- <u>Method B:</u>
  One register per node
- <u>Method C:</u>
  One register per node, two bits per *node

Notation
(Evaluating the Polynomial)

$$F(e) = F(\lambda = ..) = \Pr(e)$$

Probability of Evidence

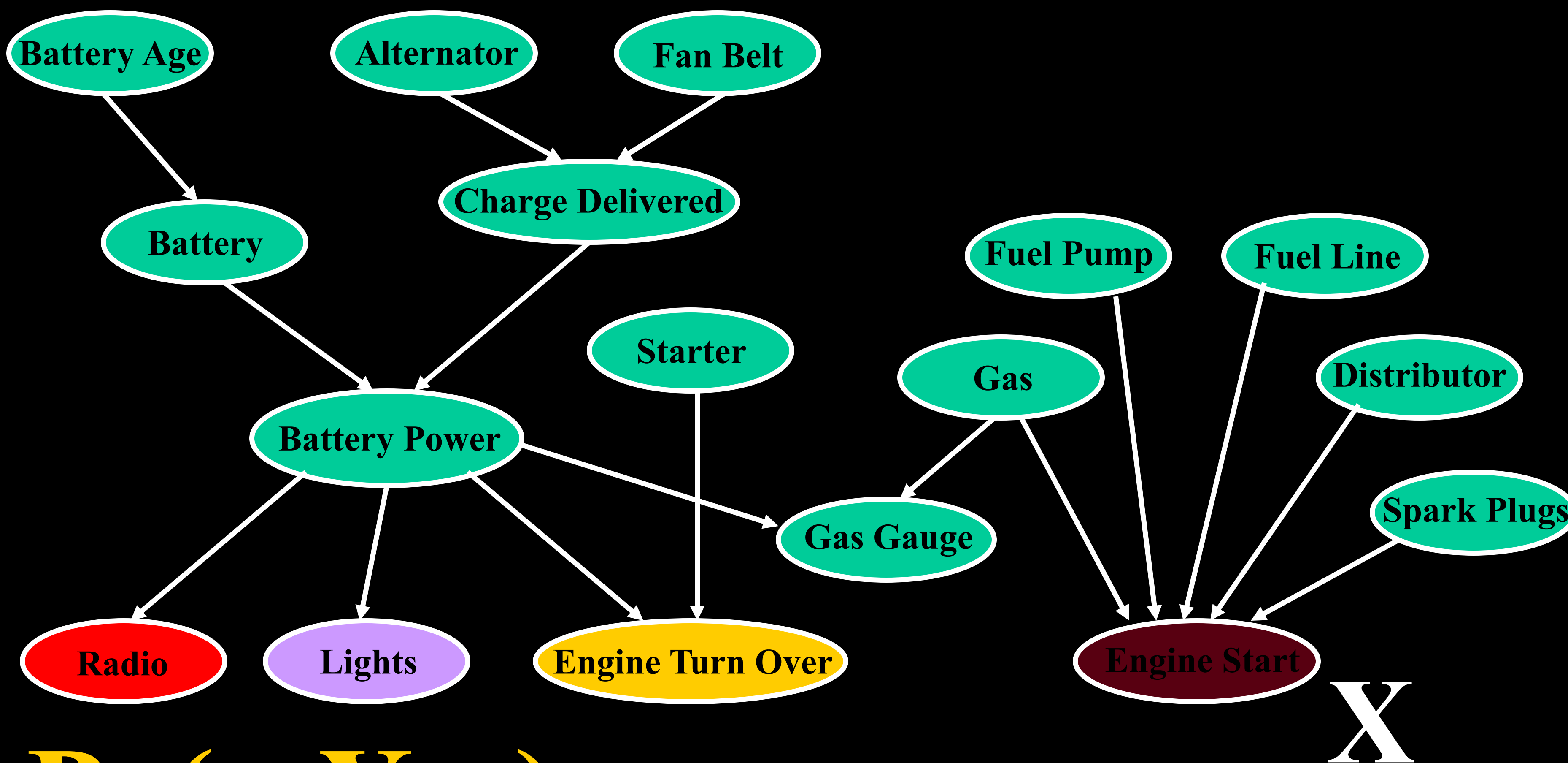


Pr(e)

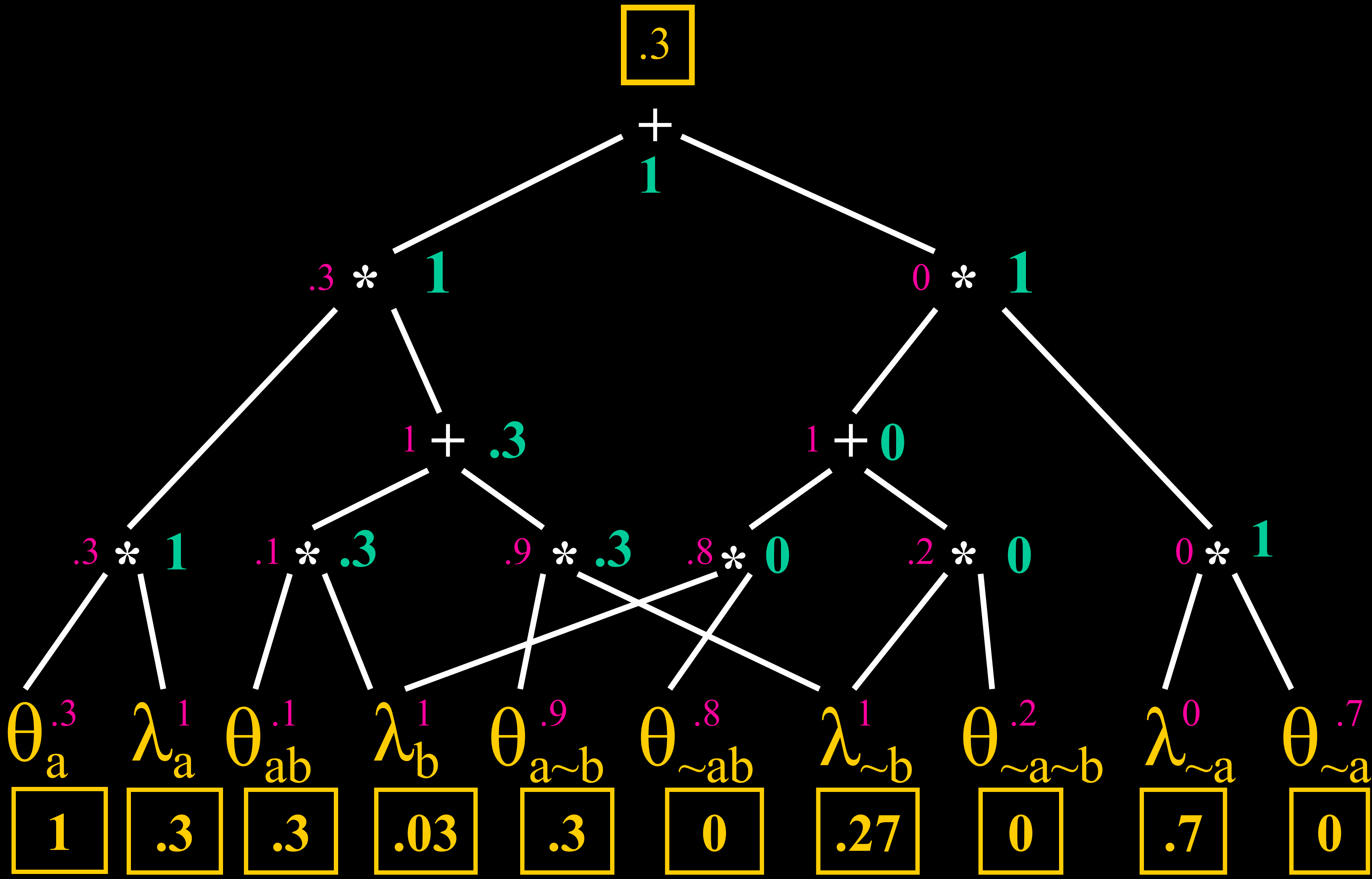
The Partial Derivatives

$$\frac{\partial F}{\partial \lambda_x}(e) = \Pr(e - X, x)$$

Probability of Evidence Flips



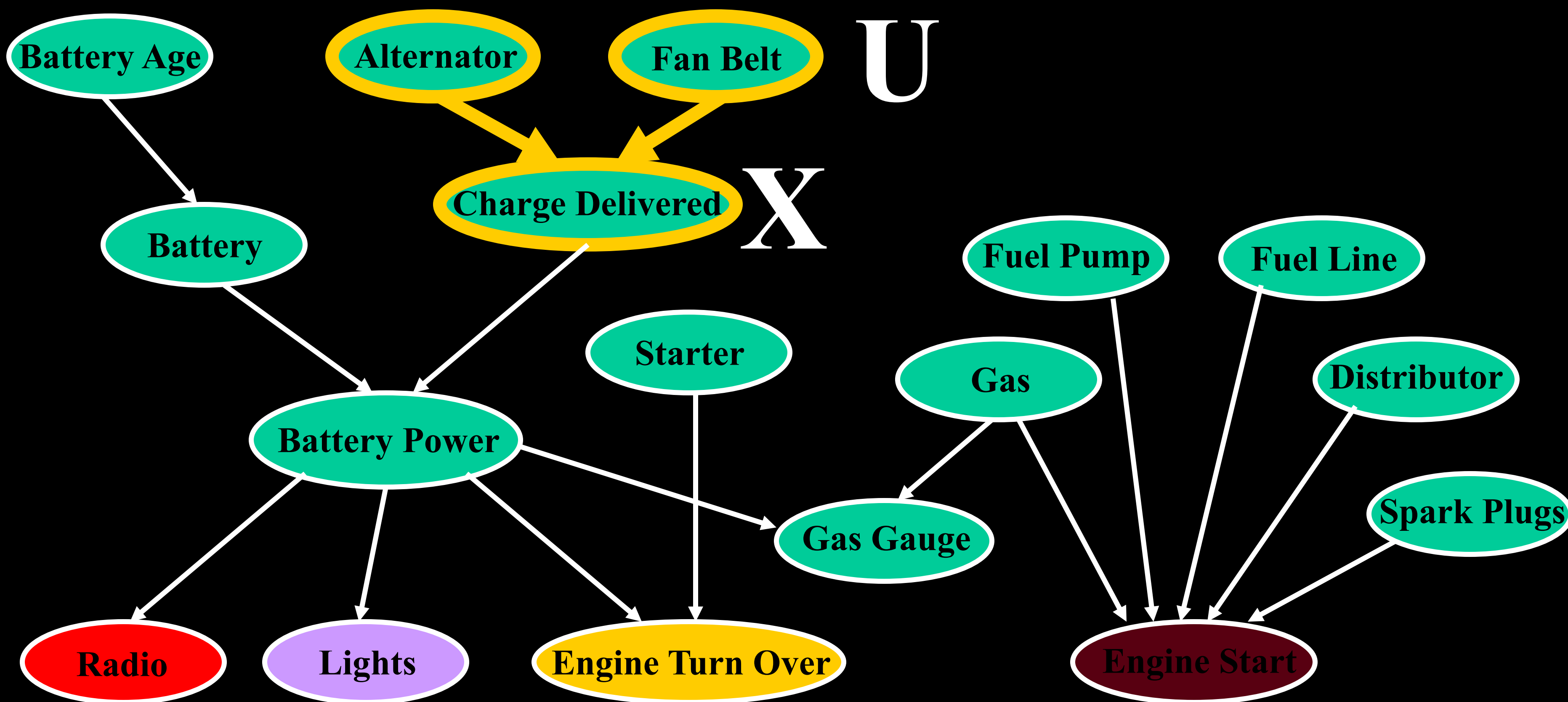
$Pr(e-X,x)$


.3
+
1
.3 * 1
0 * 1
1 + .3
1 + 0
.3 * 1
.1 * .3
.9 * .3
.8 * 0
.2 * 0
0 * 1
$\theta_a^{.3}$
λ_a^{1}
$\theta_{ab}^{.1}$
λ_b^{1}
$\theta_{a\sim b}^{.9}$
$\theta_{\sim ab}^{.8}$
$\lambda_{\sim b}^{1}$
$\theta_{\sim a\sim b}^{.2}$
$\lambda_{\sim a}^{0}$
$\theta_{\sim a}^{.7}$
1
.3
.3
.03
.3
0
.27
0
.7
0

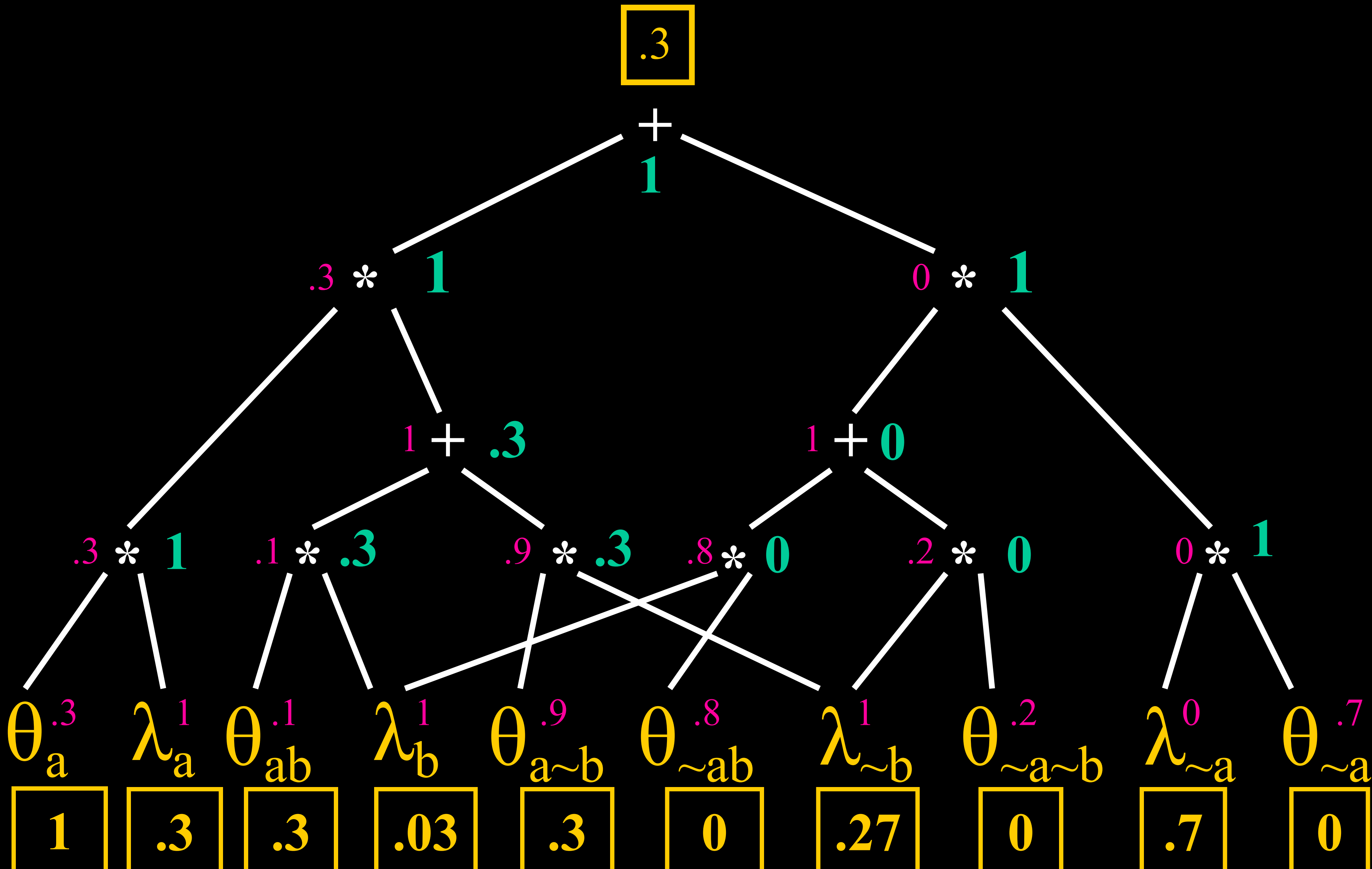
The Partial Derivatives

$$\theta_{x|u} \frac{\partial F}{\partial \theta_{x|u}}(e) = \Pr(e, x, u)$$

Family Marginals



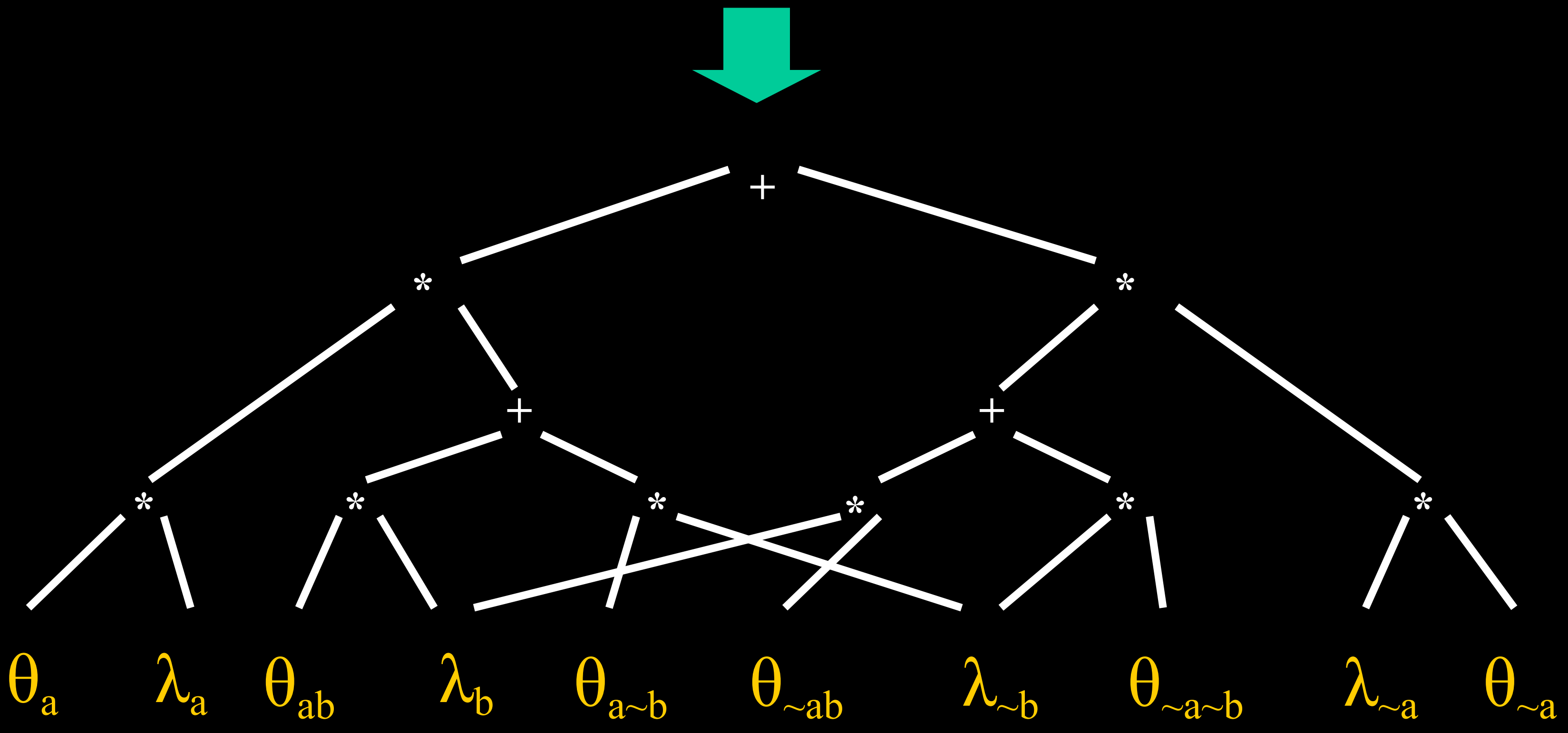
$Pr(e,x,u)$



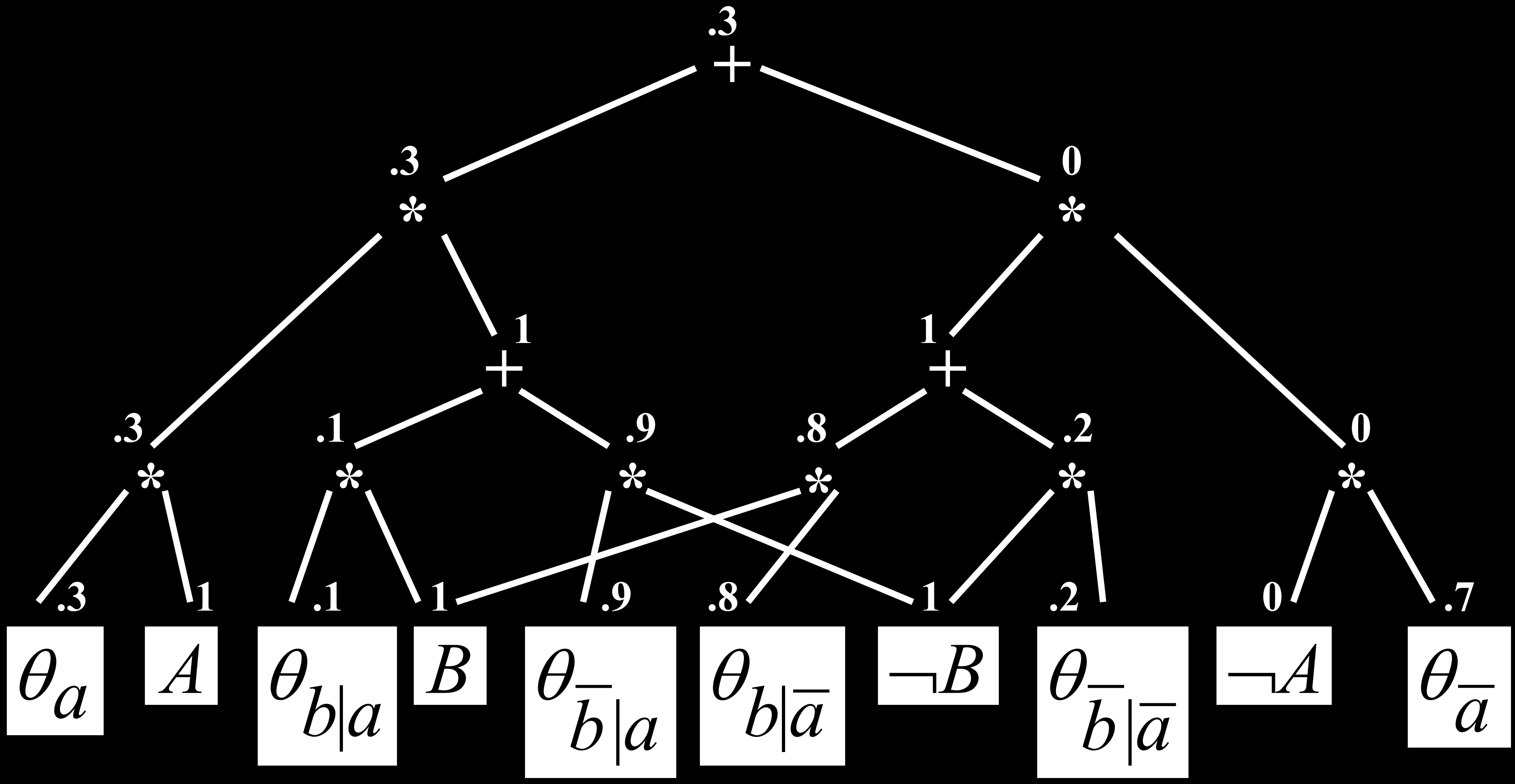
.3
+
1
.3 * 1
0 * 1
1 + .3
1 + 0
.3 * 1
.1 * .3
.9 * .3
.8 * 0
.2 * 0
0 * 1
$\theta_a^{.3}$ λ_a^{1} $\theta_{ab}^{.1}$ λ_b^{1} $\theta_{a\sim b}^{.9}$ $\theta_{\sim ab}^{.8}$ $\lambda_{\sim b}^{1}$ $\theta_{\sim a\sim b}^{.2}$ $\lambda_{\sim a}^{0}$ $\theta_{\sim a}^{.7}$
1 .3 .3 .03 .3 0 .27 0 .7 0

Polynomials → Arithmetic Circuits

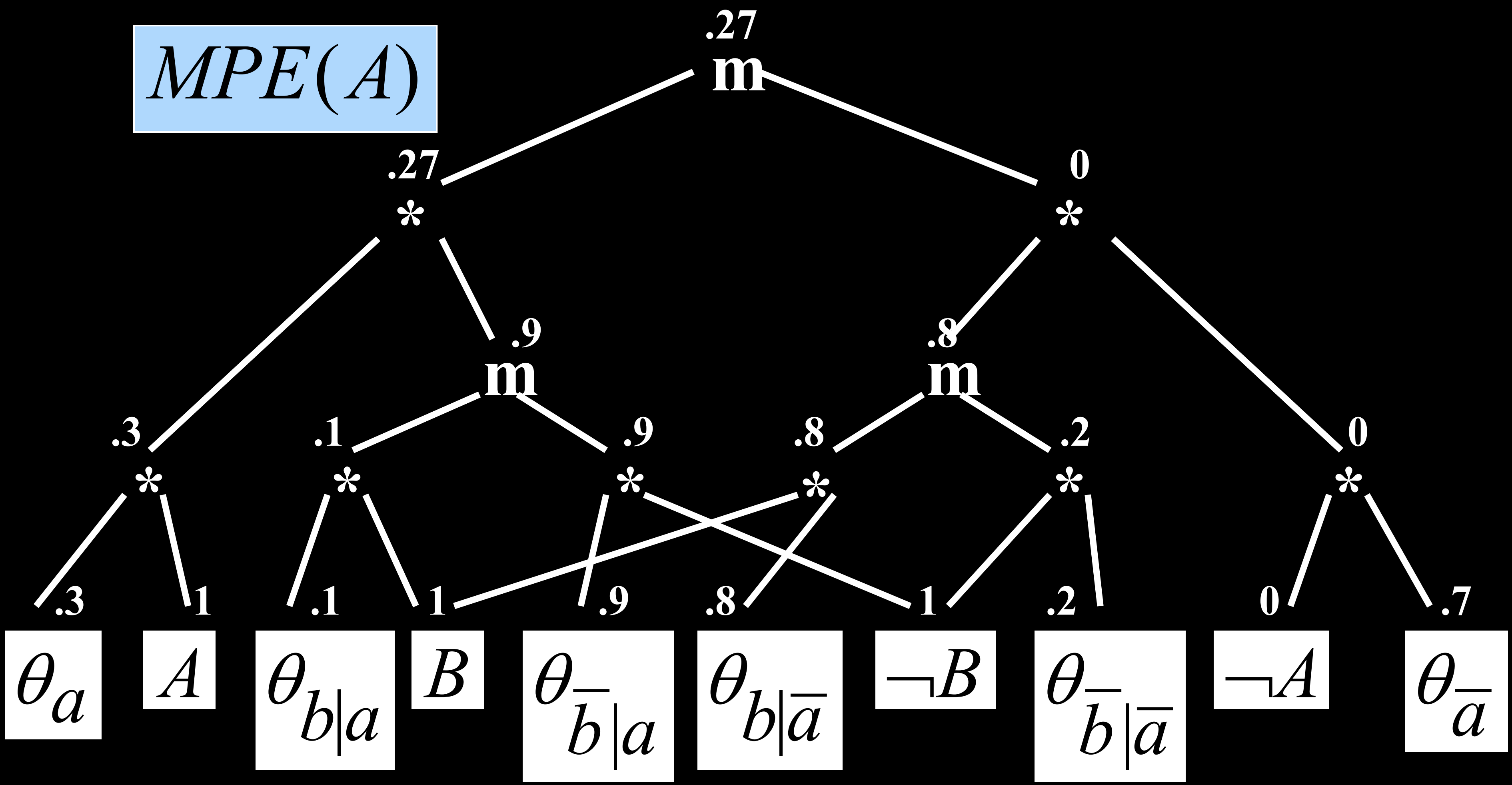
$$F = \lambda_a \lambda_b \theta_a \theta_{b|a} + \lambda_a \lambda_{\sim b} \theta_a \theta_{\sim b|a} + \lambda_{\sim a} \lambda_b \theta_{\sim a} \theta_{b|\sim a} + \lambda_{\sim a} \lambda_{\sim b} \theta_{\sim a} \theta_{\sim b|\sim a}$$



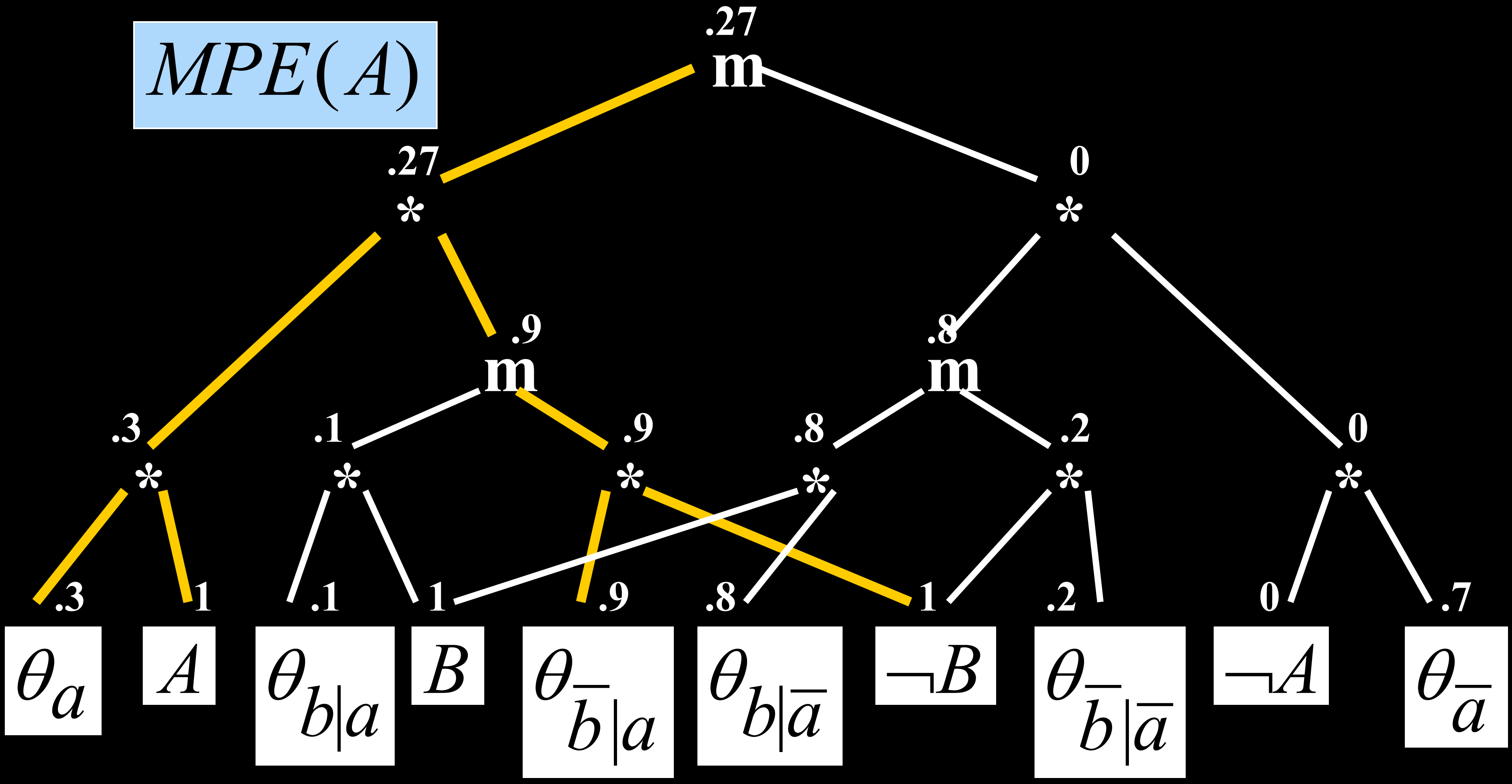
Circuit Optimization: MPE



Circuit Optimization: MPE



Circuit Optimization: MPE



Compiling Arithmetic Circuits using Variable Elimination

A	$Ø_A$
true	.3
false	.7

A	B	$Ø_B$
true	true	.1
true	false	.9
false	true	.8
false	false	.2

A	$\O_A$
true	$\theta_a\lambda_a$
false	$\theta_{\sim a}\lambda_{\sim a}$

A	B	$\O_B$
true	true	$\theta_{ab}\lambda_b$
true	false	$\theta_{a\sim b}\lambda_{\sim b}$
false	true	$\theta_{\sim ab}\lambda_b$
false	false	$\theta_{\sim a\sim b}\lambda_{\sim b}$

Eliminating B

A	B	$Ø_B$
true	true	$\theta_{ab}\lambda_b$
true	false	$\theta_{a\sim b}\lambda_{\sim b}$
false	true	$\theta_{\sim ab}\lambda_b$
false	false	$\theta_{\sim a\sim b}\lambda_{\sim b}$

Summing out B →

A	Sum_out($Ø_B$, B)
true	$\theta_{ab}\lambda_b + \theta_{a\sim b}\lambda_{\sim b}$
false	$\theta_{\sim ab}\lambda_b + \theta_{\sim a\sim b}\lambda_{\sim b}$

Eliminating A

Multiply tables containing A

A	$Ø_A$
true	$\theta_a \lambda_a$
false	$\theta_{\sim a} \lambda_{\sim a}$

*

A	Sum_out($Ø_B$, B)
true	$\theta_{ab} \lambda_b + \theta_{a\sim b} \lambda_{\sim b}$
false	$\theta_{\sim ab} \lambda_b + \theta_{\sim a\sim b} \lambda_{\sim b}$

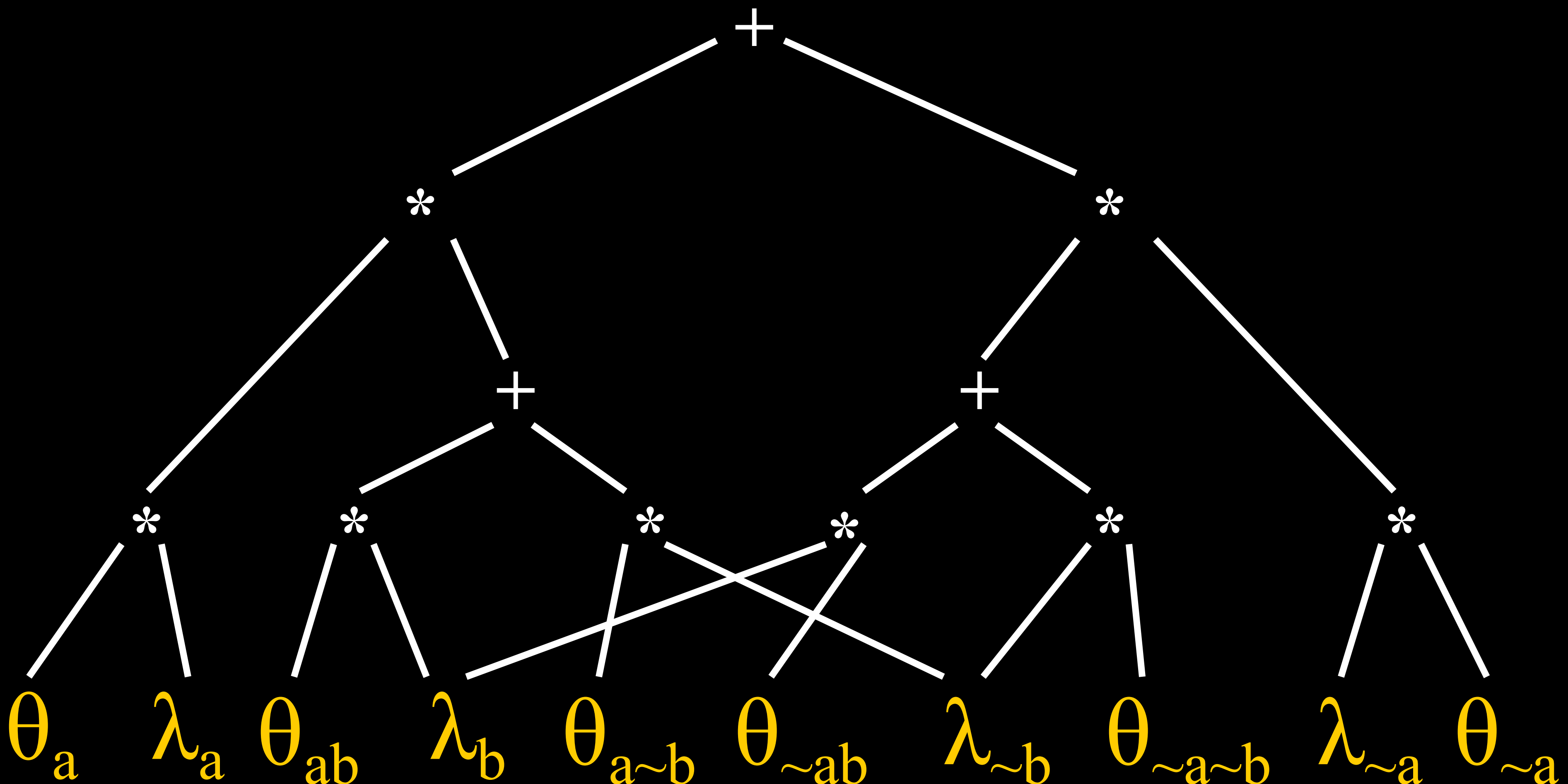
=

A	$Ø_A$Sum_out($Ø_B$, B)
true	$\theta_a \lambda_a (\theta_{ab} \lambda_b + \theta_{a\sim b} \lambda_{\sim b})$
false	$\theta_{\sim a} \lambda_{\sim a} (\theta_{\sim ab} \lambda_b + \theta_{\sim a\sim b} \lambda_{\sim b})$

A	$\text{Ø}_A \text{Sum\_out}(\text{Ø}_B, B)$
true	$\theta_a \lambda_a (\theta_{ab} \lambda_b + \theta_{a\sim b} \lambda_{\sim b})$
false	$\theta_{\sim a} \lambda_{\sim a} (\theta_{\sim ab} \lambda_b + \theta_{\sim a\sim b} \lambda_{\sim b})$

Summing out A

$\text{Sum\_out}(\text{Ø}_A \text{Sum\_out}(\text{Ø}_B, B), A)$
$\theta_a \lambda_a (\theta_{ab} \lambda_b + \theta_{a\sim b} \lambda_{\sim b}) +$ $\theta_{\sim a} \lambda_{\sim a} (\theta_{\sim ab} \lambda_b + \theta_{\sim a\sim b} \lambda_{\sim b})$



$$\theta_a\lambda_a(\theta_{ab}\lambda_b+\theta_{a\sim b}\lambda_{\sim b}) + \theta_{\sim a}\lambda_{\sim a}(\theta_{\sim ab}\lambda_b+\theta_{\sim a\sim b}\lambda_{\sim b})$$

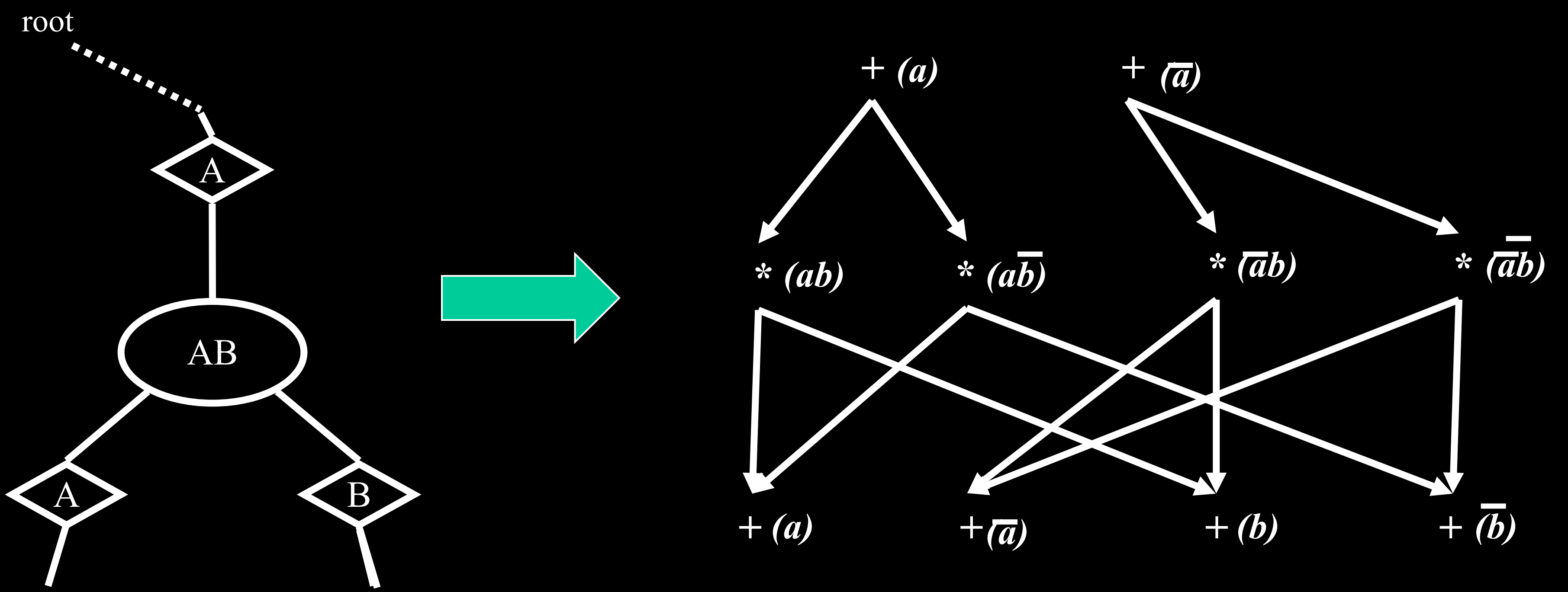
Optional Reading

An Advance on Variable Elimination with Applications to Tensor-Based Computation

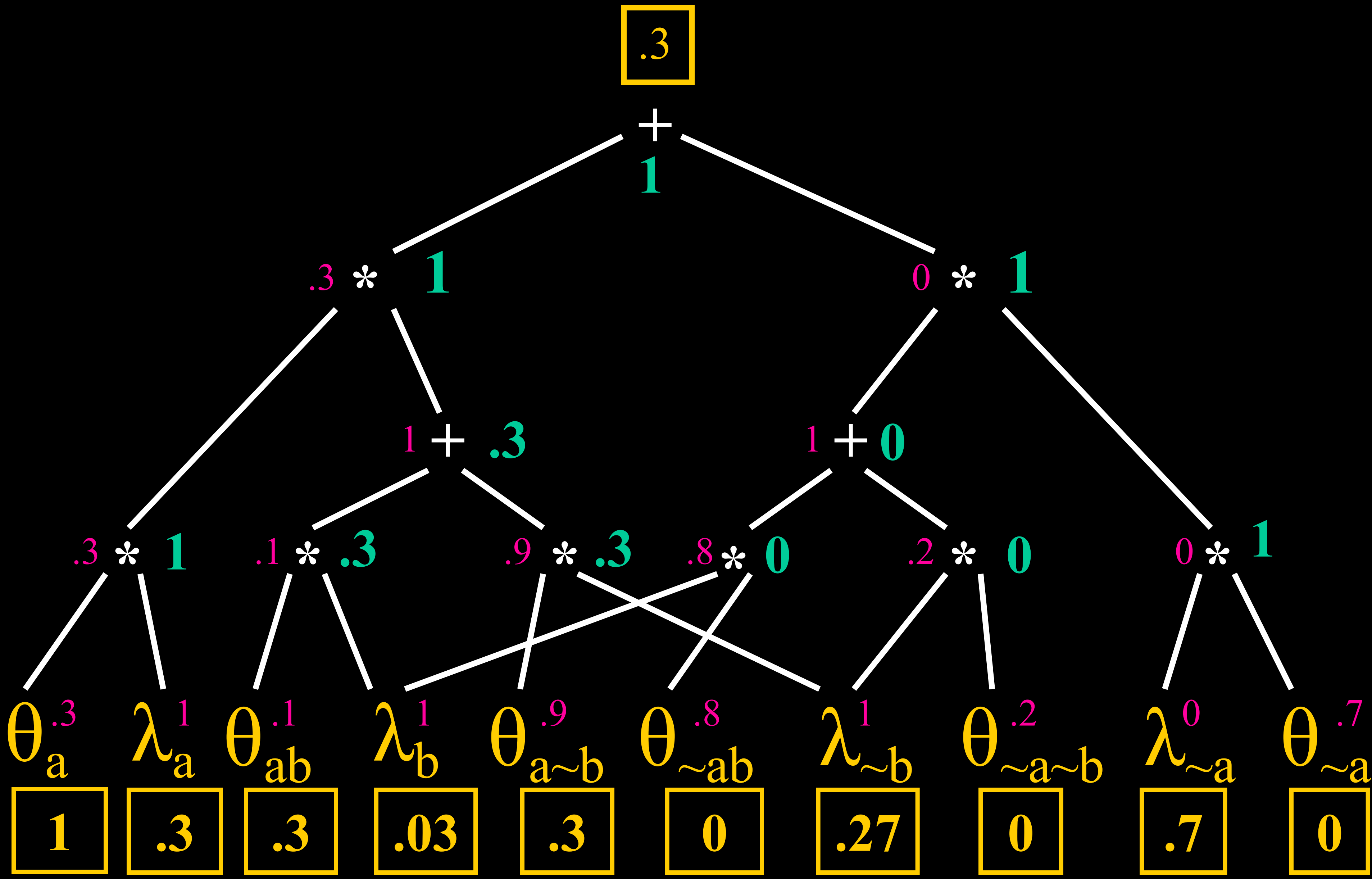
New VE Algorithm that exploits functional dependencies & outputs an AC in the form of a Tensor Graph

Extracting
Arithmetic Circuits from
Jointrees

A Jointree is an Arithmetic Circuit



Inward-pass evaluates circuit
Outward-pass differentiates circuit



.3
+
1
.3 * 1
0 * 1
1 + .3
1 + 0
.3 * 1
.1 * .3
.9 * .3
.8 * 0
.2 * 0
0 * 1
$\theta_a^{.3}$
λ_a^{1}
$\theta_{ab}^{.1}$
λ_b^{1}
$\theta_{a\sim b}^{.9}$
$\theta_{\sim ab}^{.8}$
$\lambda_{\sim b}^{1}$
$\theta_{\sim a\sim b}^{.2}$
$\lambda_{\sim a}^{0}$
$\theta_{\sim a}^{.7}$
1
.3
.3
.03
.3
0
.27
0
.7
0

Differentiation Schemes

Assume alternating levels of +/* nodes, with one parent per *node

- <u>Method A:</u>
  Two registers per +node
- <u>Method B:</u>
  One register per node
- <u>Method C:</u>
  One register per node, two bits per *node

Jointree Flavors

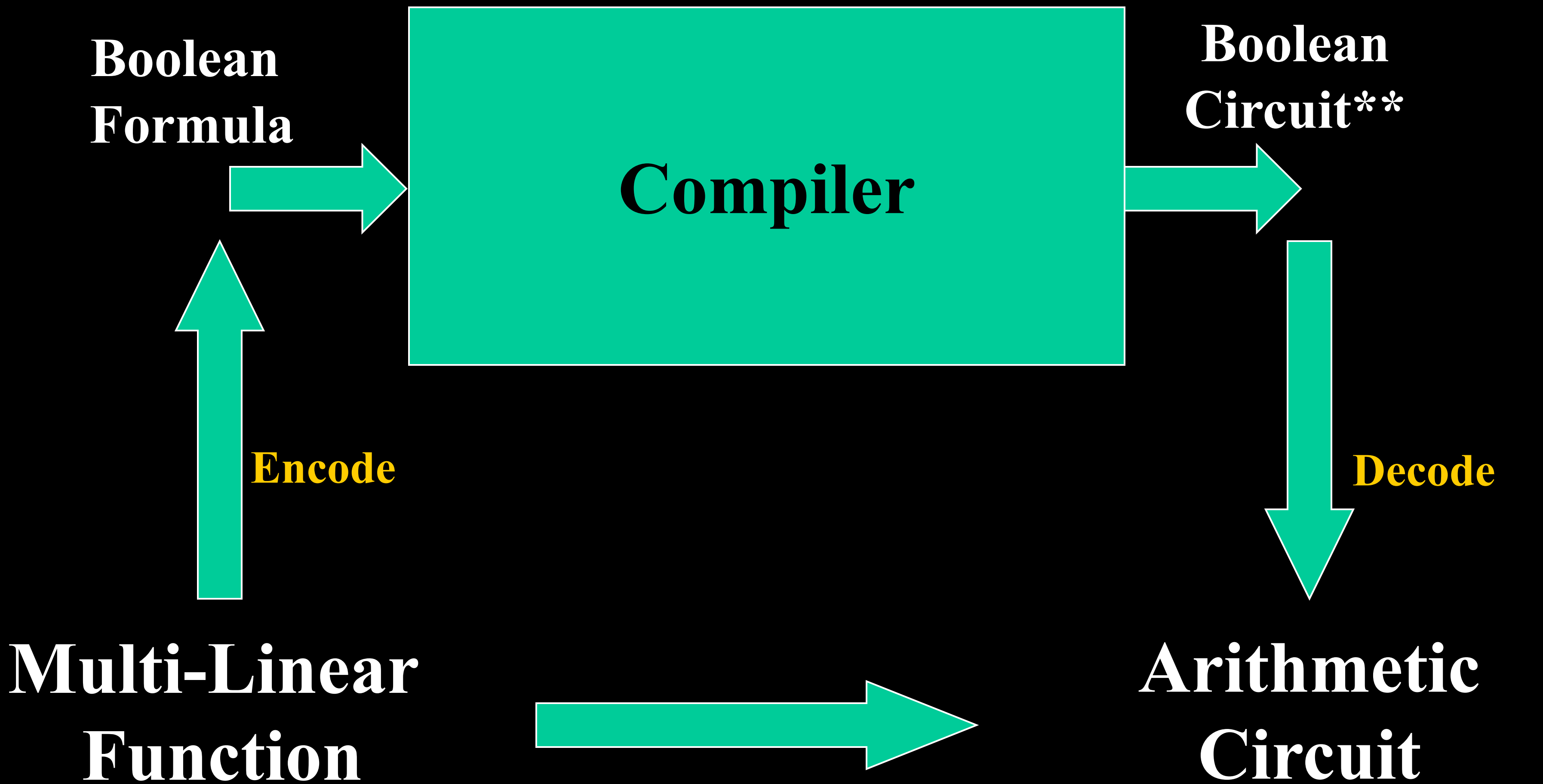
- **Shenoy-Shafer:**
  Method A **for differentiating circuits**

- **Hugin:**
  Method B **for differentiating circuits**

- **Zero-Conscious Hugin:**
  Method C **for differentiating circuits**

Compiling
Arithmetic Circuits
by Reduction to Logic

Factoring Multi-Linear Functions



Boolean Formula

$C \wedge (A \vee \neg B)$

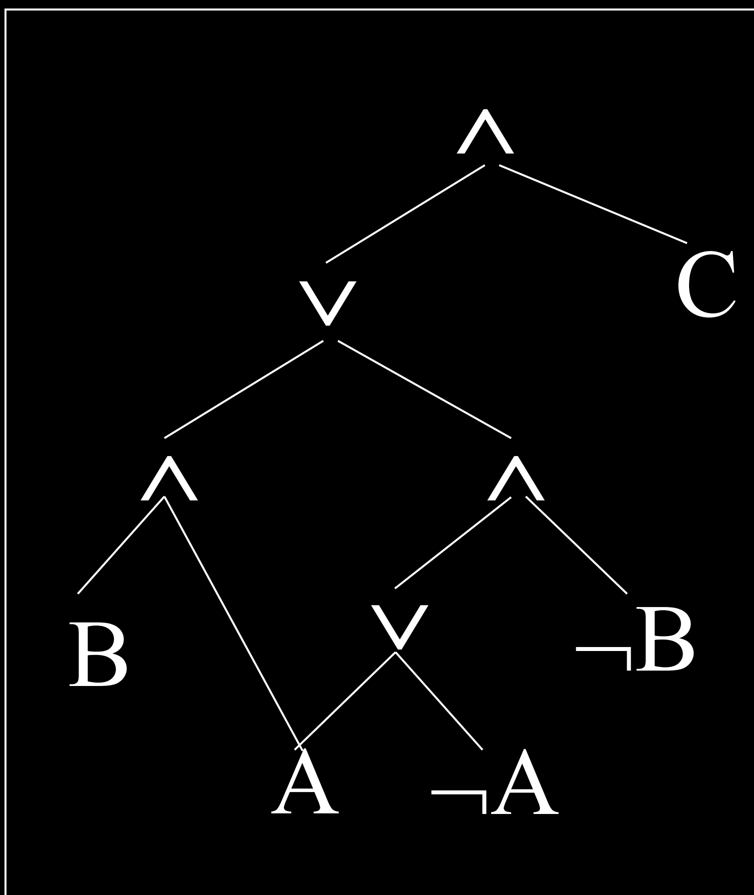
Multi-linear function

$a\,c + a\,b\,c + c$

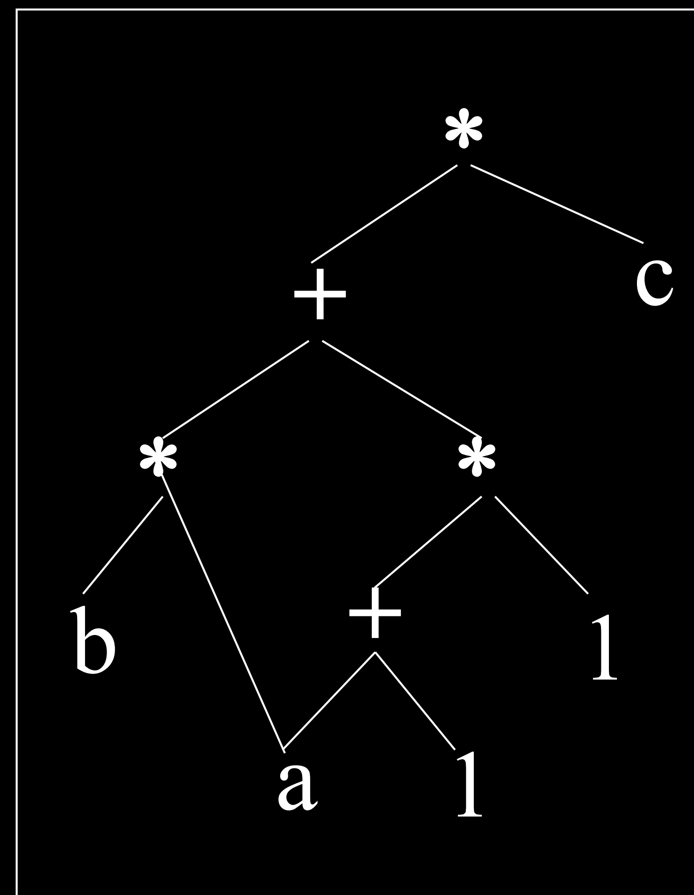
Encode ←

Compile ↓

Decode →



Boolean Circuit (sd-DNNF)



Arithmetic Circuit

Boolean Encoding of Multi-Linear Functions

Boolean Formula:

$\Delta = C \wedge (A \vee \neg B)$

Encodes:

$F = a\,c + a\,b\,c + c$

A	B	C	Encodes term
T	T	T	abc
T	T	F	ab
T	F	T	ac
T	F	F	a
F	T	T	bc
F	T	F	b
F	F	T	c
F	F	F	1

Boolean Formula

$C \wedge (A \vee \neg B)$

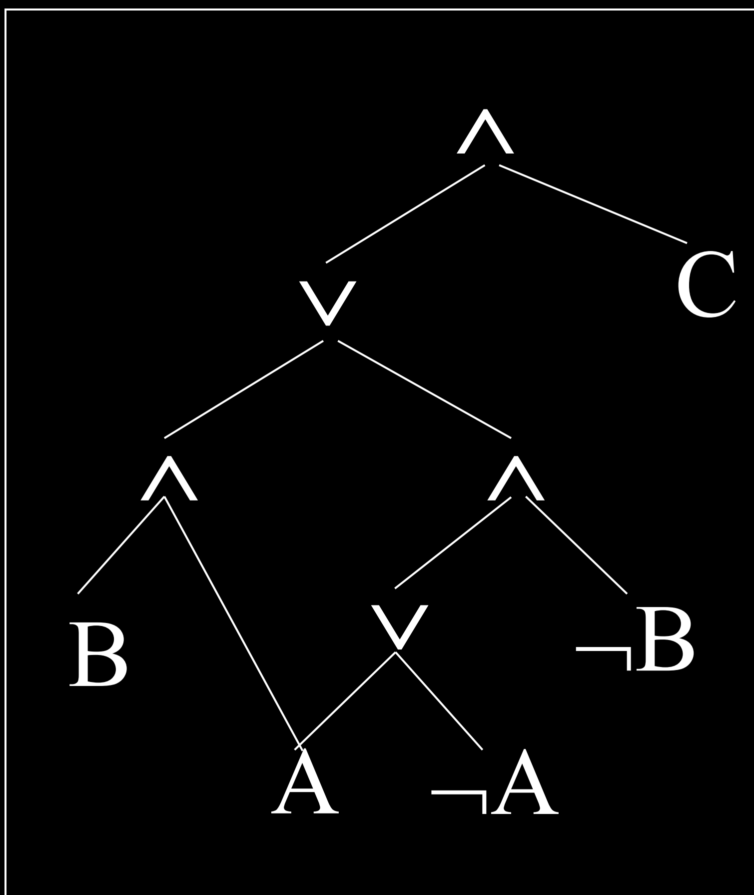
Multi-linear function

$a\,c + a\,b\,c + c$

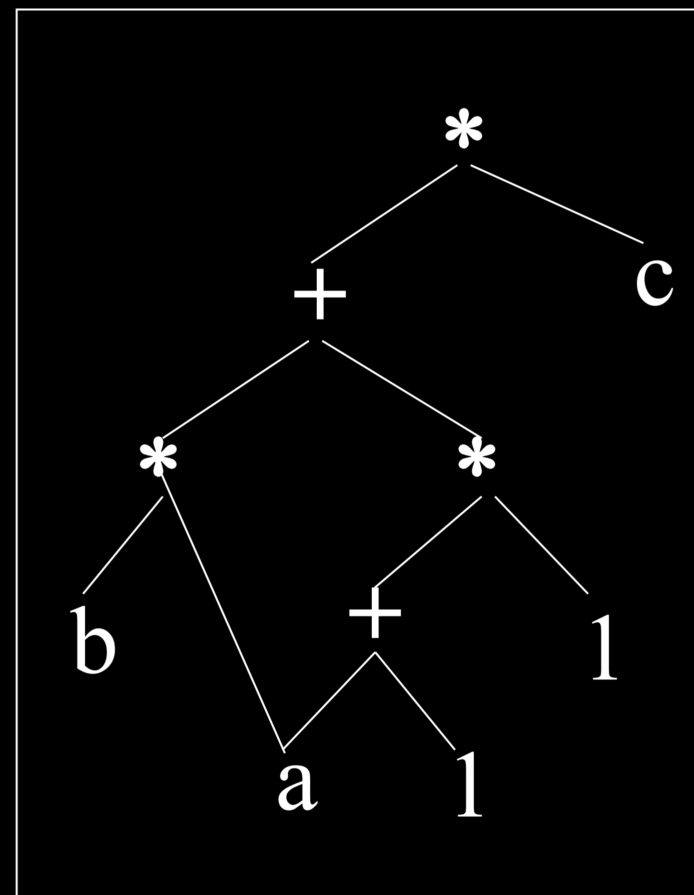
Encode

Compile

Decode



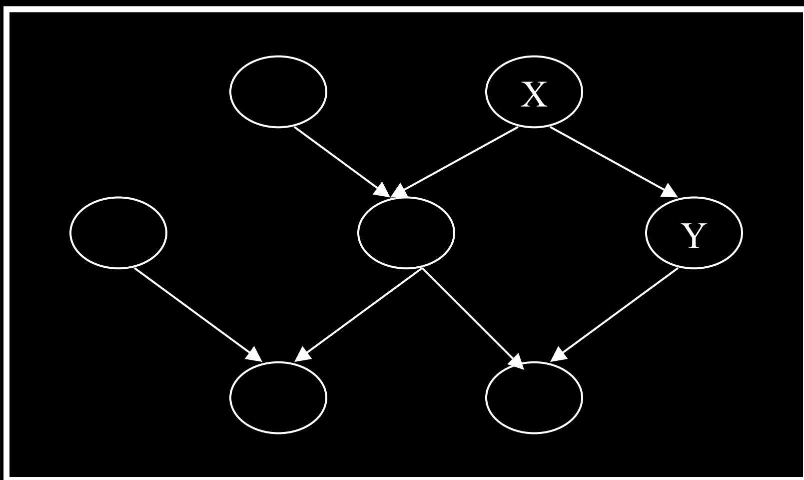
Boolean Circuit



Arithmetic Circuit

Compiling Networks

Bayesian network



Boolean Formula

$\lambda_x \vee \lambda_{\neg x}$

$\lambda_x \rightarrow \lambda_{\neg x}$

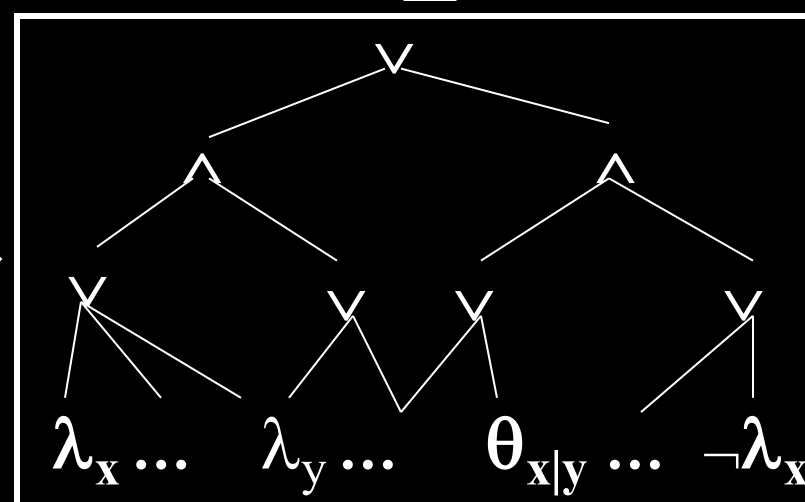
$\lambda_x \wedge \lambda_y \rightarrow \theta_{x|y}$

…

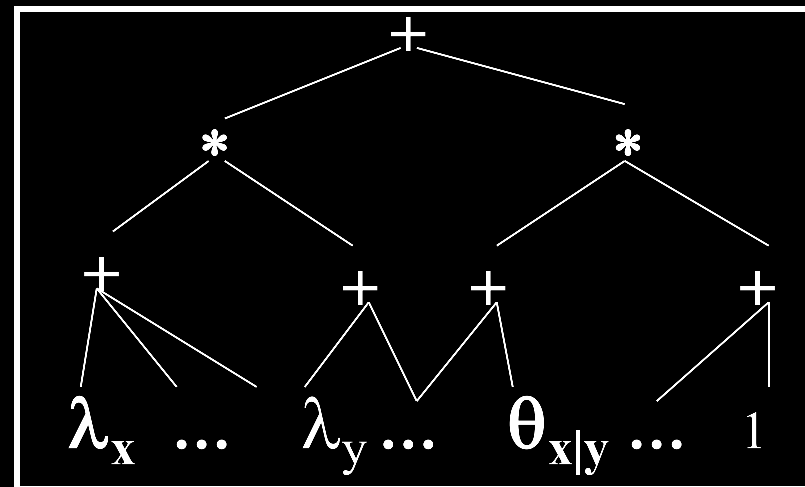
….

….

Boolean Circuit



Arithmetic Circuit



A	$Ø_A$
true	.3
false	.7

A	B	$Ø_B$
true	true	.1
true	false	.9
false	true	.8
false	false	.2

Boolean Encoding of Bayesian Networks

$$F = \lambda_a \lambda_b \theta_a \theta_{b|a} + \lambda_a \lambda_{\sim b} \theta_a \theta_{\sim b|a} + \lambda_{\sim a} \lambda_b \theta_{\sim a} \theta_{b|\sim a} + \lambda_{\sim a} \lambda_{\sim b} \theta_{\sim a} \theta_{\sim b|\sim a}$$

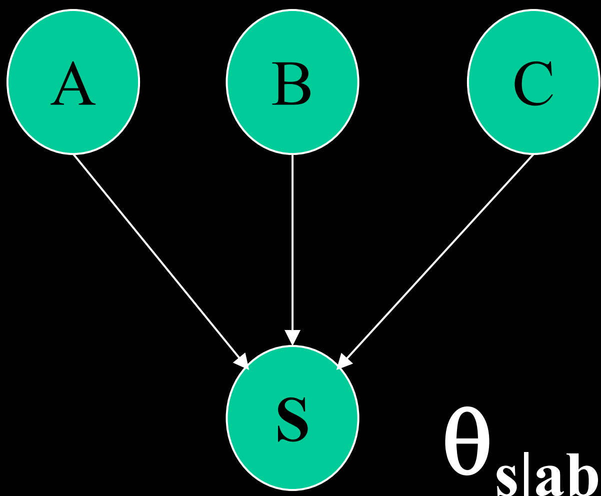
$\lambda_a \vee \lambda_{\sim a}$ $\qquad$ $\neg\lambda_a \vee \neg\lambda_{\sim a}$

$\lambda_b \vee \lambda_{\sim b}$ $\qquad$ $\neg\lambda_b \vee \neg\lambda_{\sim b}$

$\lambda_a \leftrightarrow \theta_a$ $\qquad$ $\lambda_{\sim a} \leftrightarrow \theta_{\sim a}$

$\lambda_a \wedge \lambda_b \leftrightarrow \theta_{b|a}$ $\qquad$ $\lambda_a \wedge \lambda_{\sim b} \leftrightarrow \theta_{\sim b|a}$ $\qquad$ $\lambda_{\sim a} \wedge \lambda_b \leftrightarrow \theta_{b|\sim a}$ $\qquad$ $\lambda_{\sim a} \wedge \lambda_{\sim b} \leftrightarrow \theta_{\sim b|\sim a}$

Local CPT Structure



$\theta_{s|abe}$

A	B	C	Pr(S|A,B,C)
a	b	c	0.95
a	b	$\bar{c}$	0.95
a	$\bar{b}$	c	0.20
a	$\bar{b}$	$\bar{c}$	0.05
$\bar{a}$	b	c	0.00
$\bar{a}$	b	$\bar{c}$	0.00
$\bar{a}$	$\bar{b}$	c	0.00
$\bar{a}$	$\bar{b}$	$\bar{c}$	0.00

Tabular CPT

-Functional constraints

-Context-specific independence

Functional Constraints

A	B	C	Pr(S|A,B,E)
a	b	c	0.95
a	b	$\bar{c}$	0.95
a	$\bar{b}$	c	0.20
a	$\bar{b}$	$\bar{c}$	0.05
$\bar{a}$	b	c	0.00
$\bar{a}$	b	$\bar{c}$	0.00
$\bar{a}$	$\bar{b}$	c	0.00
$\bar{a}$	$\bar{b}$	$\bar{c}$	0.00

Tabular CPT

~~$\lambda_{\sim a} \wedge \lambda_b \wedge \lambda_c \wedge \lambda_s \leftrightarrow \theta_{s|\sim abc}$~~

$\neg \lambda_{\sim a} \vee \neg \lambda_b \vee \neg \lambda_c \vee \neg \lambda_s$

Context-Specific Independence

A	B	C	Pr(S|A,B,C)
a	b	c	0.95
a	b	$\bar{c}$	0.95
a	$\bar{b}$	c	0.20
a	$\bar{b}$	$\bar{c}$	0.05
$\bar{a}$	b	c	0.00
$\bar{a}$	b	$\bar{c}$	0.00
$\bar{a}$	$\bar{b}$	c	0.00
$\bar{a}$	$\bar{b}$	$\bar{c}$	0.00

Tabular CPT

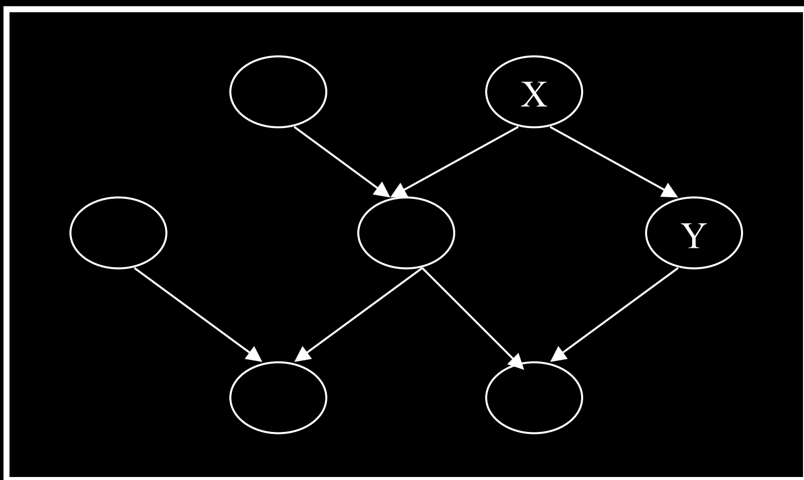
~~$\lambda_a \wedge \lambda_b \wedge \lambda_c \wedge \lambda_s \leftrightarrow \theta_{s|abc}$~~

~~$\lambda_a \wedge \lambda_b \wedge \lambda_{\sim c} \wedge \lambda_s \leftrightarrow \theta_{s|ab\sim c}$~~

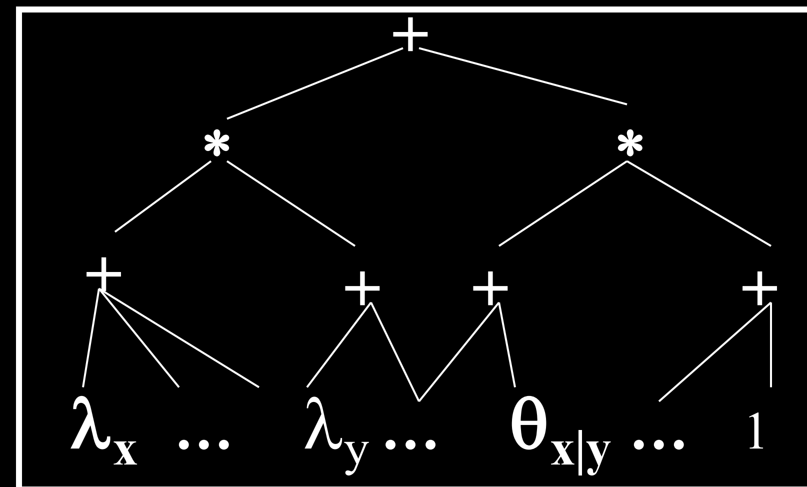
$\lambda_a \wedge \lambda_b \wedge \lambda_s \leftrightarrow \theta_{s|ab}$

Compiling Bayesian Networks

Bayesian network



Arithmetic Circuit



Boolean Formula

$\lambda_x \vee \lambda_{\neg x}$

$\lambda_x \rightarrow \lambda_{\neg x}$

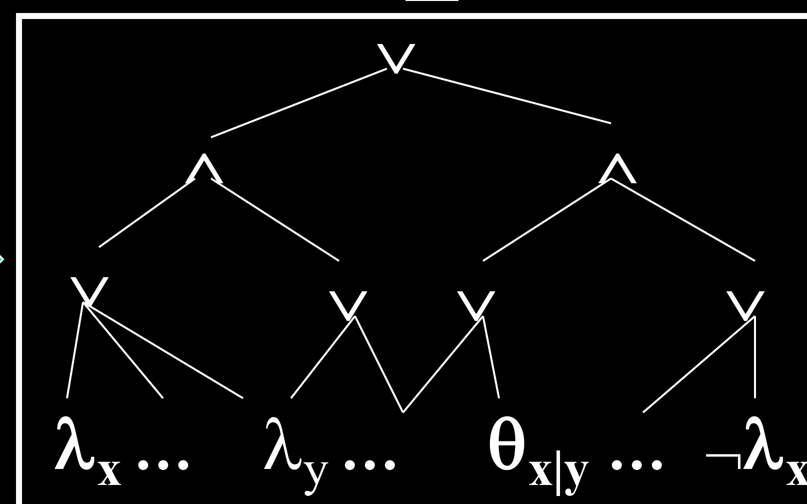
$\lambda_x \wedge \lambda_y \rightarrow \theta_{x|y}$

…

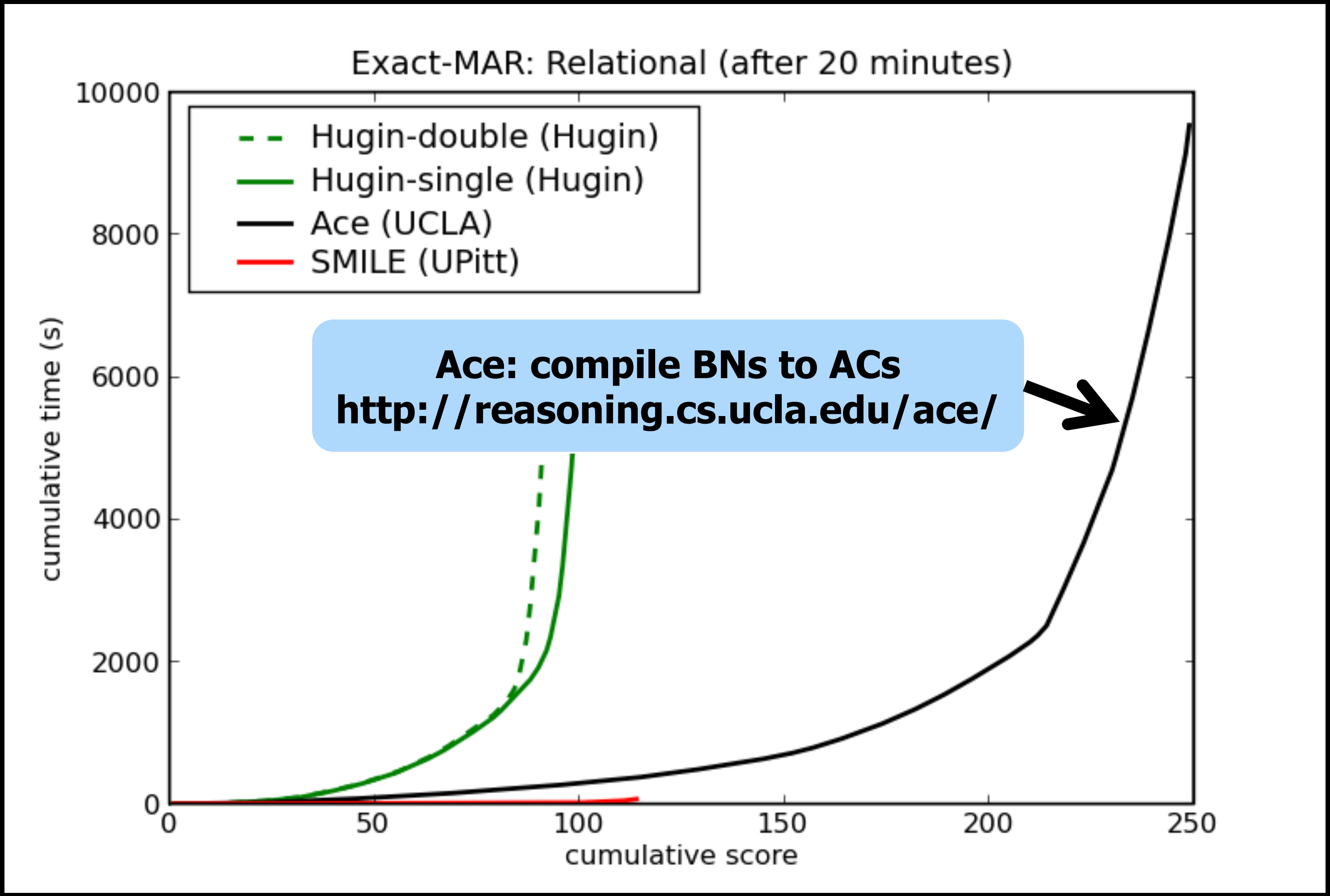
….

….

Boolean Circuit



UAI'08 Competition: Exact Inference using ACs



2021 Evaluation: Exact Inference using ACs

Durgesh Agrawal, Yash Pote, Kuldeep S. Meel:
Partition Function Estimation: A Quantitative Study.

Method Name	Problem Classes Relation-al (354)	Prome-das (65)	BN (60)	Ising (52)	Segment (50)	ObjDetect (35)	Protein (29)	Misc (27)	Total (672)
Ace	354	65	60	51	50	0	16	15	611
Fractional Belief Propagation (FBP)	293	65	58	41	48	32	29	9	575
Loopy Belief Propagation (BP)	292	65	58	41	46	32	29	10	573
Generalized Belief Propagation (GBP)	281	65	36	47	40	34	29	9	541
Edge Deletion Belief Propagation (EDBP)	245	42	56	50	49	35	28	23	528
GANAK	353	58	53	4	0	0	7	14	489
Double Loop Generalised BP (HAK)	199	65	58	43	43	35	29	14	486
Tree Expectation Propagation (TREEEP)	101	65	58	50	48	35	29	15	401
SampleSearch	89	56	33	52	37	35	29	25	356
Bucket Elimination (BE)	98	32	15	52	50	35	29	22	333
Conditioned Belief Propagation (CBP)	109	32	21	41	50	35	29	8	325
Join Tree (JT)	98	32	15	52	50	19	26	21	313
Dynamic Importance Sampling (DIS)	24	65	25	52	50	35	29	27	307
Weighted Mini Bucket Elimination (WMB)	68	13	17	50	50	20	28	12	258
miniC2D	187	1	30	31	0	0	0	1	250
WeightCount	93	0	27	0	0	0	0	0	120
WISH	0	0	0	9	0	0	0	0	9
FocusedFlatSAT	6	0	0	0	0	0	0	0	6

reasoning.cs.ucla.edu/ace

Optional Reading

An Advance on Variable Elimination with Applications to Tensor-Based Computation (2020)

New VE Algorithm that exploits unknown functional dependencies

&

outputs an AC in the form of a Tensor Graph

Functional Dependencies

X	Y	$f(XY)$
x_0	y_0	0
x_0	y_1	1
x_1	y_0	1
x_1	y_1	0

$x_0 \mapsto y_1, \quad x_1 \mapsto y_0$

X	Y	$f(XY)$
x_0	y_0	1
x_0	y_1	0
x_1	y_0	0
x_1	y_1	1

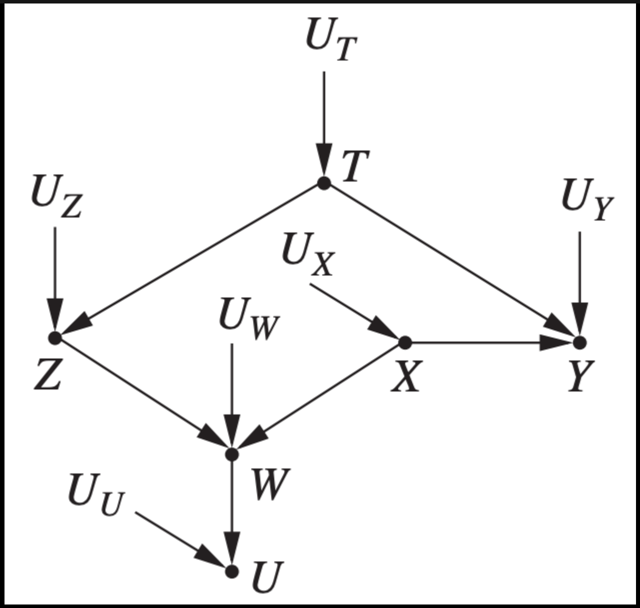
$x_0 \mapsto y_0, \quad x_1 \mapsto y_1$

We may know that Y is a function of X, Y = f(X), but may not know the identity of the function f

Unknown Functional Dependencies: Another type of Background Knowledge (BK)

Structural Causal Models (SCMs)

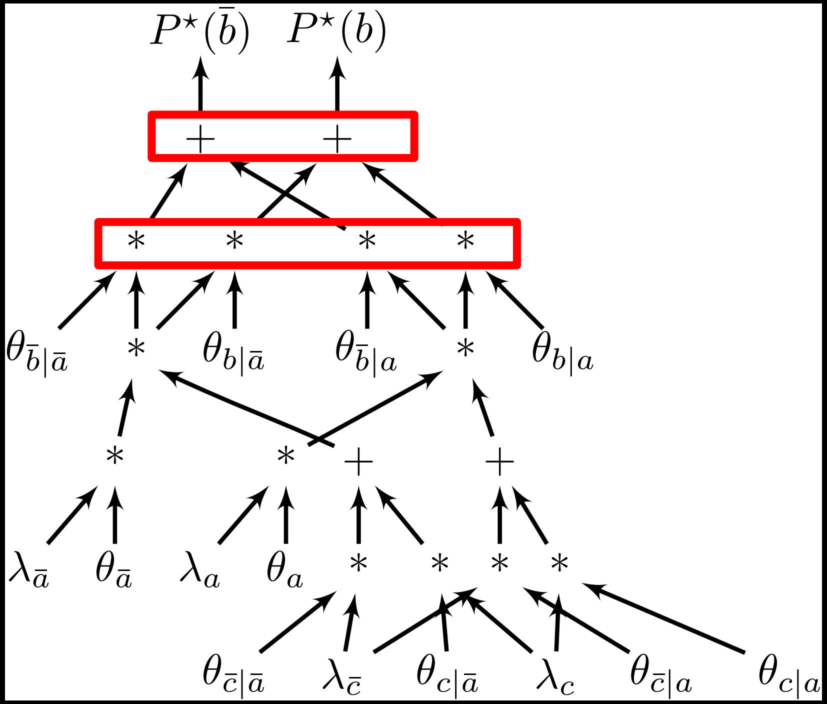
unknown functional dependencies



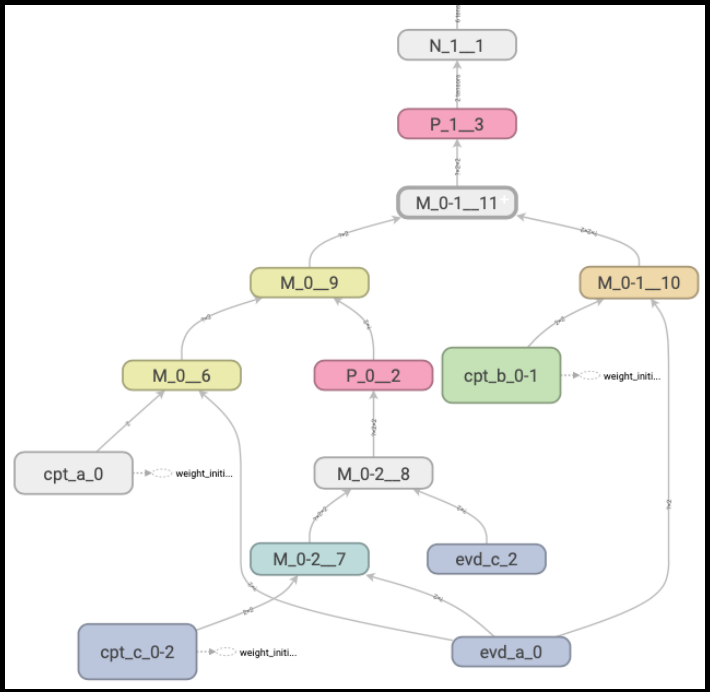
exogenous variables (distributions)

endogenous variables (functions)

Arithmetic Circuit



Tensor Graph



The Main VE Theorem

$$\sum_Y f(XY)g(XZ)h(XW)$$

expressions to be evaluated

$$\sum_Y f(XY)g(XZ)h(XW) = \sum_Y k(XYZW)$$

factor over 4 variables

$$\sum_Y f(XY)g(XZ)h(XW) = g(XZ)h(XW)\sum_Y f(XY)$$

VE theorem can help

$$\sum_X f(XY)g(XZ)h(XW)$$

VE theorem cannot help

The New VE Theorems

$$\sum_X f(XY)g(XZ)h(XW)$$

VE theorem cannot help

Suppose factor $f(XY)$ specifies a function from Y to X

$$= \sum_X \underbrace{f(XY)g(XZ)}\,\underbrace{f(XY)h(XW)}$$

Theorem 1:
replicate functional CPTs

$$= \left(\sum_X f(XY)g(XZ)\right)\left(\sum_X f(XY)h(XW)\right)$$

Theorem 2:
smaller factors

We don't need to know the identity of function f(XY)