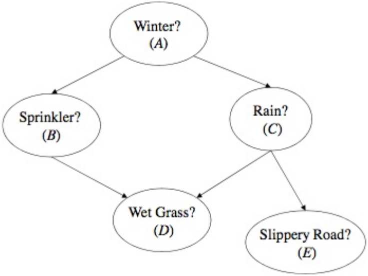
Logic and Bayesian Networks

Part 2: Logical Encoding of Bayesian Networks

Jinbo Huang

Bayesian Network



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

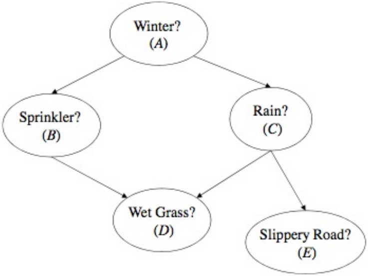
B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

$$\Pr(\mathbf{z}) \stackrel{def}{=} \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{z}} \theta_{x|\mathbf{u}}$$

$$\begin{aligned} &\Pr(\bar{e}, d, \bar{c}, b, a) \\ &= \theta_{\bar{e}|\bar{c}}\, \theta_{d|b,\bar{c}}\, \theta_{\bar{c}|a}\, \theta_{b|a}\, \theta_a \\ &= (.1)(.9)(.2)(.2)(.6) \\ &= .0216 \end{aligned}$$

Bayesian Network



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

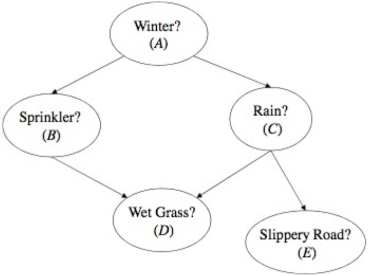
C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

$$\Pr(\mathbf{z}) \stackrel{def}{=} \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{z}} \theta_{x|\mathbf{u}}$$

$$\begin{aligned}&\Pr(\bar{e}, d, \bar{c}, b, a)\\ &= \theta_{\bar{e}|\bar{c}}\, \theta_{d|b,\bar{c}}\, \theta_{\bar{c}|a}\, \theta_{b|a}\, \theta_a\\ &= (.1)(.9)(.2)(.2)(.6)\\ &= .0216\end{aligned}$$

$$\Pr(\bar{e}) =$$

Bayesian Network



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

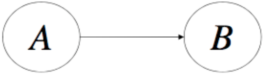
C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

$$\Pr(\mathbf{z}) \stackrel{def}{=} \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{z}} \theta_{x|\mathbf{u}}$$

$$\begin{aligned} &\Pr(\bar{e}, d, \bar{c}, b, a) \\ &= \theta_{\bar{e}|\bar{c}}\, \theta_{d|b,\bar{c}}\, \theta_{\bar{c}|a}\, \theta_{b|a}\, \theta_a \\ &= (.1)(.9)(.2)(.2)(.6) \\ &= .0216 \end{aligned}$$

$$\begin{aligned} \Pr(\bar{e}) = \quad & \Pr(\bar{e}, d, c, b, a) \\ + \quad & \Pr(\bar{e}, d, c, b, \bar{a}) \\ + \quad & \ldots \\ + \quad & \Pr(\bar{e}, \bar{d}, \bar{c}, \bar{b}, \bar{a}) \end{aligned}$$

Network Polynomial



A	Θ_A
true	$\theta_a = .3$
false	$\theta_{\bar{a}} = .7$

A	B	$\Theta_{B\|A}$
true	true	$\theta_{b\|a} = .1$
true	false	$\theta_{\bar{b}\|a} = .9$
false	true	$\theta_{b\|\bar{a}} = .8$
false	false	$\theta_{\bar{b}\|\bar{a}} = .2$

A	B	$\Pr(A, B)$
a	b	$\theta_a \theta_{b\|a}$
a	$\bar{b}$	$\theta_a \theta_{\bar{b}\|a}$
$\bar{a}$	b	$\theta_{\bar{a}} \theta_{b\|\bar{a}}$
$\bar{a}$	$\bar{b}$	$\theta_{\bar{a}} \theta_{\bar{b}\|\bar{a}}$

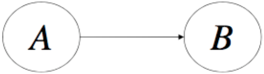
Network Polynomial

A	Θ_A
true	$\theta_a = .3$
false	$\theta_{\bar{a}} = .7$

A	B	$\Theta_{B\|A}$
true	true	$\theta_{b\|a} = .1$
true	false	$\theta_{\bar{b}\|a} = .9$
false	true	$\theta_{b\|\bar{a}} = .8$
false	false	$\theta_{\bar{b}\|\bar{a}} = .2$

A	B	$\Pr(A, B)$
a	b	$\lambda_a \lambda_b \theta_a \theta_{b\|a}$
a	$\bar{b}$	$\lambda_a \lambda_{\bar{b}} \theta_a \theta_{\bar{b}\|a}$
$\bar{a}$	b	$\lambda_{\bar{a}} \lambda_b \theta_{\bar{a}} \theta_{b\|\bar{a}}$
$\bar{a}$	$\bar{b}$	$\lambda_{\bar{a}} \lambda_{\bar{b}} \theta_{\bar{a}} \theta_{\bar{b}\|\bar{a}}$

Network Polynomial



A	Θ_A
true	$\theta_a = .3$
false	$\theta_{\bar{a}} = .7$

A	B	$\Theta_{B\|A}$
true	true	$\theta_{b\|a} = .1$
true	false	$\theta_{\bar{b}\|a} = .9$
false	true	$\theta_{b\|\bar{a}} = .8$
false	false	$\theta_{\bar{b}\|\bar{a}} = .2$

A	B	$\Pr(A, B)$
a	b	$\lambda_a \lambda_b \theta_a \theta_{b\|a}$
a	$\bar{b}$	$\lambda_a \lambda_{\bar{b}} \theta_a \theta_{\bar{b}\|a}$
$\bar{a}$	b	$\lambda_{\bar{a}} \lambda_b \theta_{\bar{a}} \theta_{b\|\bar{a}}$
$\bar{a}$	$\bar{b}$	$\lambda_{\bar{a}} \lambda_{\bar{b}} \theta_{\bar{a}} \theta_{\bar{b}\|\bar{a}}$

$$f = \lambda_a \lambda_b \theta_a \theta_{b|a} + \lambda_a \lambda_{\bar{b}} \theta_a \theta_{\bar{b}|a} + \lambda_{\bar{a}} \lambda_b \theta_{\bar{a}} \theta_{b|\bar{a}} + \lambda_{\bar{a}} \lambda_{\bar{b}} \theta_{\bar{a}} \theta_{\bar{b}|\bar{a}}$$

Network Polynomial

A	Θ_A
true	$\theta_a = .3$
false	$\theta_{\bar{a}} = .7$

A	B	$\Theta_{B\|A}$
true	true	$\theta_{b\|a} = .1$
true	false	$\theta_{\bar{b}\|a} = .9$
false	true	$\theta_{b\|\bar{a}} = .8$
false	false	$\theta_{\bar{b}\|\bar{a}} = .2$

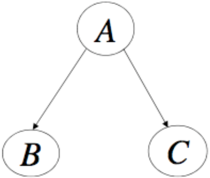
A	B	$\Pr(A,B)$
a	b	$\lambda_a\lambda_b\theta_a\theta_{b\|a}$
a	$\bar{b}$	$\lambda_a\lambda_{\bar{b}}\theta_a\theta_{\bar{b}\|a}$
$\bar{a}$	b	$\lambda_{\bar{a}}\lambda_b\theta_{\bar{a}}\theta_{b\|\bar{a}}$
$\bar{a}$	$\bar{b}$	$\lambda_{\bar{a}}\lambda_{\bar{b}}\theta_{\bar{a}}\theta_{\bar{b}\|\bar{a}}$

$$f = \lambda_a\lambda_b\theta_a\theta_{b|a} + \lambda_a\lambda_{\bar{b}}\theta_a\theta_{\bar{b}|a} + \lambda_{\bar{a}}\lambda_b\theta_{\bar{a}}\theta_{b|\bar{a}} + \lambda_{\bar{a}}\lambda_{\bar{b}}\theta_{\bar{a}}\theta_{\bar{b}|\bar{a}}$$

$$\begin{aligned} f(\mathbf{e} = \bar{a}) &= (0)(1)\theta_a\theta_{b|a} + (0)(1)\theta_a\theta_{\bar{b}|a} + (1)(1)\theta_{\bar{a}}\theta_{b|\bar{a}} + (1)(1)\theta_{\bar{a}}\theta_{\bar{b}|\bar{a}} \\ &= \theta_{\bar{a}}\theta_{b|\bar{a}} + \theta_{\bar{a}}\theta_{\bar{b}|\bar{a}} = \Pr(\mathbf{e}) \end{aligned}$$

Network Polynomial

$$f = \lambda_a \lambda_b \lambda_c \theta_a \theta_{b|a} \theta_{c|a} + \lambda_a \lambda_b \lambda_{\bar{c}} \theta_a \theta_{b|a} \theta_{\bar{c}|a} + \ldots + \lambda_{\bar{a}} \lambda_{\bar{b}} \lambda_{\bar{c}} \theta_{\bar{a}} \theta_{\bar{b}|\bar{a}} \theta_{\bar{c}|\bar{a}}$$



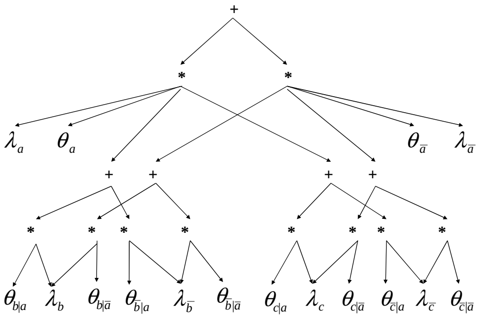
A	Θ_A
true	.5
false	.5

A	B	$\Theta_{B\|A}$
true	true	1
true	false	0
false	true	0
false	false	1

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.2
false	false	.8

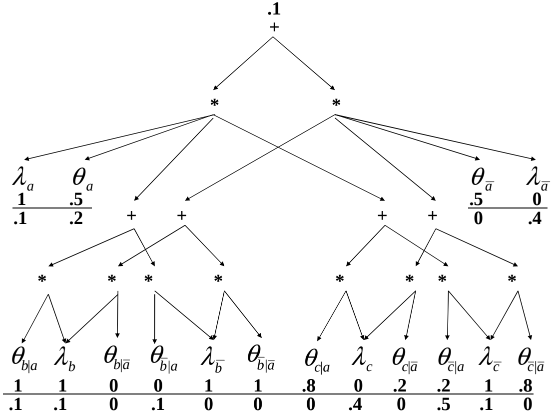
Network Polynomial as Arithmetic Circuit

$$f = \lambda_a\lambda_b\lambda_c\theta_a\theta_{b|a}\theta_{c|a} + \lambda_a\lambda_b\lambda_{\bar{c}}\theta_a\theta_{b|a}\theta_{\bar{c}|a} + \ldots + \lambda_{\bar{a}}\lambda_{\bar{b}}\lambda_{\bar{c}}\theta_{\bar{a}}\theta_{\bar{b}|\bar{a}}\theta_{\bar{c}|\bar{a}}$$



f has exponential size, circuit may not

(Ignore bottom numbers in graph)

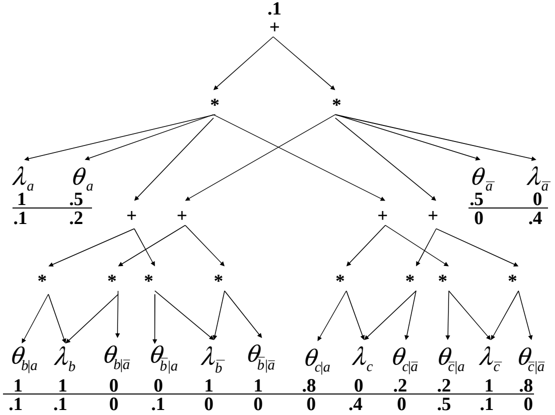


$\Pr(\mathbf{e} = a\bar{c})$

Computation of any $\Pr(\mathbf{e})$ linear in circuit size

Circuit Evaluation

(Ignore bottom numbers in graph)



$\Pr(\mathbf{e} = a\bar{c})$

Computation of any $\Pr(\mathbf{e})$ linear in circuit size

Goal: compute compact circuits

Factoring Multilinear Polynomials

$$f = \lambda_a \lambda_b \lambda_c \theta_a \theta_{b|a} \theta_{c|a} + \lambda_a \lambda_b \lambda_{\overline{c}} \theta_a \theta_{b|a} \theta_{\overline{c}|a} + \ldots + \lambda_{\overline{a}} \lambda_{\overline{b}} \lambda_{\overline{c}} \theta_{\overline{a}} \theta_{\overline{b}|\overline{a}} \theta_{\overline{c}|\overline{a}}$$

Has 2^n terms, each having $2n$ variables

Linear in every variable

Factor multilinear polynomials by compiling logic formulas

From Polynomials to Logic Formulas

$f = abc + ac + c$

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

From Polynomials to Logic Formulas

$f = abc + ac + c$

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

8 possible terms, f includes 3

8 possible models, Δ admits 3

- $A\,B\,C$ ($\Rightarrow abc$)
- $A\,\overline{B}\,C$ ($\Rightarrow ac$)
- $\overline{A}\,\overline{B}\,C$ ($\Rightarrow c$)

From Polynomials to Logic Formulas

$f = abc + ac + c$

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

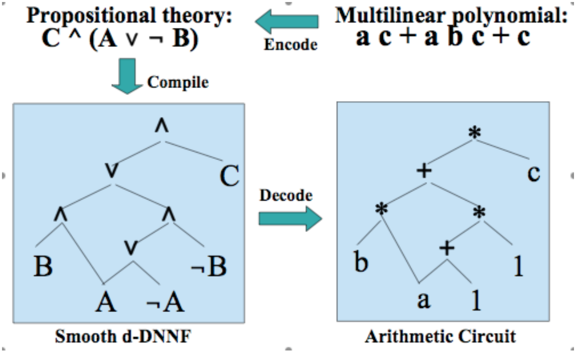
8 possible terms, f includes 3

8 possible models, Δ admits 3

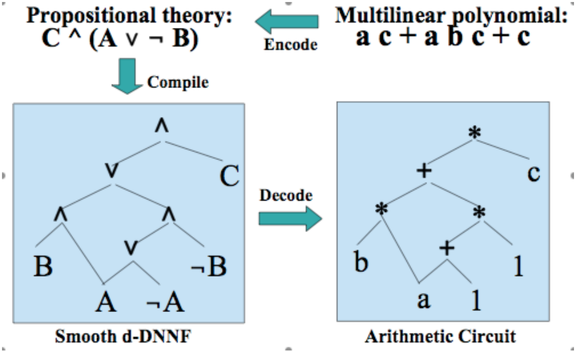
- $A\,B\,C$ ($\Rightarrow abc$)
- $A\,\overline{B}\,C$ ($\Rightarrow ac$)
- $\overline{A}\,\overline{B}\,C$ ($\Rightarrow c$)

Polynomial $\Leftrightarrow$ class of logically equivalent formulas

Encode, Compile, Decode

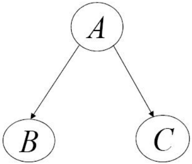


Encode, Compile, Decode



- What formula to write
- What form to compile to (so that it can be turned into AC)
- How to compile

Logical Encoding



A	Θ_A
a_1	0.1
a_2	0.9

A	B	$\Theta_{B\|A}$
a_1	b_1	0.1
a_1	b_2	0.9
a_2	b_1	0.2
a_2	b_2	0.8

A	C	$\Theta_{C\|A}$
a_1	c_1	0.1
a_1	c_2	0.9
a_2	c_1	0.2
a_2	c_2	0.8

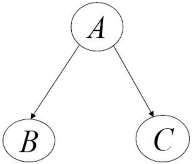
$$
\begin{aligned}
f = \; & \lambda_{a_1}\lambda_{b_1}\lambda_{c_1}\theta_{a_1}\theta_{b_1|a_1}\theta_{c_1|a_1} \\
& +\lambda_{a_1}\lambda_{b_1}\lambda_{c_2}\theta_{a_1}\theta_{b_1|a_1}\theta_{c_2|a_1} \\
& +\ldots \\
& +\lambda_{a_2}\lambda_{b_2}\lambda_{c_2}\theta_{a_2}\theta_{b_2|a_2}\theta_{c_2|a_2}
\end{aligned}
$$

Need a (compact) formula with exactly 2^n models

Write a set of formulas for each CPT, take union

Each set proportional to CPT

Exploits network structure for compactness



A	Θ_A
a_1	0.1
a_2	0.9

A	B	$\Theta_{B\|A}$
a_1	b_1	0.1
a_1	b_2	0.9
a_2	b_1	0.2
a_2	b_2	0.8

A	C	$\Theta_{C\|A}$
a_1	c_1	0.1
a_1	c_2	0.9
a_2	c_1	0.2
a_2	c_2	0.8

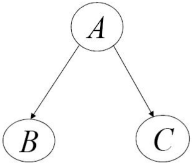
$$
\begin{aligned}
f = \; & \lambda_{a_1}\lambda_{b_1}\lambda_{c_1}\theta_{a_1}\theta_{b_1|a_1}\theta_{c_1|a_1} \\
& +\lambda_{a_1}\lambda_{b_1}\lambda_{c_2}\theta_{a_1}\theta_{b_1|a_1}\theta_{c_2|a_1} \\
& +\ldots \\
& +\lambda_{a_2}\lambda_{b_2}\lambda_{c_2}\theta_{a_2}\theta_{b_2|a_2}\theta_{c_2|a_2}
\end{aligned}
$$

Indicator variables/clauses

- $I_{a_1} \vee I_{a_2} \quad \neg I_{a_1} \vee \neg I_{a_2}$
- $I_{b_1} \vee I_{b_2} \quad \neg I_{b_1} \vee \neg I_{b_2}$
- $I_{c_1} \vee I_{c_2} \quad \neg I_{c_1} \vee \neg I_{c_2}$

Exactly 8 ways to instantiate these variables consistently

Logical Encoding



A	Θ_A
a_1	0.1
a_2	0.9

A	B	$\Theta_{B\|A}$
a_1	b_1	0.1
a_1	b_2	0.9
a_2	b_1	0.2
a_2	b_2	0.8

A	C	$\Theta_{C\|A}$
a_1	c_1	0.1
a_1	c_2	0.9
a_2	c_1	0.2
a_2	c_2	0.8

$$
\begin{aligned}
f = \; & \lambda_{a_1}\lambda_{b_1}\lambda_{c_1}\theta_{a_1}\theta_{b_1|a_1}\theta_{c_1|a_1} \\
& +\lambda_{a_1}\lambda_{b_1}\lambda_{c_2}\theta_{a_1}\theta_{b_1|a_1}\theta_{c_2|a_1} \\
& +\ldots \\
& +\lambda_{a_2}\lambda_{b_2}\lambda_{c_2}\theta_{a_2}\theta_{b_2|a_2}\theta_{c_2|a_2}
\end{aligned}
$$

Parameter variables/clauses

- $I_{a_1} \leftrightarrow P_{a_1} \quad I_{a_2} \leftrightarrow P_{a_2}$
- $I_{a_1} \wedge I_{b_1} \leftrightarrow P_{b_1|a_1}$
  $I_{a_1} \wedge I_{b_2} \leftrightarrow P_{b_2|a_1}$
  $I_{a_2} \wedge I_{b_1} \leftrightarrow P_{b_1|a_2}$
  $I_{a_2} \wedge I_{b_2} \leftrightarrow P_{b_2|a_2}$
- $I_{a_1} \wedge I_{c_1} \leftrightarrow P_{c_1|a_1}$
  $I_{a_1} \wedge I_{c_2} \leftrightarrow P_{c_2|a_1}$
  $I_{a_2} \wedge I_{c_1} \leftrightarrow P_{c_1|a_2}$
  $I_{a_2} \wedge I_{c_2} \leftrightarrow P_{c_2|a_2}$

Each instantiation of Is forces unique instantiation of Ps

Hence exactly 8 models

Zero Parameters

- $I_{a_1} \wedge I_{b_2} \leftrightarrow P_{b_2|a_1}$
- Any model sets $P_{b_2|a_1}$ to true iff it sets I_{a_1} and I_{a_2} to true
- $\theta_{b_2|a_1} = 0$: Models setting $P_{b_2|a_1}$ to true map to terms that evaluate to 0
- Might as well remove them: $\neg I_{a_1} \vee \neg I_{b_2}$
- Variable $P_{b_2|a_1}$ also removed

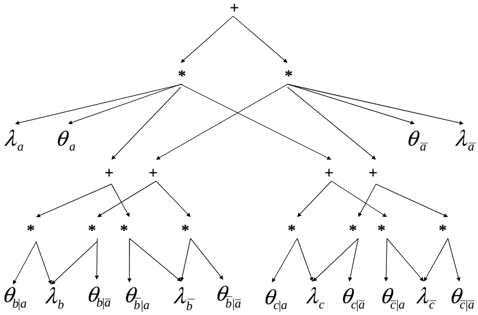
One Parameters

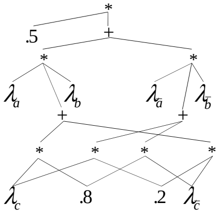
- $I_{a_1} \wedge I_{b_1} \leftrightarrow P_{b_1|a_1}$
- $\theta_{b_1|a_1} = 1$: Removing $\theta_{b_1|a_1}$ from polynomial never affects its value
- Might as well not have $P_{b_1|a_1}$ (and not have above clause) in encoding

Equal Parameters

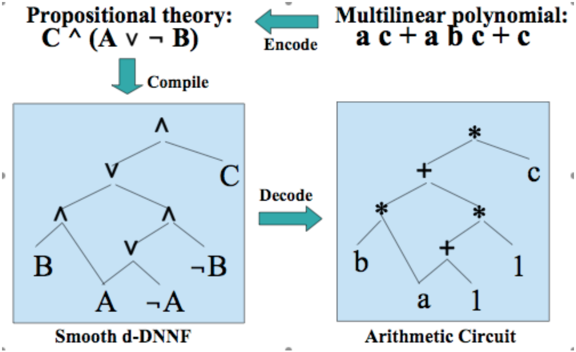
- $I_{a_1} \wedge I_{c_1} \leftrightarrow P_{c_1|a_1}$
  $I_{a_2} \wedge I_{c_2} \leftrightarrow P_{c_2|a_2}$
- $\theta_{c_1|a_1} = \theta_{c_2|a_2} = .8$: Might as well merge $P_{c_1|a_1}$ and $P_{c_2|a_2}$ into one variable
- $(I_{a_1} \wedge I_{c_1}) \vee (I_{a_2} \wedge I_{c_2}) \leftrightarrow P_1$

Effect of New Encoding





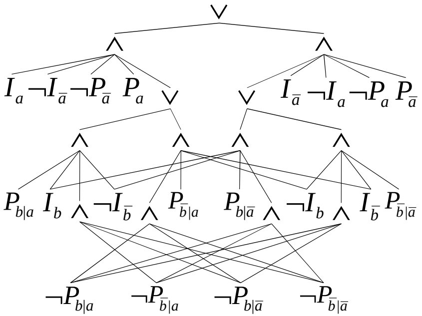
Encode, Compile, Decode

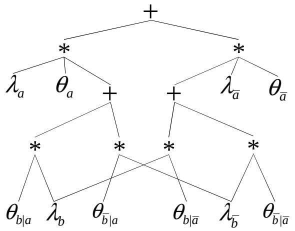


- What formula to write $\surd$
- What form to compile to (so that it can be turned into AC)
- How to compile

Target Form for Compilation

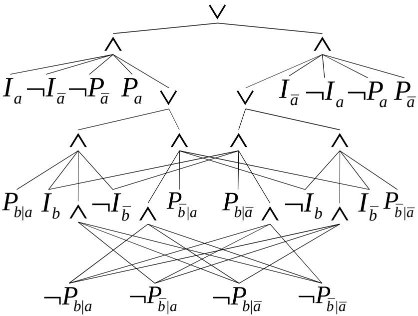
Would like to turn compiled formula (Boolean circuit) directly to arithmetic circuit

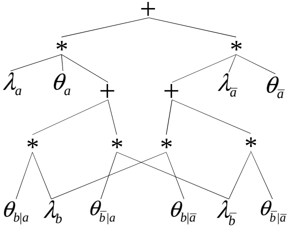




- $\vee$ to $+$, $\wedge$ to $*$, negative literals to 1
- I_x to λ_x, $P_{x|\mathbf{u}}$ to $\theta_{x|\mathbf{u}}$

Would like to turn compiled formula (Boolean circuit) directly to arithmetic circuit

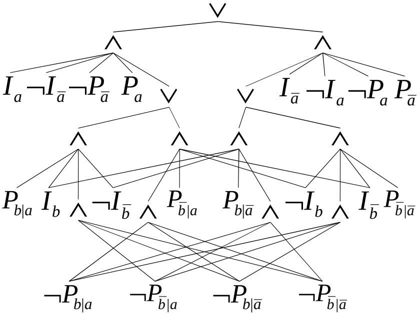


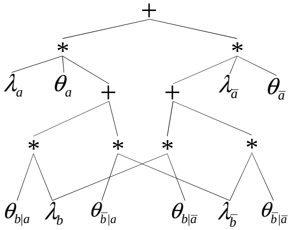


- Boolean circuit must satisfy certain properties

Target Form for Compilation

Would like to turn compiled formula (Boolean circuit) directly to arithmetic circuit

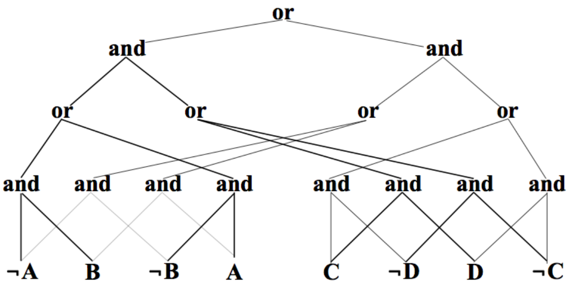




- Boolean circuit must satisfy certain properties
- Smoothness, determinism, decomposability

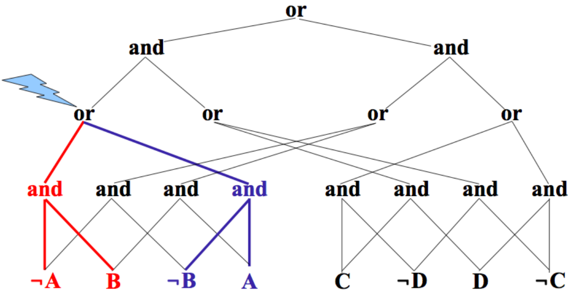
Negation Normal Form (NNF)

Negation only in leaves, other nodes AND/OR



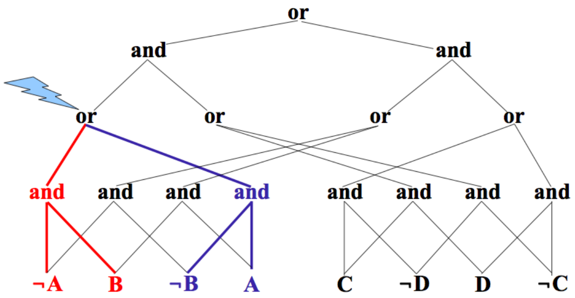
Smoothness and Determinism

Children of OR over same variables, and logically inconsistent



Smoothness and Determinism

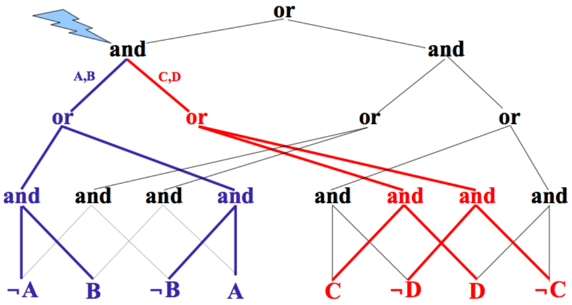
Children of OR over same variables, and logically inconsistent



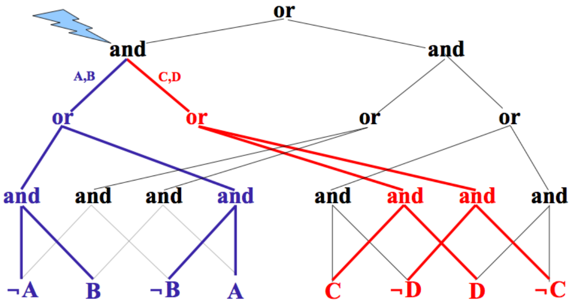
Δ_1 encodes f_1, Δ_2 encodes f_2
$\Rightarrow \Delta_1 \vee \Delta_2$ encodes $f_1 + f_2$

Decomposability

Children of AND over disjoint sets of variables

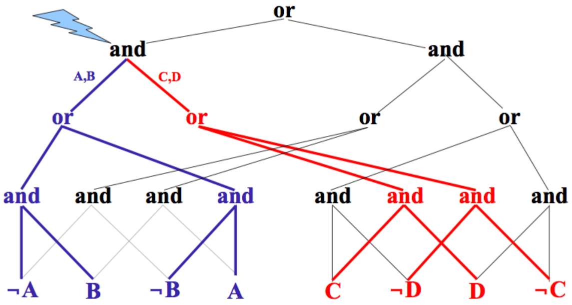


Children of AND over disjoint sets of variables



$\Delta_1 \equiv \{\overline{A}B, A\overline{B}\}$

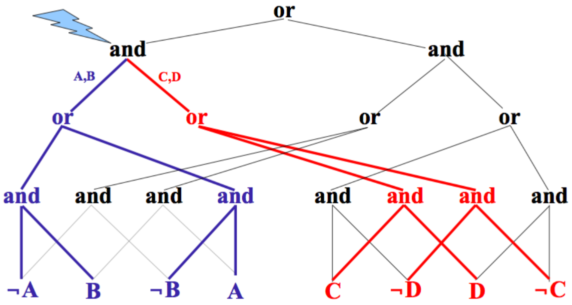
Children of AND over disjoint sets of variables



$\Delta_1 \equiv \{\overline{A}B, A\overline{B}\}$ encodes $b + a$

Decomposability

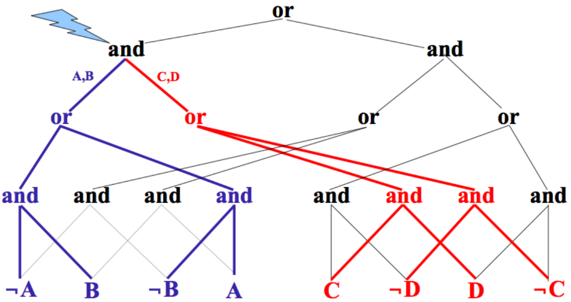
Children of AND over disjoint sets of variables



$\Delta_1 \equiv \{\overline{A}B, A\overline{B}\}$ encodes $b + a$, $\Delta_2 \equiv \{CD, \overline{CD}\}$ encodes $cd + 1$

Decomposability

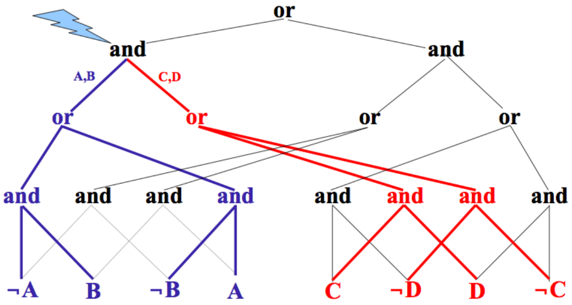
Children of AND over disjoint sets of variables



$\Delta_1 \equiv \{\bar{A}B, A\bar{B}\}$ encodes $b + a$, $\Delta_2 \equiv \{CD, \overline{CD}\}$ encodes $cd + 1$,
$\Delta_1 \wedge \Delta_2 \equiv \{\bar{A}B, A\bar{B}\} \times \{CD, \overline{CD}\}$

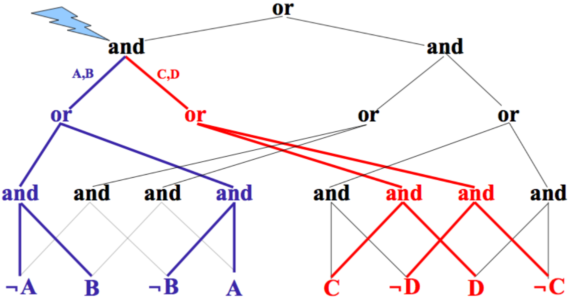
Decomposability

Children of AND over disjoint sets of variables



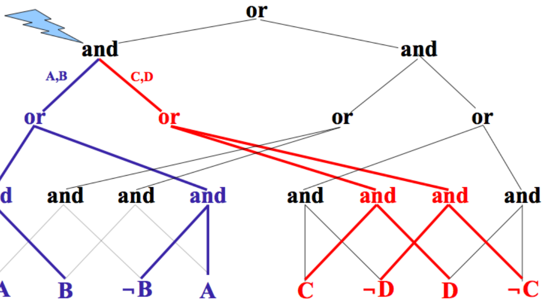
$\Delta_1 \equiv \{\overline{A}B, A\overline{B}\}$ encodes $b + a$, $\Delta_2 \equiv \{CD, \overline{CD}\}$ encodes $cd + 1$,
$\Delta_1 \wedge \Delta_2 \equiv \{\overline{A}B, A\overline{B}\} \times \{CD, \overline{CD}\} =$
$\{\overline{A}BCD, \overline{A}B\overline{C}\,\overline{D}, A\overline{B}CD, A\overline{B}\,\overline{CD}\}$

Children of AND over disjoint sets of variables



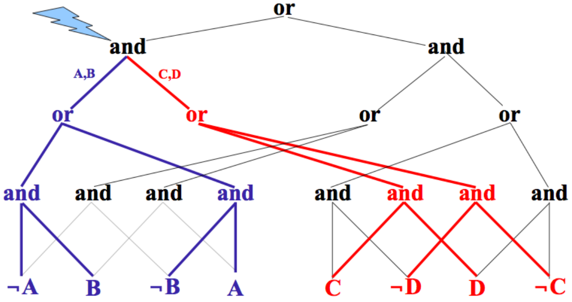
$\Delta_1 \equiv \{\overline{A}B, A\overline{B}\}$ encodes $b + a$, $\Delta_2 \equiv \{CD, \overline{C\,D}\}$ encodes $cd + 1$,
$\Delta_1 \wedge \Delta_2 \equiv \{\overline{A}B, A\overline{B}\} \times \{CD, \overline{C\,D}\} =$
$\{\overline{A}BCD, \overline{A}B\overline{C}\,\overline{D}, A\overline{B}CD, A\overline{B}\,\overline{C\,D}\}$ encodes
$bcd + b + acd + a$

Children of AND over disjoint sets of variables



$\Delta_1 \equiv \{\overline{A}B, A\overline{B}\}$ encodes $b + a$, $\Delta_2 \equiv \{CD, \overline{C}\,\overline{D}\}$ encodes $cd + 1$,
$\Delta_1 \wedge \Delta_2 \equiv \{\overline{A}B, A\overline{B}\} \times \{CD, \overline{C}\,\overline{D}\} =$
$\{\overline{A}BCD, \overline{A}B\overline{C}\,\overline{D}, A\overline{B}CD, A\overline{B}\,\overline{C}\,\overline{D}\}$ encodes
$bcd + b + acd + a = (b + a)(cd + 1)$

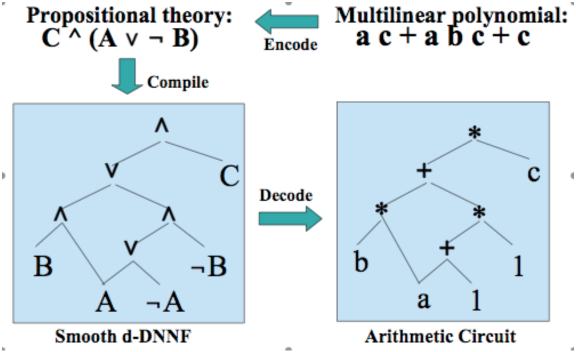
Children of AND over disjoint sets of variables



Δ_1 encodes f_1, Δ_2 encodes f_2

$\Rightarrow \Delta_1 \wedge \Delta_2$ encodes $f_1 \times f_2$

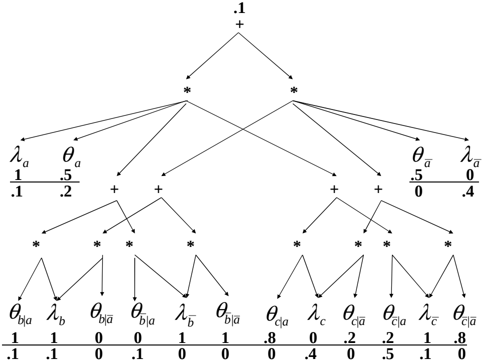
Encode, Compile, Decode



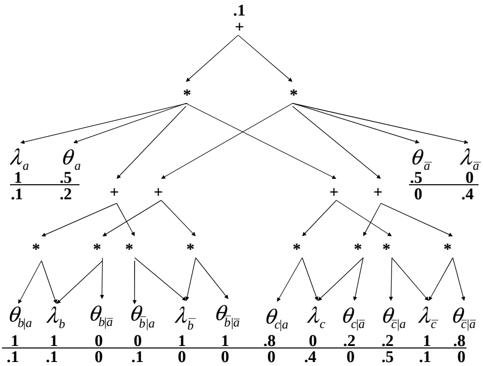
- What formula to write $\surd$
- What form to compile to (so that it can be turned into AC) $\surd$
- How to compile

Partial Derivatives

$$\Pr(\mathbf{e} = a\bar{c}) = .1$$



$\Pr(\mathbf{e} = a\bar{c}) = .1$



f is linear in every variable

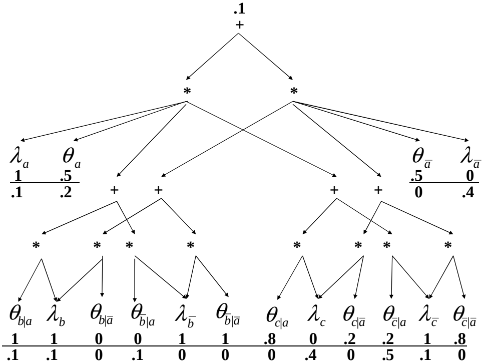
$\frac{\partial f}{\partial \lambda_{\bar{a}}}(\mathbf{e}) = .4$

f will increase by .4 if $\lambda_{\bar{a}}$ changes from 0 to 1 ($\mathbf{e}$ changes from $a\bar{c}$ to $\bar{c}$)

$\Pr(\bar{c}) = .1 + .4 = .5$

Partial Derivatives

$\Pr(\mathbf{e} = a\bar{c}) = .1$



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

Flipping variable: $\bar{x} \in \mathbf{e}$

Adding literal: $x, \bar{x} \notin \mathbf{e}$

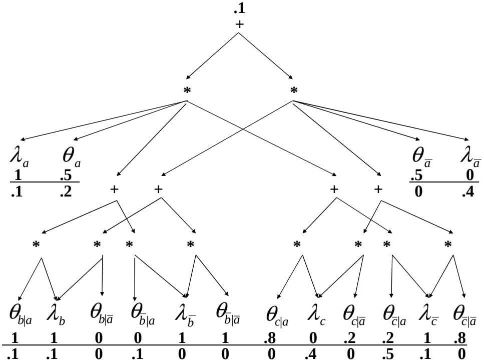
$\Pr(\mathbf{e})$: $x \in \mathbf{e}$

$\Pr(\bar{a}\,\bar{c}) = \frac{\partial f}{\partial \lambda_{\bar{a}}} = .4$

$\Pr(ab\bar{c}) = \frac{\partial f}{\partial \lambda_b} = .1$

Partial Derivatives

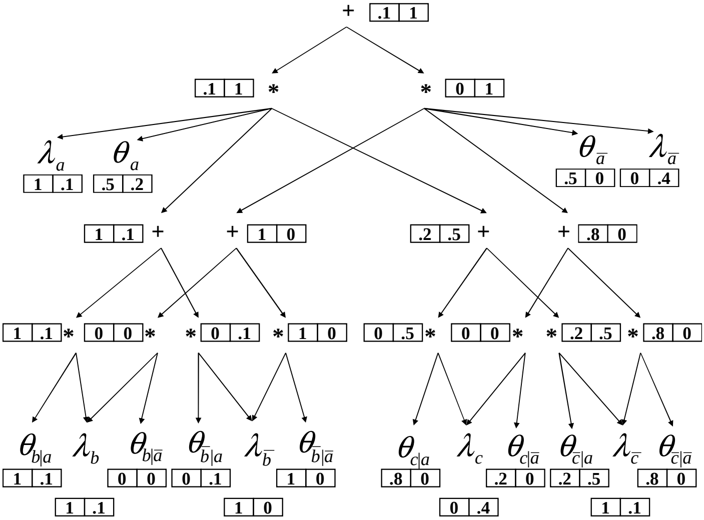
$$\Pr(\mathbf{e} = a\bar{c}) = .1$$



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

Gives family marginals $\Pr(x, \mathbf{u}, \mathbf{e})\ \forall x\mathbf{u}$

Evaluation and Differentiation



Evaluation and Differentiation

- Bottom-up pass evaluates circuit, computes $\Pr(\mathbf{e})$
- Top-down pass computes all partial derivatives
- Linear in circuit size

Summary

- Network polynomials and arithmetic circuits
- Bayesian networks encoded as logic formulas
- Target form for compiling logic formulas
- Differentiation of arithmetic circuits