Polynomial semantics of probabilistic circuits

Oliver Broadrick, Honghua Zhang, Guy Van den Broeck – UAI 2024

The limits of tractable marginalization

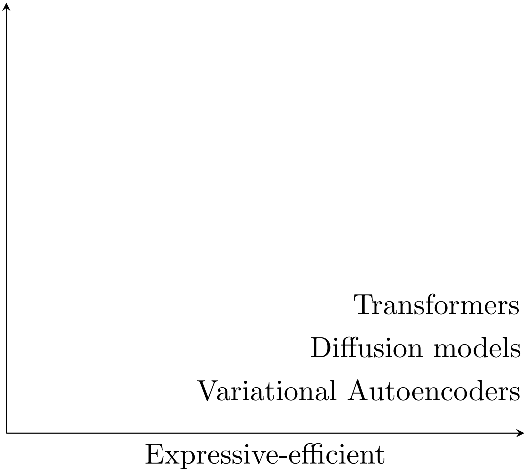
Oliver Broadrick, Sanyam Agarwal, Markus Bläser, Guy Van den Broeck – wip

Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

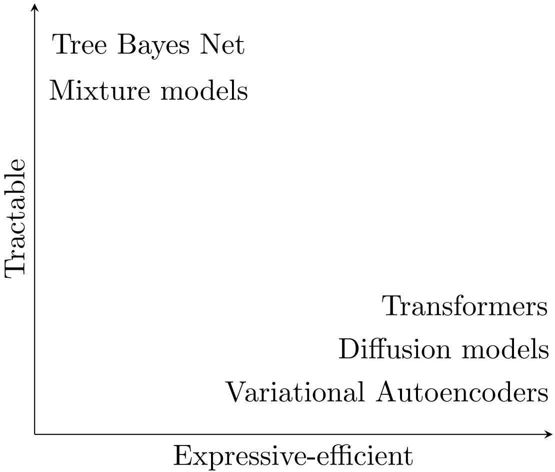
Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



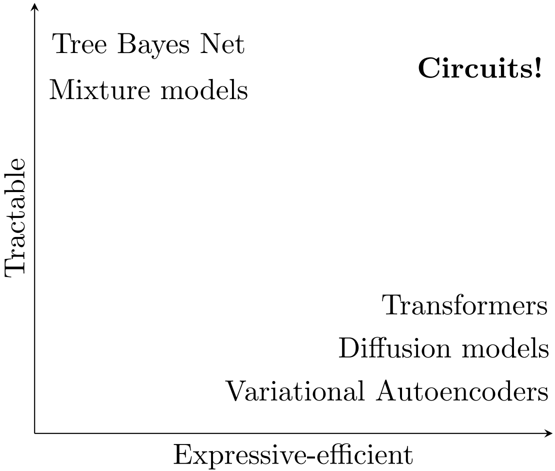
Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Marginal Inference

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= \boxed{\Pr[X_1 = 1, X_2 = 0]} + \boxed{\Pr[X_1 = 1, X_2 = 1]} \\ &= \boxed{0.3} + \boxed{0.4} \\ &= 0.7\end{aligned}$$

Marginal Inference

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= \Pr[X_1 = 1, X_2 = 0] + \Pr[X_1 = 1, X_2 = 1] \\ &= 0.3 + 0.4 \\ &= 0.7\end{aligned}$$

Goal: Find maximally expressive-efficient models that support marginal inference in time polynomial in the model size.

Tractable Marginalization is Useful

Logical control of language models (see Honghua Zhang)

Efficient causal reasoning (see Benjie Wang)

Data compression and control for diffusion models (see Anji Liu)

Probabilistic program inference (see Poorva Garg)

Neurosymbolic learning (see Kareem Ahmed)

Probabilistic logic programming (see Renato Geh)

etc. etc. etc.

Approaches

Bayes Nets (of bounded treewidth)
Determinantal Point Processes
Characteristic Circuits
Multi-Linear Representations
Probabilistic Generating Circuits
Sum-Product Networks
...

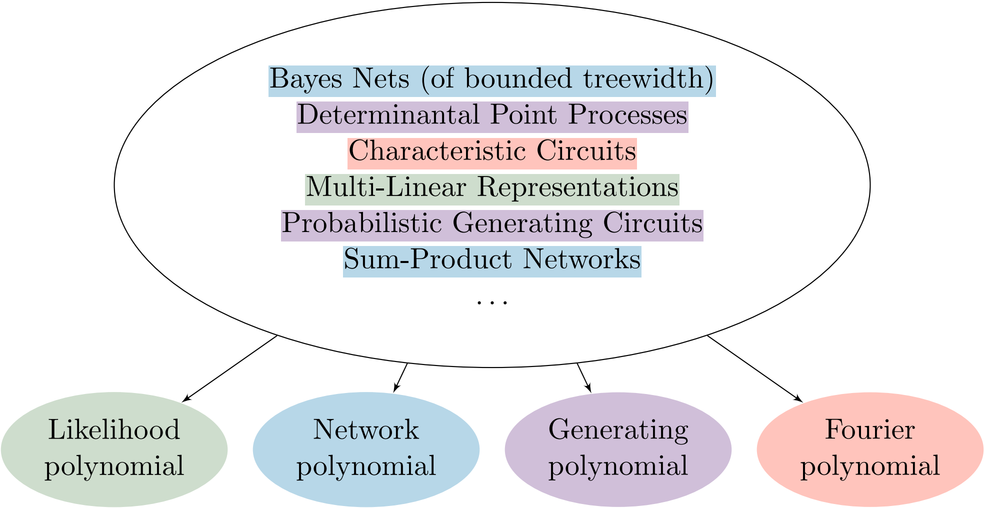
Approaches

Bayes Nets (of bounded treewidth)
Determinantal Point Processes
Characteristic Circuits
Multi-Linear Representations
Probabilistic Generating Circuits
Sum-Product Networks
...

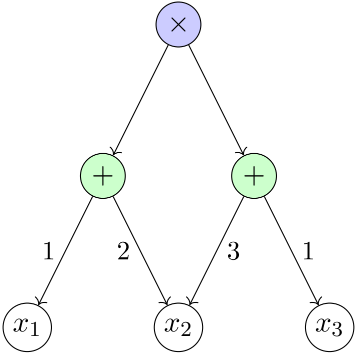
$\downarrow$

Polynomials!

How to encode distributions in polynomials?

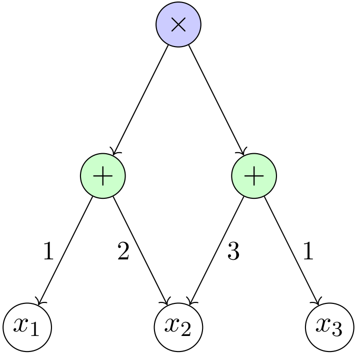


Circuits represent polynomials succinctly



$$3x_1x_2 + x_1x_3 + 6x_2^2 + 2x_2x_3$$

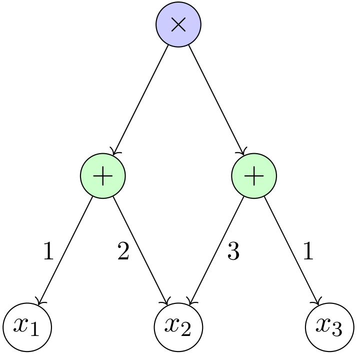
Circuits represent polynomials succinctly



$$3x_1x_2 + x_1x_3 + 6x_2^2 + 2x_2x_3$$

Circuits are *fully expressive*

Circuits represent polynomials succinctly

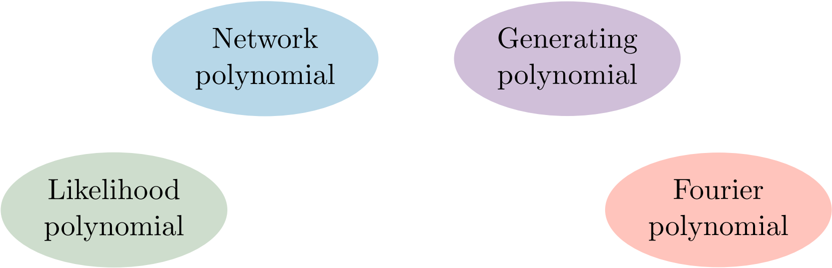


$$3x_1x_2 + x_1x_3 + 6x_2^2 + 2x_2x_3$$

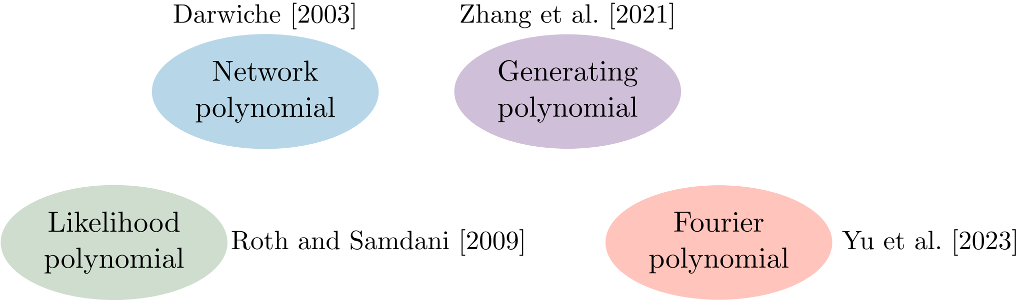
Circuits are *fully expressive*

They can also be *expressive-efficient*

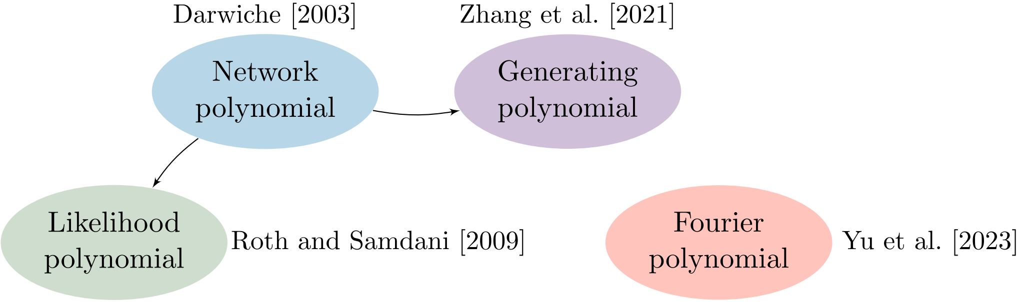
Polynomial Semantics



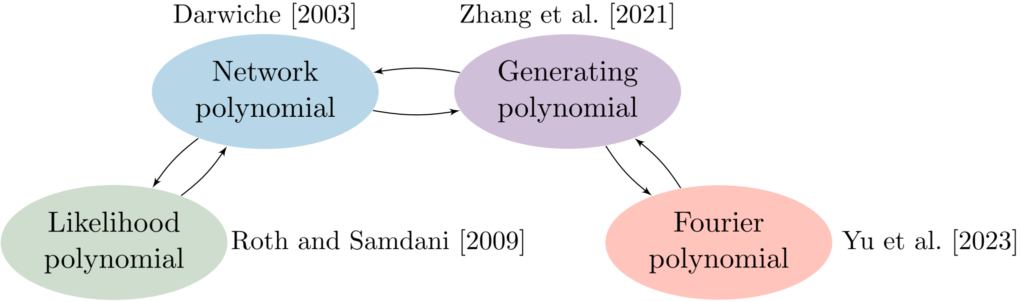
Polynomial Semantics



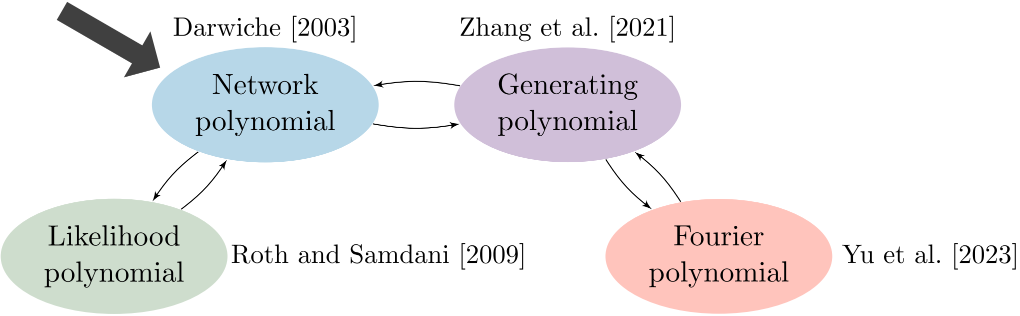
Polynomial Semantics



Polynomial Semantics



Polynomial Semantics



Network polynomial

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Network polynomial

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$\Pr[X_1 = 1]$

Network polynomial

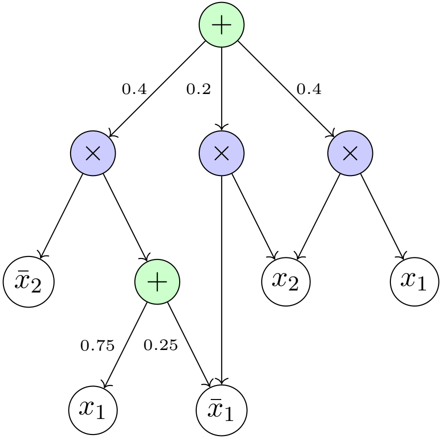
$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}
\Pr[X_1 = 1] &= p(1, 1, 0, 1) \\
&= .1(0)(1) + .2(0)(1) + .3(1)(1) + .4(1)(1) \\
&= 0 + 0 + .3 + .4 \\
&= .7
\end{aligned}$$

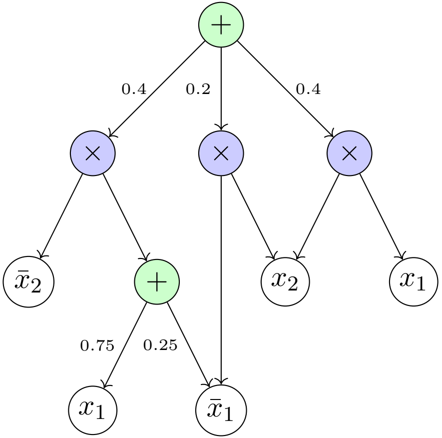
Network polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Network polynomial

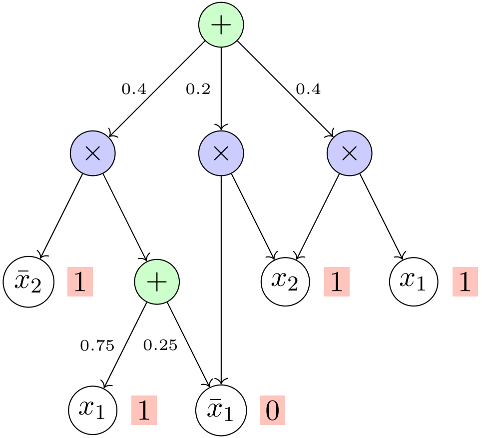
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



$\Pr[X_1 = 1]$?

Network polynomial

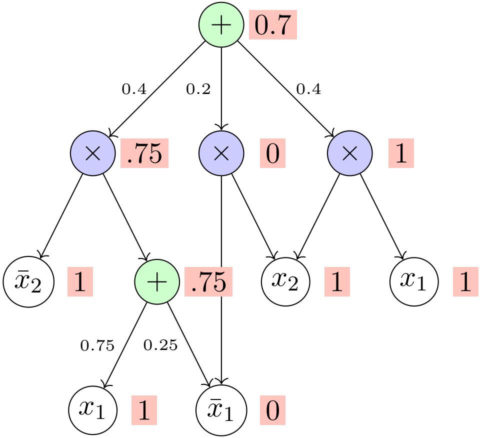
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



$\Pr[X_1 = 1]$?

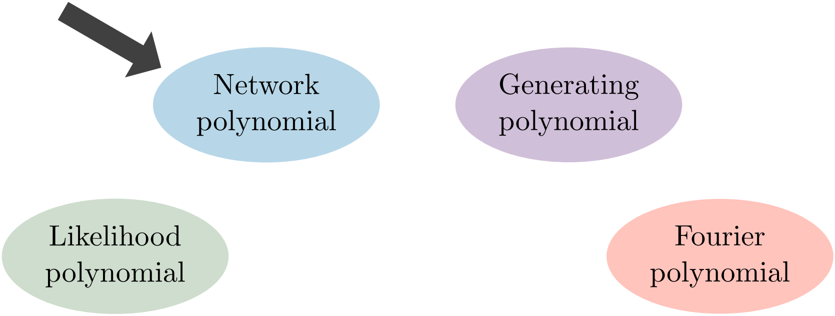
Network polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

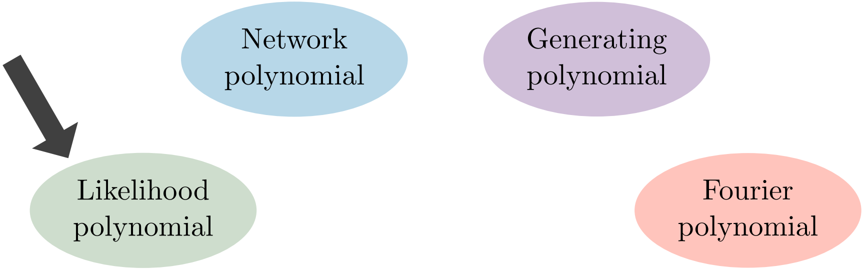


$\Pr[X_1 = 1]$?

Progress Update



Progress Update



Likelihood polynomial

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$\approx$ A neural net that for an input vector outputs its probability

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Likelihood polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$\approx$ A neural net that for an input vector outputs its probability

Marginal inference?

Likelihood polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$\approx$ A neural net that for an input vector outputs its probability

Marginal inference?

Relation to network polynomial?

Likelihood polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$\approx$ A neural net that for an input vector outputs its probability

Marginal inference?

Relation to network polynomial?

(1) Transform network to likelihood:

$$p(x, \bar{x}) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3x_1\bar{x}_2 + .4x_1x_2$$

$\rightarrow$ Replace $\bar{x}_i$ with $1 - x_i$

Likelihood polynomial

(2) Transform likelihood to network:

$$p(x_1, x_2) = \boxed{.2x_1} + .1x_2 + .1$$

Likelihood polynomial

(2) Transform likelihood to network:

$p(x_1, x_2) = \boxed{.2x_1} + .1x_2 + .1$

$$(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(\boxed{.2\frac{x_1}{x_1 + \bar{x}_1}} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1\right)$$

Likelihood polynomial

(2) Transform likelihood to network:

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$$(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(.2\frac{x_1}{x_1 + \bar{x}_1} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1 \right)$$

$$= .2x_1(x_2 + \bar{x}_2) + .1x_2(x_1 + \bar{x}_1) + .1(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)$$

Likelihood polynomial

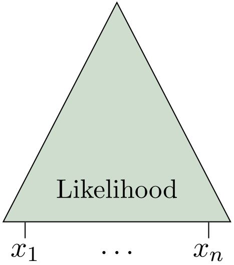
(2) Transform likelihood to network:

$$p(x_1, x_2) = \boxed{.2x_1} + .1x_2 + .1$$

$$\begin{aligned}
&(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(\boxed{.2\frac{x_1}{x_1 + \bar{x}_1}} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1\right) \\
&= \boxed{.2x_1(x_2 + \bar{x}_2)} + .1x_2(x_1 + \bar{x}_1) + .1(x_1 + \bar{x}_1)(x_2 + \bar{x}_2) \\
&= p(x_1, x_2, \bar{x}_1, \bar{x}_2)
\end{aligned}$$

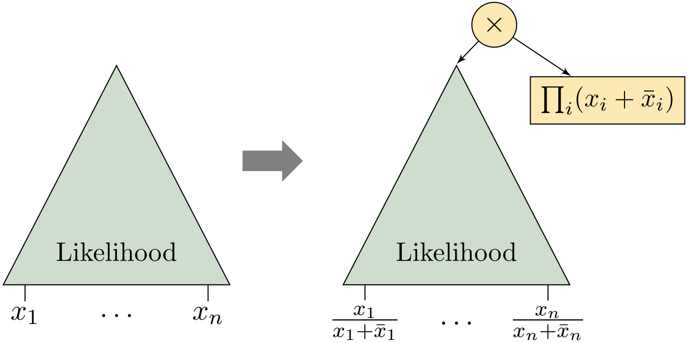
Likelihood polynomial

Transform likelihood to network:



Likelihood polynomial

Transform likelihood to network:

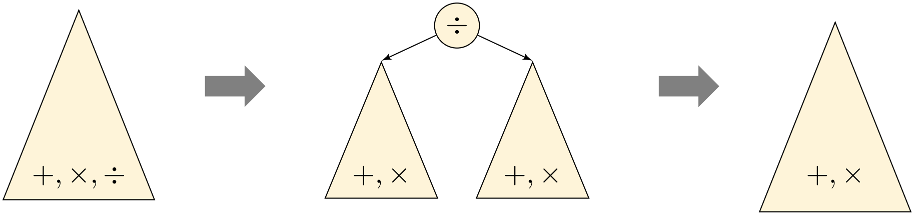


Removing Divisions

Theorem (Strassen [1973]). *You can remove divisions in polynomial time!*

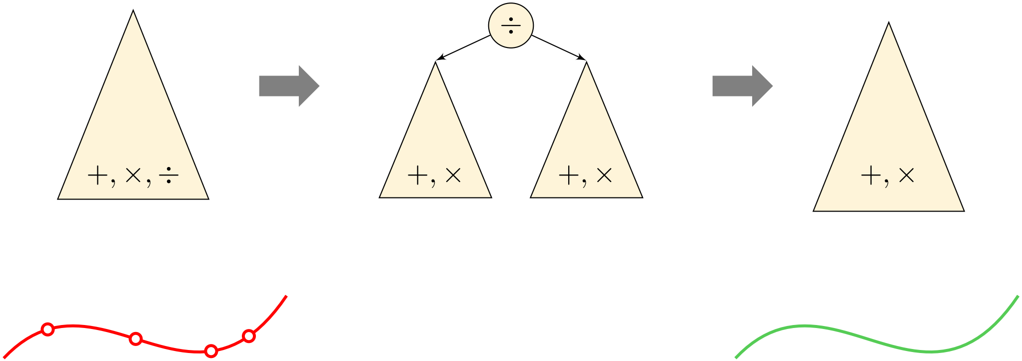
Removing Divisions

Theorem (Strassen [1973]). *You can remove divisions in polynomial time!*



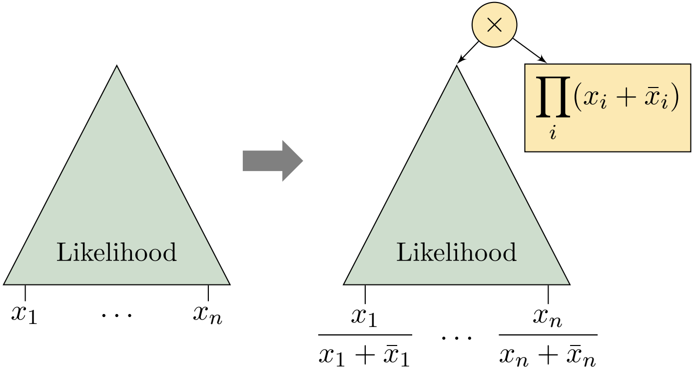
Removing Divisions

Theorem (Strassen [1973]). *You can remove divisions in polynomial time!*



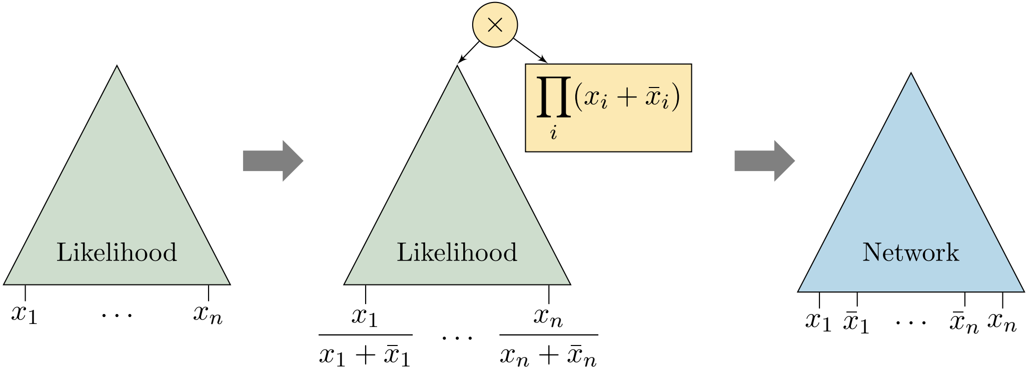
Likelihood polynomial

Transform likelihood to network:

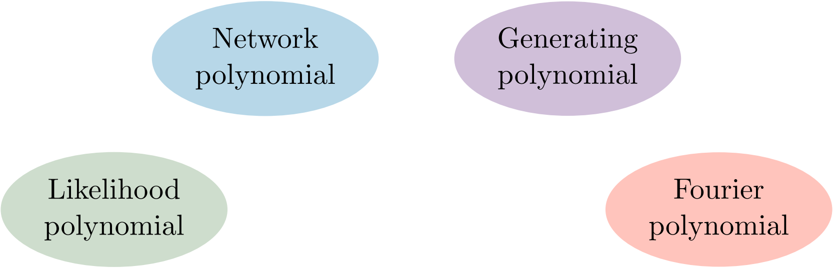


Likelihood polynomial

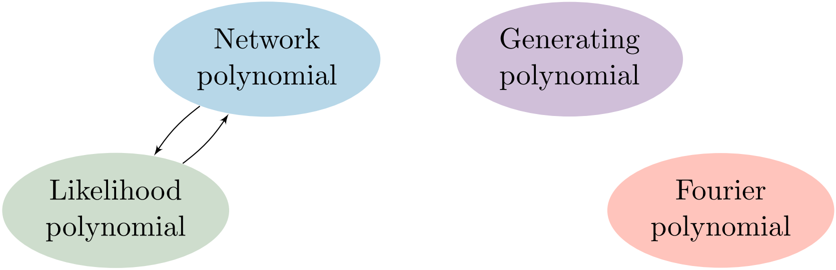
Transform likelihood to network:



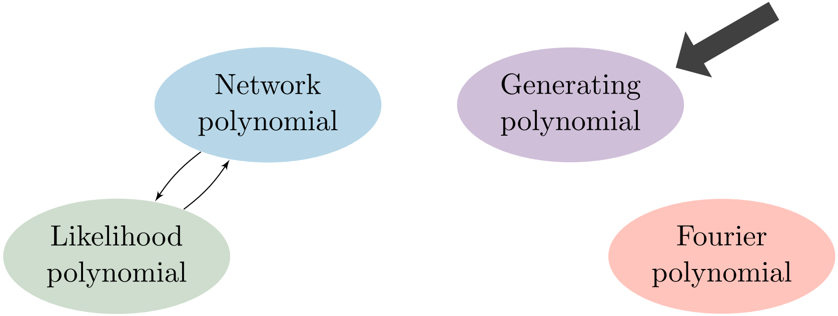
Progress Update



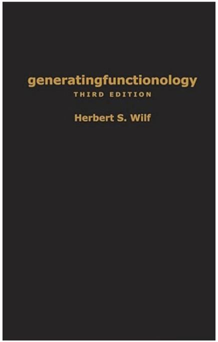
Progress Update



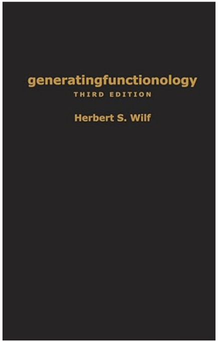
Progress Update



Generating polynomial

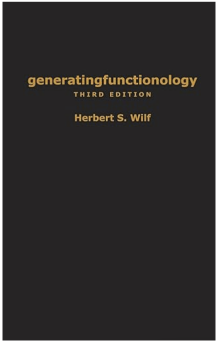


Generating polynomial

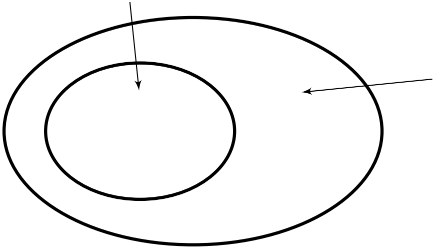


Monotone, decomposable circuits computing network polynomials (SPNs, PCs)

Generating polynomial

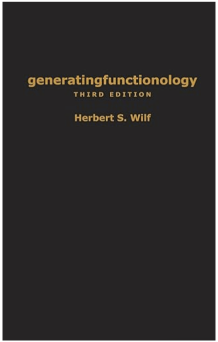


Monotone, decomposable circuits computing network polynomials (SPNs, PCs)

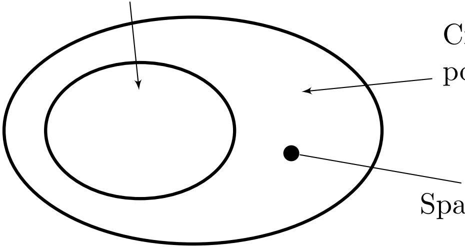


Circuits computing generating polynomials

Generating polynomial



Monotone, decomposable circuits computing network polynomials (SPNs, PCs)



Circuits computing generating polynomials

Spanning tree distribution[a]

[a]Martens and Medabalimi [2015], Zhang et al. [2021]

Generating polynomial

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Generating polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

Marginal inference: ✓ [Zhang et al., 2021]

Generating polynomial

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Marginal inference: ✓ [Zhang et al., 2021]

Relation to network polynomial?

Generating polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$

Marginal inference: ✓ [Zhang et al., 2021]

Relation to network polynomial?

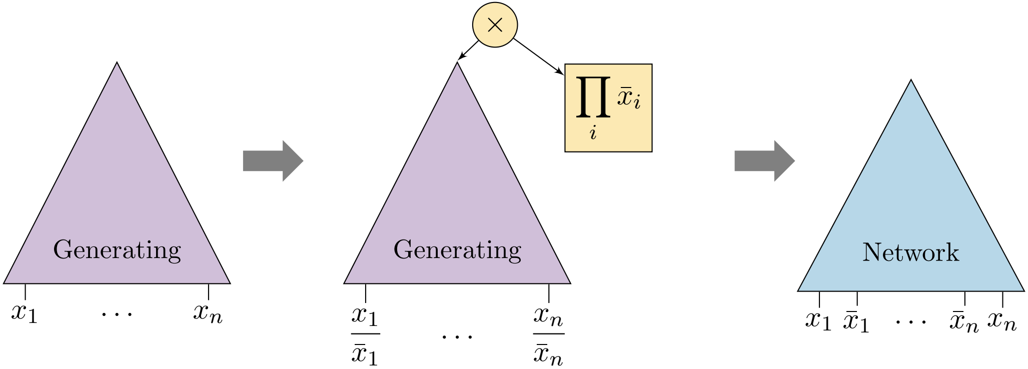
(1) Transform network to generating:

$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2$

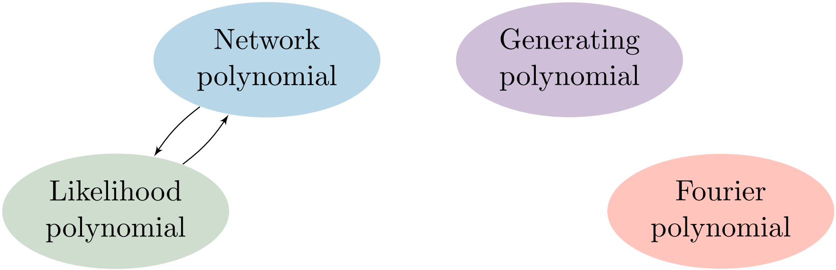
$\rightarrow$ Replace $\bar{x}_i$ with 1

Generating polynomial

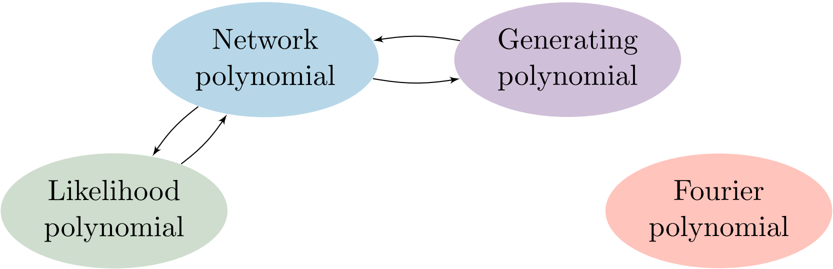
(2) Transform generating to network:



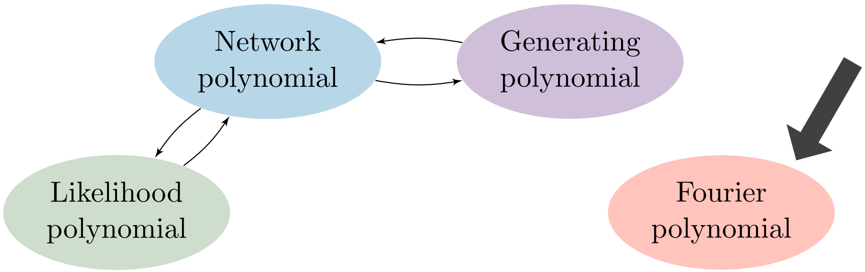
Progress Update



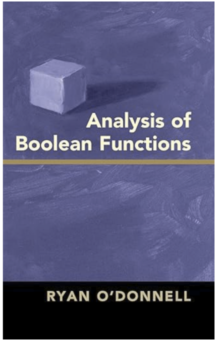
Progress Update



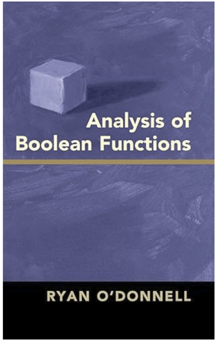
Progress Update



Fourier Polynomial

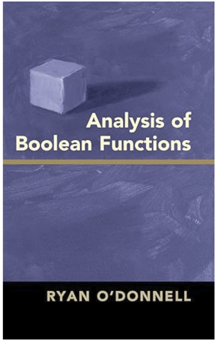


Fourier Polynomial



Fourier transform of the probability mass function

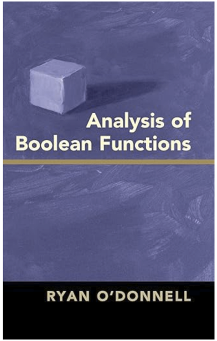
Fourier Polynomial



Fourier transform of the probability mass function

- Graphical model approximate inference
- Characteristic Circuits

Fourier Polynomial

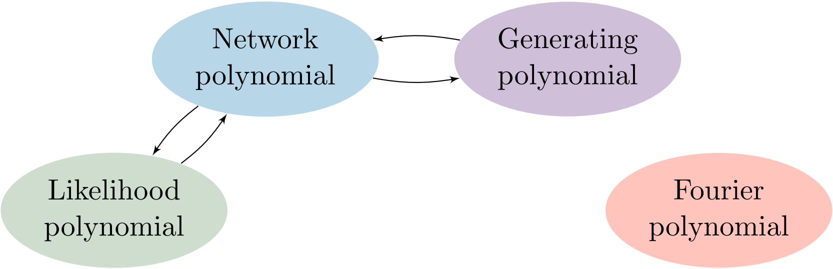


Fourier transform of the probability mass function

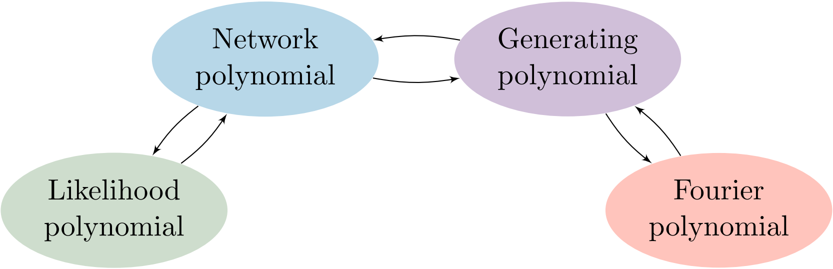
- Graphical model approximate inference
- Characteristic Circuits

Proposition. *Generating polynomials and Fourier polynomials compute* ***the same function*** *on respective domains* $\{-1, 1\}^n$ *and* $\{0, 1\}^n$.

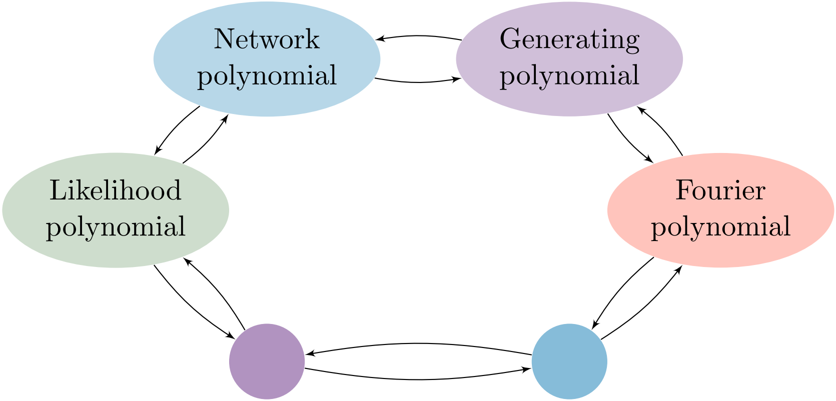
Progress Update



Progress Update



Some New Semantics



Non-binary variables?

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
$\vdots$	$\vdots$	$\vdots$

Non-binary variables?

Literature: just use a binary encoding

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
$\vdots$	$\vdots$	$\vdots$

Non-binary variables?

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
$\vdots$	$\vdots$	$\vdots$

$$g(x) = \boxed{.1x_2} + \boxed{.3x_1x_2^3} + \boxed{.2x_1^3x_2^2} + \ldots$$

Generating polynomial

Non-binary variables?

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
$\vdots$	$\vdots$	$\vdots$

$$g(x) = \boxed{.1x_2} + \boxed{.3x_1x_2^3} + \boxed{.2x_1^3x_2^2} + \dots$$

Generating polynomial

Theorem. *For $|K| \geq 4$, computing likelihoods on a circuit for $g(x)$ is #P-hard.*

Approach: Reduction from $0, 1$-permanent.

Midway Conclusion

What we've done:

- Shown several distinct circuit models are equally expressive-efficient
- Unified existing (and one new) inference algorithms
- Inference is #P-hard in generating polynomials circuits for $k \geq 4$ categories

Midway Conclusion

What we've done:

- Shown several distinct circuit models are equally expressive-efficient
- Unified existing (and one new) inference algorithms
- Inference is #P-hard in generating polynomials circuits for $k \geq 4$ categories

What's next?

- How can this theoretical progress be leveraged in practice?
- Are there more expressive-efficient tractable representations?

Polynomial semantics of probabilistic circuits

Oliver Broadrick, Honghua Zhang, Guy Van den Broeck – UAI 2024

The limits of tractable marginalization

Oliver Broadrick, Sanyam Agarwal, Markus Bläser, Guy Van den Broeck – wip

Finally Multilinear Arithmetic Circuits

$$f : \{0,1\}^n \to \mathbb{R}$$

$$p(x_1, \ldots, x_n) = \sum_{S \subseteq \{1,\ldots,n\}} f([S]) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$$

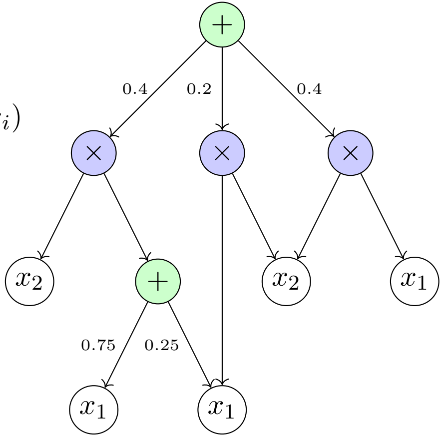
X_1	X_2	f
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Finally Multilinear Arithmetic Circuits

$$f : \{0,1\}^n \to \mathbb{R}$$

$$p(x_1, \ldots, x_n) = \sum_{S \subseteq \{1,\ldots,n\}} f([S]) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$$

X_1	X_2	f
0	0	.1
0	1	.2
1	0	.3
1	1	.4



$$p(x_1, x_2) = .1(1 - x_1)(1 - x_2) + .2(1 - x_1)x_2 + .3x_1(1 - x_2) + .4x_1x_2$$

Functions Tractable for Marginalization

Let $f : \{0,1\}^n \to \{0,1\}$ be a function (family).

Define $\mathsf{MAR}(f)$, the marginalization problem for f, which on input $m \in \{0,1,*\}^n$, asks for $\sum_{x \in M_m} f(x)$ where $M_m = \{x \in \{0,1\}^n : m_i \in \{0,1\} \implies x_i = m_i\}$.

Functions Tractable for Marginalization

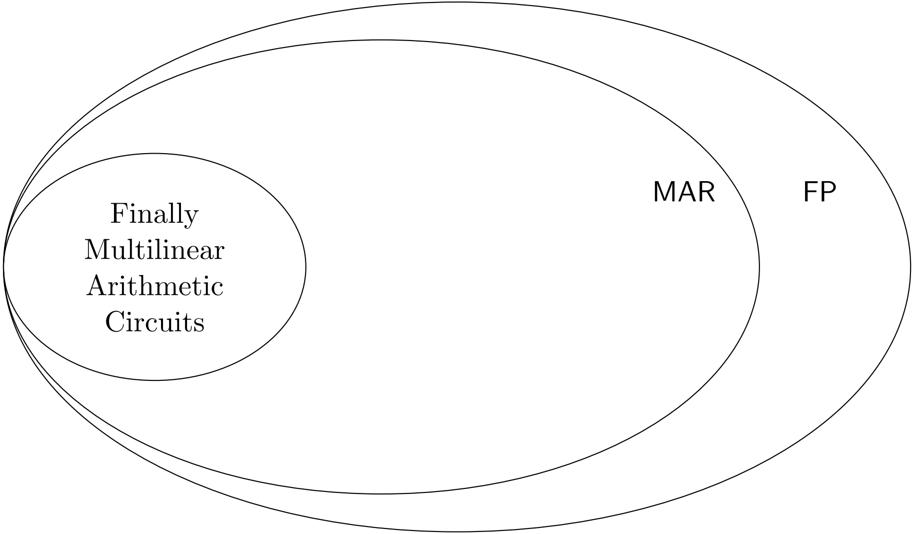
Let $f : \{0,1\}^n \to \{0,1\}$ be a function (family).

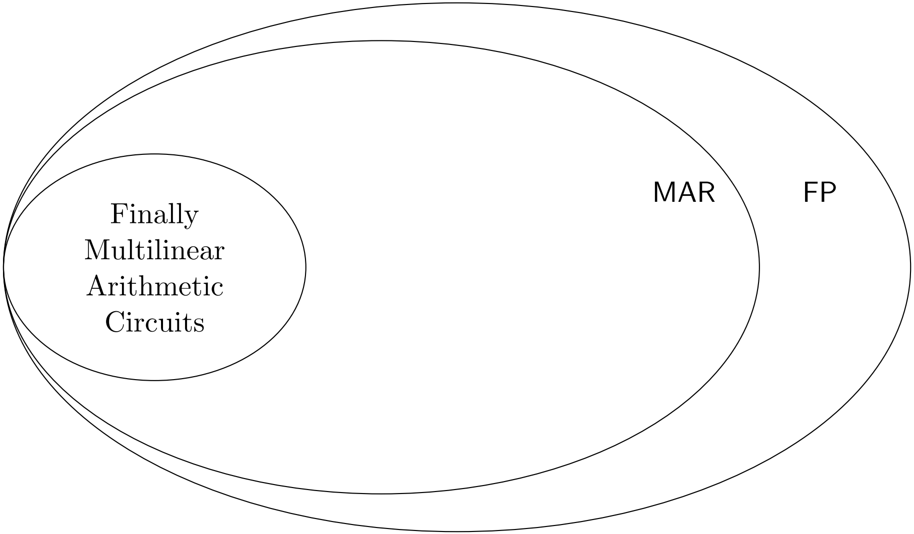
Define $\mathsf{MAR}(f)$, the marginalization problem for f, which on input $m \in \{0,1,*\}^n$, asks for $\sum_{x \in M_m} f(x)$ where $M_m = \{x \in \{0,1\}^n : m_i \in \{0,1\} \implies x_i = m_i\}$.

E.g., the earlier example was an instance of $\mathsf{MAR}(\mathrm{Pr})$ with input $m = 1*$.

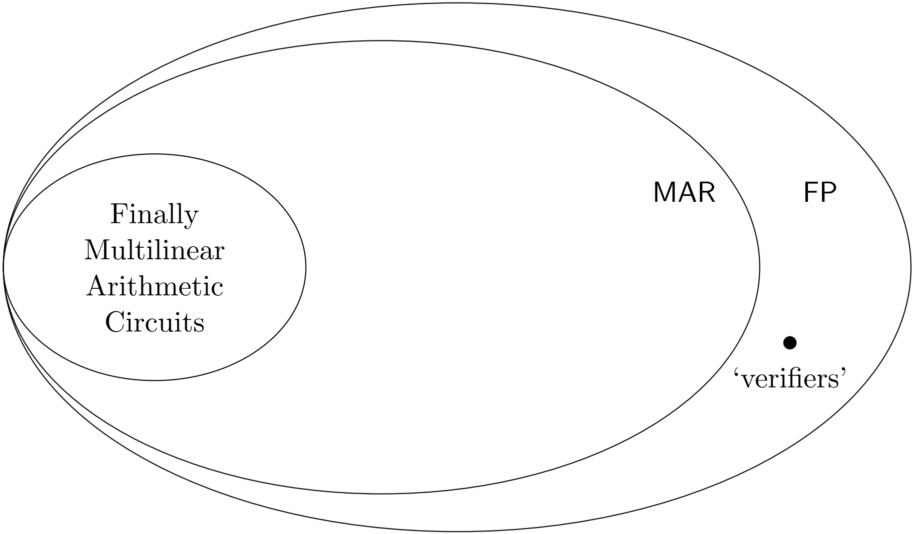
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= \Pr[X_1 = 1, X_2 = 0] + \Pr[X_1 = 1, X_2 = 1] \\ &= 0.3 + 0.4 \\ &= 0.7\end{aligned}$$


Finally
Multilinear
Arithmetic
Circuits
MAR
FP



Main Question: Does every function family with tractable marginalization have uniform finally multilinear arithmetic circuits of polynomial size?



Main Question: Does every function family with tractable marginalization have uniform finally multilinear arithmetic circuits of polynomial size?

Our Approach

Find stronger queries that are tractable for finally multilinear arithmetic circuits

Our Approach

Find stronger queries that are tractable for finally multilinear arithmetic circuits

Evaluate circuit on any real point
(Virtual evidence marginalization)

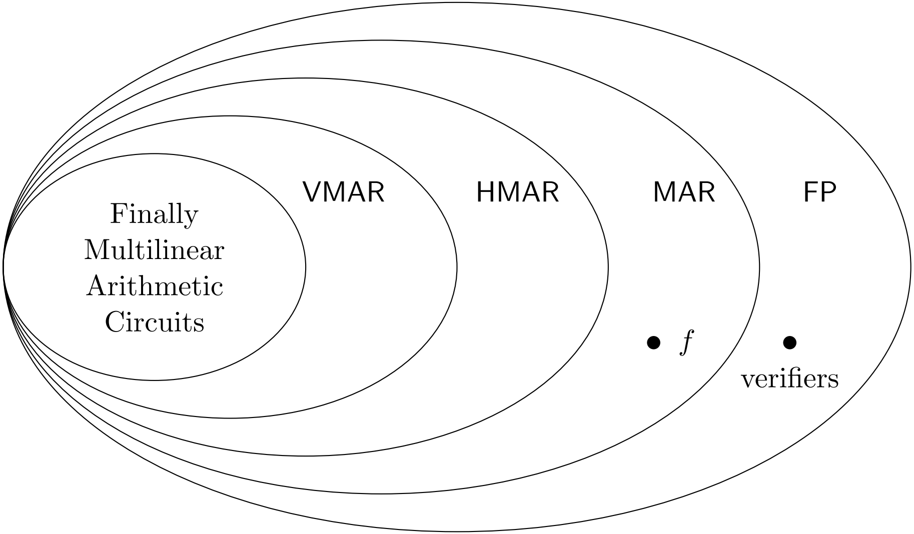
Our Approach

Find stronger queries that are tractable for finally multilinear arithmetic circuits

Evaluate circuit on any real point
(Virtual evidence marginalization)

Sum over inputs of a given Hamming weight
(Hamming weight marginalization)

Our Approach



Hamming weight marginalization

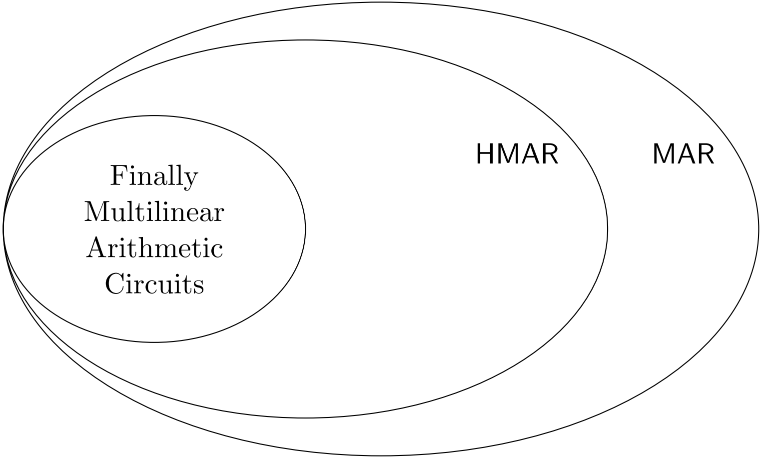
Let $f : \{0,1\}^n \to \{0,1\}$. Define $\mathsf{HMAR}(f)$ which on input $m \in \{0,1,*\}^n$ and $k \in \{0,1,2,\ldots,n\}$, asks for

$$\sum_{x \in M_{m,k}} f(x)$$

where

$$M_{m,k} = \{x \in \{0,1\}^n : (m_i \in \{0,1\} \implies x_i = m_i) \wedge (|x| = k)\}.$$

Hamming weight marginalization



Hamming weight marginalization

x_1	x_2	x_3	x_4	Pr
0	0	0	0	.25
0	1	0	1	.05
1	0	0	0	.05
1	0	0	1	.05
1	0	1	0	.10
1	0	1	1	.50

Input:
$m = 10 * *$
$k = 2$

Hamming weight marginalization

x_1	x_2	x_3	x_4	Pr
0	0	0	0	.25
0	1	0	1	.05
1	0	0	0	.05
1	0	0	1	.05
1	0	1	0	.10
1	0	1	1	.50

Input:
$m = 10 * *$
$k = 2$

Hamming weight marginalization

x_1	x_2	x_3	x_4	Pr
0	0	0	0	.25
0	1	0	1	.05
1	0	0	0	.05
1	0	0	1	.05
1	0	1	0	.10
1	0	1	1	.50

Input:
$m = 10**$
$k = 2$

Output:
$\Pr[1001] + \Pr[1010] =$
$.05 + .10 = .15$

Hamming weight marginalization

x_1	x_2	x_3	x_4	Pr
0	0	0	0	.25
0	1	0	1	.05
1	0	0	0	.05
1	0	0	1	.05
1	0	1	0	.10
1	0	1	1	.50

Prop. FMAC $\subseteq$ HMAR.

Input:
$m = 10**$
$k = 2$

Output:
$\Pr[1001] + \Pr[1010] =$
$.05 + .10 = .15$

Hamming weight marginalization

x_1	x_2	x_3	x_4	Pr
0	0	0	0	.25
0	1	0	1	.05
1	0	0	0	.05
1	0	0	1	.05
1	0	1	0	.10
1	0	1	1	.50

Prop. FMAC $\subseteq$ HMAR.
Use the *network polynomial*:

$$.25\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + .05\bar{x}_1 x_2\bar{x}_3 x_4 + .05 x_1\bar{x}_2\bar{x}_3\bar{x}_4$$
$$+ .05 x_1\bar{x}_2\bar{x}_3 x_4 + .10 x_1\bar{x}_2 x_3\bar{x}_4 + .50 x_1\bar{x}_2 x_3 x_4$$

Input:
$m = 10**$
$k = 2$

Output:
$\Pr[1001] + \Pr[1010] =$
$.05 + .10 = .15$

Hamming weight marginalization

x_1	x_2	x_3	x_4	Pr
0	0	0	0	.25
0	1	0	1	.05
1	0	0	0	.05
1	0	0	1	.05
1	0	1	0	.10
1	0	1	1	.50

Input:
$m = 10**$
$k = 2$

Output:
$\Pr[1001] + \Pr[1010] =$
$.05 + .10 = .15$

Prop. $\mathsf{FMAC} \subseteq \mathsf{HMAR}$.
Use the *network polynomial*:

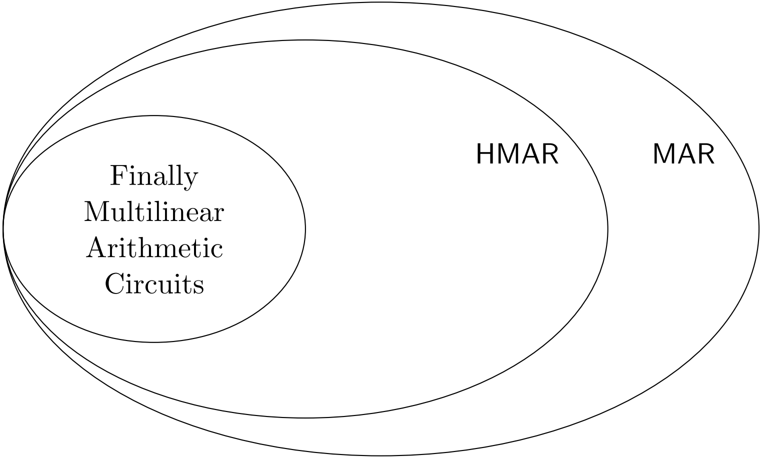
$$.25\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + .05\bar{x}_1 x_2\bar{x}_3 x_4 + .05x_1\bar{x}_2\bar{x}_3\bar{x}_4$$
$$+ .05x_1\bar{x}_2\bar{x}_3 x_4 + .10x_1\bar{x}_2 x_3\bar{x}_4 + .50x_1\bar{x}_2 x_3 x_4$$

Substitute as follows:

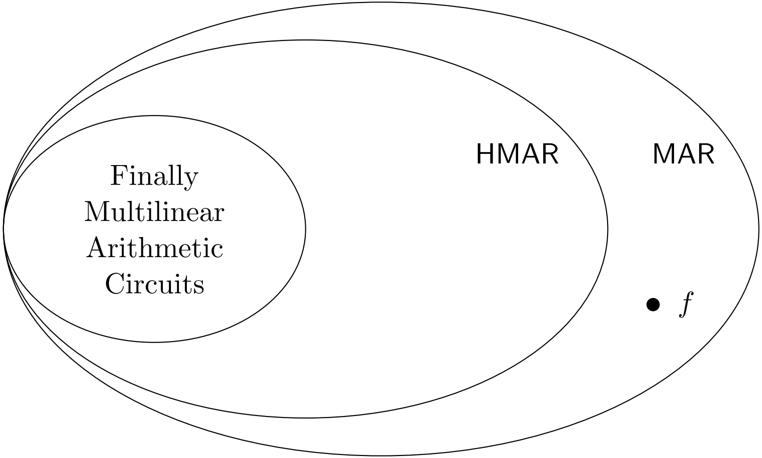
x_1	$\bar{x}_1$	x_2	$\bar{x}_2$	x_3	$\bar{x}_3$	x_4	$\bar{x}_4$
1	0	0	1	t	1	t	1

$$= .25 \cdot 0 \cdot 1 \cdot 1 \cdot 1 + .05 \cdot 0 \cdot 0 \cdot 1 \cdot t + .05 \cdot 1 \cdot 1 \cdot 1 \cdot 1$$
$$+ .05 \cdot 1 \cdot 1 \cdot 1 \cdot t + .10 \cdot 1 \cdot 1 \cdot t \cdot 1 + .50 \cdot 1 \cdot 1 \cdot t \cdot t$$
$$= .05 + .05t + .1t + .50t^2$$
$$= .05 + .15t + .50t^2$$

Hamming weight marginalization



Hamming weight marginalization



A separating example: constraint satisfaction problems

A constraint language Γ is a set of relations each of the form $R \subseteq \{0,1\}^k$ for some $k \geq 1$.

Example:
$\Gamma = \{\text{disjunctions of} \leq 3 \text{ literals}\}$

A separating example: constraint satisfaction problems

A constraint language Γ is a set of relations each of the form $R \subseteq \{0,1\}^k$ for some $k \geq 1$.

A Γ-formula is a conjunction of constraints $R(x_1, \ldots, x_k)$ for $R \in \Gamma$ where $x_1, \ldots, x_k$ are (not necessarily distinct) variables.

Example:
$\Gamma = \{\text{disjunctions of} \leq 3 \text{ literals}\}$

$$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_4 \vee \neg x_5)$$

A separating example: constraint satisfaction problems

A constraint language Γ is a set of relations each of the form $R \subseteq \{0,1\}^k$ for some $k \geq 1$.

A Γ-formula is a conjunction of constraints $R(x_1, \ldots, x_k)$ for $R \in \Gamma$ where $x_1, \ldots, x_k$ are (not necessarily distinct) variables.

Problems:

$\mathsf{CSP}(\Gamma)$

In: a Γ-formula $\phi(x)$

Q: Is there an x that satisfies ϕ?

$k - \mathsf{ONES}(\Gamma)$

In: a Γ-formula $\phi(x)$, $k \in \{0, 1, \ldots, n\}$

Q: Is there an x with k ones that satisfies ϕ?

Example:
$\Gamma = \{\text{disjunctions of} \leq 3 \text{ literals}\}$

$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_4 \vee \neg x_5)$

$\mathsf{CSP}(\Gamma) = \mathsf{3SAT}$

Dichotomy Theorems

A relation is (width-k) *affine* if it is logically equivalent to a system of linear equations over $\mathbb{F}_2$ (each mentioning at most k variables).

For example, $x_1 \oplus x_2 \oplus x_3 = 1$ and $x_3 \oplus x_4 = 0$ form a width-3 affine relation.

Dichotomy Theorems

A relation is (width-k) *affine* if it is logically equivalent to a system of linear equations over $\mathbb{F}_2$ (each mentioning at most k variables).

For example, $x_1 \oplus x_2 \oplus x_3 = 1$ and $x_3 \oplus x_4 = 0$ form a width-3 affine relation.

Theorem (Creignou and Hermann [1996]). *If Γ contains only affine relations, then $\#\mathsf{CSP}(\Gamma)$ is in PTIME. Otherwise, $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete.*

Dichotomy Theorems

A relation is (width-k) *affine* if it is logically equivalent to a system of linear equations over $\mathbb{F}_2$ (each mentioning at most k variables).

For example, $x_1 \oplus x_2 \oplus x_3 = 1$ and $x_3 \oplus x_4 = 0$ form a width-3 affine relation.

Theorem (Creignou and Hermann [1996]). *If Γ contains only affine relations, then $\#\mathsf{CSP}(\Gamma)$ is in PTIME. Otherwise, $\#\mathsf{CSP}(\Gamma)$ is $\#\mathsf{P}$-complete.*

Theorem (Creignou et al. [2010]). *If Γ contains only width-2 affine relations, then $\#k\text{-}\mathsf{ONES}(\Gamma)$ is in PTIME. Otherwise, $\#k\text{-}\mathsf{ONES}(\Gamma)$ is $\#\mathsf{P}$-complete.*

A separating function

$$f(x, y, z) = \bigwedge_{i,j,k \in [n]^3} y_{ijk} \oplus x_i \oplus x_j \oplus x_k \wedge \bigwedge_{i,j,k \in [n]^3} y_{ijk} \oplus z_{ijk}. \tag{1}$$

Theorem. $\mathsf{MAR}(f)$ *is tractable, but* $\mathsf{HMAR}(f)$ *is* $\#\mathsf{P}$*-hard.*

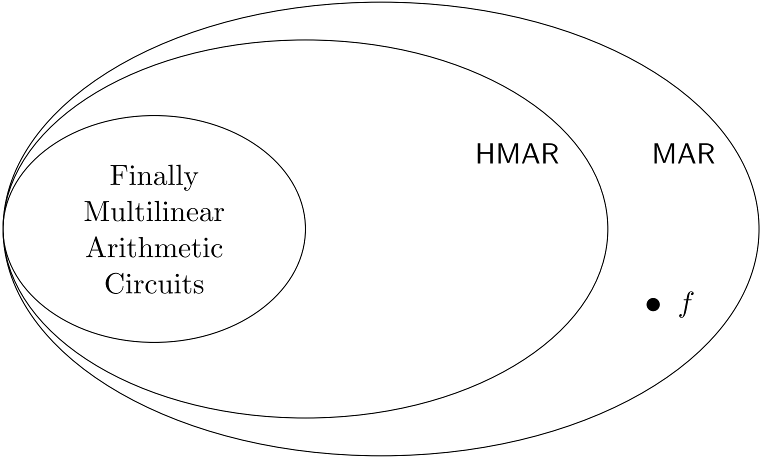
Proof summary: Tractability by Gaussian elimination. Hardness by reducing from $\#k\text{-}\mathsf{ONES}(\Gamma)$ with $\Gamma = \{a \oplus b \oplus c\} = \{\{(0,0,1),(0,1,0),(1,0,0),(1,1,1)\}\}$.

A separating function: hardness of $\mathsf{HMAR}(f)$

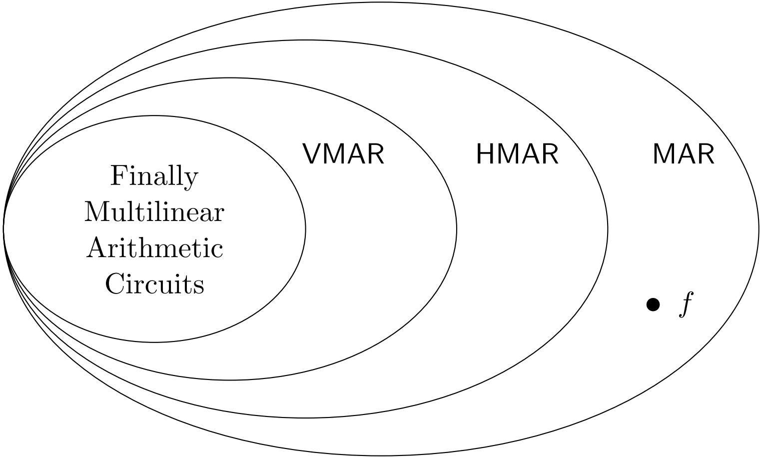
$$f(x,y,z) = \bigwedge_{i,j,k \in [n]^3} y_{ijk} \oplus x_i \oplus x_j \oplus x_k \wedge \bigwedge_{i,j,k \in [n]^3} y_{ijk} \oplus z_{ijk}.$$

Reduction: We get input a Γ-formula $\phi(x_1, \ldots, x_n)$ and an integer $k \in \{0, 1, \ldots, n\}$. For any $x_i \oplus x_j \oplus x_k = 1$ in ϕ, set $y_{ijk} = 0$ and $z_{ijk} = 1$. Call the resulting 'evidence string' m. Then $\#k\text{-}\mathsf{ONES}(\Gamma)(\phi, k) = \mathsf{HMAR}(f)(m, k + n^3)$. Suppose $\phi(x) = 1$; we find the only y and z such that $f(x, y, z) = 1$, and we then observe that $|x, y, z| = |x| + n^3$. Every constraint of ϕ is satisfied, and so every width-4 constraint in f with $y_{ijk} = 0$ is satisfied. For constraints $x_i \oplus x_j \oplus x_k$ not in ϕ, the corresponding width-4 clause in f is satisfied by setting the free variable y_{ijk} to whichever value is necessary. The values z_{ijk} are then set to the opposite of the values of y_{ijk} which satisfies the remaining width-2 clauses of f. For every $i, j, k \in [n]^3$ we have $z_{ijk} \neq y_{ijk}$, and so $|y| + |z| = n^3$.

Update



Update



Virtual evidence marginalization

Let $f : \{0,1\}^n \to \{0,1\}$. Define $\mathsf{VMAR}(f)$ which on input $x_1, \ldots, x_n \in \mathbb{Q}$ outputs $p(x_1, \ldots, x_n)$ where p is the multilinear polynomial computing f.

Hard evidence: observe that $X_i = 0$ or $X_i = 1$.
Virtual evidence instead 'scales' your belief in $X_i = 0$ and $X_i = 1$ by $\bar{\alpha}_i, \alpha_i \geq 0$.
Happens when having noisy measurements of data, many applications ...

Marginalizing with virtual evidence reduces to $\mathsf{VMAR}(f)$:

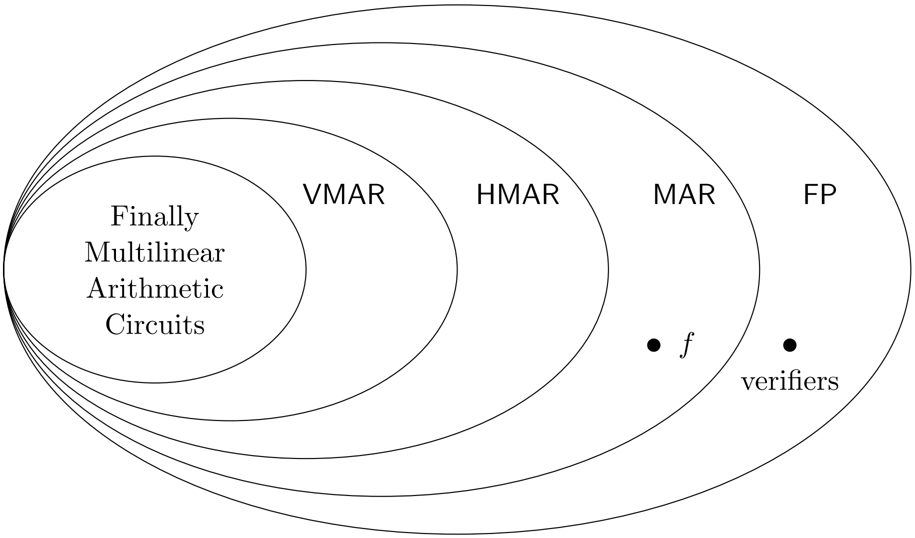
$$\left(\prod_{i=1}^{n} (\alpha_i x_i + \bar{\alpha}_i \bar{x}_i)\right) p\left(\frac{\alpha_1 x_1}{\alpha_1 x_1 + \bar{\alpha}_1 \bar{x}_1}, \ldots, \frac{\alpha_n x_n}{\alpha_n x_n + \bar{\alpha}_n \bar{x}_n}\right) \tag{2}$$

VMAR $\subseteq$ HMAR

Recall our algorithm for HMAR: we substituted the inputs of p to obtain a univariate polynomial of degree at most n

It can therefore be recovered by black-box evaluation at $n + 1$ distinct points

Update


VMAR
HMAR
MAR
FP
Finally Multilinear Arithmetic Circuits
f
verifiers

Are circuits 'complete' for virtual evidence marginalization?

real-RAM: real (and discrete) inputs followed (discrete operations and) by sums, products, and *comparisons* (i.e., $>, =$)

Proposition. *If there is a polynomial time real-RAM for* $\mathsf{VMAR}(f)$, *then there are (uniform) FMACs for* f.

Does a similar 'completeness' hold for Turing machines?
Possibly hard. [Koiran and Perifel, 2011]

Conclusion

Ongoing/open questions:

- More expressive tractable probabilistic models?
- Are finally multilinear circuits ‘complete’ for virtual evidence marginalization?
- Are there more interesting (useful?) queries living between virtual evidence and marginalization?
- Are there non-parallelizable marginalization algorithms (or is marginalization inherently parallelizable?)?

References I

Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Information and computation*, 125(1):1–12, 1996. doi: 10.1006/inco.1996.0016. URL `https://doi.org/10.1006/inco.1996.0016`.

Nadia Creignou, Henning Schnoor, and Ilka Schnoor. Nonuniform boolean constraint satisfaction problems with cardinality constraint. *ACM Trans. Comput. Logic*, 11(4), jul 2010. ISSN 1529-3785. doi: 10.1145/1805950.1805954. URL `https://doi.org/10.1145/1805950.1805954`.

Adnan Darwiche. A differential approach to inference in bayesian networks. *J. ACM*, 50(3):280–305, may 2003. ISSN 0004-5411. doi: 10.1145/765568.765570. URL `https://doi.org/10.1145/765568.765570`.

Pascal Koiran and Sylvain Perifel. Interpolation in valiant's theory. *Computational Complexity*, 20:1–20, 2011.

James Martens and Venkatesh Medabalimi. On the expressive efficiency of sum product networks, 2015.

References II

Dan Roth and Rajhans Samdani. Learning multi-linear representations of distributions for efficient inference. *Machine Learning*, 76(2):195–209, 2009. doi: 10.1007/s10994-009-5130-x. URL `https://doi.org/10.1007/s10994-009-5130-x`.

Volker Strassen. Vermeidung von divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973. URL `http://eudml.org/doc/151394`.

Zhongjie Yu, Martin Trapp, and Kristian Kersting. Characteristic circuit. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Honghua Zhang, Brendan Juba, and Guy Van den Broeck. Probabilistic generating circuits. In *International Conference on Machine Learning*, pages 12447–12457. PMLR, 2021.