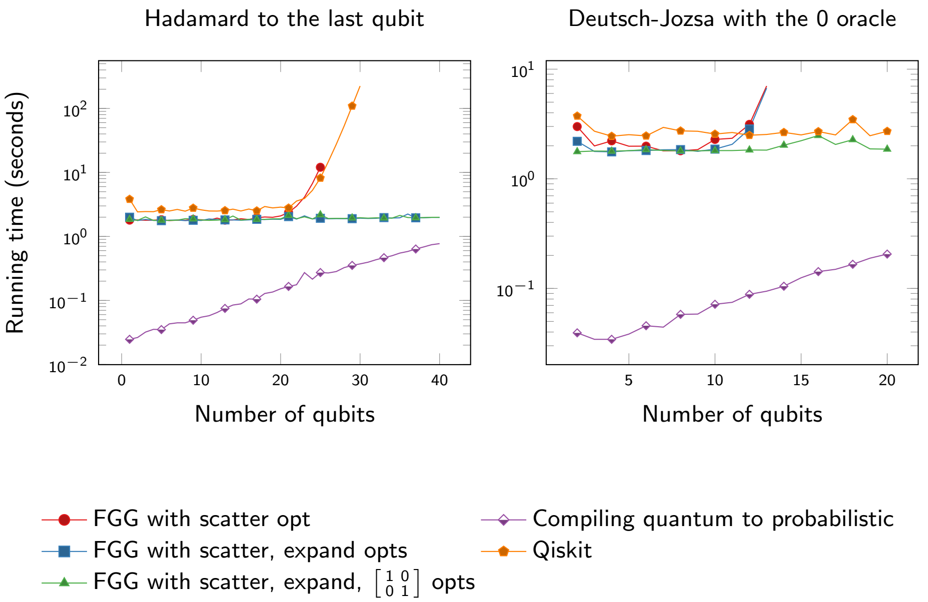
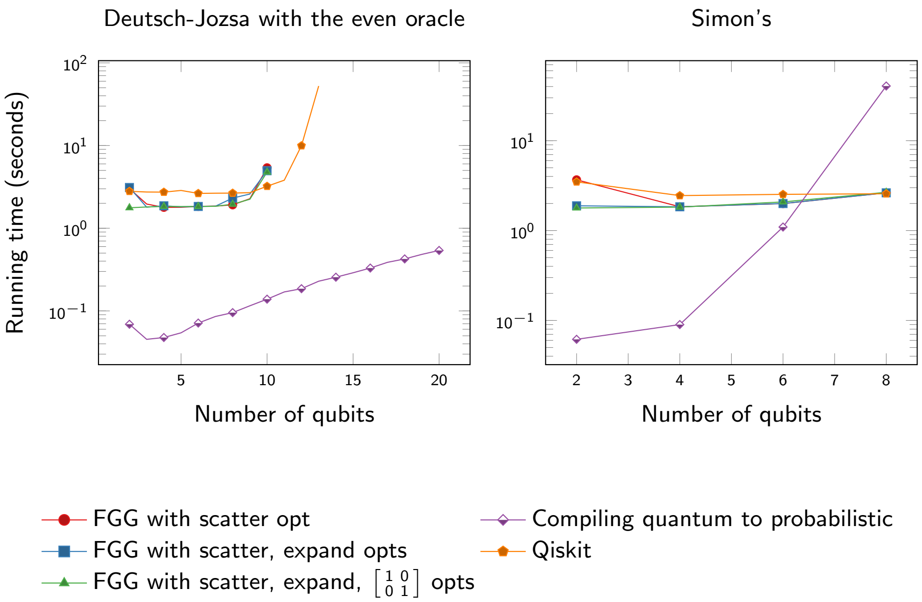
Simulating Quantum Computation Using Probabilistic Programming

Yafei Yang and Chung-chieh Shan, Indiana University

December 2025



Hadamard to the last qubit
Deutsch-Jozsa with the 0 oracle
Running time (seconds)
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
0
10
20
30
40
5
15
Number of qubits
FGG with scatter opt
FGG with scatter, expand opts
FGG with scatter, expand, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ opts
Compiling quantum to probabilistic
Qiskit



Deutsch-Jozsa with the even oracle
Simon's
Running time (seconds)
10^{2}
10^{1}
10^{0}
10^{-1}
5
10
15
20
2
3
4
5
6
7
8
Number of qubits
Number of qubits
FGG with scatter opt
FGG with scatter, expand opts
FGG with scatter, expand, $\left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$ opts
Compiling quantum to probabilistic
Qiskit

Motivations

Higher-level quantum programming

- Capture nondeterministic intuition
- Scale up from qubit tyranny
- Compose quantum computations

Simulation using probabilistic programming

- Reveal shared techniques
- Reuse existing implementation
- Relate quantum and probabilistic programming

Source language: Symmetric pattern matching

Iso	ω	$::= \{\mid v \leftrightarrow e \cdots \mid v \leftrightarrow e\}$
Extended value	e	$::= v \mid e + e \mid \alpha e \mid \mathsf{let}\ p = \omega p\ \mathsf{in}\ e$
Pure value	v	$::= \langle\rangle \mid x \mid \langle v, v\rangle \mid \mathsf{inl}\ v \mid \mathsf{inr}\ v$
Pattern	p	$::= \langle\rangle \mid x \mid \langle p, p\rangle$
Variable	x	
Weight	$\alpha \in \mathbb{C}$	

$$\texttt{Hadamard} :: \mathbb{1} \oplus \mathbb{1} \leftrightarrow \mathbb{1} \oplus \mathbb{1}$$

$$\left\{\begin{array}{l} \mid \mathsf{inl}\langle\rangle \leftrightarrow \frac{1}{\sqrt{2}}(\mathsf{inl}\langle\rangle) + \frac{1}{\sqrt{2}}(\mathsf{inr}\langle\rangle) \\ \mid \mathsf{inr}\langle\rangle \leftrightarrow \frac{1}{\sqrt{2}}(\mathsf{inl}\langle\rangle) - \frac{1}{\sqrt{2}}(\mathsf{inr}\langle\rangle) \end{array}\right\} \qquad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

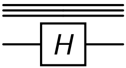
Source language: Symmetric pattern matching

Iso	ω	$::=$	$\{\mid v \leftrightarrow e \cdots \mid v \leftrightarrow e\}$
Extended value	e	$::=$	$v \mid e + e \mid \alpha e \mid \text{let } p = \omega p \text{ in } e$
Pure value	v	$::=$	$\langle\rangle \mid x \mid \langle v, v\rangle \mid \text{inl } v \mid \text{inr } v$
Pattern	p	$::=$	$\langle\rangle \mid x \mid \langle p, p\rangle$
Variable	x		
Weight	$\alpha \in \mathbb{C}$		

$\texttt{sndHadamard} :: a \otimes (\mathbb{1} \oplus \mathbb{1}) \leftrightarrow a \otimes (\mathbb{1} \oplus \mathbb{1})$

$\{\mid \langle x, y\rangle \leftrightarrow \text{let } z = \texttt{Hadamard}\, y \text{ in } \langle x, z\rangle\}$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & & & & & \\ 1 & -1 & & & & & \\ & & 1 & 1 & & & \\ & & 1 & -1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & 1 \\ & & & & & 1 & -1 \end{bmatrix}$$



Source language: Symmetric pattern matching

Iso	ω	$::= \{\mid v \leftrightarrow e \cdots \mid v \leftrightarrow e\}$
Extended value	e	$::= v \mid e + e \mid \alpha e \mid \mathsf{let}\ p = \omega p\ \mathsf{in}\ e$
Pure value	v	$::= \langle\rangle \mid x \mid \langle v, v\rangle \mid \mathsf{inl}\ v \mid \mathsf{inr}\ v$
Pattern	p	$::= \langle\rangle \mid x \mid \langle p, p\rangle$
Variable	x	
Weight	$\alpha \in \mathbb{C}$	

`Hadamard*` $:: a \oplus a \leftrightarrow a \oplus a$

$$\left\{\begin{array}{l} \mid \mathsf{inl}\ x \leftrightarrow \frac{1}{\sqrt{2}}(\mathsf{inl}\ x) + \frac{1}{\sqrt{2}}(\mathsf{inr}\ x) \\ \mid \mathsf{inr}\ x \leftrightarrow \frac{1}{\sqrt{2}}(\mathsf{inl}\ x) - \frac{1}{\sqrt{2}}(\mathsf{inl}\ x) \end{array}\right\}$$

$$\frac{1}{\sqrt{2}}\begin{array}{c} \left[\begin{array}{cc} \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} \\ \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & \begin{matrix} -1 & & \\ & \ddots & \\ & & -1 \end{matrix} \end{array}\right] \\ \underbrace{\qquad\qquad}_{|a|}\ \underbrace{\qquad\qquad}_{|a|} \end{array} \begin{array}{l} \left.\vphantom{\begin{matrix}1\\ \ddots\\ 1\end{matrix}}\right\} |a| \\ \left.\vphantom{\begin{matrix}1\\ \ddots\\ 1\end{matrix}}\right\} |a| \\ \vphantom{\underbrace{}_{|a|}} \end{array}$$

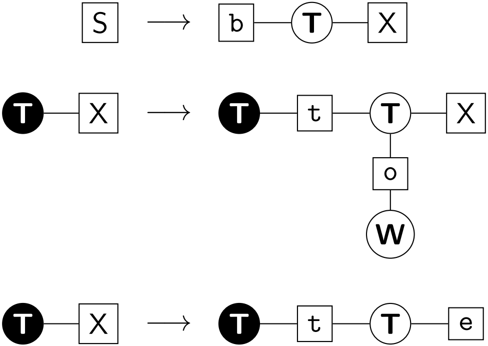
Source language: Symmetric pattern matching

$$\begin{array}{lrcl}
\text{Iso} & \omega & ::= & \{\mid v \leftrightarrow e \cdots \mid v \leftrightarrow e\} \\
\text{Extended value} & e & ::= & v \mid e + e \mid \alpha e \mid \text{let } p = \omega p \text{ in } e \\
\text{Pure value} & v & ::= & \langle\rangle \mid x \mid \langle v, v\rangle \mid \text{inl } v \mid \text{inr } v \\
\text{Pattern} & p & ::= & \langle\rangle \mid x \mid \langle p, p\rangle \\
\text{Variable} & x & & \\
\text{Weight} & \alpha \in \mathbb{C} & &
\end{array}$$

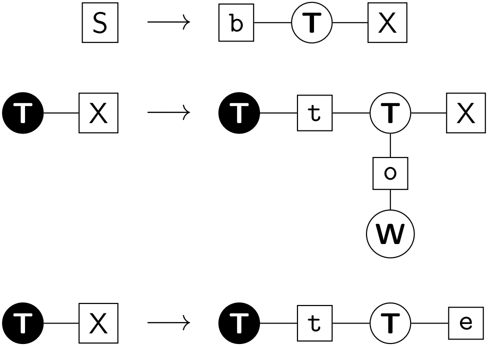
$$\begin{array}{l}
\texttt{swap} :: a \oplus b \leftrightarrow b \oplus a \\
\left\{\begin{array}{l}
\mid \text{inl } x \leftrightarrow \text{inr } x \\
\mid \text{inr } y \leftrightarrow \text{inl } y
\end{array}\right\}
\end{array}$$

$$\begin{array}{cl}
\left[\begin{array}{cc}
 & \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} \\
\begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} &
\end{array}\right] &
\begin{array}{l}
\left.\vphantom{\begin{matrix}1\\ \ddots\\ 1\end{matrix}}\right\}|a| \\
\left.\vphantom{\begin{matrix}1\\ \ddots\\ 1\end{matrix}}\right\}|b|
\end{array} \\
\begin{array}{cc}\underbrace{\hphantom{1\ \ddots\ 1}}_{|b|} & \underbrace{\hphantom{1\ \ddots\ 1}}_{|a|}\end{array} &
\end{array}$$

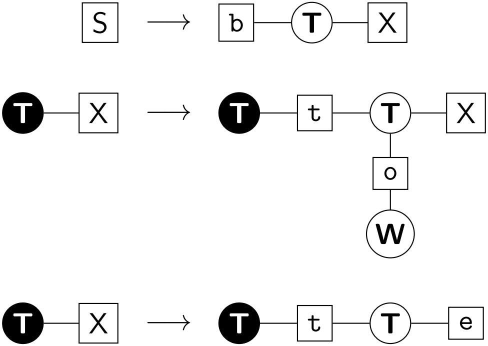
Target language: Factor graph grammar

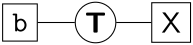


Target language: Factor graph grammar

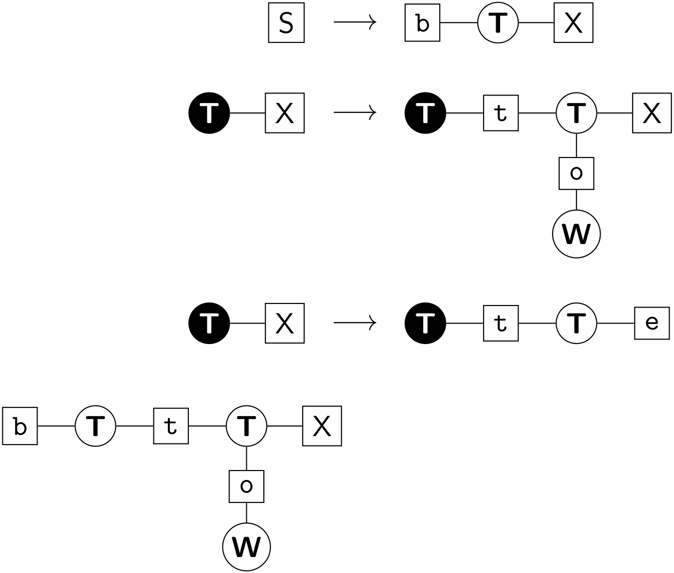


Target language: Factor graph grammar

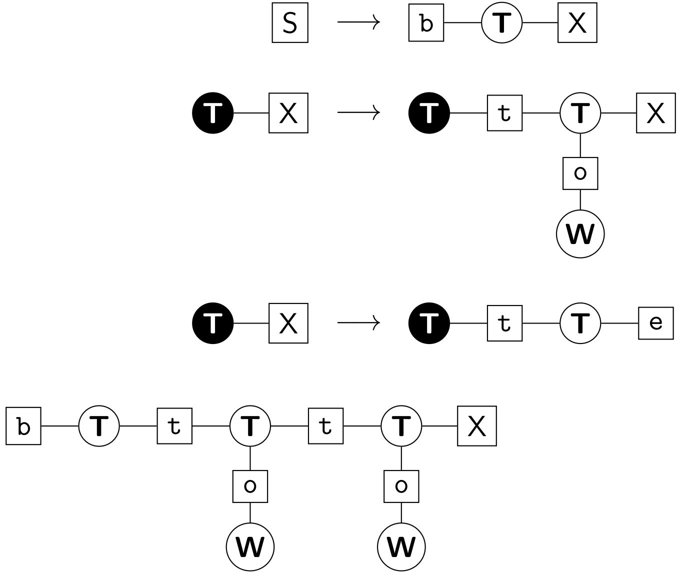




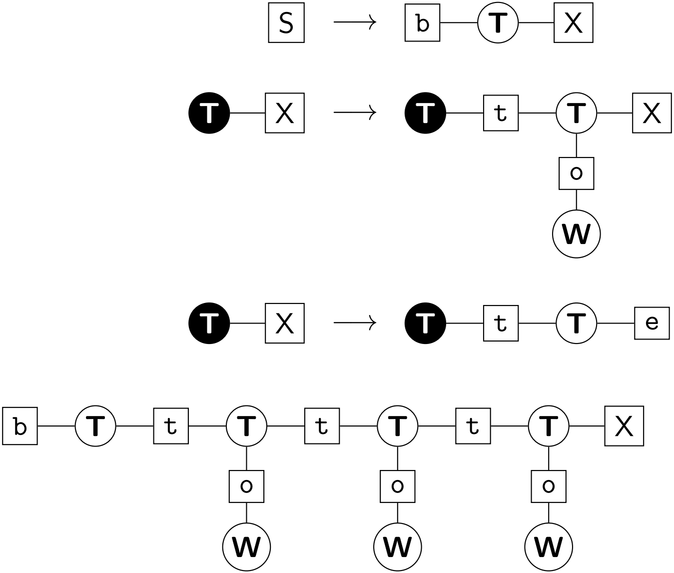
Target language: Factor graph grammar



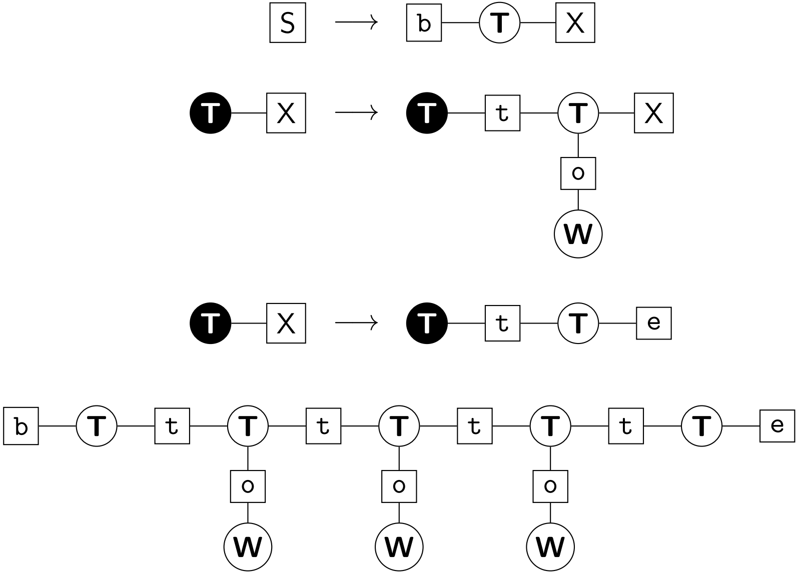
Target language: Factor graph grammar



Target language: Factor graph grammar



Target language: Factor graph grammar



Target language: Factor graph grammar

Factor graph grammar	fgg	$::= (r = E)\ldots$
Tensor expression	E, F	$::= r \mid \alpha \mid E + E \mid 0^{(s,\ldots)}$
		$\mid \text{expand}(E[\vec{x}] \to \vec{z})$
		$\mid \text{einsum}(E[\vec{x}], \ldots \to \vec{z})$
		$\mid \text{scatter}(E[\vec{x}] \to e \cdots)$
Tensor variable	r	
Weight	$\alpha \in \mathbb{C}$	
Upper bound	$s \in \mathbb{N}$	

$$\text{coin} = \frac{2}{3} \cdot \begin{bmatrix} 1 & 0 \end{bmatrix} + \frac{1}{3} \cdot \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\text{same} = \text{coin} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \text{coin}^t$$

Target language: Factor graph grammar

$$
\begin{array}{llcl}
\text{Factor graph grammar} & \mathit{fgg} & ::= & (r = E)\ldots \\
\text{Tensor expression} & E, F & ::= & r \mid \alpha \mid E + E \mid 0^{(s,\ldots)} \\
 & & \mid & \operatorname{expand}(E[\vec{x}] \to \vec{z}) \\
 & & \mid & \operatorname{einsum}(E[\vec{x}], \ldots \to \vec{z}) \\
 & & \mid & \operatorname{scatter}(E[\vec{x}] \to e \cdots) \\
\text{Index variable sequence} & \vec{x}, \vec{y}, \vec{z} & ::= & x^{(s)} \ldots \\
\text{Index variable} & & x^{(s)}, y^{(s)}, z^{(s)}, t^{(s)} & \\
\text{Upper bound} & & s \in \mathbb{N} &
\end{array}
$$

$$
\operatorname{expand}(\begin{bmatrix} 6 & 7 \end{bmatrix} [x^{(2)}] \to x^{(2)} y^{(3)}) = \begin{bmatrix} 6 & 6 & 6 \\ 7 & 7 & 7 \end{bmatrix}
$$

$$
\operatorname{einsum}(E_1[xy], E_2[yz] \to xz) = \text{matrix multiplication}
$$

Target language: Factor graph grammar

Factor graph grammar	fgg	$::=$	$(r = E)\ldots$
Tensor expression	E, F	$::=$	$r \mid \alpha \mid E + E \mid 0^{(s,\dots)}$
			$\mid \operatorname{expand}(E[\vec{x}] \to \vec{z})$
			$\mid \operatorname{einsum}(E[\vec{x}], \ldots \to \vec{z})$
			$\mid \operatorname{scatter}(E[\vec{x}] \to e \cdots)$
Index variable sequence	$\vec{x}, \vec{y}, \vec{z}$	$::=$	$x^{(s)} \ldots$
Index variable	$x^{(s)}, y^{(s)}, z^{(s)}, t^{(s)}$		
Axis expression	e	$::=$	$x^{(s)} \mid \mathbb{1} \mid e \otimes e \mid s \oplus e \oplus s$

$$\operatorname{scatter}(\begin{bmatrix}6 & 7\end{bmatrix}[x] \to xx) = \begin{bmatrix}6 & 0 \\ 0 & 7\end{bmatrix}$$

$$\operatorname{scatter}(\begin{bmatrix}5 & 6 \\ 7 & 8\end{bmatrix}[xy] \to (x \otimes y)) = \begin{bmatrix}5 & 6 & 7 & 8\end{bmatrix}$$

$$\operatorname{scatter}(\begin{bmatrix}6 & 7\end{bmatrix}[x] \to (2 \oplus x \oplus 3)) = \begin{bmatrix}0 & 0 & 6 & 7 & 0 & 0 & 0\end{bmatrix}$$

Compilation preserves denotational semantics

$$\llbracket \Delta \vdash_{\mathrm{t}} t : a \rrbracket (\delta, v) = \llbracket comp(\Delta \vdash_{\mathrm{t}} t : a) \rrbracket_{\gamma} \left[\left(comp_{\delta}(\Delta, \delta), comp_{v}(a, v) \right) \right]$$

$$\llbracket \vdash_{\mathrm{i}} \omega : a \leftrightarrow b \rrbracket (v_1, v_2) = \llbracket comp(\vdash_{\mathrm{i}} \omega : a \leftrightarrow b) \rrbracket_{\gamma} \left[\left(comp_{v}(a, v_1), comp_{v}(b, v_2) \right) \right]$$

Three optimizations

$$\llbracket \mathrm{einsum}(\mathrm{scatter}(E[x] \to xx)[xy], F[yz] \to xz) \rrbracket$$
$$= \llbracket \mathrm{einsum}(E[y], F[yz] \to yz) \rrbracket$$

$$\llbracket \mathrm{einsum}(\mathrm{expand}(E[xy] \to xyt)[xyt], F[yz] \to xyzt) \rrbracket$$
$$= \llbracket \mathrm{expand}(\mathrm{einsum}(E[xy], F[yz] \to xyz)[xyz] \to xyzt) \rrbracket$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \llbracket \mathrm{scatter}(\mathrm{expand}(1[] \to x)[x] \to xx) \rrbracket$$

Conclusion

- Compile quantum to probabilistic (proven correct)
- Get quantum simulator for free
- Optimize summation-free einsum-expand (proven correct)

Future work

- Optimize non-summation-free einsum-expand
- Higher-order isos and recursion
- Polymorphism; partial evaluation
- More benchmarks of expressivity and performance