AAQDD – Abstract additive quantum decision diagrams

M. Leroy, R. Vilmart

February 8, 2024

1 Formal definition

1.1 Abstract states

The purpose of quantum decision diagrams is to provide a more efficient way to store and manipulate quantum states of a finite number of qubits. A n-qubit state is indeed traditionally represented as an element of $\mathbb{C}^{2^n}$ (with norm 1), which takes exponential space as n grows. Abstract states will be in this part defined similarly, but with complex intervals instead of complex numbers.

The standard definition of real intervals is:

$$\forall a, b \in \mathbb{R}, [a, b] = \{x \in \mathbb{R} / \min(a, b) \leq x \leq \max(a, b)\}$$

Those can be generalised to complex intervals, commonly using the cartesian notation.

$$\forall x, y \in \mathbb{C}, [x, y] = \{a + ib; a \in [\Re(x), \Re(y)], b \in [\Im(x), \Im(y)]\}$$

Now let $\mathcal{A}_0 = \{[x, y]; x, y \in \mathbb{C}\}$, we can now define basic operations: sum, product, and join. We note that these definition do not depend on the type of complex interval chosen.

$$\forall \alpha, \beta \in \mathcal{A}_0, \alpha + \beta = \{a + b; a \in \alpha, b \in \beta\}$$

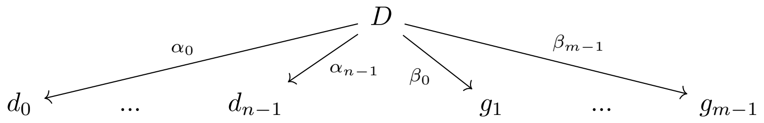
$$\forall \alpha, \beta \in \mathcal{A}_0, \alpha * \beta = \{a * b; a \in \alpha, b \in \beta\}$$

$$\forall \alpha, \beta \in \mathcal{A}_0, \alpha \sqcup \beta = \bigcap_{\alpha \subset \gamma \text{ and } \beta \subset \gamma} \gamma$$

We now have intervals, *abstract elements* of $\mathbb{C}$ represented in $\mathcal{A}_0$. Our abstract elements for a n-qubit quantum state would be in $\mathcal{A}_n = {\mathcal{A}_0}^{2^n}$ for all $n \in \mathbb{N}$. Defining a sum in $\mathcal{A}_n$, and an external product $\alpha * A$ for $\alpha \in \mathcal{A}_0$ and $A \in \mathcal{A}_n$, comes easily.

1.2 Decision diagrams

We inductively define abstract additive quantum decision diagrams, starting from zero-depth decision. The only zero-depth diagram is $\boxed{1}$. Let for every set E be the set of finite subsets of E: $\mathscr{P}_f(E) = \{A \subset E / |A| < \infty\}$. If the set $\mathcal{D}_n$ of diagrams of depth n is defined, $n+1$-depth diagrams can have a finite number of left children in $\mathcal{D}_n$ and a finite number of right children in $\mathcal{D}_n$, each being associated with an abstract amplitude in $\mathcal{A}_0$.



Defining $\mathcal{D}_{n+1} = \mathscr{P}_f(\mathcal{A}_0 \times \mathcal{D}_n) \times \mathscr{P}_f(\mathcal{A}_0 \times \mathcal{D}_n)$ thus comes naturally.

1.3 Diagram evaluation

Now that we defined our decision diagrams, we can evaluate them to get abstract elements. We inductively define our evaluation function for n quibits $\mathcal{E}_n : \mathcal{D}_n \to \mathcal{A}_n$:

$$\mathcal{E}_0(\boxed{1}) = \{1\}$$

$$\forall D, G \in \mathscr{P}_f(\mathcal{A}_0 \times \mathcal{D}_n), \mathcal{E}_{n+1}(D, G) = \begin{pmatrix} \sum\limits_{(\alpha,g)\in G} \alpha * \mathcal{E}_n(g) \\ \sum\limits_{(\beta,d)\in D} \beta * \mathcal{E}_n(d) \end{pmatrix} \quad \text{with}$$

Since there is no risk of ambiguity, defining $\mathcal{E} : \bigcup \mathcal{D}_n \to \bigcup \mathcal{A}_n$ is not problematic. With this last function, we can now evaluate all our AAQDDs.

2 Reduction algorithm

We note that multiple QDDs can be evaluated to the same abstract state. This part will aim to provide an algorithm to decrease the "size" of diagrams while not breaking their evaluation by $\mathcal{E}$. The size of a AAQDD is the number of intervals (counting them with their multiplicity). We would want, from a diagram D, to get a diagram D' such that size$(D) >$ size(D') and $\mathcal{E}(D) \subset \mathcal{E}(D')$ (the reduction is smaller in size and might be less precise that the original diagram).

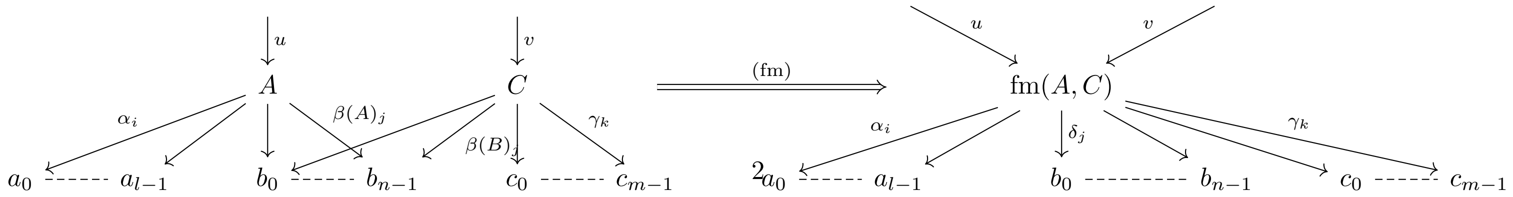
We developed a reduction formula to force the merging of two nodes. This is permitted both because of abstract interpretation (to merge without amplitudes being colinear) and The cost is zero.

2.1 Simple case

To begin, let's consider the simple case where all diagrams only have left children (this case would be useless in practice because it would result in only one interval and zeros). Let's say we want to merge the two diagrams:

$$A = (\{(\alpha_0, a_0), ..., (\alpha_{l-1}, a_{l-1}), (\beta(A)_0, b_0), ..., (\beta(A)_{n-1}, b_{n-1})\}, \emptyset) \quad \text{and}$$
$$C = (\{(\beta(C)_0, b_0), ..., (\beta(C)_{n-1}, b_{n-1}), (\gamma_0, c_0), ..., (\gamma_m, c_{m-1})\}, \emptyset)$$

with $\{a_0, ..., a_{l-1}\} \cap \{c_0, ..., c_{m-1}\} = \emptyset$. A graphic representation of our merging formula would be:



with $\delta_j = \beta(A)_j \sqcup \beta(B)_j$. More formally, the merging formula would be:

$$\mathrm{merge}(A, C) = (\{(\alpha_0, a_0), ..., (\alpha_{l-1}, a_{l-1}), (\delta_0, b_0), ..., (\delta_{n-1}, b_{n-1}), (\gamma_0, c_0), ..., (\gamma_m, c_{m-1})\}, \emptyset)$$