Mathematical methods for resource-based type theories

Aarthi Sundaram
Joint Center for Quantum Information and Computer Science, University of Maryland, College Park
aarthims@umd.edu

Brad Lackey
Joint Center for Quantum Information and Computer Science, University of Maryland, College Park
Institute for Advanced Computer Studies, University of Maryland, College Park
Departments of Computer Science and Mathematics, University of Maryland, College Park
Quantum Architectures and Computation Group, Microsoft, Redmond
bclackey@umd.edu

With the wide range of quantum programming languages on offer now, efficient program verification and type checking for these languages presents a challenge – especially when classical debugging techniques may affect the states in a quantum program. In this work, we make progress towards a program verification approach using the formalism of operational quantum mechanics and resource theories. We present a logical framework that captures two mathematical approaches to resource theory based on monoids (algebraic) and monoidal categories (categorical). We develop the syntax of this framework as an intuitionistic sequent calculus, and prove soundness and completeness of an algebraic and categorical semantics that recover these approaches. We also provide a cut-elimination theorem, normal form, and analogue of Lambek's lifting theorem for polynomial systems over the logics. Using these approaches along with the Curry-Howard-Lambek correspondence for programs, proofs and categories, this work lays the mathematical groundwork for a type checker for some resource theory based frameworks, with the possibility of extending it other quantum programming languages.

(The full version of this paper is also available at arXiv: 1812.08726)

In recent years we have seen an increasing number of practical quantum programming languages, from the high-level ones like Quipper [1] and QWIRE [2] to those that are hardware/simulator specific like Q# [3], Quil [4] and QISKit [5]. These afford users the ability to implement ever larger quantum circuits, at which point the question of checking that such quantum programs behave as intended takes prominence. One of the challenges is that, unlike classical programs, if a debugger observes the system state during program execution, it could cause the quantum state to destroyed. Moreover, characterizing the state of a quantum register is generally intractable for all but the smallest programs. The other option is to verify the program by ascertaining its correctness in a formal model. For instance QWIRE uses density matrix and circuit formalisms [6] to verify a program, but this leads to similar scalability issues. In this work we do not attempt to provide methods for full program verification, but focus on the more limited task of type checking.

Our approach uses the formalism of operational quantum mechanics via the framework of categorical quantum mechanics [7]. As the name suggest, categorical quantum mechanics, introduced by Abramsky and Coecke [7], views quantum mechanics through the lens of category theory. Specifically, it is modeled as a theory of systems, processes and their interactions i.e. as a resource theory. Further, the processes are partitioned into being free or resource conversions and objects could either be *free objects* or *resource*

Submitted to:
QPL 2019

states. The resourcefulness of a state is characterized by the monotones of a partial order applied to the states.

Example 1 (Bipartite entanglement as a resource). Two spatially separated parties, Alice and Bob, have access to local quantum computation on their respective systems together with arbitrary classical communication (LOCC), see for instance [8]. The expensive resource in this case is their use of *bipartite entanglement* while performing any task. The resource theory is then defined with: (i) separable states being free; (ii) entangled states being the resource states; (iii) LOCC operations are the free processes; (iv) the entropy of the reduced density matrix on Alice's (or Bob's) qubit represents the partial order and monotones applied to the states. △

Beyond resource theory, the "categorification" of quantum foundations [9–11], semantics [12–15], protocols [7, 16], and computation [17, 18] has led to quantum information being treated as a form of generalized probability theory [19]. From this perspective graphical languages for quantum computation [15,20] have arisen where the monoidal category [21] forms the centerpiece. An intriguing aspect of this framework is its flexibility for designing programs and proofs. For example, a diagrammatic approach was recently used to show the self-testing property of a multipartite quantum state [22].

As a resource theory, categorical quantum mechanics can be structurally defined in terms of symmetric monoidal categories[1] (SMC) [21]. To validate a program at the level of its types, there are two broad tasks to be accomplished: (i) check that the processes used take in the correct types/resources and all type conversions are valid; (ii) ensure that the processes are composed in a way that respects the rules of a monoidal category.

Example 2 (Bipartite entanglement as a type theory). In the resource theory LOCC, separated parties Alice and Bob have arbitrary local quantum operations and arbitrary classical communication. These lead to fundamental types[2] Q_A, Q_B, and C representing a qubit collocated with Alice, a qubit collocated with Bob, and a classical bit (as LOCC allows arbitrary classical communication, any classical information can be considered shared by the two parties). Note that qubits of type Q_A cannot be cast to type Q_B as quantum communication is not free. Resourceful states also become fundamental types. For example type E, representing two maximally entangled qubits, is a different type than $\mathsf{Q}_A \otimes \mathsf{Q}_B$ as states of this latter type are free. In fact, we could consider E as one in a type class parametrized by an entanglement measure (i.e. a resource monotone in LOCC). △

Resource-based reasoning is not new, the most popular being separation logic/bunched implications [23, 24] and linear logic/geometry of interactions [25, 26]. In computer science, the former has been used extensively for concurrence [27] while the later has proven more popular in quantum information [28]. Quantum thermodynamics has a long history as a resource theory, see for example [29–31]. Similarly, many ideas from quantum foundations have been recast as resource theories including purity [32], entanglement [33,34], coherence [35,36], contextuality [37,38], and nonlocality [39,40]. Foundational works on quantum information as a general resource theory such as [41, 42] have led to mathematical frameworks for abstract resource theories as monoids [43] or as monoidal categories [44, 45].

[1] A symmetric monoidal category is one equipped with a symmetric bifunctor that is associative upto a natural transformation and an object I that is both a left and right identity for the bifunctor.

[2] For simplicity we will take bits and qubits; types corresponding to arbitrary classical or quantum systems can equally be considered.

In this work, we present a type theory for program verification by building the minimal logical fragment that captures axiomatic resource theories. We refer to this logic as $\mathbb{T}$ for *tensor*, which contains only a nullary tautology $\mathbb{1}$ and a multiplicative conjunction $\otimes$. The term "multiplicative" refers to the position of $\mathbb{T}$ within linear logic or the logic of bunched implications. Opposed to this are "additive" logics, where propositions represent properties rather than resources; these are better suited for classical systems. We feel this is justified as taking, for instance, the whole of linear logic would suppose structures of our resource theory that need not exist. Namely, a sound categorical semantics for a linear logic necessarily forms a $*$-autonomous category [46], while a general resource theory does not suppose any form of duality or adjunctions. We do not even propose an implication in $\mathbb{T}$, which would recover closed categories in our semantics. Rather we follow a stricter Curry-Howard-Lambek[3] correspondence: objects correspond to terms while morphisms correspond to proofs in our logic. This is our first step towards developing strong quantum type checkers for quantum programming languages using categorical quantum mechanics.

Defining the deductive system. We consider a deductive system $\mathbb{T}$ comprised only of a multiplicative conjunction $\otimes$ as a connective. Intuitively, this connective is analogous to the tensor product in a symmetric monoidal category. We formally develop the syntax of this logic[4] as an intuitionistic logic using sequent calculus as given in Table 1. Here, the alphabets A,B,C denote *terms* and the greek alphabets Γ,Δ,Θ denote *sequents* i.e. a (possibly infinite) multi-set of terms. The various transformations that define the equivalence classes for the proofs in this logic are also described. To show that this logic is consistent we show a cut-elimination theorem [47] for system $\mathbb{T}$.

Result 1. *If $\Gamma \vdash A$ has a proof in $\mathbb{T}$, then it has a proof in $\mathbb{T}$ that does not use the Cut rule.*

One consequence of this is that we can show that every proof reduces to a *normal form* in a finite number of steps and hence, $\mathbb{T}$ is decidable.

Polynomial Systems. A critical feature of resource theories is the ability to deem some resources as free. There is no natural mechanism to express this in $\mathbb{T}$, and so we introduce polynomial systems in the spirit of [48]. This formalism splits free resources into two types: freely available or freely disposable. Logically, this translates to a term that can be arbitrarily added to the consequent or antecedent respectively of any proof statement[5] $\Gamma \vdash A$. A free resource in the usual sense has both properties. Formally, X is freely available in the system $\mathbb{T}[X]$, whose has the deduction rules of $\mathbb{T}$ with the additional rule $\vdash X$.

[3]The Curry-Howard-Lambek correspondence highlights the relationship between computer programs, mathematical proofs and category theory.

[4]Notice that this system is a fragment of Multiplicative Intuitionistic Linear Logic [25].

[5]In this notation, Γ is the consequent and A is the antecedent of this statement.

$\dfrac{}{P \vdash P}\ \text{Id}$	$\dfrac{\Gamma \vdash B \quad \Delta, B, \Theta \vdash A}{\Delta, \Gamma, \Theta \vdash A}\ \text{Cut}$	$\dfrac{\Gamma, \Delta \vdash A}{\Gamma, \mathbb{1}, \Delta \vdash A}\ \text{L-}\mathbb{1}$
$\dfrac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \otimes B, \Delta \vdash C}\ \text{R-}\otimes$	$\dfrac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \text{L-}\otimes$	$\dfrac{}{\vdash \mathbb{1}}\ \text{R-}\mathbb{1}$

Table 1: Deduction rules of System $\mathbb{T}$

Dually, X is freely disposable in the system $\mathbb{T}[\bar{X}]$, which we obtain by imposing the additional rule $X \vdash \mathbb{1}$ to those of $\mathbb{T}$. We prove that the theorems of these polynomial systems are closely related to those of $\mathbb{T}$ via analogues of Lambek's lifting theorems of [48].

Result 2. *There exists a proof of* $\Gamma \vdash A$ *in* $\mathbb{T}[X]$ *if and only if for some* $m \geq 0$ *there exists a proof of* $\Gamma, X^{\otimes m} \vdash A$ *in* $\mathbb{T}$. *Similarly, there exists a proof of* $\Gamma \vdash A$ *in* $\mathbb{T}[\bar{X}]$ *if and only if for some* $n \geq 0$ *there exists a proof of* $\Gamma \vdash A \otimes X^{\otimes n}$ *in* $\mathbb{T}$.

Tracking resource conversions can also be incorporated into our polynomial system. For example, if 2 copies of resource A get converted to 3 copies of resource B, then its corresponding logic system would be $\mathbb{T}[A \otimes A \vdash B \otimes B \otimes B]$, which is defined as the rules of $\mathbb{T}$ with the desired conversion $A \otimes A \vdash B \otimes B \otimes B$ added.

Example 3 (Bipartite entanglement as a logic) For the LOCC and entanglement scenario, we pose atomic propositions Q_A for Alice's qubit type, Q_B for Bob's qubit type, and C for a classical bit type which, as argued above, may be considered shared between Alice and Bob. Again we simply add an additional atomic proposition E for the type of a maximally entangled state shared by Alice and Bob. Needless to say, this toy model is insufficient to capture the full power of LOCC. Nonetheless we can capture simple quantum protocols; for example the type of the quantum teleportation circuit is represented logically as $E \otimes Q_A \vdash Q_B$. This is obtained by composing Alice's measurement $E \otimes Q_A \vdash C \otimes C \otimes Q_B$ with Bob's correction $C \otimes C \otimes Q_B \vdash Q_B$ using the cut rule.

As classical information may be freely created or destroyed $\vdash C$ and $C \vdash \mathbb{1}$ are theorems of this logic. Similarly local measurements are free in LOCC and thus $Q_A \vdash C$ and $Q_B \vdash C$ are theorems (and so by the cut rule so also are $Q_A \vdash \mathbb{1}$ and $Q_B \vdash \mathbb{1}$). In dealing with E there are some options. We could form a logic where $E \vdash Q_A \otimes Q_B$ is a theorem; in such a system entanglement is disposable in the technical sense given above. Namely any protocol requiring Q_A and Q_B can be run with E instead, and any protocol producing E and be transformed into one producing $Q_A \otimes Q_B$. This logic is expressed by the system $\mathbb{T}[C, \overline{C}, \overline{Q_A}, \overline{Q_B}, E \vdash Q_A \otimes Q_B]$. △

We provide two general notions of semantics for the logic.

1) Algebraic Semantics. The algebraic semantics of a logic involves mapping each term into an algebraic structure so that the connectives of the logic are realized by the operations of the algebra. A proof of an inference in the logic then corresponds to a relation between elements in the algebra. By associating a valuation function on the model, we can check the satisfaction of a logical formula using this model. The algebraic model $\mathscr{M}$ for system $\mathbb{T}$ is defined on a commutative ordered monoid[6] $\mathbb{M}$ along with a valuation function $v : \Phi \mapsto \mathbb{M}$ which takes each term in the logic to an element of the monoid. Intuitively, this valuation is meant to encode the resources associated to the term. We use a forcing relation where $m \Vdash A$ (read, $m \in \mathbb{M}$ *forces* $A \in \mathbb{T}$) implies that instances of type referred to by proposition A have sufficient resources required to instantiate m. This forcing relation respects the $\otimes$ connective that defines system $\mathbb{T}$ with the following rule: $m \Vdash A \otimes B \Leftrightarrow \exists n_1, n_2 \in \mathbb{M}$ with $m = n_1 \cdot n_2$ and $n_1 \Vdash A$ and $n_2 \Vdash B$. This relation helps define the semantic entailment over a requirement $\Gamma \vDash_m A$ and prove the soundness and completeness of the algebraic semantics of $\mathbb{T}$.

Result 3. $\Gamma \vdash A$ *in* $\mathbb{T}$ *if and only if* $\Gamma \Vdash_m A$ *over all* $m \in \mathbb{M}$ *over all models* $\mathscr{M}$.

[6]The binary monoid operation $\cdot$ is commutative, associative and satisfies a monoid order $\leq$ s.t. if $r \leq s$ and $x \leq y$ then, $r \cdot x \leq s \cdot y$.

Here we interpret $\Gamma \Vdash_m A$ to mean the following: if Γ infers A in system $\mathbb{T}$ and has sufficient resources to instantiate m then, A also has sufficient resources to instantiate m. Note that by developing semantics through ordered monoids we recover the formalism of [43].

2) Categorical Semantics. Given a language, it is possible to construct a *syntactic category* where the objects are the types in the language, the morphisms are the functions defined between the various types that satisfy the relations that can be proven in the language. The syntactic category provided by $\mathbb{T}$ is a symmetric monoidal category C. Every term in $\mathbb{T}$ becomes an object of C. Each proof of an inference $\Pi : A \vdash B$ becomes a morphism $[\![\Pi]\!] \in \mathrm{Hom}(A,B)$. The cut rule is used to define the composition of morphisms. The logical connective $\otimes$ becomes the symmetric bifunctor $\boxtimes$ where $[\![\Pi_1]\!] \boxtimes [\![\Pi_2]\!]$ is given by combining proofs Π_1 and Π_2 using the R-$\otimes$ and L-$\otimes$ rules. The bifunctor $\boxtimes$ and its natural transformations satisfy certain commutative diagrams that induce an equivalence relation on the logical proofs: $\Pi \sim \Pi'$ when $[\![\Pi_1]\!] = [\![\Pi_2]\!]$. Using transformations that maintain the equivalence of inferences, we show the following.

Result 4. *The categorical model C for system $\mathbb{T}$, constructed as described above, satisfies the hexagon, pentangon and triangle rules*[7] *and hence, can be characterized as a symmetric monoidal category.*

Note that this recovers the formalism of [44].

Related Work & Future Directions. A notion of the Curry-Howard-Lambek correspondence for quantum logic has significant literature; the most relevant constructions for our work focuses on the dagger-closed categories [49] and the quantum typed λ-calculus developed by Selinger and Valiron [28] (Qλ). The latter is a quantum analogue of Lambek's λ-calculus for classical computing. While the more complicated categorical semantics and the strongly typed quantum λ-calculus translate to a more expressive logic and get closer to the complete power of Linear Logic, implementing type checkers for them also seems like a bigger challenge [50].

On the side of practically available model checkers, QWIRE [2] is embedded into the proof assistant Coq and is enhanced with density operator denotational semantics that works well for small circuits. QPMC [51] is a model checker using density operator and quantum channel denotational semantics that has been used for some Quipper programs [52]. Proto-quipper [53, 54] tries to bridge the theoretical-practical gap in quantum type theory by formalizing some aspects of Quipper and builds on categorical semantics being closest to our approach.

In contrast, our approach takes the simplest possible logic that can still captures non-trivial aspects of resource theories and study its capability with the hope of extending the type theory to more expressive logics with more functionality in the future. To that end, our treatment of polynomial systems with respect to system $\mathbb{T}$ is the first approach of its kind in works pertaining to such logics. In ongoing work, we are working to identify if these polynomial systems correspond to an algebraic semantics that possess both completeness and soundness. Moreover, when fully implemented, polynomial systems would allow for a richer and possibly more flexible type theory than one that recognizes only bits and qubits as most of the languages currently do.

For future work, in one direction, we aim to build a practical type checking tool with strong theoretical foundations that could also be independently incorporated into the hardware-specific languages. In

[7]These are the commutative diagrams that characterize the behaviour of symmetric monoidal categories [21].

another direction, we would like to understand how to enrich the logic to add classical[8] control imperatives like if-then-else which system $\mathbb{T}$ cannot express.

References

[1] Alexander S Green, Peter LeFanu Lumsdaine, Neil J Ross, Peter Selinger, and Benoît Valiron. Quipper: a scalable quantum programming language. *ACM SIGPLAN Notices*, 48(6):333–342, 2013.

[2] Jennifer Paykin, Robert Rand, and Steve Zdancewic. Qwire: a core language for quantum circuits. *ACM SIGPLAN Notices*, 52(1):846–858, 2017.

[3] Krysta Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. Q#: Enabling scalable quantum computing and development with a high-level dsl. In *Proceedings of the Real World Domain Specific Languages Workshop 2018*, page 7. ACM, 2018.

[4] Robert S Smith, Michael J Curtis, and William J Zeng. A practical quantum instruction set architecture. *arXiv preprint arXiv:1608.03355*, 2016.

[5] Qiskit. `https://qiskit.org`, 2018.

[6] Robert Rand, Jennifer Paykin, and Steve Zdancewic. Qwire practice: Formal verification of quantum circuits in Coq. *arXiv preprint arXiv:1803.00699*, 2018.

[7] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in computer science*, pages 415–425. IEEE, 2004.

[8] Eric Chitambar, Debbie Leung, Laura Mančinska, Maris Ozols, and Andreas Winter. Everything you always wanted to know about LOCC (but were afraid to ask). *Communications in Mathematical Physics*, 328(1):303–326, 2014.

[9] Samson Abramsky and Bob Coecke. Categorical quantum mechanics. *Handbook of quantum logic and quantum structures: quantum logic*, pages 261–324, 2009.

[10] Bob Coecke, Ross Duncan, Aleks Kissinger, and Quanlong Wang. Generalised compositional theories and diagrammatic reasoning. In *Quantum Theory: Informational Foundations and Foils*, pages 309–366. Springer, 2016.

[11] Sean Tull. Operational theories of physics as categories. *arXiv preprint arXiv:1602.06284*, 2016.

[12] Bob Coecke, Chris Heunen, and Aleks Kissinger. Categories of quantum and classical channels. *Quantum Information Processing*, 15(12):5179–5209, 2016.

[13] Lucas Dixon and Ross Duncan. Graphical reasoning in compact closed categories for quantum computation. *Annals of Mathematics and Artificial Intelligence*, 56(1):23, 2009.

[14] Peter Hines. Types and forgetfulness in categorical linguistics and quantum mechanics. *arXiv preprint arXiv:1303.3170*, 2013.

[15] Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.

[16] Jamie Vicary. Higher semantics of quantum protocols. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 606–615. IEEE Computer Society, 2012.

[17] Ciarán M Lee and Jonathan Barrett. Computation in generalised probabilisitic theories. *New Journal of Physics*, 17(8):083001, 2015.

[18] Ciarán M Lee, John H Selby, and Howard Barnum. Oracles and query lower bounds in generalised probabilistic theories. *arXiv preprint arXiv:1704.05043*, 2017.

[8] By setting classical bits to be free, one can obtain the functionality of additive conjunction &.

[19] Giulio Chiribella and Carlo Maria Scandolo. An operational resource theory of purity. Submission to QPL 2016, 2016.

[20] Bob Coecke and Aleks Kissinger. *Picturing quantum processes*. Cambridge University Press, 2017.

[21] Saunders MacLane. Natural associativity and commutativity. *Rice University Studies*, 49(4):28–46, 1963.

[22] Spencer Breiner, Amir Kalev, and Carl A Miller. Parallel self-testing of the ghz state with a proof by diagrams. *arXiv preprint arXiv:1806.04744*, 2018.

[23] Peter W O'Hearn and David J Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[24] David J Pym, Peter W O'Hearn, and Hongseok Yang. Possible worlds and resources: the semantics of BI. *Theoretical Computer Science*, 315(1):257–305, 2004.

[25] Jean-Yves Girard. Linear logic. *Theoretical computer science*, 50(1):1–101, 1987.

[26] Samson Abramsky and Radha Jagadeesan. New foundations for the geometry of interaction. In *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 211–222. IEEE, 1992.

[27] Peter W O'hearn. Resources, concurrency, and local reasoning. *Theoretical computer science*, 375(1-3):271–307, 2007.

[28] Peter Selinger and Benoit Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science*, 16(3):527–552, 2006.

[29] Fernando Brandao, Michał Horodecki, Nelly Ng, Jonathan Oppenheim, and Stephanie Wehner. The second laws of quantum thermodynamics. *Proceedings of the National Academy of Sciences*, 112(11):3275–3279, 2015.

[30] Gilad Gour, Markus P Müller, Varun Narasimhachar, Robert W Spekkens, and Nicole Yunger Halpern. The resource theory of informational nonequilibrium in thermodynamics. *Physics Reports*, 583:1–58, 2015.

[31] Philippe Faist and Renato Renner. Fundamental work cost of quantum processes. *Physical Review X*, 8(2):021011, 2018.

[32] Alexander Streltsov, Hermann Kampermann, Sabine Wölk, Manuel Gessner, and Dagmar Bruß. Maximal coherence and the resource theory of purity. *New Journal of Physics*, 20(5):053058, 2018.

[33] Eric Chitambar and Min-Hsiu Hsieh. Relating the resource theories of entanglement and quantum coherence. *Physical review letters*, 117(2):020402, 2016.

[34] Alexander Streltsov, Gerardo Adesso, and Martin B Plenio. Colloquium: quantum coherence as a resource. *Reviews of Modern Physics*, 89(4):041003, 2017.

[35] Carmine Napoli, Thomas R Bromley, Marco Cianciaruso, Marco Piani, Nathaniel Johnston, and Gerardo Adesso. Robustness of coherence: an operational and observable measure of quantum coherence. *Physical review letters*, 116(15):150502, 2016.

[36] Andreas Winter and Dong Yang. Operational resource theory of coherence. *Physical review letters*, 116(12):120404, 2016.

[37] Mehdi Ahmadi, Hoan Bui Dang, Gilad Gour, and Barry C Sanders. Quantification and manipulation of magic states. *Physical Review A*, 97(6):062332, 2018.

[38] Barbara Amaral, Adán Cabello, Marcelo Terra Cunha, and Leandro Aolita. Noncontextual wirings. *Physical review letters*, 120(13):130403, 2018.

[39] Julio I De Vicente. On nonlocality as a resource theory and nonlocality measures. *Journal of Physics A: Mathematical and Theoretical*, 47(42):424017, 2014.

[40] Karol Horodecki, Andrzej Grudka, Pankaj Joshi, Waldemar Kłobus, and Justyna Łodyga. Axiomatic approach to contextuality and nonlocality. *Physical Review A*, 92(3):032104, 2015.

[41] Fernando GSL Brandão and Gilad Gour. Reversible framework for quantum resource theories. *Physical review letters*, 115(7):070503, 2015.

[42] Eric Chitambar and Gilad Gour. Quantum resource theories. *arXiv preprint arXiv:1806.06107*, 2018.

[43] Tobias Fritz. Resource convertibility and ordered commutative monoids. *Mathematical Structures in Computer Science*, pages 1–89, 2015.

[44] Bob Coecke, Tobias Fritz, and Robert W Spekkens. A mathematical theory of resources. *Information and Computation*, 250:59–86, 2016.

[45] Dan Marsden and Maaike Zwart. Quantitative foundations for resource theories. In *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[46] Paul-André Mellies. Categorical semantics of linear logic. *Panoramas et syntheses*, 27:15–215, 2009.

[47] Gerhard Gentzen. Investigations into logical deduction. *American philosophical quarterly*, 1(4):288–306, 1964.

[48] Joachim Lambek. Functional completeness of cartesian categories. *Annals of Mathematical Logic*, 6(3-4):259–292, 1974.

[49] Bob Coecke and Eric Oliver Paquette. Categories for the practising physicist. In *New Structures for Physics*, pages 173–286. Springer, 2010.

[50] Peter Selinger. Challenges in quantum programming languages (invited talk). In *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, volume 108 of *LIPIcs-Leibniz International Proceedings in Informatics*, pages 3:1–3:2. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[51] Yuan Feng, Ernst Moritz Hahn, Andrea Turrini, and Lijun Zhang. QPMC: a model checker for quantum programs and protocols. In *International Symposium on Formal Methods*, pages 265–272. Springer, 2015.

[52] Linda Anticoli, Carla Piazza, Leonardo Taglialegne, and Paolo Zuliani. Towards quantum programs verification: from quipper circuits to qpmc. In *International Conference on Reversible Computation*, pages 213–219. Springer, 2016.

[53] Neil J Ross. *Algebraic and logical methods in quantum computation*. PhD thesis, Dalhousie University Halifax, 2015.

[54] Francisco Rios and Peter Selinger. A categorical model for a quantum circuit description language. *arXiv preprint arXiv:1706.02630*, 2017.