

Efficient Formal Verification of Quantum Error Correcting Programs

量子纠错程序的高效形式化验证

黄启凡, 周立, 方望, 赵梦宇, 应明生

Proceedings of the ACM on Programming Languages 9 (PLDI, CCF-A), 1068-1093, 2025

联系方式: 黄启凡 huangqf@ios.ac.cn, 周立 zhou31416@gmail.com

问题背景

量子纠错 (Quantum Error Correction, QEC) 是目前改善量子计算机错误率的最佳方案, 通过编码逻辑量子比特, 实现对噪声的抑制; 但目前仍然缺少能够高效验证QEC程序正确性的方法。我们根据量子霍尔逻辑 [Ying, 2012] 并观察量子纠错程序的性质设计了一套具有语形的断言逻辑以及程序逻辑, 并开发了一套有实用价值的量子纠错程序验证工具Veri-qec。

技术路线

量子纠错码使用泡利群的稳定子群生成, 利用这一性质可定义泡利表达式, 进而构造断言语法; 基于经典-量子混合霍尔逻辑[Feng, 2021], 可构造如下程序语言, 并基于断言逻辑推理霍尔三元组的正确性。

$$SExp: S ::= (-1)^b \mid \sqrt{2} \mid S/2^t \mid S_1 + S_2 \mid -S \mid S_1 S_2$$

$$PExp: P ::= p_r \mid sP \mid P_1 P_2 \mid P_1 + P_2$$

$$AExp: A ::= b \in BExp \mid P \in PExp \mid \neg A \mid A \wedge A \mid A \vee A \mid A \Rightarrow A.$$

$$Prog: S ::= skip \mid q_i := |0\rangle \mid q_i * = U_1 \mid q_i q_j * = U_2$$

$$x := e \mid x := meas[P] \mid S \ ; \ S$$

$$\text{if } b \text{ then } S \text{ else } S \text{ end} \mid \text{while } b \text{ do } S \text{ end}$$

where:

$$U_1 \in \{X, Y, Z, H, S, T\}$$

$$U_2 \in \{CNOT, CZ, iSWAP\}$$

霍尔三元组 $\{P\} S \{Q\}$ 的证明方法

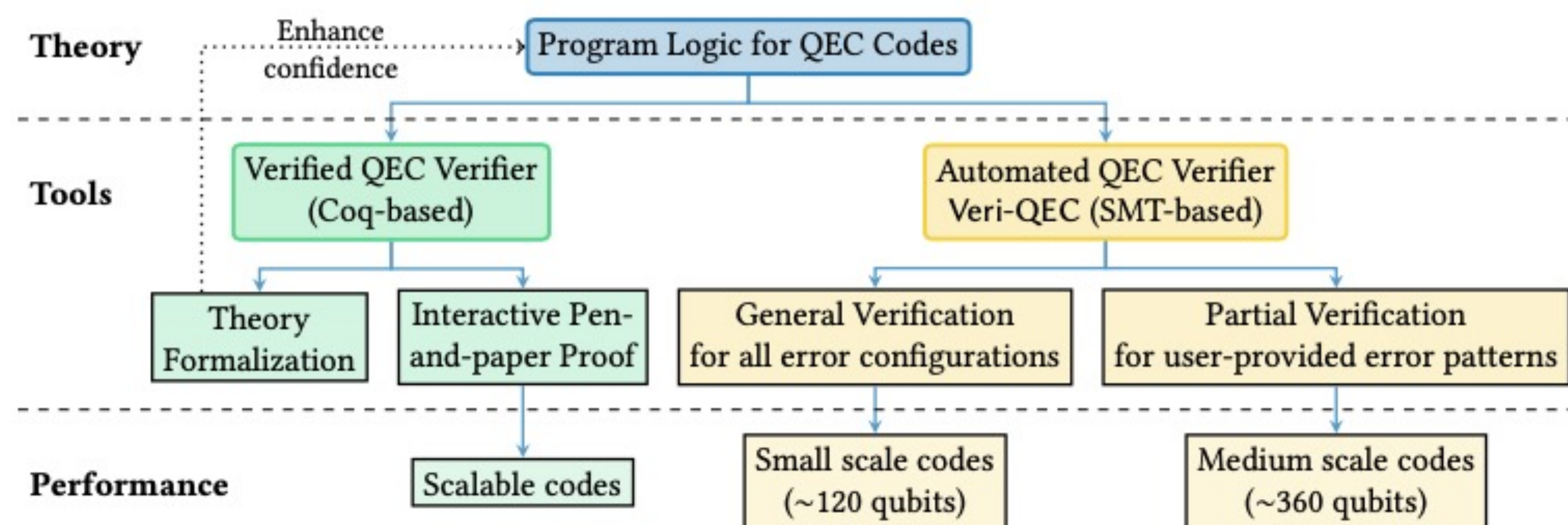
Derive P' in $\{P'\} S \{Q\}$

Proving $P \models P'$

$\{P\} S \{Q\}$ is correct

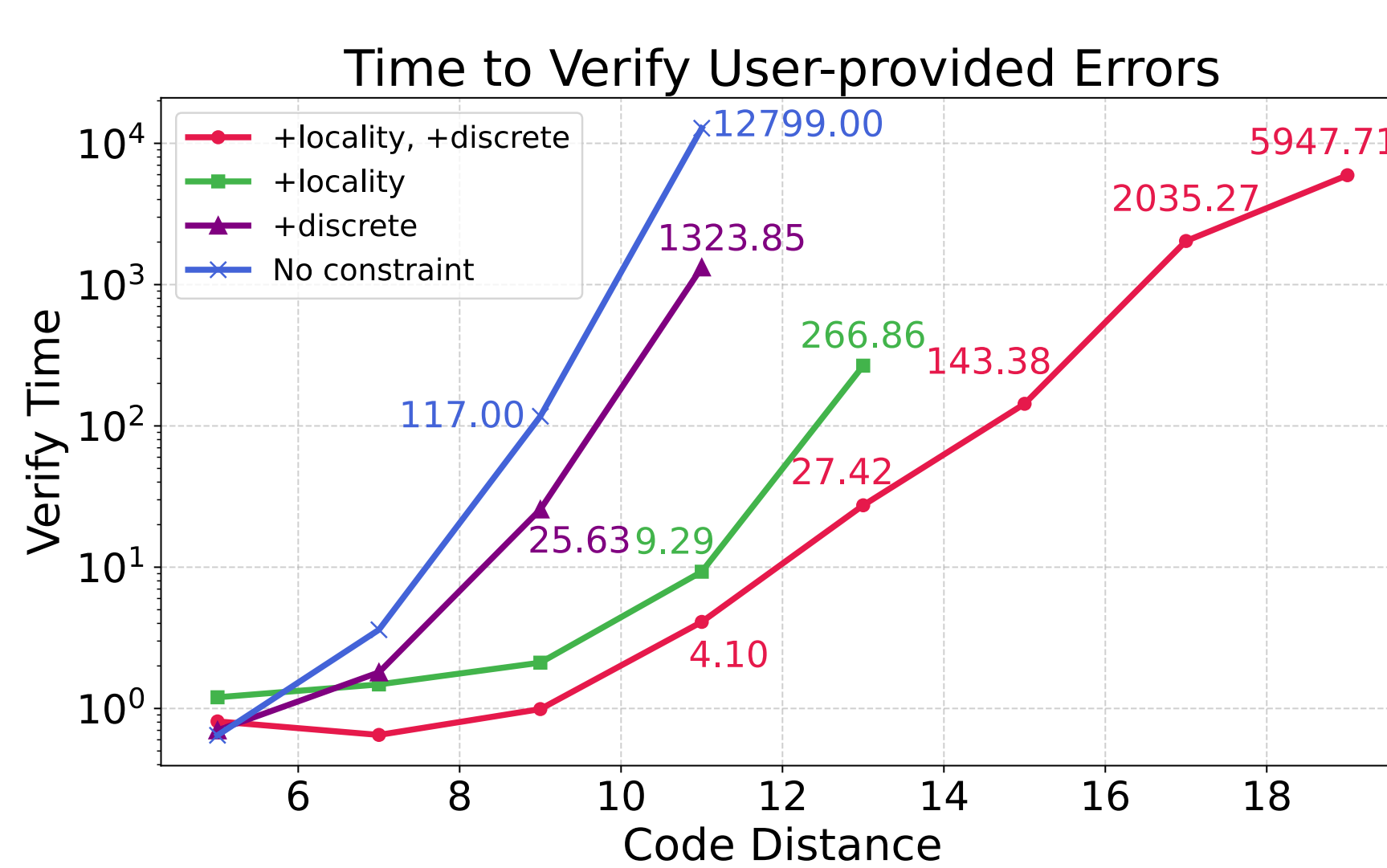
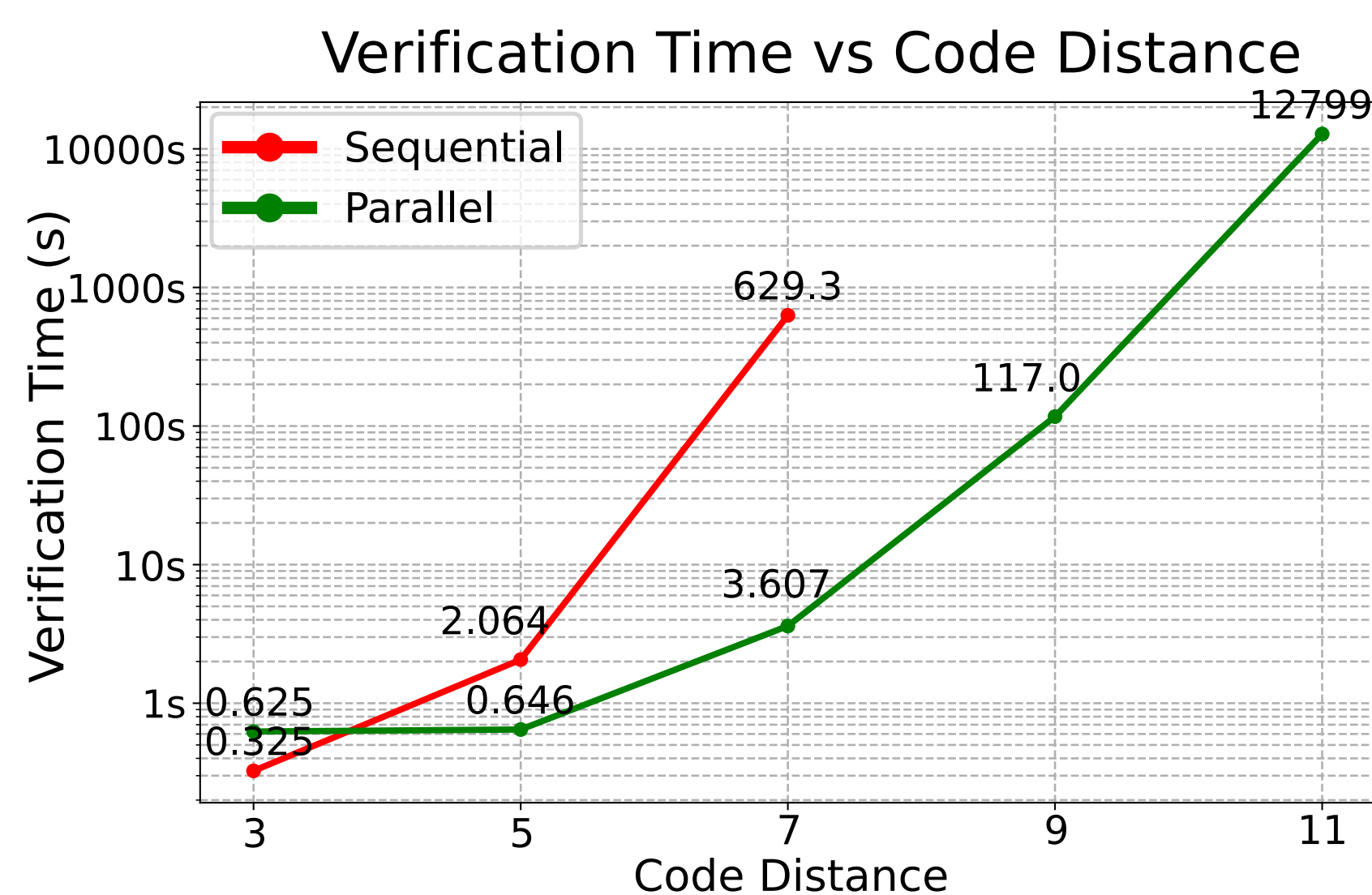
工具实现

我们开发了一套高效验证量子纠错程序的工具, 包含基于Coq的验证器以及基于Python的自动验证器。我们基于CoqQ [Zhou et al. 2023] 这一量子计算定理库, 在Coq中实现了程序逻辑并验证了证明系统的可靠性。另一方面自动验证器借助SMT求解器进行验证, 在求解规模上取得了突破。工具的开源链接: <https://github.com/Chesterhuang1999/Veri-qec>。



实验评估

我们在表面码上进行实验来评估自动化验证工具的性能; 实验结果显示通过并行求解的方案对SMT求解器进行优化后, 工具的性能具有显著提升, 同时也能够在用户提供了限制条件的情形下将可验证的规模进一步地扩大。



我们还提供了一个经过验证的量子纠错码基准集合, 共包括14种不同的码。

| Target: Accurate Correction | | |
|--|---|----------------|
| Code Name | Parameters | Verify time(s) |
| Steane code [72] | [[7, 1, 3]] | 0.095 |
| Surface code [30] ($d = 11$) | [[$d^2, 1, d$]] | 12799 |
| Six-qubit code [20] | [[6, 1, 3]] | 0.252 |
| Quantum dodecacode [20] | [[11, 1, 5]] | 0.587 |
| Reed-Muller code [73] ($r = 8$) | [[$2^r - 1, 1, 3$]] | 1868.56 |
| XZZX surface code [13] ($d_x = 9, d_z = 11$) | [[$d_x \times d_z, 1, \min(d_x, d_z)$]] | 1067.16 |
| Gottesman code [37] ($r = 8$) | [[$2^r, 2^r - r - 2, 3$]] | 587.00 |
| Honeycomb code [51] ($d = 5$) | [[19, 1, 5]] | 1.55 |
| Target: Detection | | |
| Tanner Code I [55] | [[343, 31, $d \geq 4$]] | 7086.36 |
| Tanner Code II [55] | [[125, 53, 4]] | 1667.81 |
| Hypergraph Product [18, 48, 79] | [[98, 18, 4]] | 289.37 |
| Error-Detection codes | | |
| 3D basic color code [50] ($d_z = 2$) | [[8, 3, 2]] | 2.88 |
| Triorthogonal code [17] ($k = 64$) | [[$3k + 8, k, d_x = 6, d_z = 2$]] | 144.94 |
| Carbon code [38] | [[12, 2, 4]] | 4.80 |
| Campbell-Howard code [22] ($k = 2$) | [[$6k + 2, 3k, 2$]] | 3.05 |