量子プログラムのための依存型を用いた Coupling Graph 解析

脇坂 遼
京都大学工学部情報学科

五十嵐 淳
京都大学大学院情報学研究科

目的

量子プログラムが持つ coupling graph を型システムで検証・抽出可能な体系の構築

背景 1：量子ラムダ計算 [Selinger and Valiron '06]

- No-cloning theorem のための線形型

$$q_0 : \texttt{qbit}^1 \vdash (\lambda x.\langle x, x\rangle)\ q_0 : \textbf{型エラー}$$
$$v : \texttt{unit}^\omega \vdash (\lambda x.\langle x, x\rangle)\ v : \texttt{unit}^\omega \otimes^\omega \texttt{unit}^\omega$$

- 意味論：$[Q, L, M] \to_p [Q', L', M']$

$$[|00\rangle, |q_1 q_0\rangle, \texttt{cnot}\ \langle H\ q_1, q_0\rangle]$$
$$\to_1 [\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle), |q_1 q_0\rangle, \texttt{cnot}\ \langle q_1, q_0\rangle]$$
$$\to_1 [\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), |q_1 q_0\rangle, \langle q_1, q_0\rangle]$$

背景 2：Coupling Graph

- CNOT ゲートは特定の量子ビットペアにのみ適用可能
- この制約は coupling graph $G = (Q, E)$ で表現され，アーキテクチャ毎に異なる
  - Q：量子ビットの集合
  - $E \subseteq Q \times Q$：CNOT ゲートが適用可能な量子ビットペア
- 量子プログラムの coupling graph が対象のアーキテクチャが持つ coupling graph の部分グラフに埋め込める（部分グラフ同型）ならば実行可能
- 量子プログラムの coupling graph は，それを実行したときに CNOT ゲートが適用されうる量子ビットペアを表すグラフ
- 量子プログラムが実行可能かを検証する必要がある

アプローチ

- 依存量子ビット型 $Q(i)^1$
  - ラベルを表す $\texttt{qidx}\ i$ により量子ビットを区別
  - qidx は意味論には現れない
- 型判断に coupling graph G を取り入れる
  - 量子ビット i と j が隣接することを $i \sim j$ と書く

$$G\ ; \Gamma \vdash M : T$$

グラフ G の下で項 M に型 T がつく

$$\frac{i \sim j \in G}{G; \Gamma \vdash \texttt{cnot} : Q(i)^1 \otimes^1 Q(j)^1 \to^\omega Q(i)^1 \otimes^1 Q(j)^1}$$

隣接する任意の 2 量子ビットペアに対して適用可能

Example：$\Gamma = q_0 : Q(0)^1, q_1 : Q(1)^1$ としたとき

$$0 \sim 1; \Gamma \vdash \texttt{cnot}\ \langle q_0, q_1\rangle : Q(0)^1 \otimes^1 Q(1)^1$$
$$\emptyset; \Gamma \vdash \texttt{cnot}\ \langle q_0, q_1\rangle : \textbf{型エラー}$$

qidx 上の多相性

- qidx に関する多相型 $\Pi\bar{i} : \texttt{qidx}.\bar{e} \Rightarrow T$
  - $\bar{i}$：qidx の列（束縛変数）
  - $\bar{e}$：$i \sim j$ の列（この関数を使うために必要な制約）
- let 式により導入（let 多相）
- 通常の多相型のように振る舞う
  - $\bar{i}$ が具象化されるとき，それに応じて $\bar{e}$ も具象化
  - 具象化された制約 $\bar{e}$ が G に含まれるなら OK

$$\texttt{let}\ f = \lambda x.\lambda y.\texttt{cnot}\ \langle y, x\rangle\ \texttt{in}$$
$$\langle\ f\ \ q_0\ q_1,\ \ f\ \ q_2\ q_3\rangle$$

$$\Pi i, j : \texttt{qidx}.j \sim i \Rightarrow Q(i)^1 \to^\omega Q(j)^1 \to^1 Q(j)^1 \otimes^1 Q(i)^1$$

制約 $3 \sim 2$ を要求

制約 $1 \sim 0$ を要求

$1 \sim 0 \in G$ かつ $3 \sim 2 \in G$ ならば型付け可能

意味論の変更　$G \vdash [Q, L, M] \to_p [Q', L, M']$

- CNOT ゲートの実行を G で制約付ける

$$G \vdash [Q, L, \texttt{cnot}\ \langle q_i, q_j\rangle] \to_1 \begin{cases} [Q', L, \langle q_i, q_j\rangle] & ((q_i, q_j) \in G) \\ \textbf{（実行時エラー）} & \text{(otherwise)} \end{cases}$$

- 量子ビット生成関数 new の削除により L は不変
  - 再帰関数内で new が呼ばれると解析が困難

体系の性質

- $G; \Gamma \vdash M : T$ かつ $G \vdash [Q, L, M] \to_p [Q', L, M]$ ならば $G; \Gamma \vdash M' : T$
- $G; \Gamma \vdash M : T$ ならば M は値であるか，ある項 M' があって $G \vdash [Q, L, M] \to_p [Q', L, M']$
- すなわち，型がついたプログラムは CNOT ゲートによる実行時エラーが発生しない

型推論アルゴリズムと健全性

- プログラムの持つ coupling graph を型推論により抽出可能
- 抽出されたグラフで，実際に型付け可能（証明済）

将来の課題

- 型システム改善による表現力の向上
- Qubit Allocation Problem への応用