-JP_52: Vのトランスレータとコンパイラ

著者：Víctor Estrada Díaz。枠組み：Einstein-VED - UNIHOLOG。
日付：2026年10月10日。

プロジェクトの参照資料： 基本文書 · Q-Vの構築と比較の図解資料 · コミュニティへの紹介 · リソース総索引。

デュアルライセンス — 倫理と使用

- **本文（CC BY-NC-SA 4.0）**：共有と翻案が可能。非営利利用、著作者表示、同一条件での共有。
- **コード（HL3）**：倫理原則を満たす場合に限り、商用利用を認める。 戦争、人権侵害、環境破壊、搾取、大規模な監視、差別を禁止する。

著者が宣言する倫理原則：人間の尊厳の尊重、許可のない営利利用をしないこと、有害な利用の禁止。 声明 · CC BY-NC-SA 4.0 · HL3。

目的と規則

このツールは、既知のQ回路をVの構築計画に変換する。Qの答えを問い合わせたり、測定まで隠しておくための結果をあらかじめ計算したりはしない。ゲート、制御、接続、順序を保持する。振幅レジスタを導入したり、可能な結果を走査したりはしない。

Vは著者の決定論的マシンであり、従来の電子回路またはソフトウェアで実現できる。測定前の未知性は、その動力学にランダムな機構を付け加えるものではない。

Vでは、測定前の出力は未知のままである。コントローラの時計が記録する$t = t_0$において、一括した読み取りが引き起こされる。ゲートの発展を担うのは接続された実現系であり、測定の呼び出しではない。途中の二値を得るために、その発展を観測することはしない。

$$M_V(t_0) = (v_0, \cdots, v_{n-1}), \qquad v_i \in \{0, 1\}.$$

再度の問い合わせには、記録済みの読み取りを返し、トリガーを繰り返さない。検出器でエラーが生じた場合も、その準備は消費される。別の試行や作り出した出力で置き換えることはしない。

内部モジュールの合成

構築計画は、他のマシン**V**を内部に含み、それらの関係が準備済みで、変数がまだ未確定のマシンを保持しなければならない。

$$V_t(V_1, \cdots, V_m), \qquad \mathbf{x}_i(t_i) \in \{0, 1\}^{r_i}.$$

モジュールの入力はその時点t_iで確定し、その値が接続を通って伝播する。トランスレータは、共有された依存関係を独立した出力で置き換えたり、モジュールを準備する前に内部値を要求したりしてはならない。**0**または**1**として宣言された値と、それらから導かれる値は、先に伝播させることができる。

プロトコルのトリガーが$t = t_0$の一回だけであれば、内部の確定と伝播は、そのトリガーに応じて行われ、先行する外部読み取りを加えない。関係の準備、実現系の実行、結果の観測は異なる操作である。トリガーは実現系の内部伝播を引き起こしてよく、インターフェースを

分離することは、それを禁止することではない。 著者の規則は、主要文書の20.3節で展開されている。 実行可能な状態については、現在の結果に記載している。

可搬性を備えた構成

```
ES_52/
  ES_52_Maquina_V_UNIHOLOG.md
  ES_52_Laminas_Q_V.md
  img/
  Traductores_V/
    ES_52_Traductores_y_compiladores_de_V.md
    gestor_v.py
    traductor_qasm_v.py
    v_relacional.py
    operaciones_senales.py
    v1.py
    v2.py
    v3.py
    protocolo_q3.py
    ejecucion_v.py
    requirements.txt
    v.bat
    v.ps1
    v.sh
    MaquinasQ/
      par_h_cnot.q
    MaquinasV/
      par_h_cnot/
        v.py
        v.pl
        v.cpp
        v.hpp
        v.json
```

各 `MaquinasQ/*.q` ファイルは、拡張子が `.q` でもOpenQASM **2.0**を含む。拡張子を除いた名前がマシンを識別する。HとCNOTを用いる二ユニットの例を含むが、それらのゲートの物理的実現を同梱しているわけではない。

ランチャーは、自身の配置場所を基準にパスを計算する。三つの文書は相対パスで結び付けられている。**ES_52全体**を移動すれば、その案内構造は保たれる。Traductores_Vフォルダは単独でも動作するが、基本文書へのリンクにはプロジェクトの残りの部分が必要である。

パスの可搬性は、ツールやハードウェアをインストールするものではない。移動先にはPythonとQiskitが必要であり、言語に応じてSWI-Prolog、またはGCC/Clangのインターフェースと**C++17**に対応したC++コンパイラも必要になる。

インストール

Pythonが利用できる状態で、パッケージの依存関係をインストールする。

```
python -m pip install -r requirements.txt
```

`V_PYTHON` でインタープリタ、 `V_SWIPL` でSWI-Prolog、 `V_CXX` でコンパイラを選択できる。これらの変数には実行ファイルのパスを指定し、追加の引数は含めない。既定では、インストール済みのツールを探す。自動的なダウンロードやインストールは行わない。

以下のコマンドはTraductores_Vから実行する。他のフォルダからはランチャーのパスを指定する。マシンの検索先は、そのランチャーのMaquinasQであり、利用者の作業ディレクトリではない。

Windows：PowerShellまたはBAT

```
.¥v.ps1 listar
.¥v.ps1 traducir par_h_cnot
.¥v.ps1 compilar par_h_cnot --lenguaje python
.¥v.ps1 compilar par_h_cnot --lenguaje prolog
.¥v.ps1 compilar par_h_cnot --lenguaje cpp
```

`v.bat` でも同様に使用できる。例えば、

```
v.bat traducir par_h_cnot
```

PowerShellの実行ポリシーによって、`.ps1` を開けない場合がある。このポリシーは変更しない。`.bat` を使用するか、同じ引数で `python gestor_v.py` を直接呼び出せる。

Linuxおよびその他のシェル環境

```
sh ./v.sh listar
sh ./v.sh traducir par_h_cnot
sh ./v.sh compilar par_h_cnot --lenguaje python
sh ./v.sh compilar par_h_cnot --lenguaje prolog
sh ./v.sh compilar par_h_cnot --lenguaje cpp
```

変換とコンパイル

変換は、`MaquinasV/<nombre>/` に五つの出力を生成する。既存の変換結果を上書きしない。再生成するには、既存のものを別名で保存するか、以前のファイルを明示的に取り除かなければならない。JSONに

は、計画、測定マップ、入力のハッシュが含まれ、内部状態や答えの分布は含まれない。

`compilar` 操作は、Pythonの構文を確認し、Prologモジュールをロードし、またはリンクせずにC++の翻訳単位を確認する。Vの実現系を準備したり、トリガーを発生させたりはしない。コンパイラが受理したことは実験結果ではない。

パーサがその宣言を受理する場合、H、X、CNOT、Toffoli、SWAP、CZ、Rx、Rz、P/U1、S、Sの逆、T、Tの逆、Z、CU1/CPに対応する。QASM内で定義されたゲートは、その接続とともに展開される。定義に含まれる位相を、知らせずに除去することはしない。

ユニットごとに、異なる古典的格納先を持つ終端測定が、ちょうど一回必要である。途中の測定、部分的な測定、reset、読み取りに依存する制御、測定後の操作、変換が定義されていないゲートは拒否される。この最初のバージョンでは、OpenQASM 3、U/U2/U3、Y、Ryには対応しない。これらの拒否はツールの限界を記述するものであり、Vへの反証ではない。

接続された実現系を実行する

実行には実現系を与える必要がある。それがなければ、ランチャーは明示的なメッセージを出して停止し、シミュレータや試験データを代わりに使用することはない。実現系は、利用者が実行を決めた信頼できるコードであり、Vの内部的な制約を守らなければならない。

Python

実現系のファイルは `crear_realizacion()` を定義する。この関数は、読み取りを返さずに設定する `preparar(plan)` と、ユニットのインデックス順に、ユニットごとに測定した一ビットを返す `disparar(t0)` を持つオブジェクトを返す。

```
.¥v.ps1 ejecutar par_h_cnot --lenguaje python --realizacion .¥mi_realizacion.py
```

報告には、読み取り、古典ビットのマップ、トリガー時点が含まれる。測定されていない古典的格納先には読み取りを入れず（ `null` ）、その結果を作り出すことはしない。

Prolog

実現系のファイルは、multifile述語 `qasm_v:realizar_preparacion/1` と `qasm_v:realizar_disparo/2` を実装する。コントローラはトリガー時に実時刻の時計を記録し、それを装置の変換時点の較正と混同しない。

```
.¥v.ps1 ejecutar par_h_cnot --lenguaje prolog --realizacion .¥mi_realizacion.pl
```

C++

実現系のファイルは `v.hpp` をインクルードし、インターフェース `qasm_v::Realizacion` を実装し、次のファクトリ関数を定義する。

```
std::unique_ptr<qasm_v::Realizacion> crear_realizacion();
```

管理プログラムは、接続用プログラムを生成し、その実現系とともにコンパイルしてバイナリを実行する。別のゲート動力学を導入することはしない。実現系には、それ自体の資源と依存関係が含まれる。この最初のバージョンは、外部のハードウェアライブラリを設定しない。

```
.¥v.ps1 ejecutar par_h_cnot --lenguaje cpp --realizacion .¥mi_realizacion.cpp
```

評価と適用範囲

このフォルダはプロジェクト52に属し、変換と実現系の接続については独立している。DEMUESTRAは、その外側に共通の管理プログラムとして置かれる。`documentos_proyecto.json`で主要文書と付録を識別し、派生形式を生成し、パスを確認する。文書42、45の内容やそれらの評価を、このプロジェクトと混在させることはしない。

DEMUESTRAから、次のコマンドでローカル文書を更新する。

```
python publica.py --documentos-locales ES_52
```

この操作は、HTML、PDF、ソースと派生形式のJSON報告を生成する。公開したり、Lean/Prologの評価を変更したり、証明書を発行したりはしない。プログラム`v.pl`はVトランスレータの出力であり、科学文書のProlog証明書ではない。それぞれの機能と評価は異なる。

コントローラは、読み取りに`REALIZACION_NO_VERIFICADA`というラベルを付ける。ビットを受け取ったという理由だけで認証を与えない。記録する時点はトリガーを呼び出した時点であり、較正された物理的な

同時性を保証するものではない。実現系内部の資源と動力学は、別途検討しなければならない。

Qとの比較には、同じプロトコルが必要であり、両者について対応する条件、出力、反復、資源、時間を記録しなければならない。接続の変換、コードのコンパイル、実現系の測定は異なる段階である。この管理プログラムの限界をVについての否定的結論に変えることはなく、構文試験を等価性として提示することもない。

使用するパーサは、QiskitのOpenQASM 2である。