GetCircuitsFromChannel

```
SetDirectory @ NotebookDirectory[];
Import["../Link/QuESTlink.m"] // Quiet;
CreateLocalQuESTEnv["../quest_link"];
```

Doc

```
? GetCircuitsFromChannel
```

Symbol

GetCircuitsFromChannel[channel] returns a list of all pure, analytic circuits which are admitted as possible errors of the input channel (a circuit including decoherence). Channels which are mixtures of unitaries (like Depol, Deph) become unitaries and a non−unitary Fac[] operator, while other channels (Damp, Kraus) become non−trace−preserving Matr[] operators.

The sum of the expected values of the (potentially unnormalised) state−vectors output by the returned circuits is equivalent to the expected value of the input channel.

See GetRandomCircuitFromChannel[] to randomly select one of these circuits, weighted by its probability.

See SampleExpecPauliString[] to sample such circuits in order to efficiently approximate the effect of decoherence on an expectation value.

Correctness

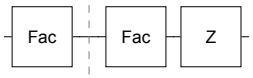
```
test[channels_] := Module[
    {circs, n, h, ψi, ψ, ϕ, ρ, ω, e1, e2, err},

    (* use random Pauli observable *)
    n = 1 + Max @ GetCircuitQubits[channels];
    h = GetRandomPauliString[n];

    (* use random initial state *)
    ψi = CreateQureg[n];
    SetQuregMatrix[ψi, RandomComplex[{0,1+ⅈ}, 2^n]];

    (* use QuESTlink's numerical backend to... *)
    {ψ,ϕ} = CreateQuregs[n, 2];
    {ρ,ω} = CreateDensityQuregs[n, 2];

    (* compute expec value via channels upon density matrix ...*)
    InitPureState[ρ, ψi];
    ApplyCircuit[ρ, channels];
    e1 = CalcExpecPauliString[ρ, h, ω];

    (* and via decomposed circuits upon statevector... *)
    circs = GetCircuitsFromChannel[channels];
    e2 = Sum[
        InitPureState[ψ, ψi];
        ApplyCircuit[ψ, circ];
        CalcExpecPauliString[ψ, h, ϕ],
        {circ, circs}];

    (* clean up *)
    DestroyAllQuregs[];

    err = e1 - e2 // Abs // Chop;
    Echo[Length[circs], "Number of decomposed circuits:"];
    Echo[err, "error: "];
    If[err =!= 0, Style["ERRONEOUS DECOMPOSITION!", Red]]
]
```

Decomposition

Unitary mixtures

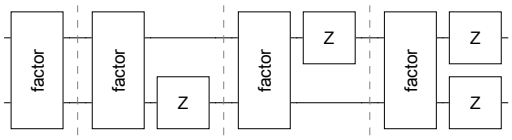
```
GetCircuitsFromChannel @ Deph0[x]
DrawCircuit @ %
```

$$\left\{\left\{\text{Fac}\left[\sqrt{1-x}\right]\right\},\left\{\text{Fac}\left[\sqrt{x}\right],Z_0\right\}\right\}$$



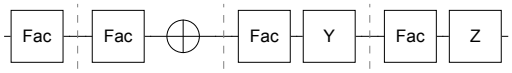

```
GetCircuitsFromChannel @ Deph0,1[x]
DrawCircuit @ %
```

$$\left\{\left\{\text{Fac}\left[\sqrt{1-x}\right]\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{3}}\right],Z_0\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{3}}\right],Z_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{3}}\right],Z_0,Z_1\right\}\right\}$$



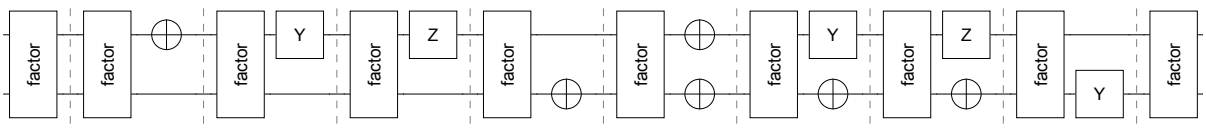

```
GetCircuitsFromChannel @ Depol0[x]
DrawCircuit @ %
```

$$\left\{\left\{\text{Fac}\left[\sqrt{1-x}\right]\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{3}}\right],X_0\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{3}}\right],Y_0\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{3}}\right],Z_0\right\}\right\}$$




```
GetCircuitsFromChannel @ Depol0,1[x]
DrawCircuit @ %
```

$$\left\{\left\{\text{Fac}\left[\sqrt{1-x}\right]\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],X_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Y_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Z_1\right\},\right.$$
$$\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],X_0\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],X_0,X_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],X_0,Y_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],X_0,Z_1\right\},$$
$$\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Y_0\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Y_0,X_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Y_0,Y_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Y_0,Z_1\right\},$$
$$\left.\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Z_0\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Z_0,X_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Z_0,Y_1\right\},\left\{\text{Fac}\left[\frac{\sqrt{x}}{\sqrt{15}}\right],Z_0,Z_1\right\}\right\}$$



Non-mixtures

```
GetCircuitsFromChannel @ Damp0[x]
```

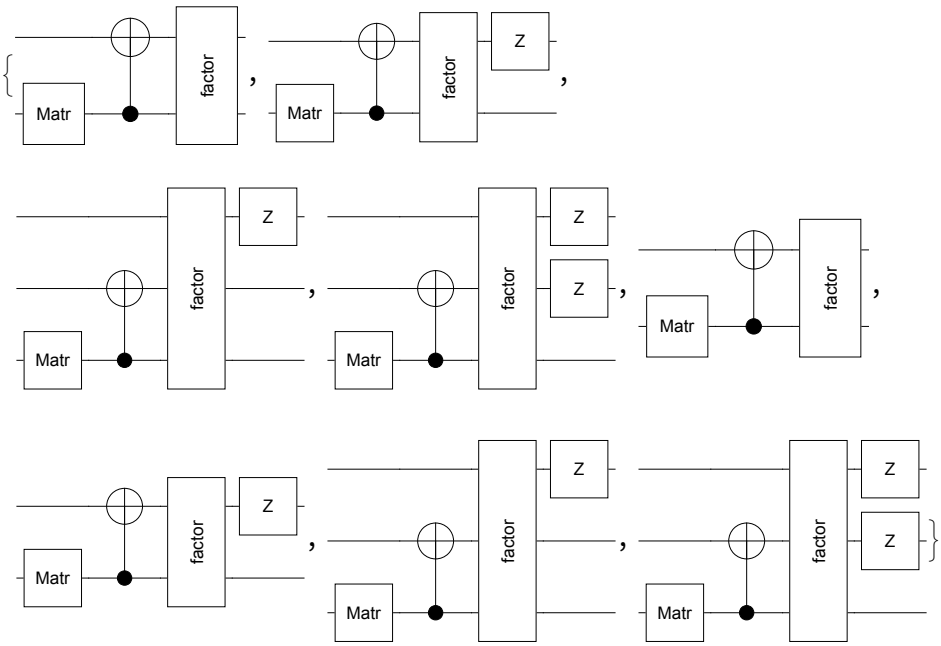
$$\{\{\text{Matr}_0[\{\{1, 0\}, \{0, \sqrt{1-x}\}\}]\}, \{\text{Matr}_0[\{\{0, \sqrt{x}\}, \{0, 0\}\}]\}\}$$

```
GetCircuitsFromChannel @ Kraus0 @ {{{a, b}, {c, d}}, {{e, f}, {g, h}}}
```

```
{{Matr0[{{a, b}, {c, d}}]}, {Matr0[{{e, f}, {g, h}}]}}
```

Circuits

```
DrawCircuit /@ GetCircuitsFromChannel @ Circuit[Damp0[x] C0[X1] Deph1,2[y]]
```



Unitaries

```
in =
  Circuit[G[x] × Fac[.1] H0 Id1 Ph0,1[x] R[.1, X0 Z1] C0,1[Rx2[x]] X0 Y1 Z2 SWAP0,1 T0 S1];
out = GetCircuitsFromChannel[in];
in === Flatten @ out
```

```
True
```

Expectation values

Unitary mixtures

```
test[Deph0 @ RandomReal[{0, 1 / 2}]]
```

» Number of decomposed circuits: 2

» error: 0

```
test[Depol0 @ RandomReal[{0, 3 / 4}]]
```

» Number of decomposed circuits: 4

» error: 0

```
test[Deph0,1 @ RandomReal[{0, 3 / 4}]]
```

» Number of decomposed circuits: 4

» error: 0

```
test[Depol0,1 @ RandomReal[{0, 15 / 16}]]
```

» Number of decomposed circuits: 16

» error: 0

Non-mixtures

```
test[Damp0 @ RandomReal[]]
```

» Number of decomposed circuits: 2

» error: 0

```
k = Table[RandomVariate @ CircularUnitaryMatrixDistribution[2], 3];
test[KrausNonTP0[k]]
```

» Number of decomposed circuits: 3

» error: 0

```
k = Table[RandomVariate @ CircularUnitaryMatrixDistribution[4], 10];
test[KrausNonTP0,1[k]]
```

» Number of decomposed circuits: 10

» error: 0

Circuits

```
in = Circuit[H0 Damp0[.1] R[.1, X0 Z1]
     Deph0,1[.3] C0,1[Rx2[.9]] Depol1[.3] SWAP0,1 T0 Depol0,1[.3] S1];
test[in]
```

» Number of decomposed circuits: 512

» error: 0

Errors

```
GetCircuitsFromChannel[a]
```

GetCircuitsFromChannel: Invalid arguments. See ?GetCircuitsFromChannel

```
$Failed
```

```
GetCircuitsFromChannel[X₀, b]
```

GetCircuitsFromChannel: Invalid arguments. See ?GetCircuitsFromChannel

```
$Failed
```