Formally Verifying your Quantum Programs

Robert Rand

with Jennifer Paykin and Steve Zdancewic

University of Pennsylvania

DeepSpec Summer School, 2017

Qubits

Vector $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Qubits

Vector $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Ket $|0\rangle$

Qubits

Vector	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
Ket	$\lvert 0 \rangle$	$\lvert 1 \rangle$

Qubits

Vector	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
Ket	$\lvert 0 \rangle$	$\lvert 1 \rangle$	$\alpha \lvert 0 \rangle + \beta \lvert 1 \rangle$

Qubits

Vector	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
Ket	$\lvert 0 \rangle$	$\lvert 1 \rangle$	$\alpha \lvert 0 \rangle + \beta \lvert 1 \rangle$

$$|\alpha|^2 + |\beta|^2 = 1$$

Unitaries

$$\mathrm{H} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Unitaries

$$\mathrm{H}\,|0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Unitaries

$$\mathrm{H}\,|0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

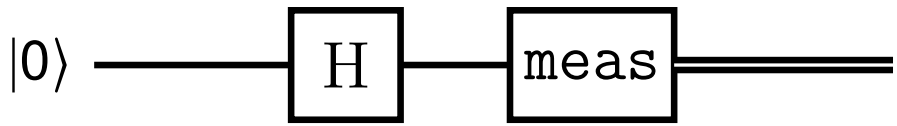
Measurement

$$\textit{measure}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{cases}|0\rangle & \text{with probability } |\alpha|^2\\ |1\rangle & \text{with probability } |\beta|^2\end{cases}$$

Measurement

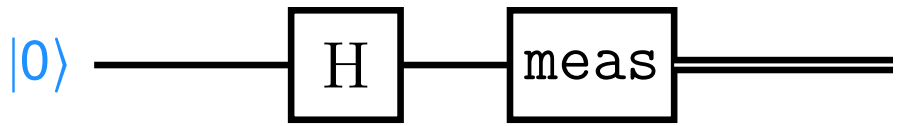
$$measure\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix} = \begin{cases}|0\rangle & \text{with probability } \frac{1}{2}\\ |1\rangle & \text{with probability } \frac{1}{2}\end{cases}$$

A Coin Flip Circuit



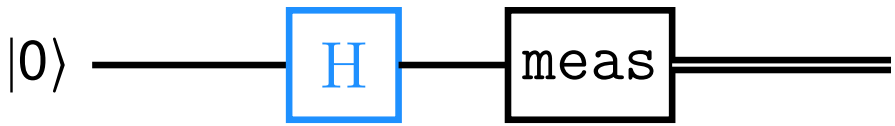

```
Definition coin_flip : Box One Bit.
  box () ⇒
    gate q ← init0 @();
    gate q ← H @q;
    gate b ← meas @q;
    output b.
Defined.
```

A Coin Flip Circuit



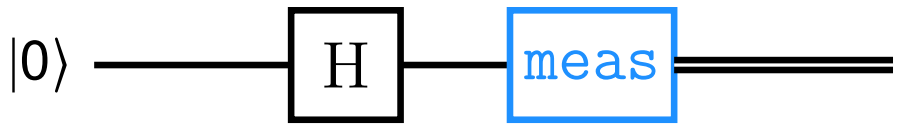

```
Definition coin_flip : Box One Bit.
  box () ⇒
    gate q ← init0 @();
    gate q ← H @q;
    gate b ← meas @q;
    output b.
Defined.
```

A Coin Flip Circuit



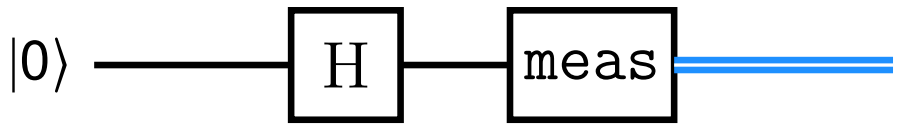

```
Definition coin_flip : Box One Bit.
  box () ⇒
    gate q ← init0 @();
    gate q ← H @q;
    gate b ← meas @q;
    output b.
Defined.
```

A Coin Flip Circuit



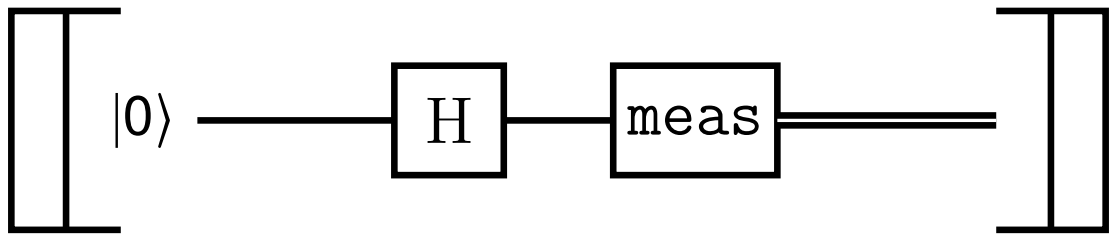

```
Definition coin_flip : Box One Bit.
  box () ⇒
    gate q ← init0 @();
    gate q ← H @q;
    gate b ← meas @q;
    output b.
Defined.
```

A Coin Flip Circuit



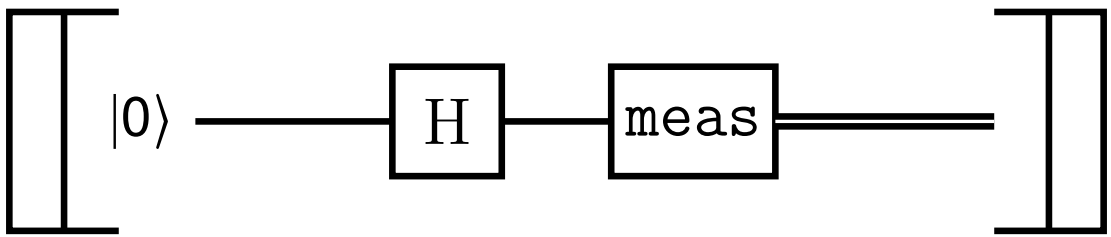

```
Definition coin_flip : Box One Bit.
  box () ⇒
    gate q ← init0 @();
    gate q ← H @q;
    gate b ← meas @q;
    output b.
Defined.
```

Denoting a Coin Flip




```
Lemma fair_toss : ⟦coin_flip⟧ [1]  = [ [1/2,  0]
                                       [0,  1/2] ].
```

Denoting a Coin Flip



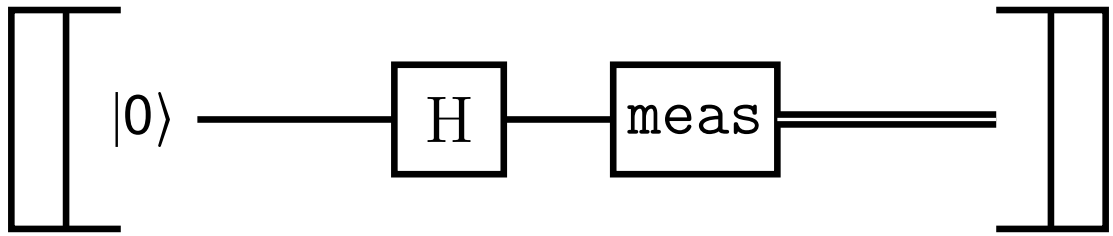
$$=$$

$$|0\rangle\langle 0|\,\mathrm{H}\,|0\rangle\langle 0|\,\mathrm{H}^\dagger\,|0\rangle\langle 0|$$

$$+$$

$$|1\rangle\langle 1|\,\mathrm{H}\,|0\rangle\langle 0|\,\mathrm{H}^\dagger\,|1\rangle\langle 1|$$

Denoting a Coin Flip



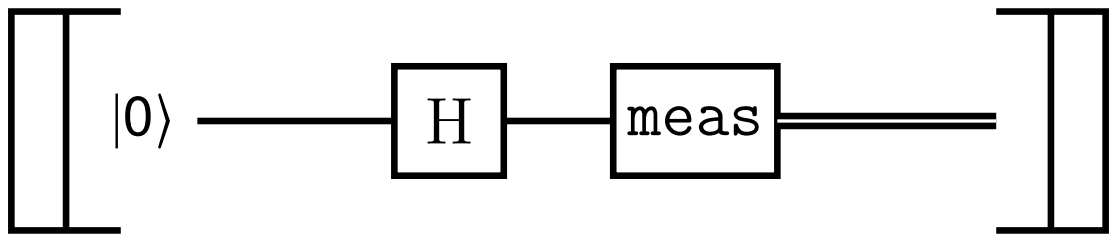
$$=$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$+$$

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Denoting a Coin Flip



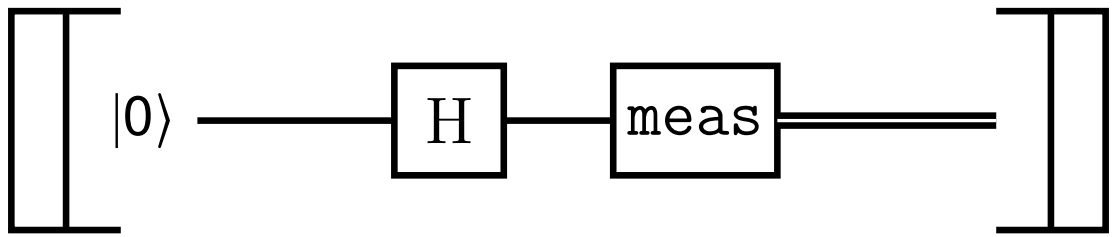
$$=$$

$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix}$$

$$+$$

$$\begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

Denoting a Coin Flip



$$=$$

$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

Verified

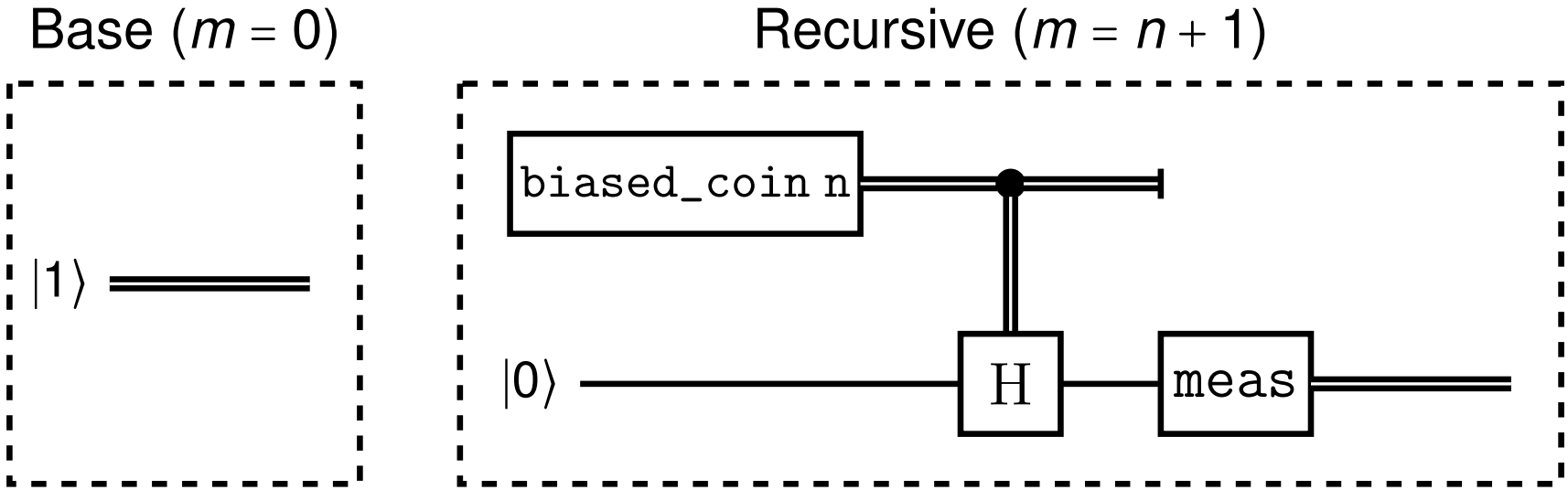
```
Definition fair_coin : Matrix 2 2 :=
  λ x y ⇒ match x, y with
          | 0, 0 ⇒ 1/2
          | 1, 1 ⇒ 1/2
          | _, _ ⇒ 0
          end.

Lemma fair_toss : ⟦ coin_flip⟧  I1  = fair_coin.
Proof.
  repeat (unfold compose_super, super, swap_list,
          swap_two, pad, apply_new0, apply_U,
          apply_meas, denote_pat_in; simpl).
  Msimpl.
  prep_matrix_equality.
  unfold even_toss, ket0, ket1, Mplus, Mmult, conj_transpose.
  Csimpl.
  destruct x, y; Csimpl; destruct_Csolve. Csolve.
Qed.
```

```
U:---  Denotation.v   86% L681  Git:NoDependencies  (Coq Script(0-) +4 company-coq yas hs company Holes Outl)
```

```
fair_toss is defined
```

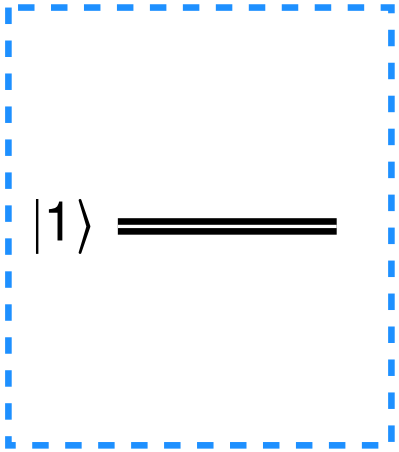
A Biased Coin



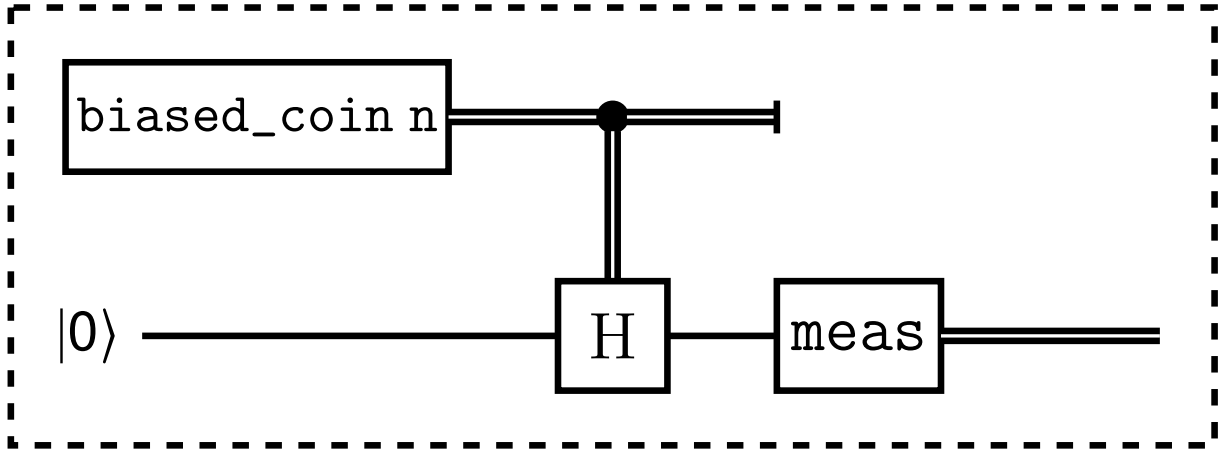

```
Fixpoint biased_coin (m : ℕ) : Box One Bit.
  box () ⇒ match m with
             | 0    ⇒ gate x ← new1 @(); output x
             | n+1  ⇒ let_ c      ← unbox (biased_coin n) ();
                      gate q      ← init0 @();
                      gate (c,q)  ← bit_ctrl H @(c,q);
                      gate ()     ← discard @c;
                      gate b      ← meas @q;
                      output b
             end.
Defined.
```

A Biased Coin

Base ($m = 0$)

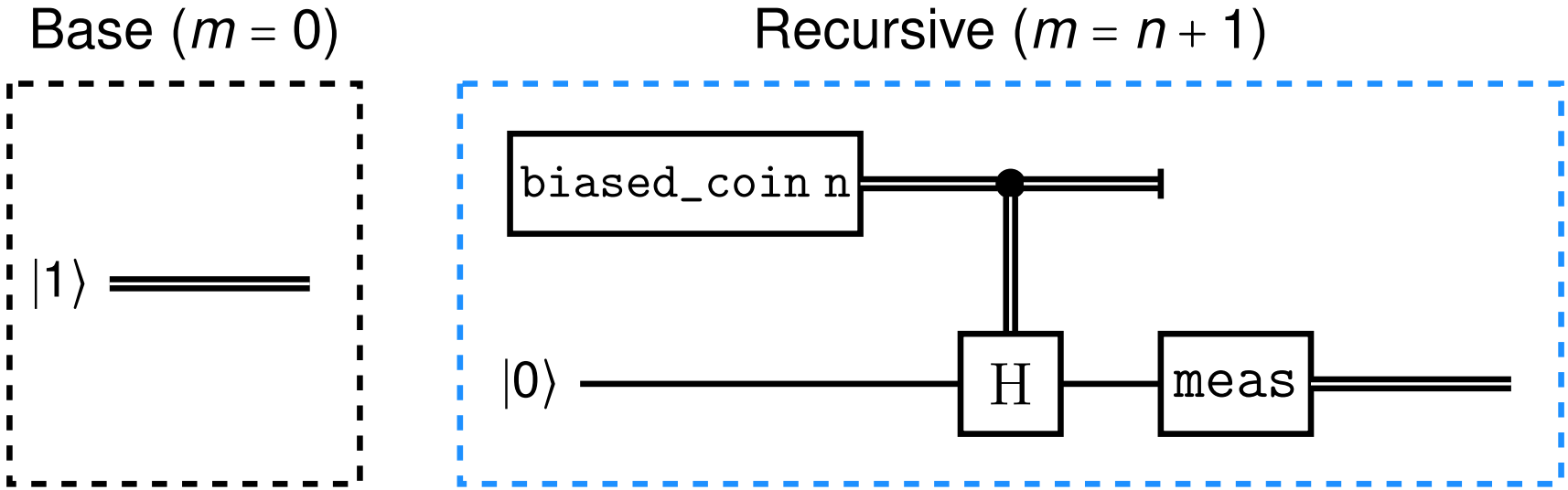


Recursive ($m = n + 1$)



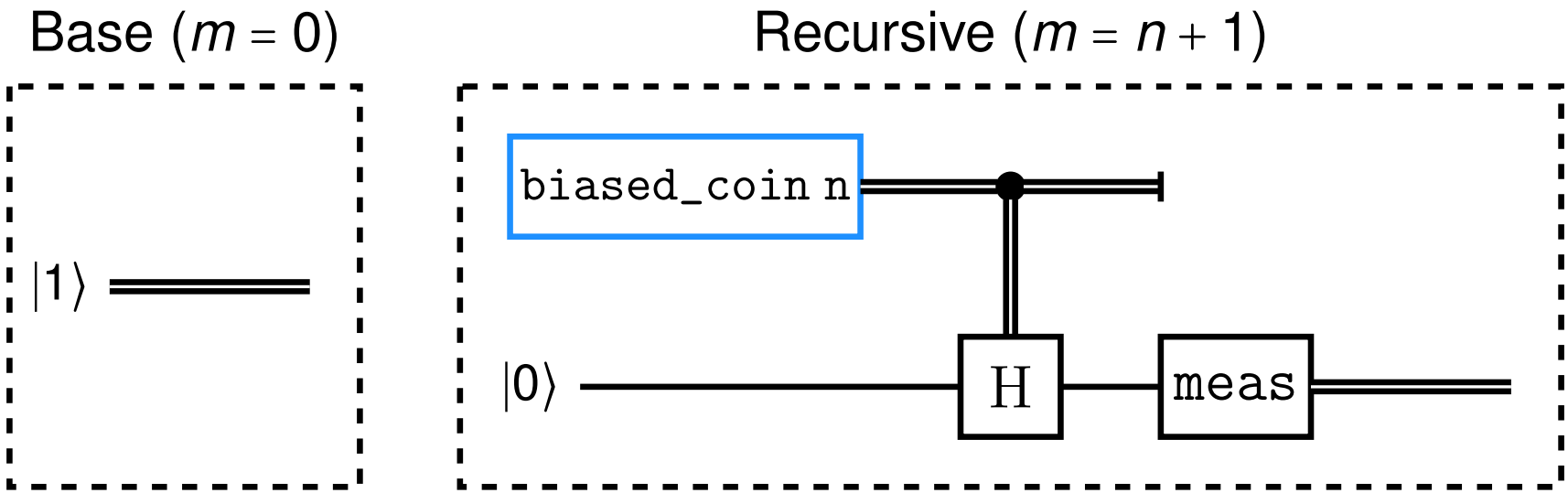

```
Fixpoint biased_coin (m : ℕ) : Box One Bit.
  box () ⇒ match m with
             | 0    ⇒ gate x ← new1 @(); output x
             | n+1  ⇒ let_ c      ← unbox (biased_coin n) ();
                      gate q      ← init0 @();
                      gate (c,q)  ← bit_ctrl H @(c,q);
                      gate ()     ← discard @c;
                      gate b      ← meas @q;
                      output b
             end.
Defined.
```

A Biased Coin



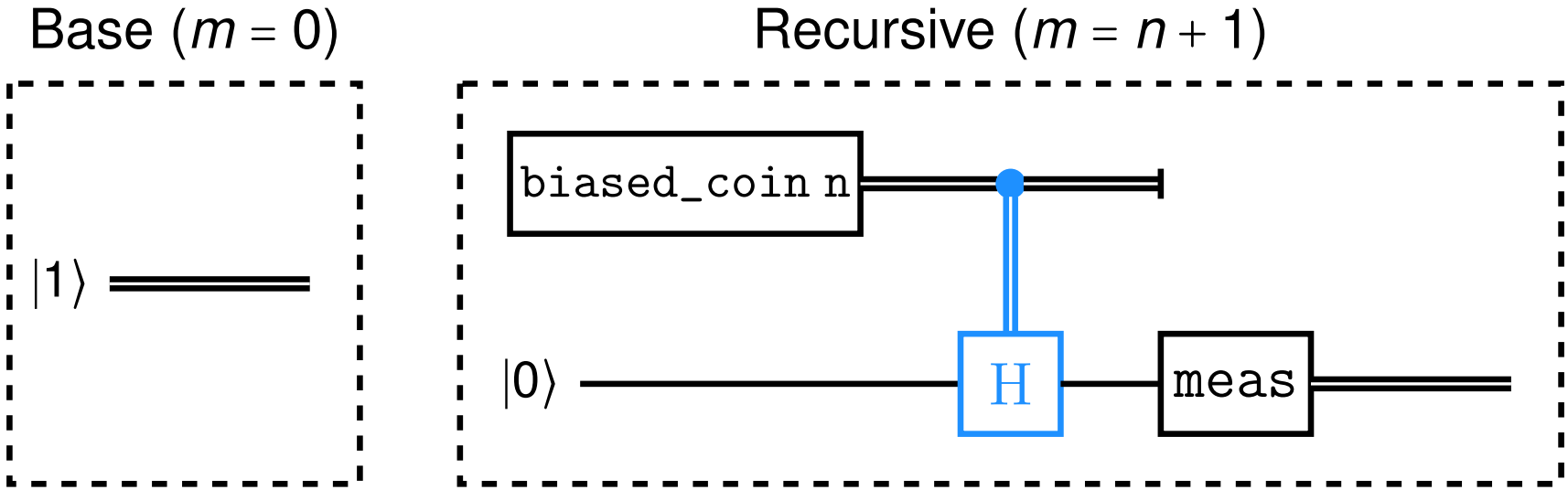

```
Fixpoint biased_coin (m : ℕ) : Box One Bit.
  box () ⇒ match m with
             | 0    ⇒ gate x ← new1 @(); output x
             | n+1  ⇒ let_ c      ← unbox (biased_coin n) ();
                      gate q      ← init0 @();
                      gate (c,q)  ← bit_ctrl H @(c,q);
                      gate ()     ← discard @c;
                      gate b      ← meas @q;
                      output b
             end.
Defined.
```

A Biased Coin



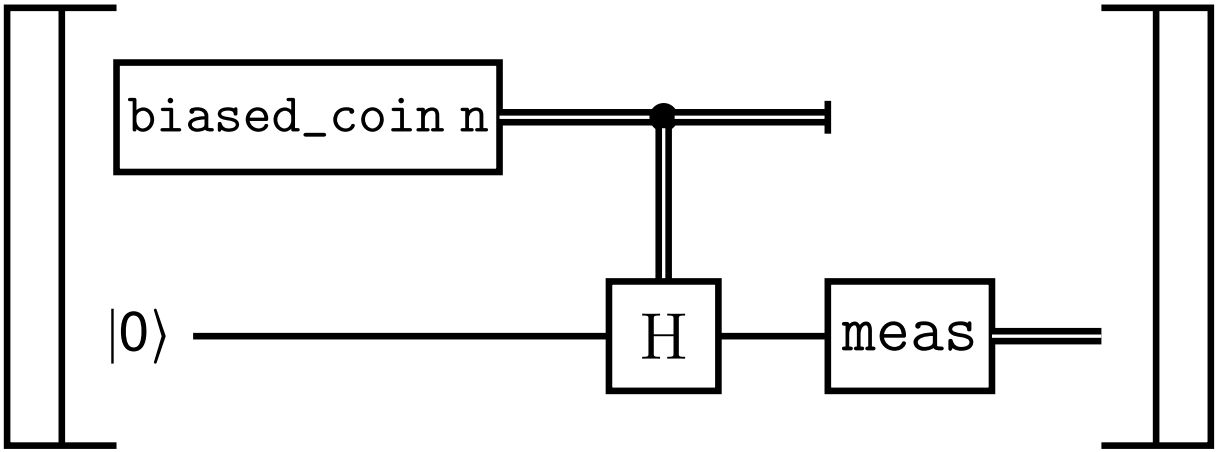

```
Fixpoint biased_coin (m : ℕ) : Box One Bit.
  box () ⇒ match m with
             | 0    ⇒ gate x ← new1 @(); output x
             | n+1  ⇒ let_ c      ← unbox (biased_coin n) ();
                      gate q      ← init0 @();
                      gate (c,q)  ← bit_ctrl H @(c,q);
                      gate ()     ← discard @c;
                      gate b      ← meas @q;
                      output b
             end.
Defined.
```

A Biased Coin

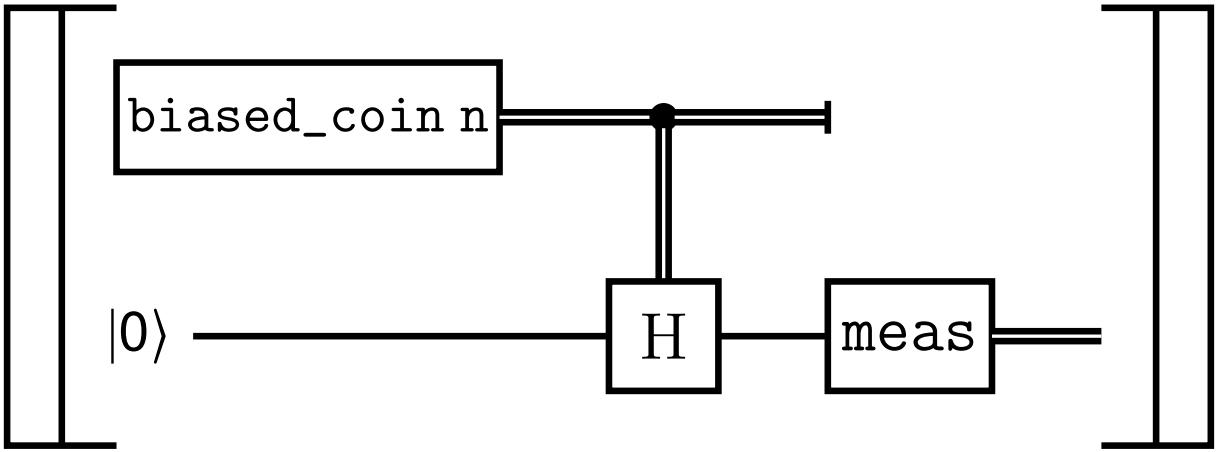



```
Fixpoint biased_coin (m : ℕ) : Box One Bit.
  box () ⇒ match m with
             | 0    ⇒ gate x ← new1 @(); output x
             | n+1  ⇒ let_ c      ← unbox (biased_coin n) ();
                      gate q      ← init0 @();
                      gate (c,q) ← bit_ctrl H @(c,q);
                      gate ()     ← discard @c;
                      gate b      ← meas @q;
                      output b
             end.
Defined.
```

A Biased Coin: Induction

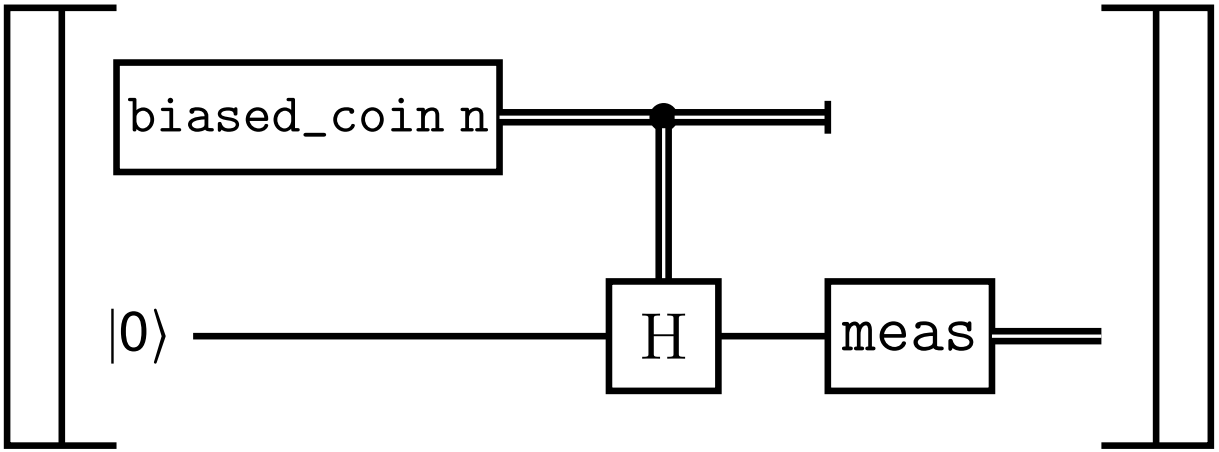


A Biased Coin: Induction



$$\begin{pmatrix} 1 - \frac{1}{2^{n+1}} & 0 \\ 0 & \frac{1}{2^{n+1}} \end{pmatrix}$$

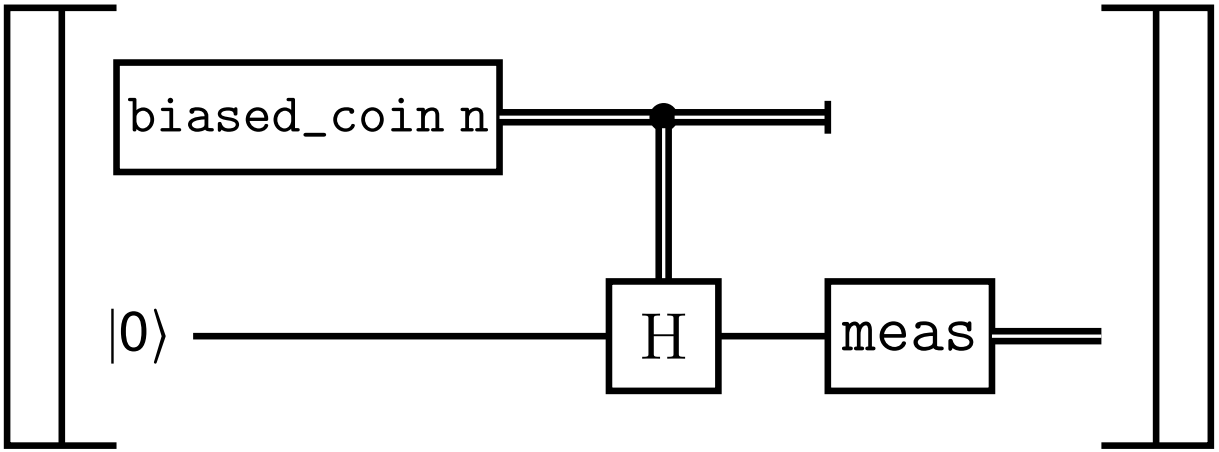
A Biased Coin: Induction



$$\begin{pmatrix} 1 - \frac{1}{2^{n+1}} & 0 \\ 0 & \frac{1}{2^{n+1}} \end{pmatrix}$$

$$[\![\texttt{biased\_coin}\ 0]\!] = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

A Biased Coin: Induction

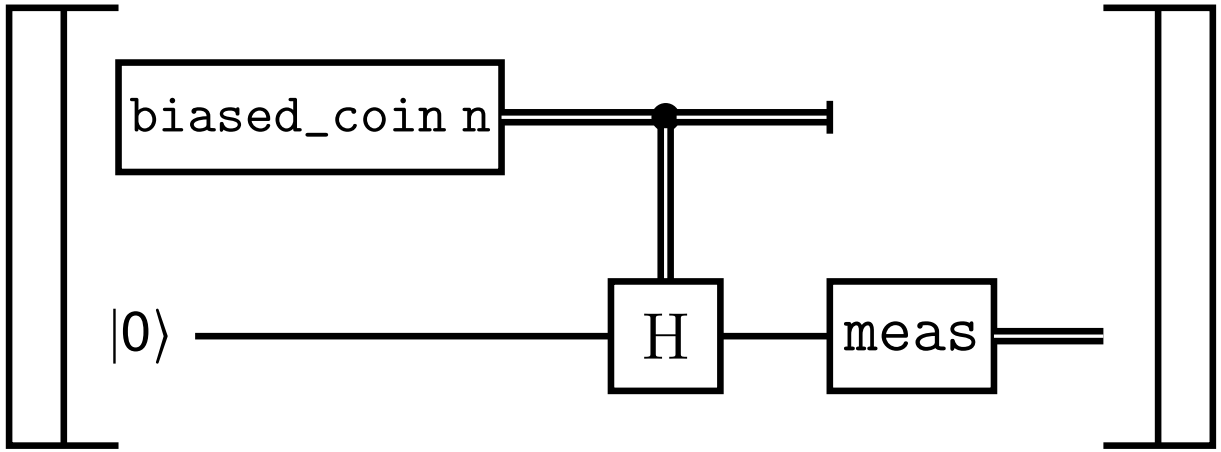


$$\begin{pmatrix} 1 - \frac{1}{2^{n+1}} & 0 \\ 0 & \frac{1}{2^{n+1}} \end{pmatrix}$$

$$[\![\texttt{biased\_coin 0}]\!] = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$[\![\texttt{biased\_coin (n+1)}]\!] = \mathit{meas}_2(\texttt{ctrl-H}([\![\texttt{biased\_coin n}]\!] \otimes |0\rangle\langle 0|))$$

A Biased Coin: Induction



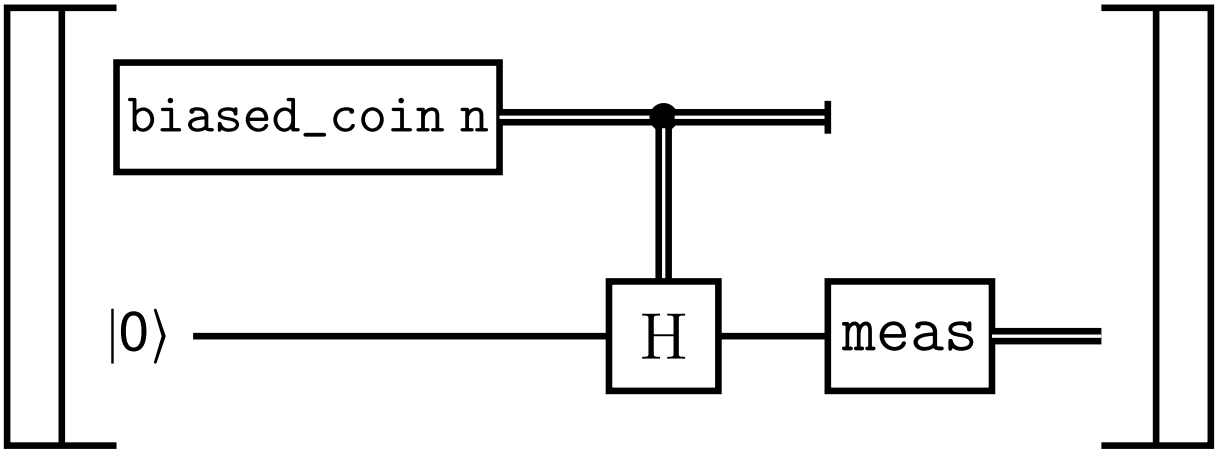
$$\begin{pmatrix} 1-\frac{1}{2^{n+1}} & 0 \\ 0 & \frac{1}{2^{n+1}} \end{pmatrix}$$

$$[\![\texttt{biased\_coin 0}]\!] = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$[\![\texttt{biased\_coin (n+1)}]\!] = \mathit{meas}_2(\texttt{ctrl-H}([\![\texttt{biased\_coin n}]\!] \otimes |0\rangle\langle 0|))$$

$$= \mathit{meas}_2(\texttt{ctrl-H}(\begin{pmatrix} 1-\frac{1}{2^n} & 0 \\ 0 & \frac{1}{2^n} \end{pmatrix} \otimes |0\rangle\langle 0|))$$

A Biased Coin: Induction



$$\begin{pmatrix} 1-\frac{1}{2^{n+1}} & 0 \\ 0 & \frac{1}{2^{n+1}} \end{pmatrix}$$

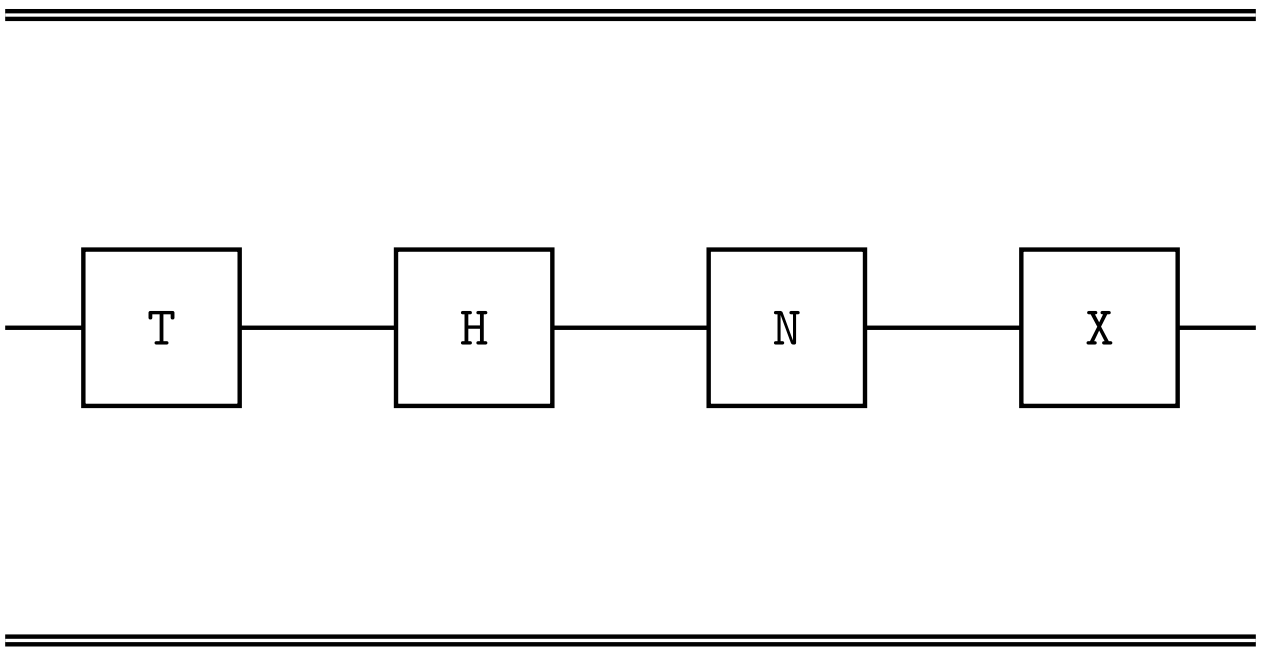
$$[\![\texttt{biased\_coin 0}]\!] = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$[\![\texttt{biased\_coin (n+1)}]\!] = \mathit{meas}_2(\texttt{ctrl-H}([\![\texttt{biased\_coin n}]\!] \otimes |0\rangle\langle 0|))$$

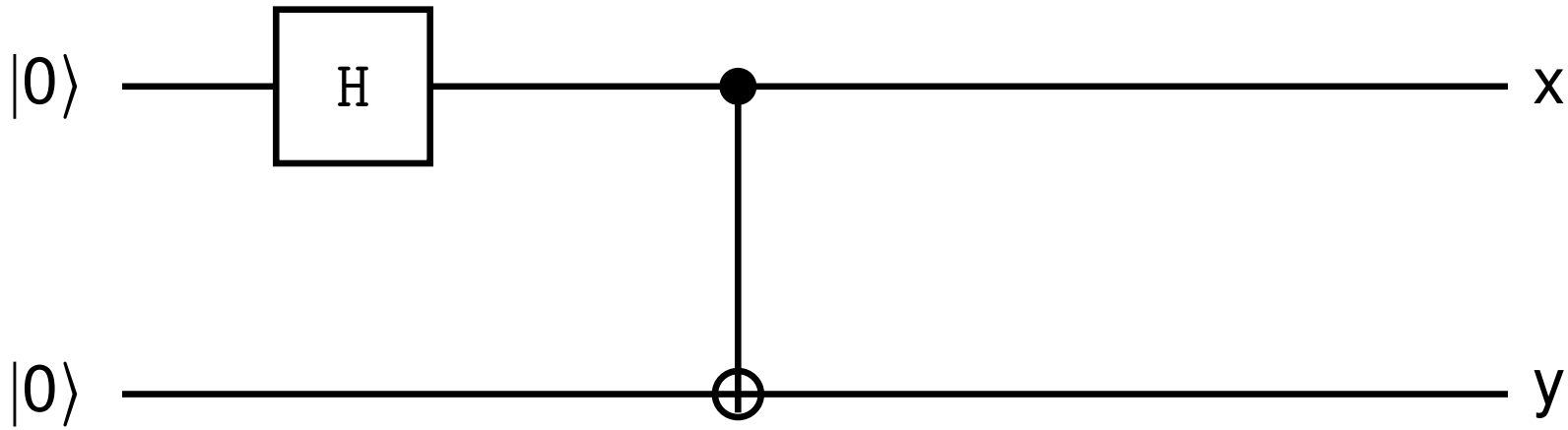
$$= \mathit{meas}_2(\texttt{ctrl-H}(\begin{pmatrix} 1-\frac{1}{2^n} & 0 \\ 0 & \frac{1}{2^n} \end{pmatrix} \otimes |0\rangle\langle 0|))$$

$$= \begin{pmatrix} 1-\frac{1}{2^{n+1}} & 0 \\ 0 & \frac{1}{2^{n+1}} \end{pmatrix}$$

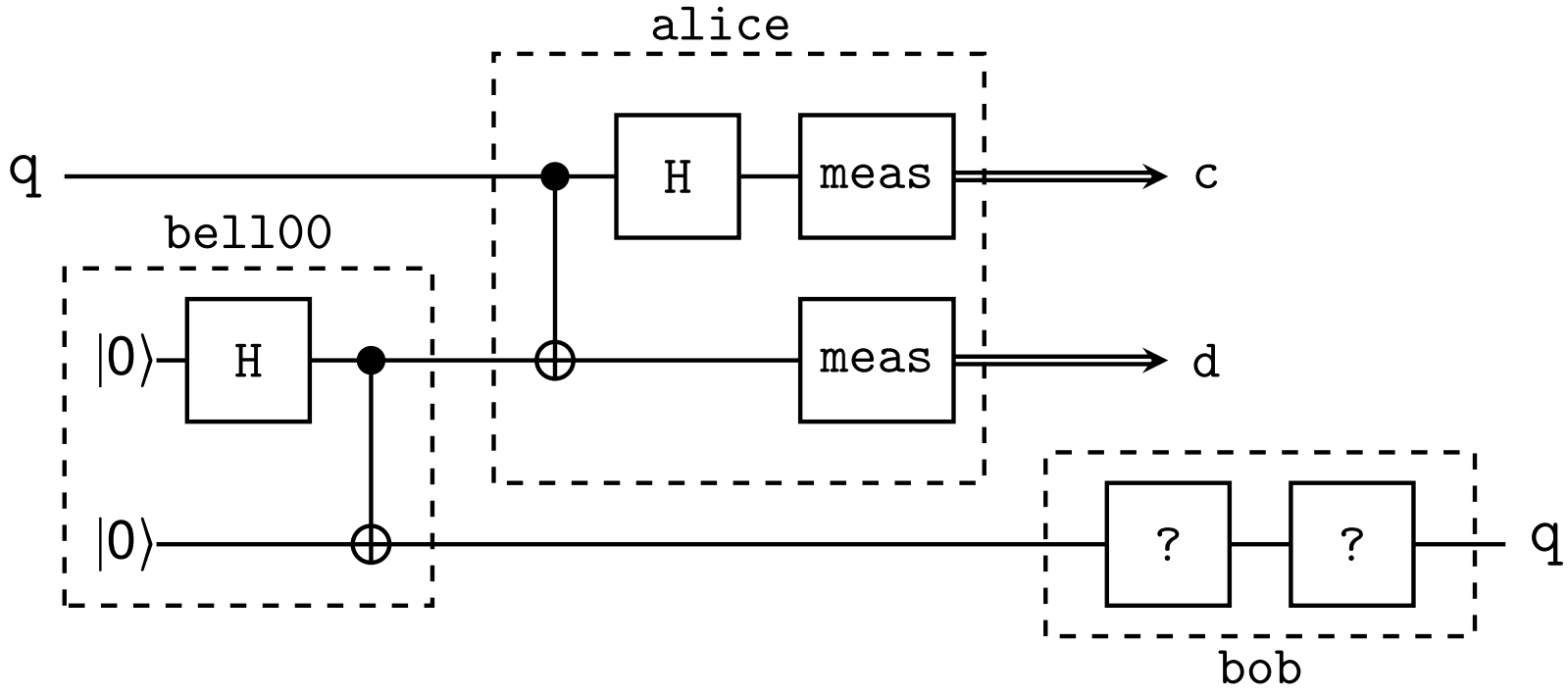
FIN



Entanglement

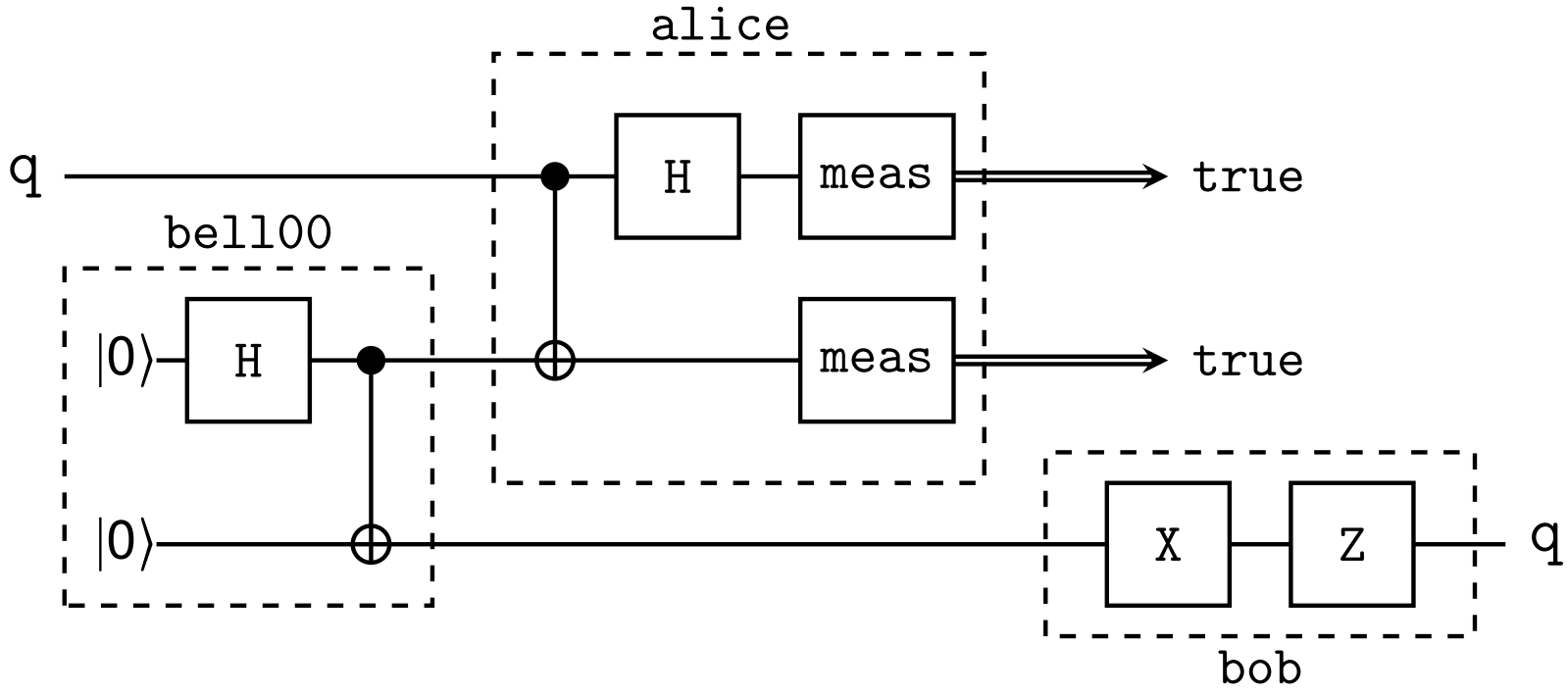


Quantum Teleportation



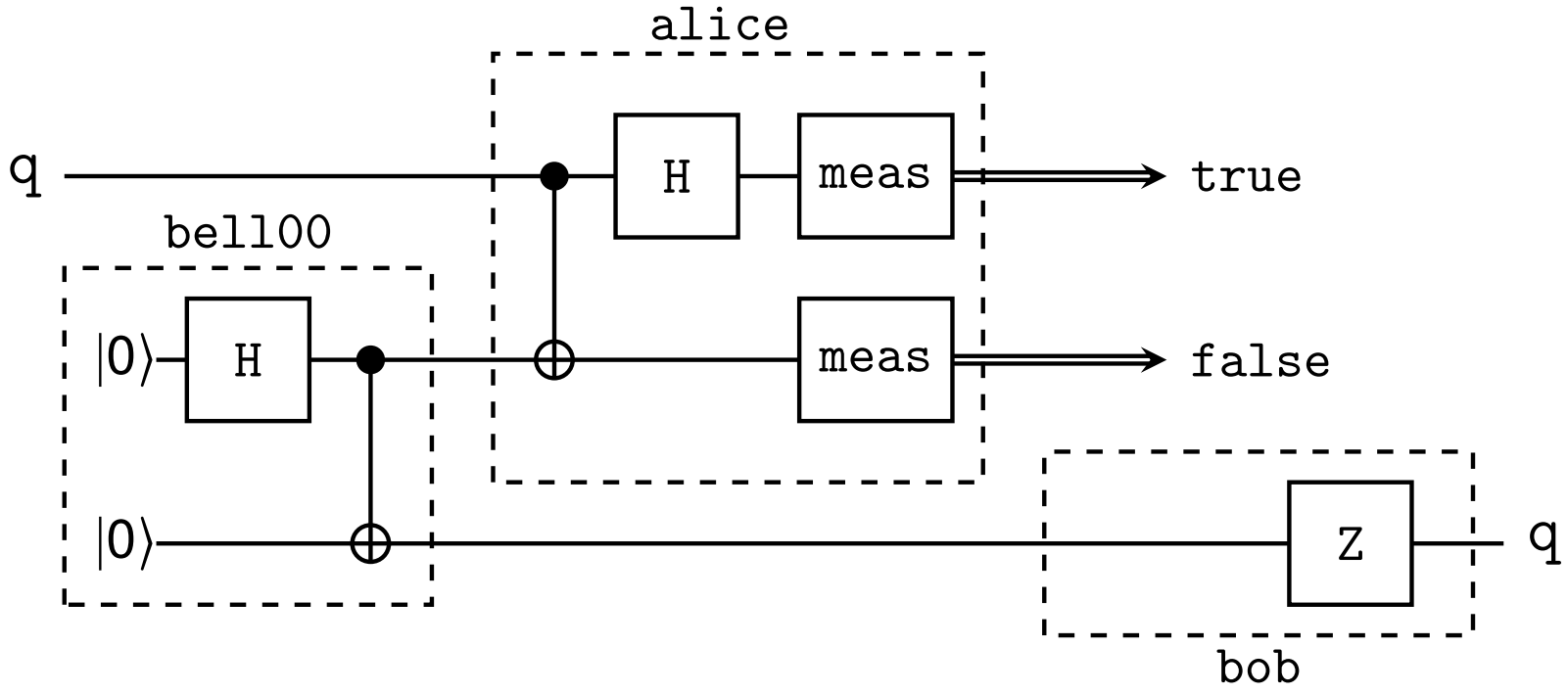

```
Definition teleport : Box Qubit Qubit.
  box_ t ⇒
    let_ (p,q)  ← unbox bell00 () ;
    let_ (a,b)  ← unbox alice (t,p) ;
    lift_ (c,d) ← (a,b) ;
    let_ t'     ← unbox (bob c d) b.
    output t'
Defined.
```

Quantum Teleportation



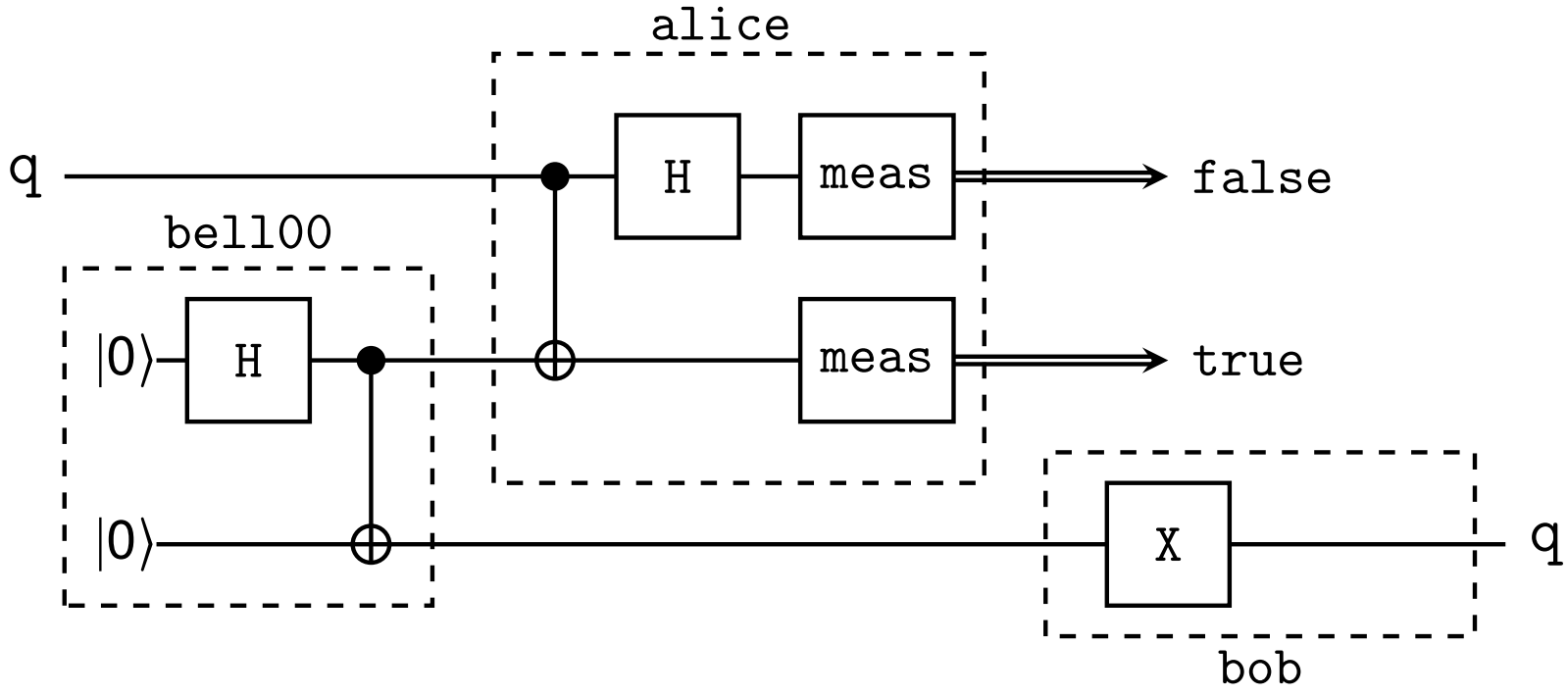

```
Definition teleport : Box Qubit Qubit.
  box_ t ⇒
    let_ (p,q)  ← unbox bell00 () ;
    let_ (a,b)  ← unbox alice (t,p) ;
    lift_ (c,d) ← (a,b) ;
    let_ t'     ← unbox (bob c d) b.
    output t'
Defined.
```

Quantum Teleportation



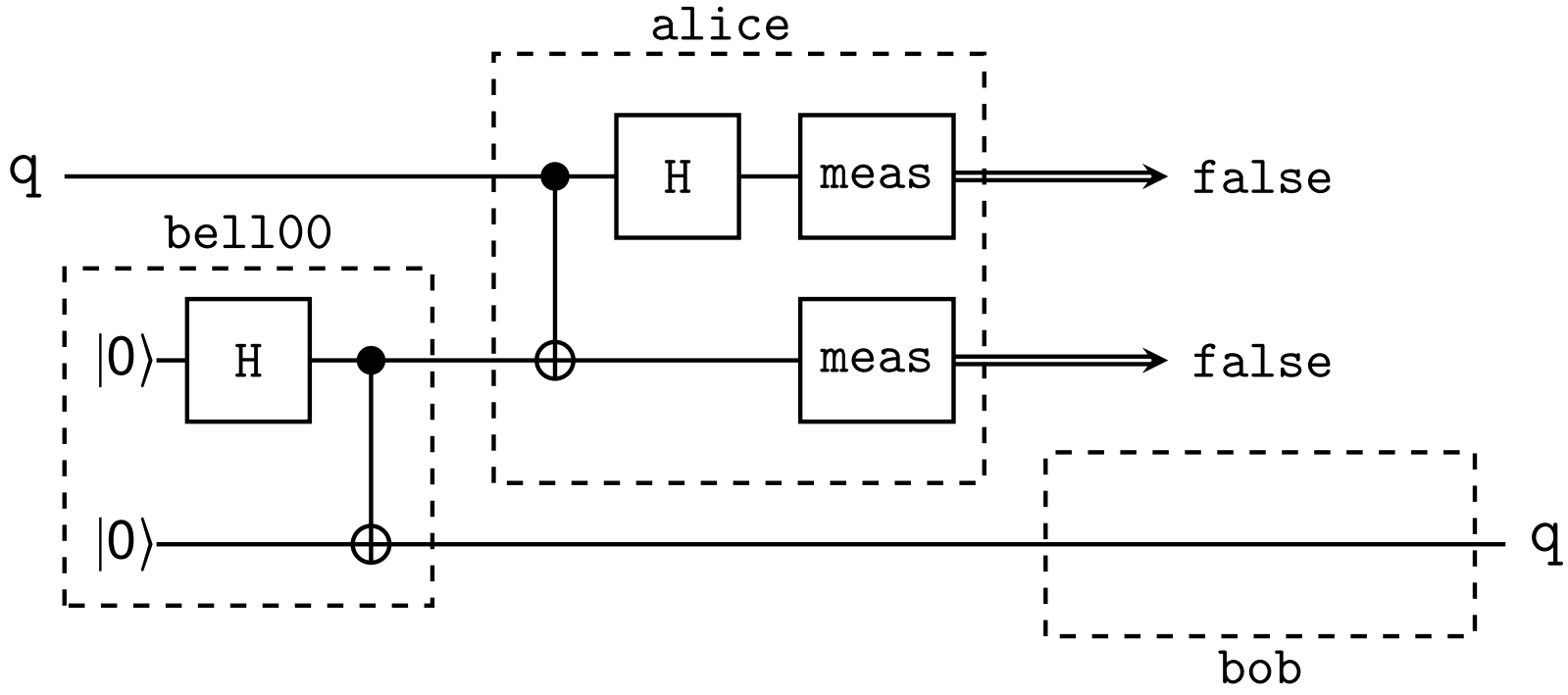

```
Definition teleport : Box Qubit Qubit.
  box_ t ⇒
    let_ (p,q)  ← unbox bell00 () ;
    let_ (a,b)  ← unbox alice (t,p) ;
    lift_ (c,d) ← (a,b) ;
    let_ t'     ← unbox (bob c d) b.
    output t'
Defined.
```

Quantum Teleportation




```
Definition teleport : Box Qubit Qubit.
  box_ t ⇒
    let_ (p,q)  ← unbox bell00 () ;
    let_ (a,b)  ← unbox alice (t,p) ;
    lift_ (c,d) ← (a,b) ;
    let_ t'     ← unbox (bob c d) b.
    output t'
Defined.
```

Quantum Teleportation




```
Definition teleport : Box Qubit Qubit.
  box_ t ⇒
    let_ (p,q)  ← unbox bell00 () ;
    let_ (a,b)  ← unbox alice (t,p) ;
    lift_ (c,d) ← (a,b) ;
    let_ t'     ← unbox (bob c d) b.
    output t'
Defined.
```

The Quantum Fourier Transform

```
Fixpoint qft (n : ℕ) : Box (S n ⊗ Qubit) (S n ⊗ Qubit).
  match n with
  | 0    ⇒ fun _ ⇒ boxed_gate H
  | S n' ⇒ fun _ ⇒ box (fun _ (q,w) ⇒
              let_ w ← unbox (qft n') w;
              unbox (rotations (S n') n') (q,w))
  end.
Defined.
```

The Quantum Fourier Transform

```
Fixpoint qft (n : ℕ) : Box (S n ⊗ Qubit) (S n ⊗ Qubit).
  match n with
  | 0    ⇒ fun _ ⇒ boxed_gate H
  | S n' ⇒ fun _ ⇒ box (fun _ (q,w) ⇒
              let_ w ← unbox (qft n') w;
              unbox (rotations (S n') n') (q,w))
  end.
Defined.
```

$$QFT\,|j\rangle = \frac{1}{2^m} \sum_{k=0}^{2^m-1} e^{2\pi ijk/2^m}\,|k\rangle$$

$\mathcal{FIN}_2$