Proof Patching

Talia Ringer, **Nate Yazdani**, John Leo, Dan Grossman
University of Washington

Proofs are brittle

```
Theorem T1:
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < m + 1.
Proof.
  P1 : T1.
Qed.
```

```
Theorem T2:
  forall (h : nat) (m : nat) (l : list nat),
    hd_error l = value h ->
    Max l m ->
    f < m + 1.
Proof.
  F T1 : T2.
Qed.
```

```
Theorem T1:
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < m + 1.
Proof.
  P1 : T1.
Qed.
```

```
Theorem T2:
  forall (h : nat) (m : nat) (l : list nat),
    hd_error l = value h ->
    Max l m ->
    f < m + 1.
Proof.
  F T1 : T2.
Qed.
```

```
Theorem T1':
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < S m.
Proof.
  P1' : T1'.
Qed.
```

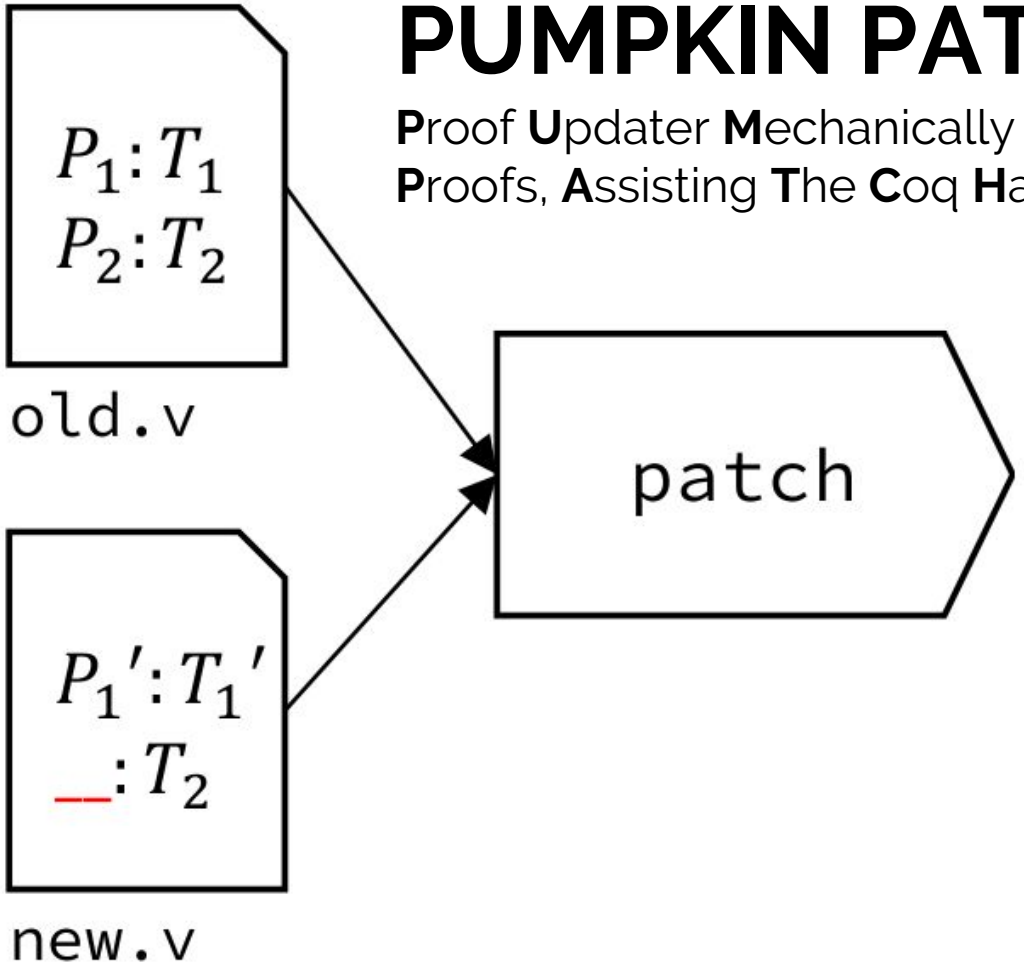
```
Theorem T2:
  forall (h : nat) (m : nat) (l : list nat),
    hd_error l = value h ->
    Max l m ->
    f < m + 1.
Proof.
  F T1' : T2.
Qed.
```

```
Theorem T1':
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < S m.
Proof.
  P1' : T1'.
Qed.
```

```
Theorem T2:
  forall (h : nat) (m : nat) (l : list nat),
    hd_error l = value h ->
    Max l m ->
    f < m + 1.
Proof.
  F T1' : T2. (* ERROR *)
Qed.
```

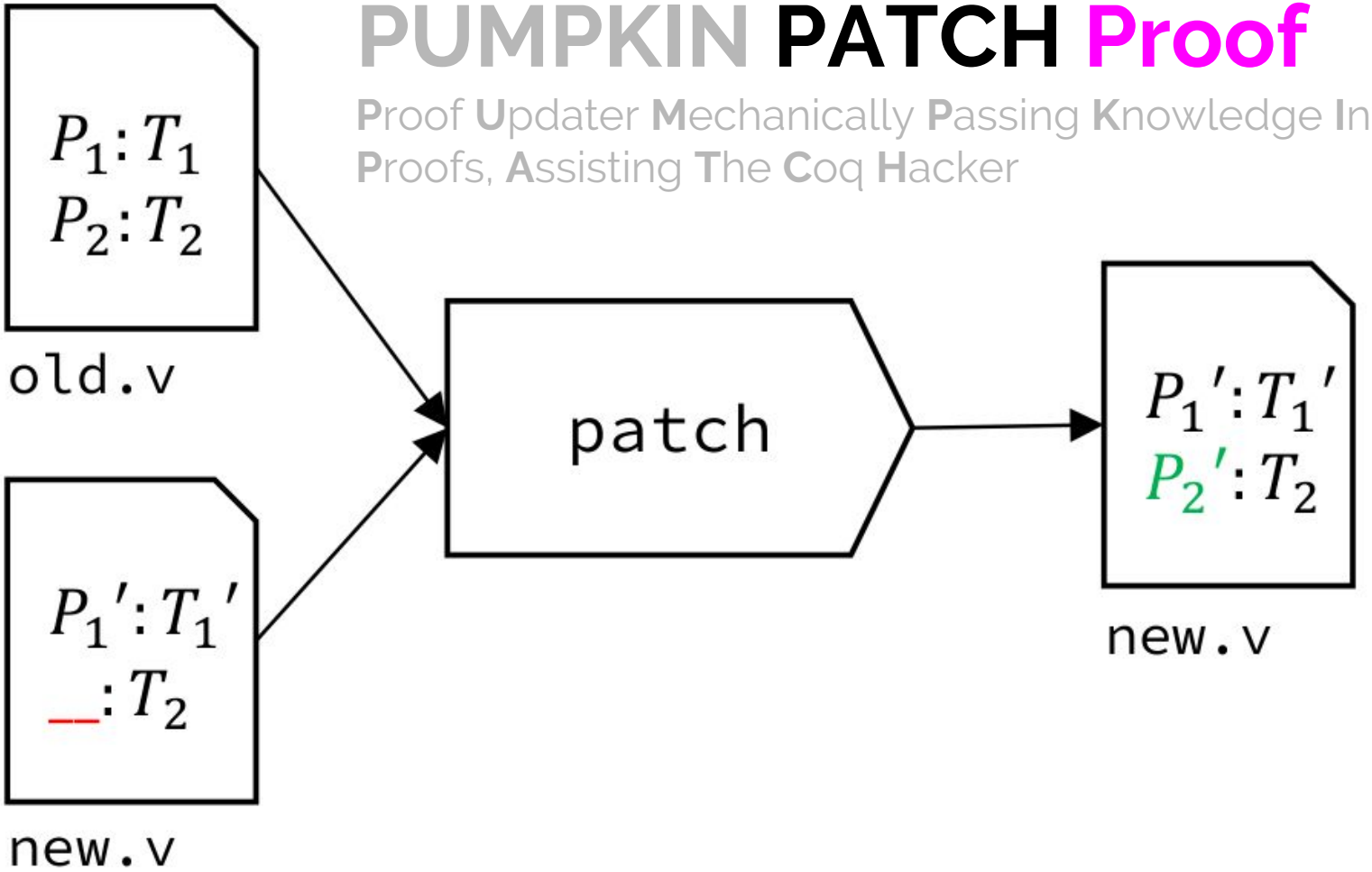
PUMPKIN PATCH

Proof **U**pdater **M**echanically **P**assing **K**nowledge **I**nto **N**ew **P**roofs, **A**ssisting **T**he **C**oq **H**acker



PUMPKIN PATCH Proof

Proof **U**pdater **M**echanically **P**assing **K**nowledge **I**nto **N**ew **P**roofs, **A**ssisting **T**he **C**oq **H**acker



```
Theorem T1':
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < S m.
Proof.
  P1' : T1'.
Qed.
```

```
Theorem T1':
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < S m.
Proof.
  P1' : T1'.
Qed.
```

rewrite plus_comm

```
Theorem T1:
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < m + 1.
Proof.
  P1 : T1.
Qed.
```

```
forall (n : nat) (m : nat) (l : list nat),
  In n l ->
  Max l m ->
  n < S m
```

rewrite plus_comm

```
n < m + 1.
```

Theorem T1'

Proof.

crush.

Qed.

Theorem T1

Proof.

crush.

Qed.

```
T1’ (...) :=
  T_ind
    t
    P’
    pt’
    (fun (...) (IH’ : P’ t) =>
      F IH’)
    ...
```

```
T1 (...) :=
  T_ind
    t
    P
    (eq_ind_r ... pt’ (plus_comm t 1))
    (fun (...) (IH: P t) =>
      F IH)
    ...
```

```
T1’ (...) :=
  T_ind
    t
    P’
    pt’
    (fun (...) (IH’ : P’ t) =>
      F IH’)
    ...
```

```
T1 (...) :=
  T_ind
    t
    P
    pt
    (fun (...) (IH: P t) =>
      (eq_ind_r … (F IH) (plus_comm t 1)))
    ...
```

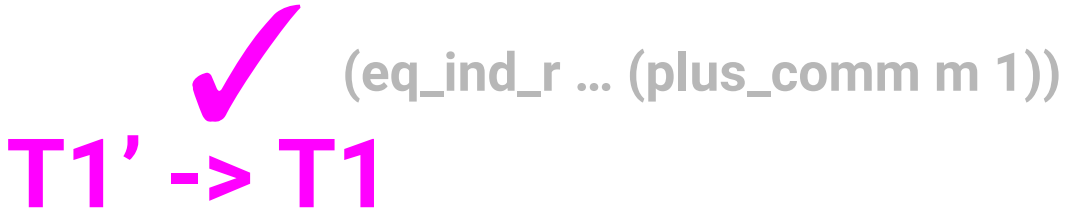
(eq_ind_r … (plus_comm t 1))

(eq_ind_r … (plus_comm t 1))

```
(fun (t0 : nat) =>
    (eq_ind_r … (plus_comm t0 1)))
```

```
(fun (t0 : nat) =>
    (eq_ind_r … (plus_comm t0 1))) m
```

(eq_ind_r … (plus_comm m 1))


✓
(eq_ind_r … (plus_comm m 1))
T1’ -> T1

✓ (eq_ind_r … (plus_comm m 1))

T1’ <- T1

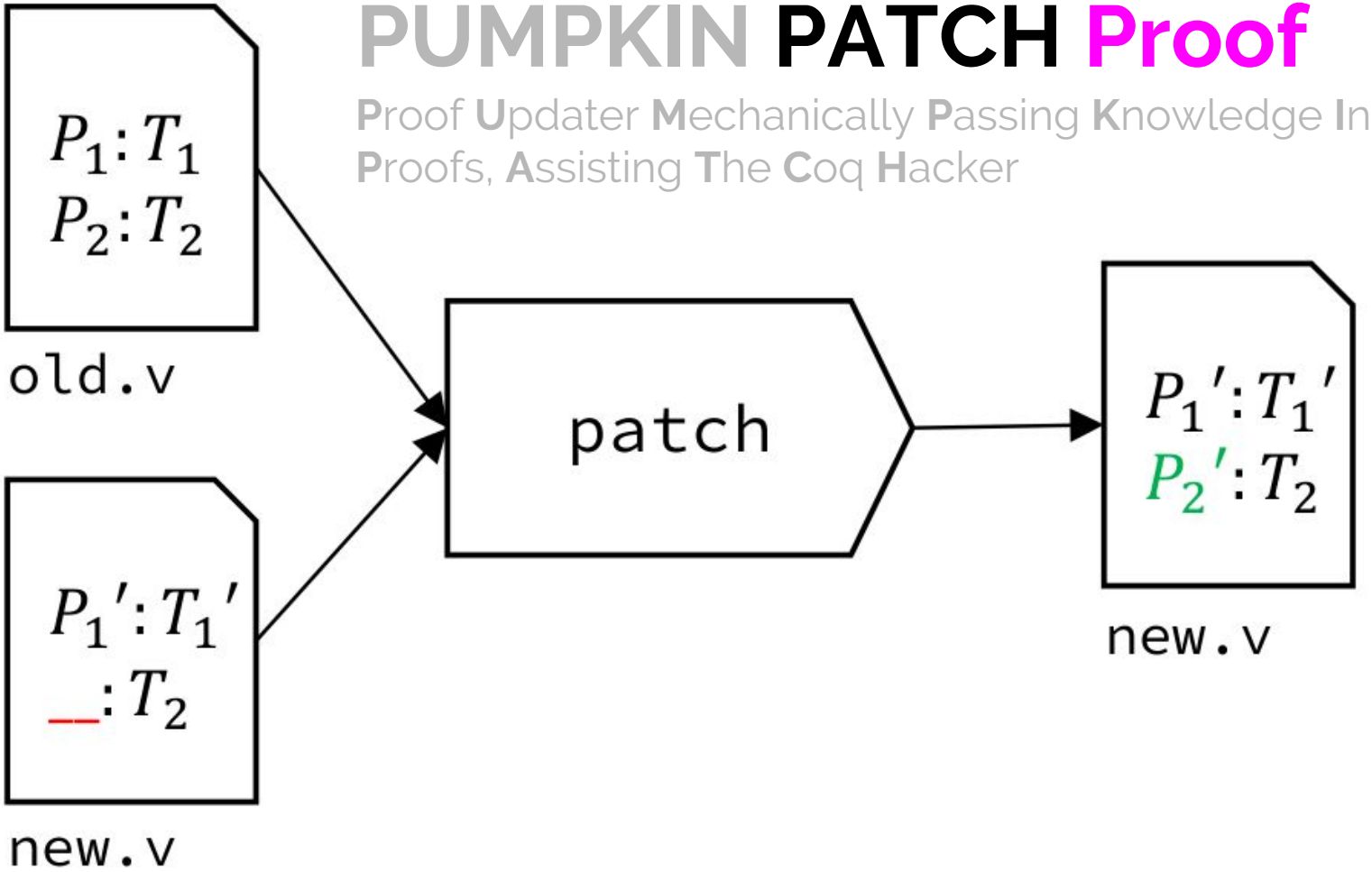
✓ (eq_ind_r … (plus_comm m 1))

T1’ <- T1


(eq_ind … (plus_comm m 1))
T1’ -> T1

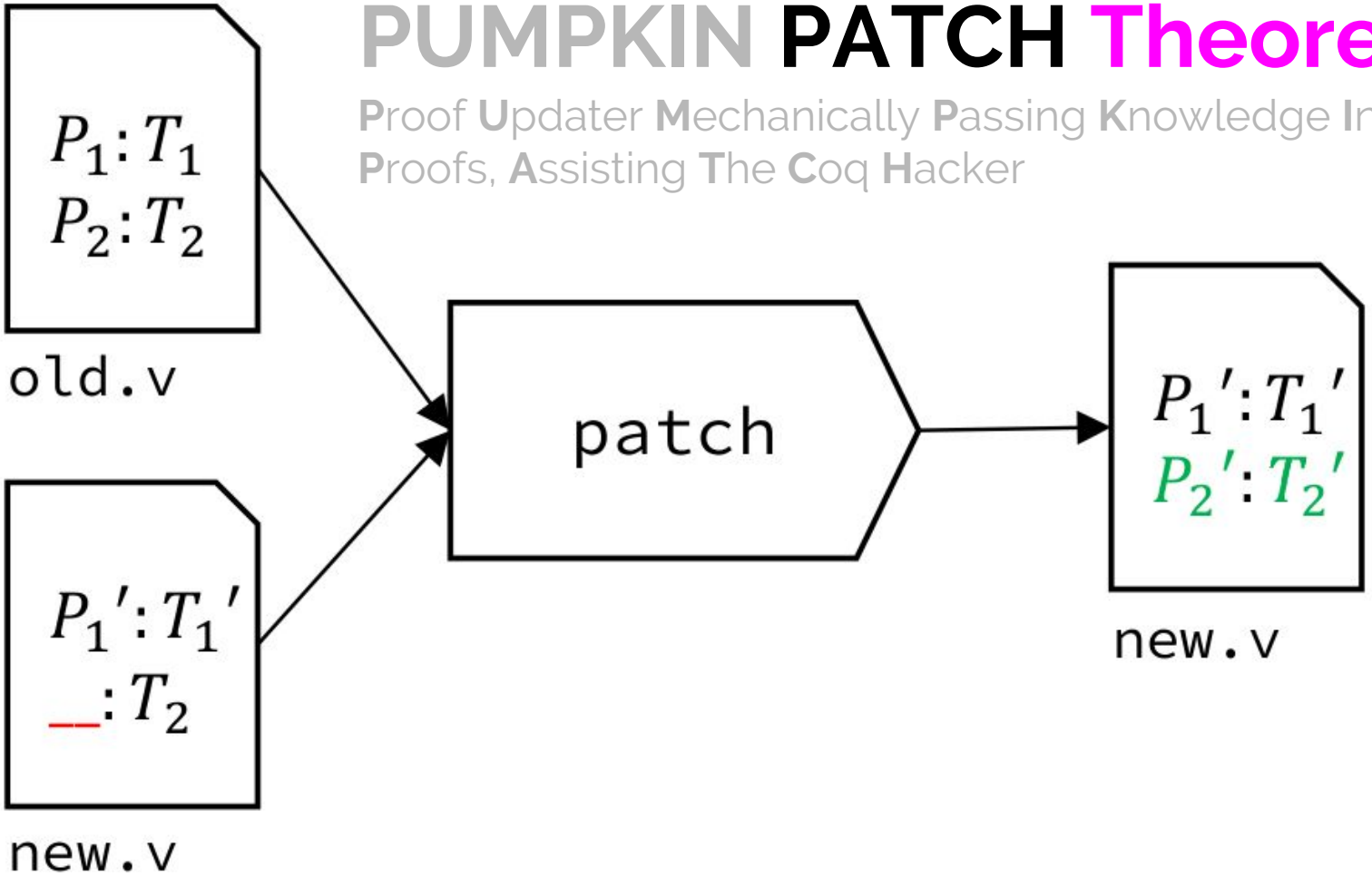
PUMPKIN PATCH Proof

Proof **U**pdater **M**echanically **P**assing **K**nowledge **I**nto **N**ew **P**roofs, **A**ssisting **T**he **C**oq **H**acker



PUMPKIN PATCH Theorem

Proof **U**pdater **M**echanically **P**assing **K**nowledge **I**nto **N**ew **P**roofs, **A**ssisting **T**he **C**oq **H**acker



```
Theorem T1':
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < S m.
Proof.
  P1' : T1'.
Qed.
```

```
Theorem T2:
  forall (h : nat) (m : nat) (l : list nat),
    hd_error l = value h ->
    Max l m ->
    f < m + 1.
Proof.
  F T1' : T2. (* ERROR *)
Qed.
```

```
Theorem T1':
  forall (n : nat) (m : nat) (l : list nat),
    In n l ->
    Max l m ->
    n < S m.
Proof.
  P1' : T1'.
Qed.
```

```
Theorem T2':
  forall (h : nat) (m : nat) (l : list nat),
    hd_error l = value h ->
    Max l m ->
    f < S m.
Proof.
  F T1' : T2'. (* :) *)
Qed.
```

P1 : T1

P2 : T2

P3 : T3

P1’ : T1’

P2 : T2

P3 : T3

PATCH Theorem

P1’ : T1’

P2’ : T2’

P3 : T3

PATCH Theorem $P1' : T1'$

$P2' : T2'$

PATCH Proof $P3' : T3$ ✓

Send us your broken proofs!

tringer@cs.washington.edu