

## 1 Sheeran らの方法 1

$$\left( (\neg \exists s_{[0..k]}. I(s_0) \wedge \text{loopFree}(s_{[0..k]})) \vee (\neg \exists s_{[0..k]}. \text{loopFree}(s_{[0..k]}) \vee \neg P(s_k)) \right) \wedge \bigwedge_{0 \leq i \leq k} (\neg \exists s_{[0..i]}. I(s_0) \wedge \text{path}(s_{[0..i]}) \wedge \neg P(s_i)) \quad (1)$$

### 1.1 証明の概要

ここでは, Sheeran らの方法 1 の健全性を証明する. つぎに, システム安全であるとき以下の条件式が成り立つ.

$$\forall i. \forall s_{[0..i]}. \neg(I(s_0) \wedge \text{path}(s_{[0..i]}) \wedge \neg P(s_i)) \quad (2)$$

さらに, 小さい  $i$  から順に検査していく場合, 閉路を含む実行パスの到達する先はすでに検証済みであることを考慮に入れると式 (2) は以下の式と等価と言える.

$$\forall i. \forall s_{[0..i]}. \neg(I(s_0) \wedge \text{loopFree}(s_{[0..k]}) \wedge \neg P(s_i)) \quad (3)$$

この式は, 初期状態  $s_0$  から性質  $P$  を満たさない状態  $s_i$  に到達するような閉路を含まない実行パスは, 存在しないことを表す. 従って, 「式 (1) ならば, 式 (3) が成り立つ」を証明すれば, Sheeran らの方法 1 の健全性を示せたことになる.

以下, 証明の流れを述べる. まず,  $i < k$  と  $i \geq k$  に場合分けを行い, それぞれ

$$\forall s_{[0..i]}. \neg(I(s_0) \wedge \text{loopFree}(s_{[0..i]}) \wedge \neg P(s_i)) \quad (4)$$

が成り立つことを証明する.

$i < k$  の場合, 式 (1) の 3 行目より,  $k$  ステップ目まで安全であることが分かっているため, 式 (9) が成り立つ.

つぎに,  $i \geq k$  の場合について証明する. 式 (1) より, (a)  $\forall s_{[0..k]}. \neg(I(s_0) \wedge \text{loopFree}(s_{[0..k]}))$  (b)  $\forall s_{[0..k]}. \neg(\text{loopFree}(s_{[0..k]}) \wedge \neg P(s_k))$  のどちらかが成り立つことがわかっている. ここで, 閉路を含まない実行パス  $\text{loopFree}(s_{[0..i]})$  は分割できることを利用すると, 式 (3) の部分式  $I(s_0) \wedge \text{loopFree}(s_{[0..i]}) \wedge \neg P(s_i)$  は以下のように書くことができる.

$$I(s_0) \wedge \text{loopFree}(s_{[0..k]}) \wedge \text{loopFree}(s_{[k..i]}) \wedge \neg P(s_i) \quad (5)$$

式 (5) は, (a) が成り立つとき否定されるので, 式 (9) が成り立つ. また,  $I(s_0) \wedge \text{loopFree}(s_{[0..i]}) \wedge \neg P(s_i)$  は, 以下のように分割することもできる.

$$I(s_0) \wedge \text{loopFree}(s_{[0..(i-k)]}) \wedge \text{loopFree}(s_{[(i-k)..i]}) \wedge \neg P(s_i) \quad (6)$$

そして,  $i \geq k$  の場合, (b) は以下のように考えて良い.

$$\left( \forall s_{[0..k]}. \neg(\text{loopFree}(s_{[0..k]}) \wedge \neg P(s_k)) \right) \Leftrightarrow \left( \forall s_{[(i-k)..i]}. \neg(\text{loopFree}(s_{[(i-k)..i]}) \wedge \neg P(s_i)) \right) \quad (7)$$

(b) が成り立つときは, 式 (6) が否定されるので, 式 (9) が成り立つ. よって,  $i \leq k$  の場合も, 式 (9) が成り立つ.

以上により, すべての  $i$  について式 (9) が成り立つので, Sheeran らのエンコード方法 1 の健全性が示された.

## 1.2 Coq 上での証明

```
Threorem Proof_Sheeran_method1
|
|_ Lemma Proof_Sheeran_method1_case1
|   |
|   |_ Lemma case1_t1
|   |   |
|   |   |_ Lemma case1_t1'
|   |   |
|   |   |_ Lemma ss_property (Admitted!)
|   |
|   |_ Lemma Proof_Sheeran_method1_case2
|   |   |
|   |   |_ Lemma case2_t1
|   |   |   |
|   |   |   |_ Lemma ss_property
|   |   |   |
|   |   |   |_ Lemma divide_loop_free
|   |   |
|   |   |_ Lemma case2_t2
|   |   |   |
|   |   |   |_ Lemma ss_property
|   |   |   |
|   |   |   |_ Lemma divide_loop_free
|   |   |   |
|   |   |   |_ Lemma case2_t2'
|   |   |   |   |
|   |   |   |   |_ Lemma case2_t2''
|   |   |   |   |   |
|   |   |   |   |   |_ Lemma a
|   |   |   |   |   |   |
|   |   |   |   |   |   |_ Lemma itl_prop (Admitted!)
|   |   |   |   |   |
|   |   |   |   |   |_ Lemma b
|   |   |   |   |   |   |
|   |   |   |   |   |   |_ Lemma itl_prop
|   |   |   |   |   |   |
|   |   |   |   |   |   |_ Lemma b'
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |_ Lemma itl_prop
|   |   |   |   |   |
|   |   |   |   |   |_ Lemma c
|   |   |   |   |   |   |
|   |   |   |   |   |   |_ Lemma itl_prop
|   |   |   |   |
|   |   |   |   |_ Lemma d (Admitted!)
```

```
Lemma divide_loop_free
|
|_ Lemma divide_path
|   |
|   |_ Lemma shift_path
|   |   |
|   |   |_ Lemma divide_tl_path
|   |   |
|   |   |_ Lemma divide_hd_path
|
|_ Lemma divide_loop_check
|   |
|   |_ Lemma divide_lc1
|   |
|   |_ Lemma divide_lc2
|   |   |
|   |   |_ Lemma divide_lc2'
|   |   |   |
|   |   |   |_ Lemma divide_lc2''
```

下に証明の概要を書く.

---

```
Theorem Proof_Sheeran_method1 :  
  ∀ (k : ℕ) (I : init) (T : trans) (P : property) (l : list ℕ),  
    Sheeran_method1 I T P l k → (∀ (i : ℕ) (s : ss), ¬(I (s 0) ∧ loop_free T s 0 i l ∧ ¬P (s i))).  
Proof.  
  intros.  
  assert (H0 : i < k ∨ i ≥ k).  
  omega.  
  destruct H0.  
  - revert H0. (* i < k → ¬(I (s 0) ∧ loop_free T s 0 i l ∧ ¬P (s i))). *)  
    apply Proof_Sheeran_method1_case1.  
    tauto.  
  - revert H0. (* i ≥ k → ¬(I (s 0) ∧ loop_free T s 0 i l ∧ ¬P (s i))). *)  
    apply Proof_Sheeran_method1_case2.  
    tauto.  
Qed.
```

---

まず, Proof\_Sheeran\_method1\_case1 を証明する.

---

**Lemma** case1\_t1:  $\forall (i\ k : \mathbb{N})(s : ss)(I : \text{init})(T : \text{trans})(P : \text{property}),$   
 $(i < k) \wedge \text{kth\_P\_safe } I\ T\ P\ k \rightarrow \neg(I(s\ 0) \wedge \text{path } T\ s\ 0\ i \wedge \neg P(s\ i)).$

**Theorem** Proof\_Sheeran\_method1\_case1 :

$\forall (k : \mathbb{N})(I : \text{init})(T : \text{trans})(P : \text{property})(l : \text{list } \mathbb{N}),$   
 $\text{Sheeran\_method1 } I\ T\ P\ l\ k$   
 $\rightarrow (\forall (i : \mathbb{N})(s : ss), (i < k)\%N \rightarrow \neg(I(s\ 0) \wedge \text{loop\_free } T\ s\ 0\ i\ l \wedge \neg P(s\ i))).$

**Proof.**

`unfold Sheeran_method1.`  
`intros.`  
`assert((i < k)  $\wedge$  kth_P_safe I T P k).`  
`tauto.`  
`apply case1_t1 with (s := s) in H1.`  
`unfold loop_free.`  
`tauto.`

**Qed.**

---

case1\_t1 を apply するために, assert を使って, 式変形を行っているだけ. kth\_P\_safe は, 式 (1) の 2 行目の部分を表している.

次に, Proof\_Sheeran\_method1\_case2 を証明する.

---

**Lemma** case2\_t1:  $\forall (i\ k : \mathbb{N})(T : \text{trans})(P : \text{property})(l : \text{list } \mathbb{N}),$   
 $(i \geq k)\%N \rightarrow$   
 $(\forall s1 : ss, \neg(\text{loop\_free } T\ s1\ (i-k)\ k\ l \wedge \neg P(s1\ i))) \leftrightarrow$   
 $(\forall s2 : ss, \neg(\text{loop\_free } T\ s2\ 0\ k\ l \wedge \neg P(s2\ k))).$

**Proof.** Admitted.

**Theorem** divide\_loop\_free:  $\forall (i\ j : \mathbb{N})(s : ss)(T : \text{trans})(l : \text{list } \mathbb{N}),$   
 $\text{loop\_free } T\ s\ 0\ (i+j)\ l \rightarrow \text{loop\_free } T\ s\ 0\ i\ l \wedge \text{loop\_free } T\ s\ i\ j\ l.$

**Theorem** Proof\_Sheeran\_method1\_case2 :

$\forall (k : \mathbb{N})(I : \text{init})(T : \text{trans})(P : \text{property})(l : \text{list } \mathbb{N}),$   
 $\text{Sheeran\_method1 } I\ T\ P\ l\ k \rightarrow$   
 $(\forall (i : \mathbb{N})(s : ss), (i \geq k)\%N \rightarrow \neg(I(s\ 0) \wedge \text{loop\_free } T\ s\ 0\ i\ l \wedge \neg P(s\ i))).$

**Proof.**

`unfold Sheeran_method1.`  
`intros.`  
`apply neg_false. (*結論部分を  $I(s\ 0) \wedge \text{loop\_free } T\ s\ 0\ i\ l \wedge \neg P(s\ i) \leftrightarrow \text{False}$  に変形*)`  
`split. (*  $\rightarrow$  と  $\leftarrow$  に分ける. *)`  
`- intros.`  
`destruct H.`  
`destruct H.`  
`+ assert (H3: i = k + (i - k)). omega.`  
`unfold lasso in H.`  
`destruct H1.`  
`destruct H4.`

```

    rewrite H3 in H4.
    apply divide_loop_free in H4.
    firstorder.
+ unfold violate_loop_free in H.
  simpl in H.
  destruct H1.
  destruct H3.
  assert (H5 : i = i - k + k). omega.
  rewrite H5 in H3.
  apply divide_loop_free in H3.
  apply (case2_t1 i k T P 1) in H0.
  rewrite ← H0 in H.
  firstorder.
- tauto.
Qed.

```

---

case2\_t1 と divide\_loop\_free を使用して, 証明した. この証明は, 前節の  $k \leq i$  の場合の証明と同じ流れで証明している. case2\_t1 は, まだ未証明.

以下, 使用した補題一覧.

---

**Lemma** `divide_path`:  $\forall (i\ j: \mathbb{N}) (s: ss) (T: trans),$   
 $path\ T\ s\ 0\ (i+j) \rightarrow path\ T\ s\ 0\ i \wedge path\ T\ s\ i\ j.$

**Lemma** `divide_loop_check`:  $\forall (i\ j: \mathbb{N}) (s: ss) (l: list\ \mathbb{N}),$   
 $loop\_check\ s\ 0\ (i+j)\ l \rightarrow loop\_check\ s\ 0\ i\ l \wedge loop\_check\ s\ i\ j\ l.$

**Lemma** `divide_loop_free`:  $\forall (i\ j: \mathbb{N}) (s: ss) (T: trans) (l: list\ \mathbb{N}),$   
 $loop\_free\ T\ s\ 0\ (i+j)\ l \rightarrow loop\_free\ T\ s\ 0\ i\ l \wedge loop\_free\ T\ s\ i\ j\ l.$

**Lemma** `divide_tl_path`:  $\forall (i: \mathbb{N}) (s: ss) (T: trans),$   
 $path\ T\ s\ 0\ (S\ i) \leftrightarrow path\ T\ s\ 0\ i \wedge T\ (s\ i)\ (s\ (S\ i)).$

**Lemma** `divide_hd_path`:  $\forall (i\ j: \mathbb{N}) (s: ss) (T: trans),$   
 $T\ (s\ i)\ (s\ (S\ i)) \wedge path\ T\ s\ (S\ i)\ j \leftrightarrow path\ T\ s\ i\ (S\ j).$

**Lemma** `shift_path`:  $\forall (i\ j: \mathbb{N}) (s: ss) (T: trans),$   
 $path\ T\ s\ 0\ i \wedge path\ T\ s\ i\ (S\ j) \leftrightarrow path\ T\ s\ 0\ (S\ i) \wedge path\ T\ s\ (S\ i)\ j.$

**Lemma** `divide_path`:  $\forall (i\ j: \mathbb{N}) (s: ss) (T: trans),$   
 $path\ T\ s\ 0\ (i+j) \rightarrow path\ T\ s\ 0\ i \wedge path\ T\ s\ i\ j.$

**Lemma** `divide_lc1`:  $\forall (j\ i: \mathbb{N}) (s: ss) (l: list\ \mathbb{N}),$   
 $loop\_check\ s\ 0\ (i+j)\ l \rightarrow loop\_check\ s\ 0\ i\ l.$

**Lemma** `divide_lc2'`:  $\forall (i\ j\ k: \mathbb{N}) (s: ss) (l: list\ \mathbb{N}),$   
 $loop\_check'\ s\ i\ (S\ k)\ (S\ j)\ l \leftrightarrow$   
 $neq\_nth\_mth\ (s\ (i + (S\ k)))\ (s\ i)\ l\ 0 \wedge loop\_check'\ s\ (S\ i)\ k\ j\ l.$

**Lemma** `divide_lc2'`:  $\forall (j\ i: \mathbb{N}) (s: ss) (l: list\ \mathbb{N}),$   
 $loop\_check\ s\ i\ (S\ j)\ l \rightarrow loop\_check\ s\ (S\ i)\ j\ l.$

**Lemma** `divide_lc2`:  $\forall (i\ j: \mathbb{N}) (s: ss) (l: list\ \mathbb{N}),$   
 $loop\_check\ s\ 0\ (i+j)\ l \rightarrow loop\_check\ s\ i\ j\ l.$

**Lemma** `divide_loop_check`:  $\forall (i\ j: \mathbb{N}) (s: ss) (l: list\ \mathbb{N}),$   
 $loop\_check\ s\ 0\ (i+j)\ l \rightarrow loop\_check\ s\ 0\ i\ l \wedge loop\_check\ s\ i\ j\ l.$

**Lemma** `lt_big_and_incl`:  $\forall (i\ k: \mathbb{N}) (P: \mathbb{N} \rightarrow Prop),$   
 $i < k \wedge big\_and\ P\ 0\ (S\ k) \rightarrow big\_and\ P\ 0\ (S\ i).$

---

## 2 Sheeran らの方法 2

$k$ -induction として知られる検査法.

$$\left( (\neg \exists s_{[0..k]}. I(s_0) \wedge \text{loopFree}(s_{[0..k]})) \vee (\neg \exists s_{[0..k]}. \text{loopFree}(s_{[0..k]}) \vee \text{all}.P(s_{[0..(k-1)]}) \vee \neg P(s_k)) \right) \wedge \bigwedge_{0 \leq i \leq k} (\neg \exists s_{[0..i]}. I(s_0) \wedge \text{path}(s_{[0..i]}) \wedge \neg P(s_i)) \quad (8)$$

### 2.1 証明の概要

以下, 証明の流れを述べる. まず,  $i < k$  と  $i \geq k$  に場合分けを行い, それぞれ

$$\forall s_{[0..i]}. \neg(I(s_0) \wedge \text{loopFree}(s_{[0..i]}) \wedge \neg P(s_i)) \quad (9)$$

が成り立つことを証明する.

$i < k$  の場合, 式 (8) の 2 行目より,  $k$  ステップ目まで安全であることが分かっている, 式 (9) が成り立つ.

つぎに,  $i \geq k$  の場合について証明する. 式 (1) より,

$$\begin{aligned} (a) & \forall s_{[0..k]}. \neg(I(s_0) \wedge \text{loopFree}(s_{[0..k]})) \\ (b) & \forall s_{[0..k]}. \neg(\text{loopFree}(s_{[0..k]}) \wedge \text{all}.P(s_{[0..(k-1)]}) \wedge \neg P(s_k)) \end{aligned}$$

のどちらかが成り立つことがわかっている. (a) が成り立つ場合の証明は, 方法 1 のときと同じなのでここでは省略し, (b) が成り立つ場合についての証明のみを行う.

(b) が成り立つ場合の証明には, 完全帰納法 (complete induction) を使用して証明する. 完全帰納法とは, 以下の式が妥当であることをいう.

$$\left( \forall i. \left( \forall m. m < i \rightarrow P(m) \right) \rightarrow P(i) \right) \rightarrow \forall i. P(i) \quad (10)$$

$k = 0$  は,

次に,  $k \geq 1$  に対しての証明を行う. 式 (9) を以下のように変形する.

$$\forall s_{[0..i]}. I(s_0) \wedge \text{loopFree}(s_{[0..i]}) \rightarrow P(s_i) \quad (11)$$

そして,  $P(s_i)$  の部分に式 (10) を適用して整理すると, 仮定は以下ようになる.

$$\forall s_{[0..i]}. I(s_0) \wedge \text{loopFree}(s_{[0..i]}) \quad (12)$$

$$\forall s_{[0..k]}. \text{loopFree}(s_{[0..k]}) \wedge \text{all}.P(s_{[0..(k-1)]}) \rightarrow P(s_k) \quad (13)$$

$$\forall m. \forall s_m. m < i \rightarrow P(s_m) \quad (14)$$

上の 3 つの式から,  $P(s_i)$  を導出できればよい. 上の式を以下のように変形する. ( $i - k \geq 0$ )

$$\forall s_{[0..i]}. I(s_0) \wedge \text{loopFree}(s_{[0..(i-k)]}) \wedge \text{loopFree}(s_{[(i-k)..i]}) \quad (15)$$

$$\forall s_{[(i-k)..i]}. \text{loopFree}(s_{[(i-k)..i]}) \wedge \text{all}.P(s_{[(i-k)..(i-1)]}) \rightarrow P(s_i) \quad (16)$$

$$\forall m. \forall s_{[0..m]}. m < i \rightarrow I(s_0) \wedge \text{loopFree}(s_{[0..(i-m)]}) \rightarrow P(s_m) \quad (17)$$

式 (16) から,  $P(s_i)$  を  $\text{loopFree}(s_{[(i-k)..i]}) \wedge \text{all}.P(s_{[(i-k)..(i-1)]})$  に変形できる. これは,  $\text{loopFree}(s_{[(i-k)..i]}) \wedge P(s_{(i-k)}) \wedge P(s_{(i-k+1)}) \wedge \dots \wedge P(s_{(i-1)})$  であるので, 式 (15), 式 (17) から成り立つ. よって, (b) が成り立つとき場合も, 式 (3) は成り立つ.