【正誤表】

38 ページ, 5 行目.

(誤) $M, (\Gamma \Rightarrow \Delta) \not\vdash \xi$

(正) $M, (\Gamma \Rightarrow \Delta) \vdash \xi$

110 ページ, 下から 11 行目.

(誤) $\{s \in S \mid s \text{ は}$

(正) $\{s \in S \mid (x, s) \text{ は}$

134 ページ, 図 6.1 のプログラム 4 行目.

(誤) $v2 := v3;$

(正) $v2 := v3$

147 ページ, 1 行目, 5 行目, 11 行目, 16 行目.

(誤) $A\{\pi\}B$

(正) $\{A\}\pi\{B\}$

【補遺】 (日付は項目ごとの更新日)

定理 3.4.4 (CTL の有限モデル性) についての注意 (2022 年 2 月)

本書で示した状態数は $O(2^{\text{Lh}(\varphi)})$ であるが, 下記の追加参考文献 [A1] 第 16 章の定理 6.14 では $O(2^{\text{Lh}(\varphi)})$ が示されている.

参考文献の追加 (2022 年 2 月)

[A1] 広瀬健, 野崎昭弘, 小林孝次郎 (監訳). 形式的モデルと意味論 (コンピュータ基礎理論ハンドブック II). 丸善, 1994. (J. van Leeuwen (ed.), *Formal Models and Semantics (Handbook of Theoretical Computer Science, Volume B)*, Elsevier 1990 の翻訳).

[A2] 徳山豪, 小林直樹 (編集), 理論計算機科学事典. 朝倉書店, 2022.

文献 [A1] では CTL とその周辺の論理が第 16 章で, PDL が第 14 章で, ホーア論理が第 15 章で, それぞれ詳しく説明されている.

文献 [A2] の第 8 章ではさまざまな種類のモデル検査が説明されている. ホーア論理も説明されている.

3.6 節における EU の長さについて (2023 年 8 月)

3.6 節では EX は $\neg AX \neg$ の省略形なので, $\text{Sub}^+(pEUq) = \{\neg AX \neg(pEUq), AX \neg(pEUq), \neg(pEUq), pEUq, p, q\}$ である ($\{EX(pEUq), pEUq, p, q\}$ ではない). そのため, $|\text{Sub}^+(\xi)| \leq \text{Lh}(\xi)$ を保証するためには EU の長さは 2 ではなく 4 にする必要がある (つまり $\text{Lh}(EX(\alpha EU \beta)) = \text{Lh}(\alpha) + \text{Lh}(\beta) + 4$).

ただし 3.6 節の最後の段落 (p.78 の一番下) で説明しているように, $\neg$ はここで作るモデルの状態数に影響しないので, 有限モデル性の文面では EU の長さは 2 としてよい.

補題 3.6.6(2) について (2023 年 8 月)

補題 3.6.6(2) には無駄がある. 補題の文面から「かつ $\alpha\text{EU}\beta \in \Gamma$ 」と「かつ $\alpha\text{EU}\beta \in \Gamma'$ 」を削除して次の主張に変えればすっきりする.

$Q = E$ ならば, ある $\Gamma' \Rightarrow \Delta'$ が存在して $\langle H, \alpha\text{EU}\beta, (\Gamma \Rightarrow \Delta) \rangle \xrightarrow{1} \langle H \cup \{\Gamma \Rightarrow \Delta\}, \alpha\text{EU}\beta, (\Gamma' \Rightarrow \Delta') \rangle \in S^+$ となる.

この補題は p.78 の 6 行目で使用されるが, 上記に変えても問題なく使用できる. この補題の証明には p.77 の式 (3.11) を用いるが, それは $\alpha\text{EU}\beta$ を削除して次にすればよい.

$$(\alpha' \wedge \langle \Gamma \Rightarrow \Delta \rangle) \text{EU}\beta, \gamma_1, \gamma_2, \dots, \gamma_n \Rightarrow \quad (3.11)$$

演習問題 3.6.7 の解答 (p.155) は, (i) を削除して (ii) から始めればよい.

ホーア論理の相対完全性の証明 (2023 年 11 月)

定理 6.3.9 の後半, すなわちホーア論理の相対完全性を, 本文と少し異なる方針で少し詳細に証明する.

任意のプログラム π と任意の表明 B に対して次の (I)(II) が成り立つことを, π に関する帰納法で示す.

(I) ある表明 A_0 が存在して次の (1)(2) が成り立つ (この A_0 を「 π と B に対する最弱前条件」と呼ぶ).

(1) $\text{Hoare} \vdash \{A_0\}\pi\{B\}$.

(2) 任意の変数解釈 J, I に対して次が成り立つ.

$J, I \not\models A_0$ ならば, ある J' に対して $J \stackrel{\pi}{\mapsto} J'$ かつ $J', I \models B$.

(II) 任意の表明 A に対して次が成り立つ.

$\models \{A\}\pi\{B\}$ ならば $\text{Hoare} \vdash \{A\}\pi\{B\}$.

この (II) が求める完全性である. ところで (I) $\Rightarrow$ (II) が以下のように証明できる.

(I) と $\models \{A\}\pi\{B\}$ を仮定する. まず $\models A \rightarrow A_0$ であることを示す. もし $\not\models A \rightarrow A_0$ ならば, ある J, I に対して $J, I \models A$ かつ $J, I \not\models A_0$ となっている. しかしこれは, (I) の (2) および仮定 $\models \{A\}\pi\{B\}$ と矛盾する. したがって $\models A \rightarrow A_0$ である. すると (I) の (1) と帰結規則によって $\text{Hoare} \vdash \{A\}\pi\{B\}$ となる.

このことから実際には (I) だけを示せばよいことになる. 以下では π に関する帰納法で, 任意の B に対して (I) が成り立つことを示す.

【 π が $vi := t$ という代入文のとき】 $B(t/vi)$ が求める A_0 になっている.

【 π が複合文のとき】 簡単のために $\pi = \text{begin } \pi_1; \pi_2 \text{ end}$ とする (3 個以上でも同様にできる). このとき帰納法の仮定により π_2 と B に対する最弱前条件 X が存在し, さらに π_1 と X に対する最弱前条件 Y が存在する. すると Y が $\text{begin } \pi_1; \pi_2 \text{ end}$ と B に対する最弱前条件になっていることは簡単に確認できる.

【 π が $\text{if } C \text{ then } \pi_1 \text{ else } \pi_2$ のとき】 帰納法の仮定により π_1 と B に対する最弱前条件 A_1 , および π_2 と B に対する最弱前条件 A_2 が存在する. すなわち次の 4 つが成り立つ. (1-1) $\text{Hoare} \vdash \{A_1\}\pi_1\{B\}$. (1-2) $(\forall J, I)(J, I \models A_1 \Rightarrow (\exists J')(J \stackrel{\pi_1}{\mapsto} J' \text{ かつ } J', I \models B))$. (2-1) $\text{Hoare} \vdash \{A_2\}\pi_2\{B\}$. (2-2)

$(\forall J, I)(J, I \not\models A_2 \Rightarrow (\exists J')(J \stackrel{\pi_2}{\dashv} J' \text{ かつ } J', I \not\models B))$. このとき $A_0 = (C \rightarrow A_1) \wedge (\neg C \rightarrow A_2)$ とすれば, これが **if** C **then** π_1 **else** π_2 と B に対する最弱前条件になっている. A_0 が最弱前条件の条件 (1) を満たすことは, (1-1), (2-1) と $\models (C \wedge A_0) \rightarrow A_1$ および $\models (\neg C \wedge A_0) \rightarrow A_2$ という事実と, 帰結規則と **if** 文規則で言える. A_0 が最弱前条件の条件 (2) を満たすことは, $J, I \not\models A_0$ ならば $J, I \models C \wedge \neg A_1$ または $J, I \models \neg C \wedge \neg A_2$ であることと, (1-2), (2-2) および **if** 文の動作の定義から言える.

【 π が **while** C **do** π' のとき】 簡単のためこのプログラム中には変数がただひとつ (v と書く) しか使われていないとする (変数が複数あっても同様にできる). このとき次が求める A_0 である.

$$(\forall k \geq 0) \forall x_0 \forall x_1 \cdots \forall x_k \left[\left(\begin{array}{l} v = x_0 \wedge \\ (\forall i) \left((0 \leq i < k) \rightarrow C(x_i/v) \wedge "(v = x_i) \stackrel{\pi'}{\dashv} (v = x_{i+1})" \right) \end{array} \right) \rightarrow C(x_k/v) \vee B(x_k/v) \right]$$

ただしここでは以下の事実を使用している.

- 可変個数の表明変数を用いた記法 $\forall x_0 \forall x_1 \cdots \forall x_k$ は表明言語 (算術の一階述言語) を逸脱しているが, 任意の長さの整数列をひとつの自然数で表現する技法 (ゲーデル数化とも言う) を用いれば, 固定した表明変数で同じ意味を表現できる.
- “ $(v = x_i) \stackrel{\pi'}{\dashv} (v = x_{i+1})$ ” は「変数 v の値が x_i のときにプログラム π' を実行開始すると, v の値が x_{i+1} になって停止する」を表す表明であり, 算術の一階述言語で記述することができる.

これらの詳細は参考文献 [29,2,3] やゲーデルの不完全性定理を解説している教科書を見てほしい. A_0 の直観的な意味は次の通りである:「変数 v の現在の値を x_0 としてその後 $x_1, \dots, x_k$ とどのように変化しても, それがプログラム **while** C **do** π' の実行 (の途中までまたは最後まで) に従った変化であるならば, $v = x_k$ のときに条件 C または B が成り立つ」. 以下では A_0 が **while** C **do** π' と B に対する最弱前条件の条件 (1) と (2) を満たすことを示す. (1): まず A_0 の内容をよく考えれば $\models \{C \wedge A_0\} \pi' \{A_0\}$ であることが言える. したがって帰納法の仮定 (II) により **Hoare** $\vdash \{C \wedge A_0\} \pi' \{A_0\}$ となり, **while** 文規則で **Hoare** $\vdash \{A_0\} \text{while } C \text{ do } \pi' \{-C \wedge A_0\}$ を得る. 一方 $\models \neg C \wedge A_0 \rightarrow B$ も言える (A_0 で $k=0, x_0=v$ とすればよい). したがって帰結規則により **Hoare** $\vdash \{A_0\} \text{while } C \text{ do } \pi' \{B\}$ が得られる. (2): $J, I \not\models A_0$ であるとする. これはすなわち「プログラム **while** C **do** π' の実行に従った変数 v の値の遷移 $x_0, x_1, \dots, x_k$ が存在して, $v = x_k$ のときに C も B も成り立たない」ということである. すなわち $J \stackrel{\pi_2}{\dashv} J'$ かつ $J', I \not\models B$ となる J' が存在する.