Errata

Last updated May 6, 2020


This list is a work in progress. Some of the following corrections are tentative and may be revised, and additional corrections will probably be added.

---

**Section 2.2.7.** In the first bullet, *Children* should be *Frontier*.

**Section 2.3.2.** Figure 2.8 should be as shown below. The dashed lines indicate situations where an assertion in an r-state \( \hat{s}_i \) is used to r-satisfy the goal \( \hat{g}_{i+1} \) of a later iteration.

**Section 2.3.3.** In Step 1 of RPG-landmark, replace the phrase “and the only landmark is \( \phi \) itself, so return \( \phi \)” with “and there are no intermediate states, so return \( \emptyset \).”

**Example 2.28.** The last assignment statement should be

\[
\pi \leftarrow \text{move}(r1, d3, d1), \text{load}(r1, c1, d1), \text{move}(r1, d1, d3)).
\]

**Exercise 3.19.** Part (a) should be

What sequence of commands will Refine-lookahead, Refine-lazy-lookahead, and Refine-concurrent-lookahead execute?
**Definition 4.4.** The first sentence of the definition should be

A ground instance of \((\mathcal{T}', \mathcal{C}')\) of \((\mathcal{T}, \mathcal{C})\) is **consistent** if \(\mathcal{T}'\) satisfies \(\mathcal{C}'\) and does not specify two different values for a state variable at the same time.

**Example 4.5.** The second paragraph should be

The assertions \([t_1, t_2] \\text{loc}(r1) = \text{loc1}\) and \([t_2, t_3] \\text{loc}(r1) : (\text{loc1, loc2})\) are nonconflicting: they have no inconsistent instances.

**Example 4.11.** For consistency with Examples 4.12 and 4.17, put \((k', r, c, p')\) and take \((k', r, c, p')\), respectively.

**Example 4.12.** In \(m\)-move1, \text{navigate}(w, w)\) should be \text{navigate}(r, w, w).

**Section 4.2.1, near the end of the section.**

Let \((\mathcal{T}, \mathcal{C}) = (\mathcal{T}_1, \mathcal{C}_1) \cup \ldots \cup (\mathcal{T}_k, \mathcal{C}_k)\). If (i) each timeline \((\mathcal{T}_i, \mathcal{C}_i)\) is secure and (ii) no pair of timelines \((\mathcal{T}_j, \mathcal{C}_j)\) and \((\mathcal{T}_j, \mathcal{C}_j)\) have any unground variables in common, then \((\mathcal{T}, \mathcal{C})\) is secure. The book omits part (ii).

**Exercise 4.8.** The reference to Exercise 4.4 should instead be a reference to Exercise 4.3.

**Section 5.2.3.** The definition of a reachability graph should be this:

\[
\text{Graph}(s, \pi) = (\hat{\gamma}(s, \pi), \{(s', s'') \mid s' \in \hat{\gamma}(s, \pi) \text{ and } s'' \in \gamma(s', \pi(s'))\})
\]

or perhaps more clearly,

\[
\text{Graph}(s, \pi) = (V, E), \text{ where } V = \hat{\gamma}(s, \pi), E = \{(s', s'') \mid s' \in \hat{\gamma}(s, \pi) \text{ and } s'' \in \gamma(s', \pi(s'))\}
\]

**Section 5.2.3.** The last line before Example 5.5 should be

We let \(\hat{\Gamma}(s)\) be the set of all states that are reachable from \(s\), i.e., \(\Gamma(s) = \bigcup\pi \hat{\gamma}(s, \pi)\).

**Exercise 5.7(b).** Remove the words “by drawing the And/Or search tree.”

**Section 6.2.1.** Where definition 6.3 says \(\text{leaves}(s_0, \pi) \cap S_g \neq \emptyset\), it should instead say \(\hat{\gamma}(s_0, \pi) \cap S_g \neq \emptyset\).
Section 6.2.3. The paragraph after Equation 6.3 should be

A closed policy $\pi'$ dominates a close policy $\pi$ if and only if $V^{\pi'}(s) \leq V^\pi(s)$ at every state $s$ where both $\pi$ and $\pi'$ are defined. A closed policy $\pi^*$ is optimal if it dominates all other closed policies. At every state $s$ where $\pi^*$ is defined, it has a minimal expected cost: $V^*(s) = \min_\pi V^\pi(s)$. Under our assumption of probabilistic planning in a domain without dead ends, $\pi^*$ is guaranteed to exist.

Algorithm 6.8. $V$, $\pi$, and Envelope should be global variables. Also, the following line should be added at the beginning of the algorithm:

$$V_0(s_0) \leftarrow V(s_0)$$

Section 6.4.2. RFF, Algorithm 6.16, should be as follows:

<table>
<thead>
<tr>
<th>Algorithm 1: A determinization planning algorithm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RFF(\Sigma, s_0, S_g, \theta)$</td>
</tr>
<tr>
<td>$\pi \leftarrow $ Det-Plan$(\Sigma_d, s_0, S_g)$</td>
</tr>
<tr>
<td>if $\pi = \text{failure}$ then return failure</td>
</tr>
<tr>
<td>while $\exists s \in \tilde{\gamma}(s_0, \pi) \setminus (\text{Dom}(\pi) \cup S_g)$ such that $\Pr(s</td>
</tr>
<tr>
<td>$\pi' \leftarrow $ Det-Plan$(\Sigma_d, s, S_g \cup \text{Targets}(\pi, s))$</td>
</tr>
<tr>
<td>if $\pi = \text{failure}$ then return failure</td>
</tr>
<tr>
<td>$\pi \leftarrow \pi \cup {(s, a) \in \pi' \mid s \notin \text{Dom}(\pi)}$</td>
</tr>
</tbody>
</table>