Errata for Automated Planning and Acting

Dana Nau, University of Maryland

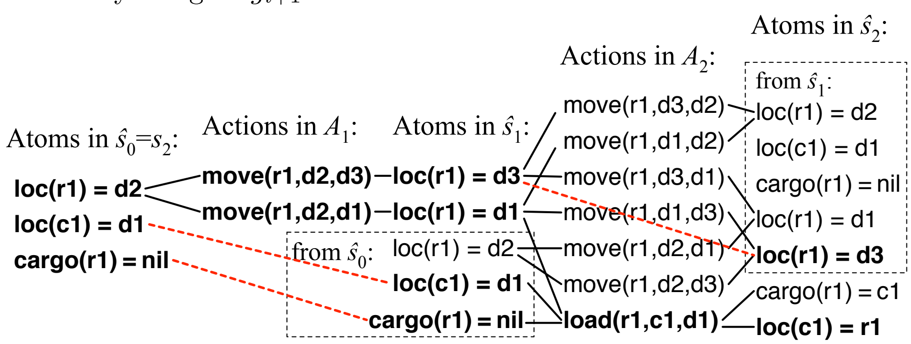
Last updated May 28, 2021

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

Because the printed book and the authors' manuscript have different pagination, the list uses section numbers, etc., instead of page numbers.

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 ϕ itself, so return ϕ" with "and there are no intermediate states, so return $\varnothing$."

Section 2.3.3. Step 5 of RPG-landmark should be performed only on the landmarks found in Step 4 that are not subsumed by other landmarks found in Step 4.

Example 2.28. The last assignment statement should be

$$\pi \leftarrow \mathsf{move}(\mathsf{r1}, \mathsf{d3}, \mathsf{d1}), \mathsf{load}(\mathsf{r1}, \mathsf{c1}, \mathsf{d1}), \mathsf{move}(\mathsf{r1}, \mathsf{d1}, \mathsf{d3})\rangle.$$

Section 2.6.1. In the first bullet of the bulleted list, the phrase "Run-Lookahead is a simple version of the receding-horizon approach ..." should be

If Run-Lookahead is used with a Lookahead procedure that searches to a fixed depth, then this is an example of the receding-horizon approach ...

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

Example 4.11. For consistency with Examples 4.12 and 4.17, $\mathsf{put}(k', r, c, p')$ and $\mathsf{take}(k', r, c, p')$ should be $\mathsf{load}(k', r, c, p')$ and $\mathsf{unload}(k', r, c, p')$, respectively.

Example 4.12. In m-move1, $\mathsf{navigate}(w', w)$ should be $\mathsf{navigate}(r, w', w)$.

Section 4.2.1. In the paragraph before the bullets at the end of the section, "$(\mathcal{T}, \mathcal{C})$ is consistent or secure if each of its timelines is" should be

$(\mathcal{T}, \mathcal{C})$ is consistent if each of its timelines is, and $(\mathcal{T}, \mathcal{C})$ is secure if each of its timelines is secure and its timelines have no variables in common.

Exercise 4.8. "Exercise 4.4" should be "Exercise 4.3".

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

$$\mathit{Graph}(s, \pi) = (\widehat{\gamma}(s, \pi), \{(s', s'') \mid s' \in \widehat{\gamma}(s, \pi) \text{ and } s'' \in \gamma(s', \pi(s'))\})$$

or perhaps more clearly, this:

$\mathit{Graph}(s, \pi) = (V, E)$, where

$$V = \widehat{\gamma}(s, \pi),$$
$$E = \{(s', s'') \mid s' \in \widehat{\gamma}(s, \pi) \text{ and } s'' \in \gamma(s', \pi(s'))\}$$

Section 5.2.3. The last line before Example 5.5 should be

> We let $\widehat{\Gamma}(s)$ be the set of all states that are *reachable* from s, i.e., $\Gamma(s) = \bigcup_\pi \widehat{\gamma}(s, \pi)$.

Exercise 5.3. "a policy π" should be "a solution policy π".

Exercise 5.7(b). "by drawing the And/Or search tree" should be removed.

Section 6.2.1. In Definition 6.3, a *solution* should be defined as a policy π for Σ such that $\widehat{\gamma}(s_0, \pi) \cap S_g \neq \varnothing$. This differs from the definition used in Chapter 5, which required that $leaves(s_0, \pi) \cap S_g \neq \varnothing$.

Section 6.2.3. The paragraph after Equation 6.3 should be

> A closed policy π' *dominates* a close policy π if and only if $V^{\pi'}(s) \leq V^{\pi}(s)$ at every state s where both π and π' are defined. A closed policy π^* is *optimal* if it dominates all other closed policies. At every state s where π^* 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, π^* is guaranteed to exist.

Section 6.3.2. In Algorithm 6.8, AO*, the following lines should be inserted after the first line:

> global V, π, *Envelope*
> $\pi \leftarrow \varnothing$
> $V(s_0) \leftarrow V_0(s_0)$

Also, each occurrence of $\widehat{\gamma}(s_0, \pi)$ should be replaced with $\widehat{\gamma}(s_0, \pi) \setminus S_g$.

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

```
RFF(Σ, s_0, S_g, θ)
    π ← Det-Plan(Σ_d, s_0, S_g)
    if π = failure then return failure
    while ∃s ∈ γ̂(s_0, π) \ (Dom(π) ∪ S_g) such that Pr(s|s_0, π) ≥ θ, do
        π' ← Det-Plan(Σ_d, s, S_g ∪ Targets(π, s))
        if π = failure then return failure
        π ← π ∪ {(s, a) ∈ π' | s ∉ Dom(π)}
```

Exercises 6.12 and 6.14. FF-Replan should be FS-Replan.