zero, it follows from the law of total probability that $\Pr[i^{*(t)} = i' = \alpha] = 1/(n-t-1)$.

For (2.i), a closed form equation for $\Pr[i^{*(t)} = i = \alpha]$ is complex, but we can observe that it depends only on $\Pr[\hat{R}[\alpha] = 1]$ and if so whether α has the greatest $idxs_{\pi_2}$ value of all unvisited 'made ready' indices. The former is uniformly distributed across all indices by π_1 regardless of model and formula structure, and the latter is uniformly distributed by the independent π_1 and π_2. As such, the probability must be the same for all indices. Then, as before by (1) and the law of total probability we have $\Pr[i^{*(t)} = i = \alpha] = 1/(n-t-1)$. □

The oblivious nature of the algorithms then follows quickly from the prior lemma.

Lemma A.5 (Obliviousness of obcheckEU and obcheckAU). *If $P \in \{\text{obcheckEU}, \text{obcheckAU}\}$ and for all $x \in X$ we may write $x = \langle \mathcal{M}, l^{\psi}, r^{\phi} \rangle$ for $\mathcal{M} \in \boldsymbol{\mathcal{M}}$ and $l^{\psi}, r^{\phi} \in \mathbf{BitVec}_n$ for some $n \in \mathbb{N}$ with $\mathcal{M}.n = n$, then P is data-oblivious with respect to X.*

Proof. By an identical argument to Lemma A.3 the number of instructions issued in an execution of P is a deterministic injective function of n. So, for $x_1, x_2 \in X$, $|\mathcal{AP}(\langle P, x_1 \rangle)|$ and $|\mathcal{AP}(\langle P, x_2 \rangle)|$ are identically distributed only if x_1 and x_2 represent models $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively such that $\mathcal{M}_1.n = \mathcal{M}_2.n$.

Let $x_1, x_2 \in X$ be an arbitrary pair of such inputs. The argument reduces to showing that for such x_1 and x_2 their access patterns are identically distributed. Further, let $I^{*(n)}$ be as in Lemma A.4. By loop unrolling, it follows that for a given $I^{*(n)}$ the access pattern of P is fixed. So the argument may be reduced further to showing that $I^{*(n)}$ is identically distributed for x_1 and x_2. But, by Lemma A.4 the choice of $i^{*(t)}$ for all $t \in [1..n]$ is always uniformly distributed over all unvisited indices regardless of model structure and prior choices. It follows that whole sequences are also uniformly — and so identically — distributed for x_1 and x_2. □

The proof of Theorem 3.1 now follows immediately from the conclusions of these lemmas.

Proof. Apply Lemma A.3 and Lemma A.5. □

Our second theorem establishing the functional correctness and efficiency of the oblivious checking algorithm is also restated here.

Theorem 3.2. *For any Kripke structure $\mathcal{M} = (S, I, \delta, L)$ and CTL formula ϕ*

(1) $\text{obcheck}_{\text{CTL}}(\mathcal{M}, \phi) = 1$ *if and only if* $\mathcal{M} \models \phi$*; and*
(2) $\text{obcheck}_{\text{CTL}}(\mathcal{M}, \phi)$ *runs in time* $O(mn^2)$ *where* $|\mathcal{M}| = O(n^2)$ *and* $|\phi| = m$.

We will not provide a detailed proof of this theorem, but rather sketch the proof by arguing (somewhat informally) that certain invariants between the original checkEU and checkAU subroutines and their oblivious variants hold. This implies the functional equivalence of the $\text{obcheck}_{\text{CTL}}$ algorithm to $\text{check}_{\text{CTL}}$, at which point Theorem 2.1 and the additive $O(n)$ cost of permutations completes the argument.

Proof Sketch. As the differences between $\text{obcheck}_{\text{CTL}}$ and $\text{check}_{\text{CTL}}$ lie exclusively within obcheckEU and obcheckAU, we argue these subroutines are functionally equivalent to their non-oblivious variants and retain complexity $O(n^2)$. The complexity follows immediately for both subroutines due to their nested loop structure with both inner and outer iterating over $[\mathcal{M}.n]$.

As for functional equivalence, the core of the argument is that (i) we process all states 'made ready' before any others; that (ii) we process those states in an order consistent with the use of R in the original algorithms; that (iii) while processing 'made ready' states under identical selection we, as compared to the original algorithms, update $\hat{R}$ to be identical to o, $\hat{R} - \hat{K}$ to be an exact representation of inclusion into R, and $\hat{K}$ to be an exact representation of inclusion into K. That all processing done on states not 'made ready' in the oblivious algorithm does not modify $\hat{R}$ then establishes the equivalency. Once we have shown that $\text{obcheck}_{\text{CTL}}$ runs in time $O(mn^2)$ and is functionally equivalent to $\text{check}_{\text{CTL}}$, Theorem 2.1 completes a proof. □