Theorem 1: Let ϕ be a TWTL formula, then for any trace t, Algorithm 1 returns $\bot/\top$ *iff* $[t[i,j] \models \phi] = \bot/\top$.

Proof: We proceed to prove the theorem above. The proof is by a structural induction on the structure of the formula $\phi \in$ TWTL.

<u>Base Case</u>: $\phi \in \{\top, \bot, \mathbf{H}^{\mathbb{D}}p, \mathbf{H}^{\mathbb{D}}\neg p\}$

Case 1: $\phi = \top$. The proof for this case is trivial based on the definition Algorithm 1, since given trace t, `Monitor` ($\top$, t) = $\top$. From the semantics, $t[i,j] \models \top = \top$.

Case 2: $\phi = \bot$. Same as the previous case.

Case 3: $\phi = \mathbf{H}^{\mathbb{D}}p$. For Algorithm 1 to be correct, then $\forall t.[t[i,j] \models \mathbf{H}^{\mathbb{D}}p] \equiv$ `Monitor` $(\mathbf{H}^{\mathbb{D}}p,\ t[i,j])$
Performing a case analysis on the length of trace $|t|$.
Given $|t|$= 1, i.e. $t = \{e_i\}$ while performing further case analysis on the value of $\mathbb{D}$.

- Assuming $\mathbb{D} = 0$ or $\mathbb{D} = 1$: Based on TWTL semantics $t[i,j] \models \mathbf{H}^{\mathbb{D}}p$ iff $p \in e_i, \forall n \in \{i\} \wedge (t[j].\tau - t[i].\tau) \geq \mathbb{D}$. With $(t[j].\tau - t[i].\tau) \geq \mathbb{D}$, the above specification will be satisfied if $p \in e_i$ as stated above. Hence, $[t[i,j] \models \mathbf{H}^{\mathbb{D}}p]$ =$\top$ if $p \in e_i$ else $\bot$. According to Algorithm 1
`Monitor` $(p, t[i,j]) = \top$ after applying Algorithm 2, `Progress` (p, e_i) if $p \in e_i$.
- Assuming $\mathbb{D} > 1$: Based on TWTL semantics, with $|t|$= 1, the case where $\mathbb{D} > 1$ violates the condition $(t[j].\tau - t[i].\tau) \geq \mathbb{D}$. Hence, $t[i,j] \models \mathbf{H}^{\mathbb{D}}p$ = $\bot$. Furthermore, from the Algorithm 1, `Monitor` $(\mathbf{H}^{\mathbb{D}}p, t[i,j]) = \bot$. This is because `Progress` $(\mathbf{H}^{\mathbb{D}}p, e_i)$ returns $\top$ for every $p \in e_i$ and then subsequently $\mathbf{H}^{\mathbb{D}-1}p$. Hence, it is required that $\mathbb{D} \leq |t|$.

Case 4: $\phi = \mathbf{H}^{\mathbb{D}}\neg p$. Same as previous case.

We proceed with the induction on the structure of the formula $\phi \in \{\varphi_1 \wedge \varphi_2,\ \varphi_1 \odot \varphi_2,\ \neg\varphi,\ [\varphi]^{[\tau,\tau']}\}$. The induction hypothesis requires that given a TWTL formula and a trace, the theorem based on the recursion in Algorithm 1 returns $\bot/\top$ demonstrating whether the TWTL formula is satisfied or not.

Case 5: $\phi = \phi_1 \wedge \phi_2$. For Algorithm 1 to be correct, then $\forall t.t[i,j] \models \phi_1 \wedge \phi_2 \equiv$ `Monitor` $(\phi_1 \wedge \phi_2, t[i,j])$.
Based on TWTL semantics, $t[i,j] \models \phi_1 \wedge \phi_2$ = $(t[i,j] \models \phi_1) \wedge (t[i,j] \models \phi_2)$. Likewise, from Algorithm 1, `Monitor` $(\phi_1 \wedge \phi_2, t[i,j])$ = `Monitor` $(\phi_1, t[i,j]))$ $\wedge$ `Monitor` $(\phi_2, t[i,j]))$. Performing the induction on both ϕ_1: $\phi_1 = \top$ and ϕ_2: $\phi_2 = \top$:

- **Subcase 5.1**: $\phi = \top \wedge \phi_2$. From the TWTL semantics on the $\wedge$ operator, $t[i,j] \models (\top \wedge \phi_2) = \phi_2$. Likewise, from Algorithm 1, `Monitor` $(\top \wedge \phi_2, t[i,j])$. By applying Algorithm 3, `Reduce` $(\top \wedge \phi_2)$ = ϕ_2
- **Subcase 5.2**: $\phi = \phi_1 \wedge \top$. Again, based on TWTL semantics, $t[i,j] \models \phi_1 \wedge \top$ = ϕ_1. From Algorithm 1, `Monitor` $(\phi_1 \wedge \top, t[i,j])$. By applying Algorithm 3, `Reduce` $(\phi_1 \wedge \top)$ = ϕ_1.

Case 6: $\phi = \phi_1 \odot \phi_2$. For Algorithm 1 to be correct, then $\forall t.\ [t[i,j] \models \phi_1 \odot \phi_2 \equiv$ `Monitor` $(\phi_1 \odot \phi_2, t[i,j])$. According to the TWTL semantics $\exists k = \arg\min_{i \leq k < j}\{t[i,k] \models \phi_1\} \wedge (t[k+1,j] \models \phi_2)$, where k is time between i and j i.e. $i \leq k < j$. Hence, $t[i,j] \models \phi_1 \odot \phi_2 = (t[i,k] \models \phi_1) \wedge (t[k+1,j] \models \phi_2)$. Likewise, based on Algorithm 1, `Monitor` $(\phi_1 \odot \phi_2, t[i,j])$ = `Monitor` $(\phi_1 t[i,j])$ $\odot$ `Monitor` $(\phi_2, t[i,j])$.
First performing the induction on ϕ_1: $\phi_1 = \top$ and then ϕ_2: $\phi_2 = \top$:

- **Subcase 6.1**: $\phi = \top \odot \phi_2$. From the TWTL semantics, $t[i,k] \models \top \wedge$ $(t[k+1,j] \models \phi_2)$ = ϕ_2. Likewise, from the Algorithm 1, `Monitor` $(\top \odot \phi_2, t[i,j])$. By applying Algorithm 3, `Reduce` $(\top \odot \phi_2) = \phi_2$.
- **Subcase 6.2**: $\phi = \phi_1 \odot \top$. From the TWTL semantics, $t[i,k] \models \phi_1 \wedge (t[k+1,j] \models \top)$ = $t[i,j] \models \phi_1 \odot \top$. Based on semantics $\phi = \phi_1$. Furthermore, from the Algorithm 1, `Monitor` $(\phi_1 \odot \top, t[i,j])$. Applying Algorithm 3, `Reduce` $(\phi_1 \odot \top) = \phi_1$.

Case 7: $\phi = \neg\phi$. For Algorithm 1 to be correct, then $\forall t.t[i,j] \models \neg\phi \equiv$ `Monitor` $(\neg\phi, t[i,j])$. From the TWTL semantics, $t[i,j] \models \neg\phi$ iff $t[i,j] \not\models \phi$. Likewise, Based Algorithm 1, `Monitor` $(\neg\phi, t[i,j]) = \neg$`Progress` (φ, e_i), by applying Algorithm 2.

Case 8: $[\phi]^{[\tau,\tau']}$. For Algorithm 1 to be correct, then $\forall t.[t[i,j] \models [\phi]^{[\tau,\tau']} \equiv$ `Monitor` $([\phi]^{[\tau,\tau']}, t[i,j])$. Based on TWTL semantics, $t[i,j] \models [\phi]^{[\tau,\tau']} = \top$ given that $\exists k \geq i + \tau.\, t[k, i+\tau'] \models \phi \wedge (t[j].\tau - t[i].\tau) \geq \tau'$ and all other cases in relation to ϕ as stated in the semantics is satisfied. According to Algorithm 1, `Monitor` $([\phi]^{[\tau,\tau']}, t[i,j])$ is based on 2 scenarios. Given $\phi = \mathbf{H}^{\mathbb{D}}p$ or $\phi = \mathbf{H}^{\mathbb{D}}\neg p$, the conditions of the base case must hold as well as $\mathbb{D} \leq (\tau' - \tau)$ else $\bot$. In the other scenario, `Monitor` $([\phi]^{[\tau,\tau']}, t[i,j]) = \top$ if Algorithms 2 and 3 are applied on all the respective induction cases as presented in the algorithms 2 and 3 otherwise $\bot$.