1)

i.
$Sat(\phi_1) = \{s_0, s_1, s_2, s_3, s_4\}$
$TS \vDash \phi$

ii.
$Sat(\phi_2) = \{s_4\}$
$\neg TS \vDash \phi$

iii.
$Sat(\phi_1) = \{s_0, s_1, s_2, s_3, s_4\}$
$TS \vDash \phi$

2)

i.
$\phi_1 = \exists \Diamond \forall \Box c$
Subformulas: $\forall \Box c$, c
$Sat(c) = \{s_2, s_3, s_4\}$
$Sat(\forall \Box c) = \{s_2, s_3, s_4\}$
$Sat(\exists \Diamond \forall \Box c) = \{s_0, s_1, s_2, s_3, s_4\}$

Therefore $TS \vDash \phi_1$

ii.
$\phi_2 = \forall (a U \forall \Diamond c)$

Subformulas: $\forall \Diamond c$, c
$Sat(c) = \{s_2, s_3, s_4\}$
$Sat(\forall \Diamond c) = \{s_0, s_1, s_2, s_3, s_4\}$
$Sat(\forall (a U \forall \Diamond c)) = \{s_0, s_1, s_2, s_3, s_4\}$

Or alternatively normalizing the formula for the algorithm: $\forall (a U \neg \exists \Box \neg c)$
Subformulas: c, $\neg c$, $\exists \Box \neg c$, $\neg \exists \Box \neg c$
$Sat(c) = \{s_2, s_3, s_4\}$
$Sat(\neg c) = \{s_0, s_1\}$
$Sat(\exists \Box \neg c) = \{\}$
$Sat(\neg \exists \Box \neg c) = \{s_0, s_1, s_2, s_3, s_4\}$

And we get the same result.
Therefore $TS \vDash \phi_1$

3) (TS ⊨ ∃(ϕUψ)) = (TS' ⊨ ∃◇ψ). Theorem

Since TS has more or equal transitions than TS' and everything else the same, for any formula ϕ we have that

```
  (TS' ⊨ ϕ) ⇒ (TS ⊨ ϕ) ---------------- assump_0
```

I formalize the removal of outgoing transitions

```
   ∀π:TS'·∀i·∀j·j<i ⇒ (π_j⊨¬(¬ϕVψ) V π_i=π_j)  ---- assump_1
```

Proving (TS ⊨ ∃(ϕUψ)) ⇒ (TS' ⊨ ∃◇ψ):

```
  TS ⊨ ∃◇ψ      def of ◇
= TS ⊨ ∃(TUψ)  base law
⇐ TS ⊨ ∃(ϕUψ)  by assump_1, this path is unchanged until ψ hold.
= TS' ⊨ ∃(ϕUψ)
```

Proving (TS' ⊨ ∃◇ψ) ⇒ (TS ⊨ ∃(ϕUψ)):

```
 TS'⊨∃◇ψ
=TS'⊨∃π:Paths(TS)·∃i·π_i⊨ψ
=TS'⊨∃π:Paths(TS)·∃i·π_i⊨ψ ∧ T
=TS'⊨∃π:Paths(TS)·∃i·π_i⊨ψ ∧ (∀j·j<i ⇒ (π_j⊨¬(¬ϕVψ) V π_i=π_j))
⇒TS'⊨∃π:Paths(TS)·∃i·π_i⊨ψ ∧ ((∀j·j<i ⇒ π_j⊨¬(¬ϕVψ)) V (∀j·j<i ⇒ π_i=π_j))
=TS'⊨∃π:Paths(TS)·∃i·π_i⊨ψ ∧ ∀j·j<i ⇒ π_j⊨¬(¬ϕVψ)
   V ∃π:Paths(TS)·∃i·π_i⊨ψ ∧ ∀j·j<i ⇒ π_i=π_j
=TS'⊨∃(ϕUψ) V ∃π:Paths(TS)·∃i·π_i⊨ψ ∧ ∀j·j<i ⇒ π_i=π_j
⇒TS'⊨∃(ϕUψ) V ∃π:Paths(TS)·∃i·π_i⊨ψ ∧ ∀j·j<i ⇒ π_i⊨ψ = π_j⊨ψ
=TS'⊨∃(ϕUψ) V ∃π:Paths(TS)·∃i·π_i⊨ψ ∧ ∀j·j<i ⇒ π_j⊨ψ
=TS'⊨∃(ϕUψ) V ∃π:Paths(TS)·∃i·∀j·j≤i ⇒ π_j⊨ψ
⇒TS'⊨∃(ϕUψ) V ∃(FUψ)
⇒TS'⊨∃(ϕUψ) V ∃(ϕUψ)
=TS'⊨∃(ϕUψ)
⇒TS⊨∃(ϕUψ)
```

4)

i. ∀○∀◇ϕ = ∀◇∀○ϕ. Not a theorem. Example TS:

```
      s_0⊨¬a <---------.
       /|\              |
        |               |
       \|/             \|/
      s_1⊨a  <---->  s_2⊨a
```

ii. $\exists\bigcirc\exists\Diamond\phi = \exists\Diamond\exists\bigcirc\phi$. Theorem.

$\exists\bigcirc\exists\Diamond\phi = \exists\Diamond\exists\bigcirc\phi$ double negation
$= \exists\bigcirc\exists\Diamond\phi\neg\neg\phi = \exists\Diamond\exists\bigcirc\neg\neg\phi$ duality laws
$= \exists\bigcirc\neg\forall\Box\neg\phi = \exists\Diamond\neg\forall\bigcirc\neg\phi$ duality laws
$= \neg\forall\bigcirc\forall\Box\neg\phi = \neg\forall\Box\forall\bigcirc\neg\phi$ eliminate double negation in equality
$= \forall\bigcirc\forall\Box\neg\phi = \forall\Box\forall\bigcirc\neg\phi$ theorem iii. from below
$= T$

iii. $\forall\bigcirc\forall\Box\phi = \forall\Box\forall\bigcirc\phi$. Theorem

Informally: The right hand side says that for all paths starting at this state, it is always true that in any path starting from the next state ϕ is always true. In other words, starting from the next state, ϕ always holds. This is the same as what the left hand side says.

More formally it is easier to first convert to an equivalent LTL formula and prove $\bigcirc\Box\phi = \Box\bigcirc\phi$.

$\bigcirc\Box\phi$
$= \pi\_1 \vDash \Box\phi$
$= \forall i \cdot \pi\_(i+1) \vDash \phi$
$= \forall i \cdot \pi\_i \vDash \bigcirc\phi$
$= \Box\bigcirc\phi$

iv. $\exists\bigcirc\exists\Box\phi = \exists\Box\exists\bigcirc\phi$. Not a theorem. Example TS:

```
s_0⊨¬a <---------.
 /|\             |
  |              |
 \|/            \|/
s_1⊨¬a  ---->  s_2⊨a
```

Bonus

Informally:

CTL formula: $\forall\Diamond\exists\bigcirc\forall\Diamond\neg a$
To get a LTL possibly equivalent formula we remove the quantifiers: $\Diamond\bigcirc\Diamond\neg a$

Consider this transition system:

```
        s_0⊨a   <------->  s_1⊨¬a
       /  /|\              /  /|\
       |   |               |   |
        \_/                  \_/
```

The CTL formula holds. On the path that starts and stays at s_0 the LTL formula does not hold.