CSC410 Assignment 4

Joshua Han 1005109669
Zhi Yiqi 1004892725
Liang Chen 1005735126

Problem 0:

a. Whenever Alice is waiting for Bob's help, Alice is not using the tool t

Solution:

$$\square(\mathrm{aw} \Longrightarrow \neg \mathrm{at})$$

b. If Bob requests help, then either Alice will eventually helps or Bob will be waiting forever

Solution:

$$\square(\mathrm{bw} \Longrightarrow \bigcirc(\Diamond \mathrm{aw} \vee \mathrm{bw}))$$

c. If Alice requests Bob's help, then Bob cannot request Alice's help until he provides help to Alice.

Solution:

$$\square(\mathrm{aw} \Longrightarrow \neg \mathrm{bw} \wedge \bigcirc(\neg \mathrm{bw}\ U\ bh))$$

d. Neither Bob nor Alice can request the other person's help if their help has already been requested until they provide the requested help

Solution:

$$\square(\mathrm{aw} \Longrightarrow \neg \mathrm{bw} \wedge \bigcirc(\neg \mathrm{bw}\ U\ bh)\ ) \wedge \square(\mathrm{bw} \Longrightarrow \neg \mathrm{aw} \wedge \bigcirc(\neg \mathrm{aw}\ U\ ah))$$

Problem 1:

a. Solution:

$$(\mathrm{r}\ U\ (\neg \mathrm{r} \wedge \bigcirc(\neg \mathrm{r}\ U\ \square \mathrm{r})))$$
$$\vee\ (\mathrm{r}\ U\ (\neg \mathrm{r} \wedge \bigcirc(\neg \mathrm{r}\ U\ (r \wedge \bigcirc(\mathrm{r}\ U\ \square \neg \mathrm{r})))))$$
$$\vee(\square \mathrm{r})$$
$$\vee(\square \neg \mathrm{r})$$

We have 6 scenarios to take into consideration:
(starting with red won't be considered as "turning red")
Case 1: starts red, then turn to some other colour than red, and then turns to red and stay being red forever: (r U ($\neg$r $\wedge$ $\bigcirc$($\neg$r U $\square$r)))
Case 2: starts with some other colour than red, and then turns to red and stay being red forever: same as case 1
Case 3: starts red, then turn to some other colour than red, and then turns to red and turns to other colour without backing to red anymore:
(r U ($\neg$r $\wedge$ $\bigcirc$($\neg$r U ($r\wedge\bigcirc$(r U $\square\neg$r)))))
Case 4: starts some other colour than red, and then turns to red and turns to other colour without backing to red anymore: same as case 3
Case 5: starts with red, and being red forever($\square$r)
Case 6: never being red ($\square\neg$r)

b. Solution:

$$(\mathrm{r}\ U\ (\neg\mathrm{r} \wedge \bigcirc(\neg\mathrm{r}\ U\ \square\mathrm{r})))$$
$$\vee\ (\mathrm{r}\ U\ (\neg\mathrm{r} \wedge \bigcirc(\neg\mathrm{r}\ U\ (r\wedge\bigcirc(\mathrm{r}\ U\ \square\neg\mathrm{r})))))$$

c. Solution:

$$\square((\mathrm{r}\wedge\bigcirc\mathrm{w})\vee(\mathrm{w}\wedge\bigcirc\mathrm{r}))$$

d. Solution:

$$\square((\mathrm{r}\wedge\bigcirc(\mathrm{r}\ U\ b)) \vee (\mathrm{b}\wedge\bigcirc\ (b\ U\ w)) \vee (\mathrm{w}\wedge\bigcirc(\mathrm{w}\ U\ r)))$$

e. Solution:

$$\square\mathrm{w} \vee (w\ U \neg(\neg(\mathrm{b} \vee w) U\ w)\ )$$

f. Solution:

$$\square\mathrm{w} \vee (w\ U \neg((\neg(\mathrm{b} \vee w) U\ w) \vee\ (\neg(\mathrm{r} \vee w) U\ w) \vee\ (\neg(\mathrm{g} \vee w) U\ w))$$

Problem 2:
(For those whose equivalence do not hold; we will give examples of paths that satisfy one LTL formulae but not the other one)

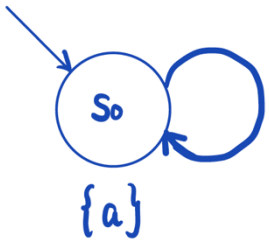
a. The equivalence **DOESN'T** hold

Proof:
Fixing some set of atomic predicates AP
By the definition, we have:
$\varphi\ U\ \neg\ \varphi \equiv true$ iff $\{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \varphi\ U\ \neg\varphi\} = \{\sigma \in (2^{AP})^{\omega} | \sigma \vDash true\}$.
Let $\sigma \in (2^{AP})^{\omega}$.
By extensionality, it must meet the following statement if the LTL equivalence holds:

$$\sigma \vDash \varphi\ U\ \neg\varphi \iff \sigma \vDash true$$

And if only either direction doesn't hold, the equivalence doesn't hold.

We can prove the direction $\sigma \vDash true \longrightarrow \sigma \vDash \varphi\ U\ \neg\varphi$ is not true with an counterexample as below:



AP set = {a}.
Initial state is S_0.
S_0 satisfies LTL property φ where $\varphi = a$
The only path emanating from S_0 is infinite loop over state S_0
This means that it forever satisfies the LTL property φ and never reach the state $\neg\varphi$
By LTL semantics, we have the following always holds: $\sigma \vDash true$
However, $\sigma \nvDash \varphi\ U\ \neg\varphi$.
By the definition of LTL *until* semantics, we have:
$\sigma \vDash \varphi\ U\ \neg\varphi$ iff $\exists\ j\ \geq 0.\ \sigma[j..] \vDash \neg\varphi$ and $\sigma[i..] \vDash \varphi$ for all $0\ \leq i < j$.
In this given counterexample, $\nexists\ such\ j\ \geq 0\ that\ \sigma[j..] \vDash \neg\varphi$ and $\sigma[i..] \vDash \varphi$ for all $0\ \leq i < j$.
Thus, $\sigma \nvDash \varphi\ U\ \neg\varphi$. i.e. $\sigma \vDash true$ cannot imply $\sigma \vDash \varphi\ U\ \neg\varphi$

b. The equivalence **DOES** hold

Proof:
Fixing some set of atomic predicates AP
By the definition, we have:
$(\Diamond\Box\varphi_1) \wedge (\Diamond\Box\varphi_2) \equiv \Diamond(\Box\varphi_1 \wedge \Box\varphi_2)$
Iff $\{\sigma \in (2^{AP})^{\omega} | \sigma \vDash (\Diamond\Box\varphi_1) \wedge (\Diamond\Box\varphi_2)\} = \{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \Diamond(\Box\varphi_1 \wedge \Box\varphi_2)\}$.
Let $\sigma \in (2^{AP})^{\omega}$.
By extensionality, we can prove this LTL equivalence if we can prove:

$$\sigma \vDash (\Diamond\Box\varphi_1) \wedge (\Diamond\Box\varphi_2) \iff \sigma \vDash \Diamond(\Box\varphi_1 \wedge \Box\varphi_2)$$

Forward direction:

1) Assumption:
   $\sigma \vDash (\Diamond\Box\varphi_1) \wedge (\Diamond\Box\varphi_2)$.
2) By 1) and definition of conjunction & eventually $\Diamond$:
   $\exists\, i \geq 0.\, \sigma[i..] \vDash \Box\varphi_1$ and $\exists\, j \geq 0.\, \sigma[j..] \vDash \Box\varphi_2$.
3) By 2) and definition of always $\Box$:
   $\exists\, i \geq 0, \forall\, i' \geq i.\, \sigma[i'..] \vDash \varphi_1$ and $\exists\, j \geq 0, \forall\, j' \geq j.\, \sigma[j'..] \vDash \varphi_2$.
4) Take $k = \max(i, j)$, combine with 3):
   $\exists\, k \geq 0, \forall\, k' \geq k.\, \sigma[k'..] \vDash \varphi_1$ and $\sigma[k'..] \vDash \varphi_2$
5) By 4) and definition of always $\Box$:
   $\exists\, k \geq 0, \sigma[k..] \vDash \Box\varphi_1 \wedge \Box\varphi_2$
6) By 5) and definition of eventually $\Diamond$:
   $\sigma \vDash \Diamond(\Box\varphi_1 \wedge \Box\varphi_2)$

Reverse Direction:

1) Assumption:
   $\sigma \vDash \Diamond(\Box\varphi_1 \wedge \Box\varphi_2)$
2) By 1) and definition of eventually $\Diamond$:
   $\exists\, k \geq 0.\, \sigma[k..] \vDash \Box\varphi_1 \wedge \Box\varphi_2$
3) By 2) and definition of always $\Box$:
   $\exists\, k \geq 0.\, \sigma[k..] \vDash \Box\varphi_1 \wedge \sigma[k..] \vDash \Box\varphi_2$
   i.e. $\exists\, k \geq 0, \forall\, k' \geq k.\, \sigma[k'..] \vDash \varphi_1 \wedge \sigma[k'..] \vDash \varphi_2$.
   i.e. $\exists\, k \geq 0, \forall\, k' \geq k.\, \sigma[k'..] \vDash \varphi_1 \wedge \varphi_2$.
4) Take $i = j = \mathrm{k}$, combine with 3):
   $\exists\, i \geq 0, \forall\, i' \geq i.\, \sigma[i'..] \vDash \varphi_1$ and $\exists\, j \geq 0, \forall\, j' \geq j.\, \sigma[j'..] \vDash \varphi_2$
5) By 4) and definition of always $\Box$:
   $\exists\, i \geq 0, \sigma[i..] \vDash \Box\varphi_1$ and $\exists\, j \geq 0.\, \sigma[j..] \vDash \Box\varphi_2$
6) By 5) and definition of eventually $\Diamond$:
   $\sigma \vDash (\Diamond\Box\varphi_1) \wedge (\Diamond\Box\varphi_2)$

c. The equivalence **DOESN'T** hold

Proof:
Fixing some set of atomic predicates AP
By the definition, we have: $\Box\Diamond\varphi \implies \Box\Diamond\psi \equiv \Box(\varphi \implies \Diamond\psi)$
Iff $\{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \Box\Diamond\varphi \implies \Box\Diamond\psi\} = \{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \Box(\varphi \implies \Diamond\psi)\}$.
Let $\sigma \in (2^{AP})^{\omega}$.
By extensionality, it must meet the following statement if the LTL equivalence holds:

$$\sigma \vDash \Box\Diamond\varphi \implies \Box\Diamond\psi \Leftrightarrow \sigma \vDash \Box(\varphi \implies \Diamond\psi)$$

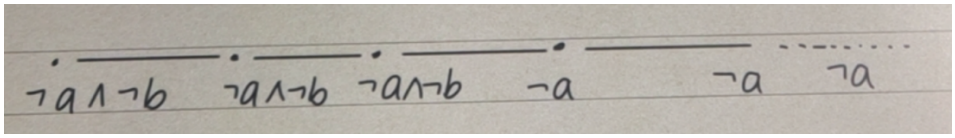
And if only either direction doesn't hold, the equivalence doesn't hold.
Let us first do some rewriting:

$$\sigma \vDash \Box\Diamond\varphi \implies \Box\Diamond\psi \Leftrightarrow$$
$$\sigma \vDash (\neg(\Box\Diamond\varphi)) \vee (\Box\Diamond\psi) \Leftrightarrow$$

$$\sigma \vDash (\Diamond(\neg(\Diamond\varphi)) \vee (\Box\Diamond\psi) \ (\#\ duality\ law) \Leftrightarrow$$
$$\sigma \vDash (\Diamond\Box\neg\varphi) \vee (\Box\Diamond\psi) \ (\#\ duality\ law)$$

$$\sigma \vDash \Box(\varphi \Longrightarrow \Diamond\psi) \Leftrightarrow$$
$$\sigma \vDash \Box(\neg\varphi \vee \Diamond\psi)$$

Now we prove the forward direction $\sigma \vDash (\Diamond\Box\neg\varphi)) \vee (\Box\Diamond\psi) \longrightarrow \sigma \vDash \Box(\neg\varphi \vee \Diamond\psi)$ DOESN'T hold via a counterexample:



In this example, we have:
$\varphi = a$
$\psi = b$
This path $\sigma \vDash (\Diamond\Box\neg\varphi) \vee (\Box\Diamond\psi)$ since from a certain step, always be $\neg a$, i.e. $\sigma \vDash \Diamond\Box\neg a$,i.e. $\sigma \vDash (\Diamond\Box\neg a) \vee (\Box\Diamond b)$
However, it doesn't satisfy $\Box(\neg a \vee \Diamond b)$, since never satisfy $\Diamond b$, and from certain step it always satisfy a

d. The equivalence **DOES** hold

Proof:
WTS $\varphi\ U(\psi \vee \neg\varphi) \equiv \Box\varphi \Longrightarrow \Diamond\psi$
Fixing some set of atomic predicates AP
By the definition, we have:
$$\varphi\ U(\psi \vee \neg\varphi) \equiv \Box\varphi \Longrightarrow \Diamond\psi$$
if and only if
$$\{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \varphi\ U(\psi \vee \neg\varphi)\} = \{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \Box\varphi \Longrightarrow \Diamond\psi\}.$$

Let $\sigma \in (2^{AP})^{\omega}$.
By extensionality, we can prove this LTL equivalence if we can prove
$$\sigma \vDash \varphi\ U(\psi \vee \neg\varphi) \ \Leftrightarrow\ \sigma \vDash \Box\varphi \Longrightarrow \Diamond\psi$$
Rewrite the LTL formula $\Box\varphi \Longrightarrow \Diamond\psi$
we have:
$$\sigma \vDash \varphi\ U(\psi \vee \neg\varphi) \ \Leftrightarrow\ \sigma \vDash (\neg\Box\varphi) \vee \Diamond\psi$$

Forward direction:
WTS
$$\sigma \vDash \varphi\ U(\psi \vee \neg\varphi) \ \longrightarrow\ \sigma \vDash (\neg\Box\varphi) \vee \Diamond\psi$$
1) Assumption:
$\sigma \vDash \varphi\ U(\psi \vee \neg\varphi)$

2) By 1) and definition of until:
$\Leftrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash (\psi \vee \neg\, \varphi\,)\ and\ \sigma[i..] \vDash \varphi\ for\ all\ i < j$

Case 1: $j = 0$
$\Leftrightarrow\ \sigma \vDash (\psi \vee \neg\, \varphi)$
$\Leftrightarrow (\sigma \vDash \psi) \vee (\sigma \vDash \neg\, \varphi)$
$\Leftrightarrow (\sigma[0..] \vDash \psi\,) \vee (\sigma[0..] \vDash \neg\, \varphi)$
$\Leftrightarrow (\exists\ such\ a\ j \geq 0\ (j = 0), \sigma[j..] \vDash \psi\,) \vee (\exists\ such\ a\ j \geq 0\ (j = 0), \sigma[j..] \vDash \neg\, \varphi)$
$\Leftrightarrow (\sigma \vDash \Diamond\psi\,) \vee (\sigma \vDash \Diamond\neg\, \varphi)$
$\Leftrightarrow (\sigma \vDash \Diamond\psi\,) \vee (\sigma \vDash \neg\, \Box\varphi)$
$\Leftrightarrow\ \sigma \vDash (\neg\, \Box\varphi) \vee \Diamond\psi$

Case 2: $j = j' + 1$ for some $j' \epsilon$ N
$\Leftrightarrow \exists j' \geq 0\,, \sigma[j' + 1..] \vDash (\psi \vee \neg\, \varphi\,)\ and\ \sigma[i..] \vDash \varphi\ for\ all\ i < j' + 1$
$\Leftrightarrow \sigma[0..] \vDash \varphi$ and
$\sigma[1][i'..] \vDash \varphi\ for\ all\ i' < j'$ and
$\exists j' \geq 0\,, \sigma[j' + 1..] \vDash \psi \vee \sigma[j' + 1..] \vDash \neg\, \varphi\,)$
$\Longrightarrow \exists\ k \geq 0\ (where\ k = j' + 1), \sigma[k..] \vDash \psi \vee \sigma[k..] \vDash\ \neg\, \varphi$
$\Longrightarrow \sigma \vDash \Diamond\psi \vee \sigma \vDash\ \Diamond\neg\, \varphi$
$\Longrightarrow \sigma \vDash \Diamond\psi \vee \sigma \vDash\ \neg\, \Box\varphi$
$\Leftrightarrow\ \sigma \vDash (\neg\, \Box\varphi) \vee \Diamond\psi$

Backward direction:
WTS
$$\sigma \vDash (\neg\, \Box\varphi) \vee \Diamond\psi \longrightarrow\ \sigma \vDash \varphi\ U(\psi \vee \neg\, \varphi)$$
Assumption:
$\sigma \vDash (\neg\, \Box\varphi) \vee \Diamond\psi$

case 1:
$\sigma \vDash (\neg\, \Box\varphi)$
$\Longrightarrow\ \sigma \vDash (\Diamond\neg\varphi)$
$\Longrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \neg\varphi$ and for all k < j, $\sigma[k..] \vDash\ \varphi$
(there must a smallest k such that $\sigma[k..] \vDash\ \neg\, \varphi$ and for all i < k, $\sigma[k..] \vDash\ \varphi$)
$\Longrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \neg\varphi \vee\psi$ and for all i < k, $\sigma[k..] \vDash\ \varphi$
$\Longrightarrow \sigma \vDash \varphi U(\neg\varphi \vee\psi)$

Case 2:
$\sigma \vDash \Diamond\psi$
$\Longrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \psi$ and for all k < j, $\sigma[k..] \vDash\ \neg\psi$
(let j be the smallest natural number that satisfy the above formula)

- case 2.1:
$\sigma \vDash (\Diamond\neg\varphi)$ (i.e. $\sigma \vDash (\neg\, \Box\varphi)$)
then it is the same scenario as case 1, but let's write it down
$\Longrightarrow \exists\, l \geq 0\,, \sigma[l..] \vDash \neg\varphi$ and for all m < l, $\sigma[m..] \vDash \varphi$

(there must a smallest l such that $\sigma[l..] \vDash \neg\varphi$ and for all i < l, $\sigma[l..] \vDash \varphi$)
$\Longrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \neg\varphi \vee \psi$ and for all i < k, $\sigma[k..] \vDash \varphi$
$\Longrightarrow \sigma \vDash \varphi U(\neg\varphi \vee \psi)$

- case 2.2:
$\sigma \nvDash (\Diamond\neg\varphi)$ (i.e. $\sigma \nvDash (\neg\, \Box\varphi)$)
$\Longrightarrow \sigma \vDash \neg(\Diamond\neg\varphi)$
$\Longrightarrow \sigma \vDash \Box(\neg(\neg\varphi))$
$\Longrightarrow \sigma \vDash \Box(\varphi)$

Since we have:
$\exists\, j \geq 0\,, \sigma[j..] \vDash \psi$ and for all k < j, $\sigma[k..] \vDash \neg\psi$
Combine with $\sigma \vDash \Box(\varphi)$
$\Longrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \psi$
and for all k < j, $\sigma[k..] \vDash \neg\psi$
and $\sigma[k..] \vDash \varphi$
$\Longrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \psi \vee \neg\varphi$
and $\sigma[k..] \vDash \varphi$
$\Longrightarrow \sigma \vDash \varphi U(\neg\varphi \vee \psi)$

e. The equivalence **DOES** hold

Proof:
WTS $\bigcirc\Diamond\varphi \equiv \Diamond \bigcirc \varphi$.
Fixing some set of atomic predicates AP
By the definition, we have:

$$\bigcirc\Diamond\varphi \equiv \Diamond\bigcirc \varphi$$

if and only if

$$\{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \bigcirc\Diamond\varphi\} = \{\sigma \in (2^{AP})^{\omega} | \sigma \vDash \Diamond\bigcirc \varphi\}\,.$$

Let $\sigma \in (2^{AP})^{\omega}$.
By extensionality, we can prove this LTL equivalence if we can prove

$$\sigma \vDash \bigcirc\Diamond\varphi \iff \sigma \vDash \Diamond\bigcirc \varphi$$

Forward direction:

1) Assumption:
   $\sigma \vDash \bigcirc\Diamond\varphi$
2) By 1) and definition of next
   $\Leftrightarrow \sigma[1\,..] \vDash \Diamond\varphi$
3) By 2) and definition of eventually
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[1\,][j..] \vDash \varphi$
4) By 3), rewriting
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[j+1..] \vDash \varphi$

5) By 4), rewriting
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[j][1..] \vDash \varphi$
6) By 5) and definition of next
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[j] \vDash \bigcirc \varphi$
7) By 6) and definition of eventually
   $\Leftrightarrow \sigma \vDash \Diamond \bigcirc \varphi$

Reverse Direction:

1) Assumption:
   $\sigma \vDash \Diamond \bigcirc \varphi$
2) By 1) and definition of eventually
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[j..] \vDash \bigcirc \varphi$
3) By 2) and definition of next
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[j][1..] \vDash \varphi$
4) By 3), rewriting
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[j+1..] \vDash \varphi$
5) By 4), rewriting
   $\Leftrightarrow \exists\, j \geq 0\,, \sigma[1][j..] \vDash \varphi$
6) By 5) and definition of eventually
   $\Leftrightarrow\ \sigma[1..] \vDash \Diamond \varphi$
7) By 6) and definition of next
   $\Leftrightarrow \sigma \vDash \bigcirc \Diamond \varphi$

Thus,

$$\sigma \vDash \bigcirc \Diamond \varphi \ \Leftrightarrow\ \sigma \vDash \Diamond \bigcirc \varphi$$

Thus,

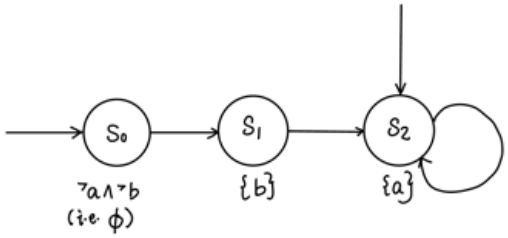
$$\bigcirc \Diamond \varphi \ \equiv \Diamond \bigcirc \varphi$$

Problem 3:

a. $\Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$ is satisfiable

Proof:
WTS $\Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$ is satisfiable
ETS $\exists\, \pi_1 \colon \pi_1 \vDash \Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$ and $\exists\, \pi_2 \colon \pi_2 \nvDash \Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$
$\Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$ is equivalent to $\neg(\Diamond b) \vee (\mathrm{a}\ U\ \mathrm{b})$
$\neg(\Diamond b) \vee (\mathrm{a}\ U\ \mathrm{b})$ equivalent to $\Box(\neg b) \vee (\mathrm{a}\ U\ \mathrm{b})$

We can prove by giving an example TS:



In this TS, we have:
S = {S_0, S_1, S_2}
I = {S_0, S_2}
Label for each state is marked in the image.

First, let's check the path starting from initial state S_2:
$\Longrightarrow$ The only path emanates from S_2, called π_1, is infinite loop over S_2.
$\Longrightarrow$ S_2 has only one atomic propositional logic a in its label.
$\Longrightarrow \pi_1 \vDash \square(\neg b)$
$\Longrightarrow \pi_1 \vDash \square(\neg b) \vee (\mathrm{a}\ U\ \mathrm{b})$
$\Longrightarrow$ the implication "$\Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$" is true (#false always implies true)
$\Longrightarrow \pi_1 \vDash \Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$
Thus, $\exists\ \pi_1 : \pi_1 \vDash \Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$

Next, let's check the path starting from initial state S_0:
$\Longrightarrow$ The only path emanates from S_0 is called π_2
$\Longrightarrow$ S_0 has empty label, i.e. $\neg a \wedge \neg b\}$ and S_1 have label {b}
$\Longrightarrow \pi_2 \nvDash \square(\neg b)$ since $\pi_2[1..] \vDash b$
Also $\pi_2 \nvDash (\mathrm{a}\ U\ \mathrm{b})$ since $\pi_2[0..] \nvDash a$ and $\pi_2[1..] \vDash b$
Thus, $\exists\ \pi_2 : \pi_2 \nvDash \Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$
Thus, $\Diamond b \Longrightarrow (\mathrm{a}\ U\ \mathrm{b})$ is satisfiable

b. $\bigcirc(\mathrm{a} \vee \Diamond \mathrm{a}) \Longrightarrow \Diamond \mathrm{a}$ is valid

Proof:
WTS $\bigcirc(\mathrm{a} \vee \Diamond \mathrm{a}) \Longrightarrow \Diamond \mathrm{a}$ is valid
ETS $\forall\ \pi : \pi \vDash \bigcirc(\mathrm{a} \vee \Diamond \mathrm{a}) \Longrightarrow \Diamond \mathrm{a}$

Let π be an arbitrary (infinite) path.

Case 1: $\pi \nvDash \bigcirc(\mathrm{a} \vee \Diamond \mathrm{a})$
Then $\pi \vDash \bigcirc(\mathrm{a} \vee \Diamond \mathrm{a}) \Longrightarrow \Diamond \mathrm{a}$.

Case 2: $\pi \vDash \bigcirc(\mathrm{a} \vee \Diamond \mathrm{a})$
WTS $\pi \vDash \Diamond \mathrm{a}$

ETS $\exists\, k \geq 0, , \pi[k..] \vDash \mathrm{a}$

(1) $\pi \vDash \bigcirc(\mathrm{a} \vee \Diamond \mathrm{a})$ (# by assumption)

(2) $\Leftrightarrow \pi[1..] \vDash (\mathrm{a} \vee \Diamond \mathrm{a})$. (# by (1) and def. of next)

(3) $\Leftrightarrow (\pi[1..] \vDash \mathrm{a}) \vee (\pi[1..] \vDash \Diamond \mathrm{a})$. (# by (2) and def. of or)

(4) $\Leftrightarrow (\pi[1..] \vDash \mathrm{a}) \vee (\pi[1..] \vDash \Diamond \mathrm{a})$. (# by (3) and def. of or)

(5) $\Leftrightarrow (\pi[1..] \vDash \mathrm{a}) \vee (\exists\, j \geq 0, \pi[1][j..] \vDash \mathrm{a})$. (# by (3) and def. of eventually)

(6) $\Leftrightarrow (\pi[1..] \vDash \mathrm{a}) \vee (\exists\, j \geq 0, \pi[j+1..] \vDash \mathrm{a})$.

We have $\pi[1..] \vDash \mathrm{a}$, or $\pi[j+1..] \vDash \mathrm{a}$ for some $j \geq 0$

In either case, there always exists some $k \geq 1$ such that $\pi[k..] \vDash \mathrm{a}$

Thus, we proved:

$$\exists\, k \geq 0, , \pi[k..] \vDash \mathrm{a}$$

i.e. :

$$\pi \vDash \bigcirc(\mathrm{a} \vee \Diamond \mathrm{a}) \implies \Diamond \mathrm{a}$$

CTL Problems

Problem 5

a. Bob cannot ask for Alice's help unless he has already helped Alice at least once. But doesn't have to ask for Alice's help at all, but if he does, it should be after having helped Alice before.

Solution:

$$\forall \Box\, (\neg \mathrm{bw} \Rightarrow \neg(\exists \neg bh\ \cup bw))$$

Explanation:

$\forall \Box$: since we want it applies to every state in the infinite execution tree

$\exists \neg bh\ \cup bw$: the path which we don't hope to occur

However, b can always ask a for help as long as he helped a at least once.

Thus, for any state we currently into consideration as the initial state, if he doesn't ask Alice for help, we hope no path emanating from current state satisfy $\neg bh\ \cup bw$; however, if Bob asks Alice's help at the current state, we have to use its predecessor states to decide if he can do so, which is why we use " $\neg \mathrm{bw} \Rightarrow$"

(Consider the TS where s0 -> s1, s1->s2, and s2->s0 and s0 is the initial state. s0 has aw true, s1 has bh true and s2 has bw true. This TS should be satisfied, but at s2, $(\neg(\exists \neg bh\ \cup bw))$ is not true. bw trivially satisfies $\exists \neg bh\ \cup bw$. That's why the implies was added)

b. If Alice asks for Bob's help, then it is future possibility (but not necessary) that the tool remains available from this moment that the help was requested until the help is delivered

Solution:

$$\forall\Box(aw \Longrightarrow \exists((\neg(at \vee \text{bt})) \cup bh))$$

Explanation:
$aw \Longrightarrow$: Have to use imply since it uses "if aw" to describe current state S_0
$\exists$: Future possible means there exist such a path, but not necessarily execute through this path
tool remains available at this moment means $\neg(at \vee \text{bt})$
until the help is delivered means $(\neg(at \vee \text{bt})) \cup bh$
the above case should hold at any state when Alice asks Bob for help, and thus we add for all always

c. The light bulb has a possible future in which it is never indefinitely stuck on any one color

Solution:

$$\exists\Diamond((\forall\Diamond\neg\text{w}) \wedge (\forall\Diamond\neg\text{r}) \wedge (\forall\Diamond\neg\text{g}) \wedge (\forall\Diamond\neg\text{b}) \wedge (\forall\Diamond\neg\text{o}))$$

Explanation:
$\exists\Diamond$: A possible future means at least one path has a successor state would satisfy the formular which represents never stuck on a colour
May indefinitely stuck on a colour: $\exists\Box\text{w} \vee \exists\Box\text{r} \vee \exists\Box\text{g} \vee \exists\Box\text{b} \vee \exists\Box\text{o}$
Never indefinitely stuck on a colour:

$$\neg(\exists\Box\text{w} \vee \exists\Box\text{r} \vee \exists\Box\text{g} \vee \exists\Box\text{b} \vee \exists\Box\text{o})$$

Equivalent to say:

$$\neg(\exists\Box\text{w}) \wedge \neg(\exists\Box\text{r}) \wedge \neg(\exists\Box\text{g}) \wedge \neg(\exists\Box\text{b}) \wedge \neg(\exists\Box\text{o})$$

equivalent to say:

$$(\forall\Diamond\neg\text{w}) \wedge (\forall\Diamond\neg\text{r}) \wedge (\forall\Diamond\neg\text{g}) \wedge (\forall\Diamond\neg\text{b}) \wedge (\forall\Diamond\neg\text{o})$$

d. If the light bulb has ever switched from white to blue in the past, then it cannot switch from blue to white in the future

Solution:

$$\forall\Box\,(\neg(\text{w} \wedge \exists\bigcirc(\text{b} \wedge \exists\Diamond(\text{b} \wedge \exists\bigcirc\text{w}))))$$

or saying:

$$\forall\Box\,(\neg\text{w} \vee \neg\exists\bigcirc(\text{b} \wedge \exists\Diamond(\text{b} \wedge \exists\bigcirc\text{w}))))$$

Explanation:
the forbidden scenario: switch from white to blue in the past and switch from blue to white in the future
switch from means two consecutive states, use $\bigcirc$

switch from white to blue: $w \wedge \exists\bigcirc b$
switch from blue to white: $b \wedge \exists\bigcirc w$
if two states which satisfies these two states formula are located in the same possible path, it breaks the rule, a.k.a. satisfy the forbidden scenario
$w \wedge \exists\bigcirc (b \wedge \exists\Diamond(b \wedge \exists\bigcirc w))$
this makes sure current state is w
exists a path that next state is b
this path will eventually reach a state that is b and its consecutive successor state is
we want this never happens, thus we need $\forall\Box (\neg(\dots)$

$$\forall\Box (\neg(w \wedge \exists\bigcirc (b \wedge \exists\Diamond(b \wedge \exists\bigcirc w)))$$
$$\forall\Box (\neg w \vee \neg\exists\bigcirc (b \wedge \exists\Diamond(b \wedge \exists\bigcirc w)))$$
$$\forall\Box (\neg w \vee \neg\exists\bigcirc (b \wedge \exists\Diamond(b \wedge \exists\bigcirc w)))$$

Problem 6:

TS is an infinite transfer system
AP
CTL state formula Ψ and Φ
WTS $TS \models \exists(\Phi \cup \Psi) \Leftrightarrow TS' \models \exists\Diamond\Psi$
TS' is a transfer system eliminating all outgoing transition from states s such that
$s \models \Psi \vee \neg\Phi$

Proof:

Let I be the set of initial states of TS
Let S be the set of states of TS
Let I' be the set of initial states of TS'
Let S' be the set of states of TS'
By definition of TS', $S' \subseteq S$, $I' = I$, since those eliminated states cannot be initial states (those eliminated states are transited from state s where $s \models \Psi \vee \neg\Phi$)

- **Prove forward direction:**
  WTS $TS \models \exists(\Phi \cup \Psi) \longrightarrow TS' \models \exists\Diamond\Psi$
  Assume $TS \models \exists(\Phi \cup \Psi)$.
  By definition:
  $$TS \models \exists(\Phi \cup \Psi) \Leftrightarrow \forall s \in I: s \models \exists(\Phi \cup \Psi)$$
  Let s be an arbitrary initial state such that $s \in I$.
  We have $s \models \exists(\Phi \cup \Psi)$.
  By CTL semantics:
  $$\exists\pi \in Path(s), \pi \models (\Phi \cup \Psi)$$
  By definition of until:
  $$\exists\pi \in Path(s), \exists\, j \geq 0, \pi[j] \models \Psi \text{ and } \forall\, 0 \leq i < j, \pi[i] \models \Phi$$
  Let π be such a path emanating from s.
  Since $\forall\, 0 \leq i < j, \pi[i] \models \Phi$ and $\pi[j] \models \Psi$, we have
  $$\forall\, 0 \leq i < j,\ \pi[i] \not\models \Psi \vee \neg\Phi, \text{ and } \pi[j] \models \Psi \vee \neg\Phi$$

(if there is another k < j that $\pi[k] \vDash \Psi$, we would let the k be the new j – that is why we say $\forall\ 0 \leq i < j,\ \pi[i] \vDash \neg\Psi$, combining with $\forall\ 0 \leq i < j,\ \pi[i] \vDash \Phi$, we know $\forall\ 0 \leq i < j,\ \pi[i] \vDash \neg\Psi \wedge \Phi$, i.e. $\forall\ 0 \leq i < j,\ \pi[i] \nvDash \Psi \vee \neg\Phi$)
Thus, we know by definition of TS':
All states that path π transited to from $\pi[0]$ to $\pi[j]$ won't be eliminated
Thus, $\pi[0..j]$(including j) is still partial of a valid path in TS', let's call this path π'
The path π' emanates from s
Also, s is in I' too, we can write it as s' although s and s' refer to the same initial state.
Thus, for TS':
$\exists\ \pi' \in Path(s'), \exists\ j \geq 0, \forall\ 0 \leq i < j, \pi'[i..] \vDash \Phi$ and $\pi'[j..] \vDash \Psi$
By definition of eventually, we have: $\exists\ \pi' \in Path(s')$ such that $\pi' \vDash \Diamond\Psi$
Thus, we have:

$$s' \vDash \exists\Diamond\Psi$$

Since it holds for arbitrary s'∈ I, combine with I = I', we know it holds for arbitrary s'∈ I'
Thus, we have:

$$TS' \vDash \exists\Diamond\Psi$$

- **Prove reverse direction:**
  WTS $TS' \vDash \exists\Diamond\Psi \longrightarrow TS \vDash \exists(\Phi\ \cup\ \Psi)$
  Assume $TS' \vDash \exists\Diamond\Psi$.
  By definition:

$$TS' \vDash \exists\Diamond\Psi \Leftrightarrow \forall\ s' \in I' : s' \vDash \exists\Diamond\Psi$$

  Let s' be an arbitrary initial state such that $s' \in I'$.
  We have $s' \vDash \exists\Diamond\Psi$.
  By CTL semantics:

$$\exists\pi' \in Path(s'), \pi' \vDash \Diamond\Psi$$

  By definition of eventually:

$$\exists\pi' \in Path(s'), \exists\ j \geq 0, \pi'[j..] \vDash \Psi$$

  Let π' be such a path emanating from s'.

  Now consider two cases.

  Case 1: $j = 0$

$$\pi'[0] \vDash \Psi$$
$$s' \vDash \Psi$$

  By definition of until, exists such a j = 0 that $s' \vDash \Psi$
  Thus, for all paths emanating from s', they will have $\pi'[0] \vDash \Psi$
  This implies that for all paths emanating from s', they will have $\pi' \vDash \Phi \cup \Psi$
  Thus, we have:

$$s' \vDash \forall\ \Phi \cup \Psi$$

  s' is arbitrary initial state that $s' \in I'$, also I= I', we have:
  For arbitrary initial state that $s \in I$:

$$s \vDash \forall\ \Phi \cup \Psi$$

(∀ quantifier here only applies for scenario that j=0, if it doesn't hold for all paths starting from s when j > 0, we will change quantifier)

Case 2: $j > 0$

$$\exists\, j > 0, \pi'[j..] \vDash \Psi \; and \; \forall\, 0 \leq \mathrm{i} < \mathrm{j}, \pi'[i..] \vDash \Phi$$

Thus, we have:

$$\pi'[0] \vDash \Phi$$
$$\exists\, j \geq 0, \pi'[1][j..] \vDash \Psi$$
$$\forall\, 0 \leq \mathrm{i} < \mathrm{j}, \pi'[1][i..] \vDash \Phi$$

Rewrite,

$$\exists\, j \geq 0, \pi'[j+1] \vDash \Psi$$
$$\forall\, 0 \leq \mathrm{i} < \mathrm{j}+1, \pi'[i..] \vDash \Phi \wedge \neg\Psi$$

(when there are multiple such j's that $\pi'[j+1] \vDash \Psi$, we always can choose the smallest such j and thus $\pi'[i..] \vDash \neg\Psi$ for all i smaller than this chosen j)

By the definition of until, we have:

$$\exists \pi' \in Path(s'), \exists\, j \geq 0, \pi'[j+1] \vDash \Psi \text{ and } \forall\, 0 \leq \mathrm{i} < \mathrm{j}+1, \pi'[i..] \vDash \Phi$$

i.e.

$$\exists \pi' \in Path(s'), \exists\, j \geq 0, \pi'[j] \vDash \Psi \text{ and } \forall\, 0 \leq \mathrm{i} < \mathrm{j}, \pi'[i..] \vDash \Phi$$

i.e.

$$\exists \pi' \in Path(s')\,, \qquad \pi' \vDash \Phi \cup \Psi$$

By definition of TS', we know, there must be a path π that has exactly the same states as $\pi'[0..j+1]$

(if any of the states between $\pi'[0]$ to $\pi'[j]$, saying k satisfying the elimination policy $\Psi \vee \neg\Phi$, then all states transit out from that state k will disappear including successors after k in this path π' ,and thus $\pi'[j+1]$ won't exist then.)

s' is an arbitrary initial state that $s' \in I'$, also I= I', we have:

For arbitrary initial state that $s \in I$,

$$\exists \pi \in Path(s), \pi \vDash \Phi \cup \Psi$$

Thus, we have:

$$s \vDash \exists \Phi \cup \Psi$$

Combine two cases, we have:

$$s \vDash \exists \Phi \cup \Psi$$

Also, s is an arbitrary initial state in TS, we have:

$$TS \vDash \exists \Phi \cup \Psi$$

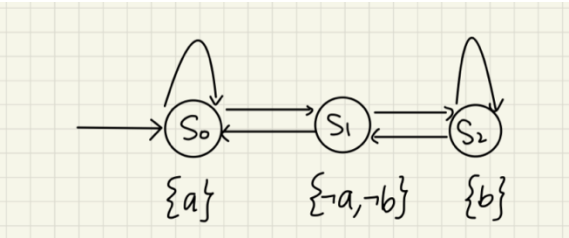
Thus, we conclude:

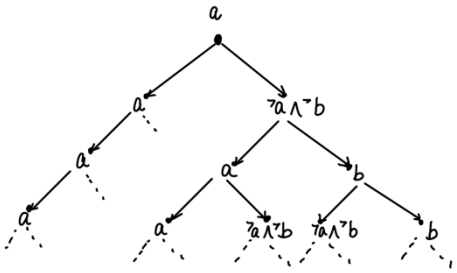
$$TS \vDash \exists(\Phi \cup \Psi) \iff TS' \vDash \exists \Diamond \Psi$$

Problem 7:

(a) Solution: $\forall\Box\exists\Diamond(\varphi_1 \wedge \varphi_2) \equiv \forall\Box\exists\Diamond\varphi_1 \wedge \forall\Box\exists\Diamond\varphi_2$ DOESN'T hold

Counter example:
This example picture below shows an TS and corresponding infinite execution tree
Let $\varphi_1 = a$ and $\varphi_2 = b$

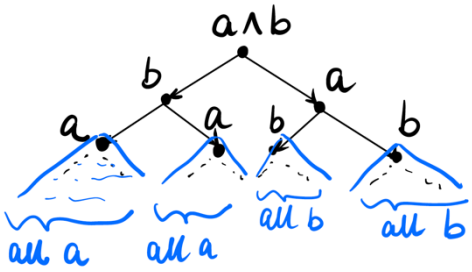




At any arbitrary state in the infinite tree, we can tell that there is always a path emanating from this state that can eventually have state {a};
Also, there exists a path emanating from this state that can be in state b eventually
However, there is no such a state in the execution tree have a ∧ b;
Thus, the tree satisfies $\forall\Box\exists\Diamond\varphi_1 \wedge \forall\Box\exists\Diamond\varphi_2$; but it doesn't satisfy $\forall\Box\exists\Diamond(\varphi_1 \wedge \varphi_2)$

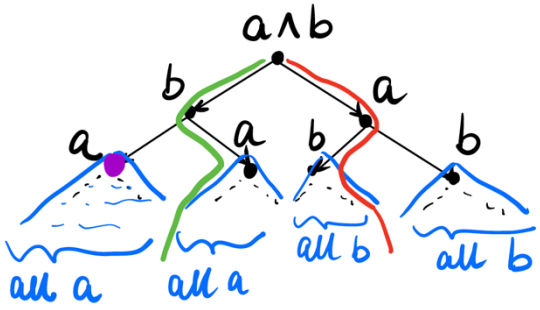
(b) Solution: $\exists\Box\forall\Diamond(\varphi_1 \wedge \varphi_2) \equiv \exists\Box\forall\Diamond\varphi_1 \wedge \exists\Box\forall\Diamond\varphi_2$ DOESN'T hold

Counter example:
Let $\varphi_1 = a$ and $\varphi_2 = b$
This example picture below shows an infinite execution tree



At level 0: state is a ∧ b
At level 1: left child of root has b, right child of root has a

Starting from level 2, left subtrees all a, right subtrees all b



If we choose the green path, we can find that for any states on this path, all paths emanating from this state eventually reaches a (and for states on red path, some of them don't always hold $\forall\Diamond\varphi_1$), i.e. we have $\exists\Box\forall\Diamond\varphi_1$
Similarly, when choosing the red path, we can find that for any states on this path, all paths emanating from this state eventually reaches b (and for states on green path, some of them don't always hold $\forall\Diamond\varphi_2$), i.e. we have $\exists\Box\forall\Diamond\varphi_2$
Thus, the tree satisfies $\exists\Box\forall\Diamond\varphi_1 \wedge \exists\Box\forall\Diamond\varphi_2$
However, if we take a state at level 3 (the purple colour state for example, but also applies to other states at level 3), there is no such a path emanating from it can eventually reach a ∧ b (but any path emanating from the root in our example execution tree is infinite long and always need to pass through a state at level 3)
Thus, does not satisfy $\exists\Box\forall\Diamond(\varphi_1 \wedge \varphi_2)$

(c) Solution: $\exists\Box\varphi \equiv \varphi \wedge \exists\bigcirc\exists\Box\varphi$ holds
Forward direction: $\exists\Box\varphi \longrightarrow \varphi \wedge \exists\bigcirc\exists\Box\varphi$
assume TS $\vDash \exists\Box\varphi$
$\Leftrightarrow \forall s \in I: s \vDash \exists\Box\varphi$
$\Leftrightarrow \forall s \in I, \exists \pi \in Path(s): \pi \vDash \Box\varphi$ (#By def of ∃)
$\Leftrightarrow \forall s \in I, \exists \pi \in Path(s), \forall j \geq 0, \pi[j..] \vDash \varphi$ (#By def of ∀)
$\Leftrightarrow \forall s \in I, \exists \pi \in Path(s), \pi[0] \vDash \varphi \wedge \forall j \geq 0, \pi[1+j..] \vDash \varphi$ (#By basic logic)
$\Leftrightarrow \forall s \in I,\ s \vDash \varphi \wedge \exists \pi \in Path(s), \forall j \geq 0,\ \pi[1+j..] \vDash \varphi$
(# By π emanates from s and thus $\pi[0]$ is s)
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge \exists \pi \in Path(s), \forall j \geq 0, \pi[0+1+j..] \vDash \varphi$
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge \exists \pi \in Path(s), \forall j \geq 0, \pi[0][1][j..] \vDash \varphi$ (#By algebra of words)
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge \exists \pi \in Path(s), \pi[0][1][0..] \vDash \Box\varphi$ (#By def of □)
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge \exists \pi \in Path(s), \pi[0][1] \vDash \exists\Box\varphi$ (#By def of state)
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge \exists \pi \in Path(s), \pi[0] \vDash \bigcirc\exists\Box\varphi$ (#By def of next)
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge s \vDash \exists\bigcirc\exists\Box\varphi$ (#By def of ∃)
$\Leftrightarrow \forall s \in I, s \vDash \varphi \wedge \exists\bigcirc\exists\Box\varphi$ (#By def of ∧)
$\Leftrightarrow$ TS $\vDash \varphi \wedge \exists\bigcirc\exists\Box\varphi$

Reverse direction: (the above proving process is iff, which actually proved the reverse direction already, but to make the proof process formal, we write down the reverse

direction proof steps)
WTS: $\varphi \wedge \exists \bigcirc \exists \Box \varphi \rightarrow \exists \Box \varphi$
Assume: TS $\models \varphi \wedge \exists \bigcirc \exists \Box \varphi$
$\Leftrightarrow \forall s \in I, s \models \varphi \wedge \exists \bigcirc \exists \Box \varphi$
$\Leftrightarrow \forall s \in I, s \models \varphi \wedge s \models \exists \bigcirc \exists \Box \varphi$ (#By def of $\wedge$)
$\Leftrightarrow \forall s \in I, \forall \pi \in Path(s): \pi[0] \models \varphi \wedge \exists \pi' \in Path(s), \pi' \models \bigcirc \exists \Box \varphi$
$\Rightarrow \forall s \in I, \exists \pi \in Path(s): \pi[0] \models \varphi \wedge \pi \models \bigcirc \exists \Box \varphi$
$\Leftrightarrow \forall s \in I, \exists \pi \in Path(s): \pi[0] \models \varphi \wedge \pi[1] \models \exists \Box \varphi$
$\Leftrightarrow \forall s \in I, \exists \pi \in Path(s), \exists \pi' \in Path(\pi[1]): \pi[0] \models \varphi \wedge \pi' \models \Box \varphi$
$\Rightarrow \forall s \in I, \exists \pi'' \in Path(s): \pi''[0] \models \varphi \wedge \pi''[1..] \models \Box \varphi$ (if we take such a path π'' that $\pi''[0] = \pi[0]$ and $\pi''[1..] = \pi'[0..]$)
$\Rightarrow \forall s \in I, \exists \pi'' \in Path(s): \pi'' \models \Box \varphi$
$\Leftrightarrow \forall s \in I, s \models \exists \Box \varphi$
$\Leftrightarrow$ TS $\models \exists \Box \varphi$