

Formal Verification and Specification
Lab Session (TP) 05

1. CTL formulas:

- Always, if a train passes, the gate is closed
- Despite the current situation, I will eventually do a in the future.
- It exists currently a choice, just after it I could eventually choose b
- For all executions, every event p will be eventually followed by q .
- For all executions, every event p will be eventually followed by q in an exact future.
- Every q implies that p would be true, where p must not be preceded by r
- I do not enter only if the crossing is free, and I will close the door once I'm not in the crossing.
- If two processes are waiting infinitely often, they would be served the one after the other.

2. CTL equivalence:

Express the equivalences for the following CTL formulae:

- $AX\varphi$
- $AF\varphi$.
- $EF\varphi$
- $AG\varphi$
- $AG\phi1U\phi2$
- $EG\varphi$

Answers

1. CTL formulas:

- Always, if a train passes, the gate is closed
- Despite the current situation, I will eventually do a in the future.
- It exists currently a choice, just after it I could eventually choose b
- For all executions, every event p will be eventually followed by q .
- For all executions, every event p will be eventually followed by q in an exact future.
- Every q implies that p would be true, where p must not be preceded by r
- I do not enter only if the crossing is free, and I will close the door once I'm not in the crossing.
- If two processes are waiting infinitely often, they would be served the one after the other.

Answer: • $AGpasse \implies gate_closed$

- $AXEFa$
- $EXEFb$
- $AG(p \implies AFq)$
- $AG(p \implies AXAFq)$
- $AG(q \implies A(\neg rUp))$
- $(\neg EF(enter \wedge \neg free)) \wedge (AG(enter \implies AF(\neg stay \wedge closed)))$
- $A[GFatt1 \wedge GFatt2 \implies \neg F((s1 \wedge (\neg s2)Us1 \wedge s2) \vee ((\neg s1)Us2))]$

2. CTL equivalence:

Express the equivalences for the following CTL formulae:

- $AX\varphi$
- $AF\varphi$.
- $EF\varphi$
- $AG\varphi$
- $AG\phi1U\phi2$
- $EG\varphi$

Answer: • $AX\varphi \equiv \neg EX\neg\varphi$

- $AF\varphi \equiv \neg EG\neg\varphi$.
- $EF\varphi \equiv EtrueU\varphi$
- $AG\varphi \equiv \neg EF\neg\varphi$
- $AG\phi1U\phi2 \equiv \neg((E(\phi1 \wedge \neg\phi2)U(\neg\phi1 \wedge \neg\phi2)) \vee (EG\neg\phi2))$
- $EG\varphi \equiv \neg AF\neg\varphi$