Homework 1

Exercise 2 Prove or give a counterexample: $((\forall y : P(y)) \vee (\forall z : Q(z))) \to (\forall x : P(x) \vee Q(x))$.

The implication holds as it is proven below:

1. Let us assume $((\forall y : P(y)) \vee (\forall z : Q(z)))$. Then we apply reasoning based on cases; i.e. we consider first the case where $((\forall y : P(y))$ holds but not $(\forall z : Q(z))$, and then the case where $(\forall z : Q(z))$ holds but not $((\forall y : P(y))$.
2. first case: From $((\forall y : P(y))$, we get $P(x)$ for any arbitrary x.
3. $P(x) \vee Q(x)$ holds for any predicate $Q(x)$ on any arbitrary x.
4. $(\forall x : P(x) \vee Q(x))$ follows by introducing $\forall$.
5. second case: From $((\forall z : Q(z))$, we get $Q(x)$ for any arbitrary x.
6. $P(x) \vee Q(x)$ holds for any predicate $P(x)$ on any arbitrary x.
7. $(\forall x : P(x) \vee Q(x))$ follows by introducing $\forall$.

Exercise 3 Prove or give a counterexample: $(\forall x : P(x) \vee Q(x)) \to ((\forall y : P(y)) \vee (\forall z : Q(z)))$.

The implication does not hold. A counterexample can be the interpretation $I = \{D = \{1,\ 2\},\ P(1),\ Q(2)\}$.

Homework 2

Exercise 1 Prove the following equivalence: $\forall v \forall v' : H(v) \wedge R(v, v') \to H(v')$ if and only if forall s and for all s' it holds if $s \models H(v)$ and $(s, s') \models R(v, v')$ then $s' \models H(v)$.

The proof depends mainly of the semantics of the satisfaction statements such as $s \models H(v)$ which states that some s is an instance of $H(v)$ or $H(s)$ holds. The equivalence is shown by proving that one follows from the other in both directions.

$\Longrightarrow$ proof

1. Let us start by assuming $\forall v \forall v' : H(v) \wedge R(v, v') \to H(v')$.
2. To show that $s \models H(v)$ and $(s, s') \models R(v, v') \implies s' \models H(v)$ for all s and s', we also assume $s \models H(v)$ and $(s, s') \models R(v, v')$ for arbitrary s and s'.
3. By the semantics of $\models$, $H(s)$ and $R(s, s')$ follows from $s \models H(v)$ and $(s, s') \models R(v, v')$.
4. $H(s')$ follows from (1), which is equivalent to $s' \models H(v)$.

$\Leftarrow$ proof

1. Let us start by assuming that forall s and for all s', if it holds $s \models H(v)$ and $(s, s') \models R(v, v')$ then it also holds $s' \models H(v)$.
2. Let us also assume $H(v) \wedge R(v, v')$ for arbitrary v and v'.
3. From the conjunction, we have $H(v)$ which is equivalent to $v \models H(s)$ and $R(v, v') which is equivalent to (v, v') \models R(s, s')$.
4. We get $v' \models H(s)$ from (1). But this is equivalent to $H(v')$.

Exercise 2 Prove that if a program is safe then there exists H(v) (in an expressive assertion language) such that

$$
\begin{array}{ll}
\forall v : \varphi_{init}(v) \to H(v) & : C_1 \\
\forall v \forall v' : H(v) \wedge R(v, v') \to H(v') & : C_2 \\
\forall v : H(v) \wedge \varphi_{err}(v) \to \bot & : C_3
\end{array}
$$

Let $H(v) \equiv \varphi_{reach}(v)$ where $\varphi_{reach}(v)$ is the set of reachable states of the program. By the definition of reachability, $H(v)$ satisfies C_1 and C_2. But since it is given that the program is safe, $\varphi_{reach}(v)$, and hence $H(v)$ satisfy C_3.

Homework 3

Exercise 2 Prove that upon termination of BRA, if a node n is reachable from the initial node n_0 via the set of edges E, i.e., $(n_0, n) \in E^*$, then $n \in C$.

We prove the theorem by induction on the length of the path k from n_0 to n i.e. k = $|(n_0, n)|$.

Our induction hypothesis Hyp(k) is: Each node n that was reached from n_0 through a path of length k is in C, i.e., $n \in C$.

Base case: For k = 0, we have $n = n_0$ and $(n_0, n_0) \in E^*$. Since $n_0 \in C$, Hyp(0) holds.

Step: We assume that for k the induction hypothesis $Hyp(k)$ holds, i.e., if a node n is reachable from the initial node n_0 in k steps, i.e., $(n_0, n) \in E^*$ such that $k = |(n_0, n)|$, then $n \in C$.

We prove $Hyp(k + 1)$, which amounts to proving that n reached during the $(k + 1)_{th}$ step from n_0 by following the if branch is in C. The case when the $(k + 1)_{th}$ iteration goes through the else branch does not change the reachable states. For any n_{k+1} that is reachable from n_0 in $k + 1$ steps, i.e. $|(n_0, n_{k+1})| = k + 1$, there exists n_k reachable from n_0 such that $|(n_0, n_k)| = k$ and $(n_k, n_{k+1}) \in E$. By the induction hypothesis, $n_k \in C$. $n_{k+1} \in C$ follows immediately.

Exercise 3 Extend the BRA algorithm to detect the existence of cycles in a given finite graph. The extended algorithm returns true iff there exists $n \in N$ such that $(n_0, n) \in E^*$ and $(n, n) \in E^+$.

This is the algorithm to detect existence of cycles in a given finite graph. It makes use of a modified version of BRA.

```
algorthm detect_cycle
input
         N : set of nodes
         n0 : start node, where n0 \in N
         E : set of edges, where E \subseteq N \times N

begin
        (C, D) := BRA (N, N0, Edges)
        foreach s in C
(Cs,Ds) := BRA (N, s, Edges)
if  (s \in Ds) then
    return true
 return false
end
```

This is the moified version of BRA that is used in defining the cycle detection algorithm above.

```
algorithm BRA
       input
         N : set of nodes
         n0 : start node, where n0 \in N
         E : set of edges, where E \subseteq N \times N
      var
         C : nodes reached so far
         done : Boolean flag
         D : auxiliary set of nodes
       begin
         C := {n0}
         done := false
         while \neg done do
           D := { d \in N | \exists c \in C: (c, d) \in E }
           if \neg (D \subseteq C) then
             C := C \cup D
           else
             done := true
         od
 return (C, D)
end
```

Homework 4

Exercise 2

1. $\forall \phi : \phi \models \alpha(\phi)$
   (a) we have $(\phi \models p_1 \wedge \phi \models p_2) \rightarrow \phi \models p_1 \wedge p_2$
   (b) $\alpha(\phi) \equiv \wedge\{p_i \in P : \phi \models p_i\}$
   (c) Let the set $\{p_i \in P : \phi \models p_i\}$ has n elements such that $\alpha(\phi) = p_1 \wedge p_2 \wedge \cdots \wedge p_n$. By definition of $\alpha(\phi)$ we have $\phi \models p_1, \phi \models p_2, \ldots,$ and $\phi \models p_n$
   (d) By (a), we have $\phi \models p_1 \wedge p_2 \wedge \cdots \wedge p_n$, and $p_1 \wedge p_2 \wedge \cdots \wedge p_n \equiv \alpha(\phi)$ follows from (b). Therefore, $\phi \models \alpha(\phi)$.
2. α is monotonic, i.e. $\forall \phi \forall \psi : (\phi \models \psi) \rightarrow (\alpha(\phi) \models \alpha(\psi))$
   (a) $\forall \sigma_1 \sigma_2 : \sigma_1 \wedge \sigma_2 \models \sigma_1$
   (b) Let's assume $\phi \models \psi$
   (c) By the definition of α, we have $\alpha(\phi) \equiv \wedge\{p_i \in P : \phi \models p_i\}$ and $\alpha(\psi) \equiv \wedge\{p_i \in P : \psi \models p_i\}$.
   (d) An important observation here is that any predicate in the set $\{p_i \in P : \psi \models p_i\}$ is also in $\{q_i \in P : \phi \models q_i\}$ since from $\phi \models \psi$ by (b) and each $\psi \models p_i$ we always have $\phi \models p_i$. i.e $\{p_i \in P : \phi \models p_i\}$ is the superset of $\{p_i \in P : \psi \models p_i\}$.
   (e) Therefore, we have $\alpha(\phi) \equiv \alpha(\psi) \wedge q_1 \wedge \cdots \wedge q_m$ where $q_1, \ldots, q_m$ are predicates that are in $\{p_i \in P : \phi \models p_i\}$ but not in $\{p_i \in P : \psi \models p_i\}$.
   (f) From $\alpha(\phi) \models \alpha(\psi) \equiv \alpha(\psi) \wedge q_1 \wedge \cdots \wedge q_m \models \alpha(\psi)$, and by (a) we can see that $\alpha(\psi) \wedge q_1 \wedge \cdots \wedge q_m \models \alpha(\psi)$ holds. i.e. $\alpha(\phi) \models \alpha(\psi)$.
   (g) We now introduce the implication since the satisfaction was proven based on the assumption in (b). i.e $(\phi \models \psi) \rightarrow (\alpha(\phi) \models \alpha(\psi))$.
   (h) And finally, universal quanitification is done on both variables: $\forall \phi \forall \psi : (\phi \models \psi) \rightarrow (\alpha(\phi) \models \alpha(\psi))$
3. $\forall \phi \forall R_1 \forall R_2 : \mathit{post}(\phi, R_1 \vee R_2) \iff \forall \phi \forall R_1 \forall R_2 : (\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2))$
   (a) assume $\forall \phi \forall R_1 \forall R_2 : \mathit{post}(\phi, R_1 \vee R_2)$
   (b) $\mathit{post}(\phi, R_1 \vee R_2)$ by applying $\forall$ elimination
   (c) $\exists V" : \phi(V)[V"/V] \wedge (R_1(V, V')[V"/V][V/V'] \vee R_2(V, V')[V"/V][V/V'])$ by reducing *post* into its definition
   (d) $\exists V" : (\phi(V") \wedge (R_1(V", V)) \vee \exists V" : (\phi(V") \wedge R_2(V", V))$ by distributing the conjunction and the existential quantifier over the disjunction
   (e) $\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2)$ by rewriting back in terms of *post*
   (f) $\forall \phi \forall R_1 \forall R_2 : (\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2))$ by applying $\forall$ introduction
   (g) $\forall \phi \forall R_1 \forall R_2 : \mathit{post}(\phi, R_1 \vee R_2) \rightarrow \forall \phi \forall R_1 \forall R_2 : (\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2))$ by implication introduction
   (h) assume $\forall \phi \forall R_1 \forall R_2 : (\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2))$
   (i) $(\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2))$ by applying $\forall$ elimination
   (j) $(\exists V" : \phi(V)[V"/V] \wedge R_1(V, V')[V"/V][V/V']) \vee (\exists V" : \phi(V)[V"/V] \wedge R_2(V, V'))[V"/V][V/V'])$ by reducing *post* into its definition
   (k) $\exists V" : \phi(V") \wedge (R_1(V", V) \vee R_2(V", V))$ by collecting terms over the existential quantifier and ϕ
   (l) $\mathit{post}(\phi, R_1 \vee R_2)$ by rewriting back in terms of *post*
   (m) $\forall \phi \forall R_1 \forall R_2 : \mathit{post}(\phi, R_1 \vee R_2)$ by applying $\forall$ introduction
   (n) $\forall \phi \forall R_1 \forall R_2 : (\mathit{post}(\phi, R_1) \vee \mathit{post}(\phi, R_2)) \rightarrow \forall \phi \forall R_1 \forall R_2 : \mathit{post}(\phi, R_1 \vee R_2)$ by implication introduction
4. [1] *post* is monotonic, i.e. $\forall \phi \forall \psi : (\phi \models \psi) \rightarrow (\mathit{post}(\phi, \rho) \models \mathit{post}(\psi, \rho))$
   (a) $\forall \phi \forall \psi \forall \sigma : \phi \wedge \sigma \models \psi \wedge \sigma$ conjuction on both sides will not affect satisfaction of the entailment.
   (b) assume $\phi \models \psi$
   (c) we have $\phi(V) \models \psi(V)$ by explicitly putting the variable V with the formulas
   (d) $\phi(V) \wedge \rho(V, V') \models \psi(V) \wedge \rho(V, V')$ by (a)
   (e) $\exists V" : (\phi(V)[V"/V] \wedge \rho(V", V)[V"/V][V/V']) \models \exists V" : \psi(V")[V"/V] \wedge \rho(V", V)[V"/V][V/V']$ variable substitution and $\exists$ introduction
   (f) $\mathit{post}(\phi, \rho) \models \mathit{post}(\psi, \rho))$ by the definition of post
   (g) $(\phi \models \psi) \rightarrow (\mathit{post}(\phi, \rho) \models \mathit{post}(\psi, \rho))$ by implication introduction from (b) and (f).
   (h) $\forall \phi \forall \psi : (\phi \models \psi) \rightarrow (\mathit{post}(\phi, \rho) \models \mathit{post}(\psi, \rho))$ by introducing $\forall$
5. [2] $\mathit{post}^{\#}$ is monotonic, i.e. $\forall \phi \forall \psi : (\phi \models \psi) \rightarrow (\mathit{post}^{\#}(\phi, \rho) \models \mathit{post}^{\#}(\psi, \rho))$

[1] Not in the homework
[2] Not in the homework but important for proving exercise 3

The proof follows directly from the monotonicity proofs for *post* and α above.

(a) assume $\phi \models \psi$

(b) $post(\phi, \rho) \models post(\psi, \rho))$ follows since *post* is monotonic

(c) $\alpha(post(\phi, \rho)) \models \alpha(post(\psi, \rho))$ follows since α is monotonic

(d) $post^{\#}(\phi, \rho) \models post^{\#}(\psi, \rho))$ by definition of $post^{\#}$

(e) $(\phi \models \psi) \rightarrow (post^{\#}(\phi, \rho) \models post^{\#}(\psi, \rho))$ by implication introduction from (a) and (d).

(f) $\forall\phi\forall\psi : (\phi \models \psi) \rightarrow (post^{\#}(\phi, \rho) \models post^{\#}(\psi, \rho))$ by introducing $\forall$

Exercise 3

Let R be a transition relation over V and V'. We define $post(\phi) = \exists V" : \phi[V"/V] \wedge R[V"/V][V/V']$.
Prove that $\bigvee_{i \geq 0} post^{\#^i}(\phi_{init}) \models \bigvee_{j \geq 0} post^{\#^j}(\alpha(\phi_{init}))$.
Here we use the fact that proving $\forall A_i \exists B_j : A_i \models B_j$ is enough to prove that $A_0 \vee A_1 \vee \ldots \models B_0 \vee B_1 \vee \ldots$

1. $\phi \models \alpha(\phi)$ (as proven in the previous exercise)
2. $post^{\#}(\phi) \models post^{\#}(\alpha(\phi))$ follows since $post^{\#}$ is monotonic
3. $post^{\#^i}(\phi) \models post^{\#^i}(\alpha(\phi))$ follows from the fact that applying $post^{\#}$ i times on both side keeps the entailment since $post^{\#}$ is monotonic.
4. Therefore, $\forall i \exists j : post^{\#^i}(\phi) \models post^{\#^j}(\alpha(\phi))$ holds when $j = i$.

Homework 5

Question 1

We define a constraint C_1 over ϕ_1, ϕ_2, ϕ_3 as follows:

$$
\begin{aligned}
C_1(\phi_1,\phi_2,\phi_3) \equiv \\
\varphi_{init} &\models \phi_1 \quad \wedge \\
post(\phi_1,\rho_1) &\models \phi_2 \quad \wedge \\
post(\phi_2,\rho_2) &\models \phi_3 \quad \wedge \\
\phi_3 \wedge \varphi_{err} &\models false
\end{aligned}
$$

Prove that for each ϕ_1, ϕ_2, and ϕ_3, if $C_1(\phi_1,\phi_2,\phi_3)$ then $\varphi_{init} \wedge (\rho_1 \circ \rho_2) \wedge \varphi_{err}[V'/V] \models false$.

Let us assume $C_1(\phi_1,\phi_2,\phi_3)$ holds.

1. $\varphi_{init} \models \phi_1$ from the first conjunct.
2. $post(\varphi_{init},\rho_1) \models post(\phi_1,\rho_1)$ since $post$ is monotonic.
3. From the second conjuct and by (2), we have $post(\varphi_{init},\rho_1) \models \phi_2$.
4. $post(post(\varphi_{init},\rho_1),\rho_2) \models post(\phi_2,\rho_2)$ since $post$ is monotonic.
5. From the third conjunct and by (4), we have $post(post(\varphi_{init},\rho_1),\rho_2) \models \phi_3$.
6. By adding the same conjunct φ_{err} on both sides, we have $post(post(\varphi_{init},\rho_1),\rho_2) \wedge \varphi_{err} \models \phi_3 \wedge \varphi_{err}$.
7. From the forth conjunct and by (6), we have $post(post(\varphi_{init},\rho_1),\rho_2) \wedge \varphi_{err} \models false$.
8. But since $post(post(\phi,\rho_1),\rho_2)$ is equivalent to $post(\phi,\rho_1 \circ \rho_2)$ (check out Question 3 below for the proof), $post(\varphi_{init},\rho_1 \circ \rho_2) \wedge \varphi_{err} \models false$.

Question 2

We define a constraint C_2 over ϕ_1, ϕ_2 as follows:

$$
\begin{aligned}
C_2(\phi_1,\phi_2) \equiv \\
\varphi_{init} &\models \phi_1 \quad \wedge \\
post(\phi_1,\rho_1) &\models \phi_2 \quad \wedge \\
post(\phi_2,\rho_2) \wedge \varphi_{err} &\models false
\end{aligned}
$$

Prove that C_1 is satisfiable if and only if C_2 is satisfiable, where C_1 is defined in the previous question.

$C_1(\phi_1,\phi_2,\phi_3) \Longrightarrow C_2(\phi_1,\phi_2)$

1. Let us assume $C_1(\phi_1,\phi_2,\phi_3)$ holds. The first and second conjuncts of C_2 follows directly from the first and second conjuncts of C_1.
2. We take the third conjunct from C_1 and add the same conjunct φ_{err} on both sides of the entailment to get $post(\phi_2,\rho_2) \wedge \varphi_{err} \models \phi_3 \wedge \varphi_{err}$.
3. From the forth conjunct of C_1 and (2), we get $post(\phi_2,\rho_2) \wedge \varphi_{err} \models false$ which is the thrid conjunct of C_2. This completes the proof for $C_1(\phi_1,\phi_2,\phi_3) \Longrightarrow C_2(\phi_1,\phi_2)$.

$C_2(\phi_1,\phi_2) \Longrightarrow C_1(\phi_1,\phi_2,\phi_3)$

1. Let us assume $C_2(\phi_1,\phi_2)$ holds. The first and second conjuncts of C_1 follows directly from the first and second conjuncts of C_2.
2. Let $\phi_3 = post(\phi_2,\rho_2)$. Since $\forall \psi : \psi \models \psi$, we get $post(\phi_2,\rho_2) \models \phi_3$. This proves the third conjunct of C_1.
3. In addition, from $\phi_3 = post(\phi_2,\rho_2)$ and the third conjunct of C_2, we get $\phi_3 \wedge \varphi_{err} \models false$ which is the forth conjunct of C_1. This completes the proof for $C_2(\phi_1,\phi_2) \Longrightarrow C_1(\phi_1,\phi_2,\phi_3)$.

Question 3 Prove that $post(post(\phi, \rho_1), \rho_2)$ is equivalent to $post(\phi, \rho_1 \circ \rho_2)$.

1. From $post(post(\phi, \rho_1), \rho_2)$, we reach $\exists v'' : (\exists v' : \phi(v') \wedge \rho_1(v', v'')) \wedge \rho_2(v'', v)$
2. For arbitrary constants a and b, this gives us $(\phi(a) \wedge \rho_1(a, b)) \wedge \rho_2(b, v)$ which is equivalent to $\phi(a) \wedge (\rho_1(a, b) \wedge \rho_2(b, v))$ since *conjunction* is associative.
3. Introducting an existential quantifier on the right conjunct gives $\phi(a) \wedge (\exists v'' : \rho_1(a, v'') \wedge \rho_2(v'', v))$. But the right conjunct defines $\rho_1 \circ \rho_2(a, v)$, and hence we have $\phi(a) \wedge (\rho_1 \circ \rho_2(a, v))$.
4. Introducing an existential quantifier again gives $\exists v' : \phi(v') \wedge (\rho_1 \circ \rho_2(v', v))$ which is equivalent to $post(\phi, \rho_1 \circ \rho_2)$.

Homework 6

Question 1 Compute interpolants for:

a) $x \leq 5, y \leq x$ and $y \geq 10 = y \leq a$ where $a \in \{5, 6, 7, 8, 9\}$

b) $x \leq 5$ and $x \geq y, y \geq 10 = x \leq a$ where $a \in \{5, 6, 7, 8, 9\}$

c) $x + 1 \leq z$ and $x \geq y, y \geq z = x \leq z$

Question 2 Prove that our interpolation algorithm respects the condition imposed on the variables that occur in the computed interpolant.

Our interpolation algorithm is given below:

$$\begin{aligned}
&\exists i\ \exists i_0 \\
&\quad \exists \lambda\ \exists \mu : \\
&\qquad \lambda \geq 0 \wedge \mu \geq 0 \wedge \\
&\qquad \begin{pmatrix} \lambda & \mu \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = 0 \wedge \qquad (conjunct1) \\
&\qquad \begin{pmatrix} \lambda & \mu \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \leq -1 \wedge \qquad (conjunct2) \\
&\qquad i = \lambda A \wedge \qquad (conjunct3) \\
&\qquad i_0 = \lambda a\ .
\end{aligned} \tag{1}$$

Let ϕ_A and ϕ_B be the formulals that we want to compute an interpolant for, and whose coefficient matrices are give as A and B after rewriting all expressions ϕ_A and ϕ_B as inequalities over $\leq$. Let m be the number of inequalities in ϕ_A, n be the number of inequalities in ϕ_B, and k be the number of variables that appear either in ϕ_A or ϕ_B (or both). A has m rows each for each inequality in ϕ_A. B has n rows each for each inequality in ϕ_B. Both A and B are of k columns where each column contains array of values for each variable (one value per inequality).

What we need to show here is that if j^{th} column of A is 0 (the j^{th} variable is not in A) or j^{th} column of B is 0 (the j^{th} variable is not in B), then the j^{th} column of an interpolant i should be 0. i.e. an interpolant is defined only over the variables in both A and B.

From the assumptions on the number of inequalities in ϕ_A and ϕ_B, the total number of variables in ϕ_A and ϕ_B, and the conjuncts in our interpolatione algorithm abov, we conclude:

- λ is a row-matrix of m columns.
- μ is a row-matrix of n columns.
-

$$\begin{pmatrix} \lambda & \mu \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \lambda A + \mu B = 0 \tag{2}$$

If the j^{th} column of A is 0, then the j^{th} column of any interpolant i should be 0 by the conjunct (3) of equation (1).

If the j^{th} column of B is 0, then the j^{th} column of μB is 0. From equation (2), then it follows that the j^{th} column of λA is 0 since $\lambda A + \mu B = 0$. Like the first case, j^{th} column of any interpolant i should be 0 by the conjunct (3) of equation (1).

Question 4 Represent inference rules describing summarization as entailments.
The inference rules describing summarization are:

1. $$\frac{(g, l_{main}) \models init(V_G, V_{main})}{((g, l_{main}), (g, l_{main})) \in summ_{main}}$$

2. $$\frac{((g, l_p), (g', l'_p)) \in summ_p \qquad ((g', l'_p), (g'', l''_p)) \models step_p(V_G, V_p, V'_G, V'_p)}{((g, l_p), (g'', l''_p)) \in summ_p}$$

3.

$$\frac{((g, l_p), (g', l'_p)) \in summ_p \qquad ((g', l'_p, l_q)) \models call_{p,q}(V_G, V_p, V_q)}{((g', l_q), (g', l_q)) \in summ_q}$$

4.

$$\frac{\begin{array}{c} ((g, l_p), (g', l'_p)) \in summ_p \qquad\qquad ((g', l'_p, l_q)) \models call_{p,q}(V_G, V_p, V_q) \\ ((g', l_q), (g'', l'_q)) \in summ_q \qquad (g'', l'_q, q''') \models ret_q(V_G, V_q, V'_G) \qquad (l'_p, l''_p) \models loc_p(V_p, V'_p) \end{array}}{((g, l_p), (g''', l''_p)) \in summ_p}$$

Representation of these inference rules as entailments is given below:

1.

$$init(V_G, V_{main}) \models summ_{main}((V_G, V_{main}), (V_G, V_{main}))$$

2.

$$summ_p((V_G, V_p), (V'_G, V'_p)) \wedge step_p((V'_G, V'_p), (V''_G, V''_p)) \models summ_p((V_G, V_p), (V''_G, V''_p))$$

3.

$$summ_p((V_G, V_p), (V'_G, V'_p)) \wedge call_{p,q}((V'_G, V'_p, V_q)) \models summ_q((V'_G, V_q), (V'_G, V_q))$$

4.

$$\begin{array}{c} summ_p((V_G, V_p), (V'_G, V'_p)) \wedge call_{p,q}((V'_G, V'_p, V_q)) \wedge summ_q((V'_G, V_q), (V''_G, V'_q)) \wedge \\ ret_q(V''_G, V'_q, V'''_G) \wedge loc_p(V'_p, V''_p) \models summ_p((V_G, V_p), (V'''_G, V''_p)) \end{array}$$

Homework 7

Question 1 For the program producer-consumer with semaphores, prove the stability of the inductive invariant given in class under transitions $PW \to PM$ and $CI \to CL$.

The given inductive invariant is:

$$\Big(0 \le empty \wedge 0 \le full \wedge 0 \le in \wedge 0 \le out \wedge \forall k < out : B[k] = f(g(A[k]))\Big) \wedge \qquad C_1$$

$$\begin{pmatrix} ((pc_1 \le PM \vee pc_1 \ge PL) \wedge (pc_2 \le CA \vee pc_2 \ge CL) \wedge (full = in - out))\vee \\ ((pc_1 = PI) \wedge (CR \le pc_2 \le CI) \wedge (full = in - out))\vee \\ ((pc_1 \le PM \vee pc_1 \ge PL) \wedge (CR \le pc_2 \le CI) \wedge (full = in - out - 1))\vee \\ ((pc_1 = PI) \wedge (pc_2 \le CA \vee pc_2 \ge CL) \wedge (full = in - out + 1)) \end{pmatrix} \wedge \qquad C_2$$

$$\begin{pmatrix} ((pc_1 \le PA \vee pc_1 \ge PI) \wedge (pc_2 \le CA \vee pc_2 \ge CW) \wedge (empty + full = N))\vee \\ ((pc_1 \le PA \vee pc_1 \ge PI) \wedge (CR \le pc_2 \le CM) \wedge (empty + full = N - 1))\vee \\ ((PW \le pc_1 \le PM) \wedge (pc_2 \le CA \vee pc_2 \ge CW) \wedge (empty + full = N - 1))\vee \\ ((PW \le pc_1 \le PM) \wedge (CR \le pc_2 \le CM) \wedge (empty + full = N - 2))\vee \end{pmatrix} \wedge \qquad C_3$$

$$\begin{pmatrix} (pc_1 = PS \wedge in = 0)\vee \\ (pc_1 = PR \wedge in < M)\vee \\ (PA \le pc_1 \le PW \wedge in < M \wedge x = g(A[in]))\vee \\ (PM \le pc_1 \le PI \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]))\vee \\ PL \le pc_1 \le PF \end{pmatrix} \wedge \qquad C_4$$

$$\begin{pmatrix} (pc_2 = CS \wedge out = 0 \wedge \forall k \in [out, in) : buf[k \ mod \ N] = g(A[k]))\vee \\ (CA \le pc_2 \le CR \wedge out < M \wedge \forall k \in [out, in) : buf[k \ mod \ N] = g(A[k]))\vee \\ (pc_2 = CM \wedge out < M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge y = g(A[out]))\vee \\ (pc_2 = CW \wedge out < M \wedge (\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge y = g(A[out]))\vee \\ (pc_2 = CI \wedge out < M \wedge (\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])))\vee \\ (pc_2 = CL \wedge out \le M \wedge \forall k \in [out, in) : buf[k \ mod \ N] = g(A[k]))\vee \\ (pc_2 = CF \wedge out = M) \end{pmatrix} \qquad C_5$$

To make application of *post* straight forward later when checking stability of the invariant, assume that the invariant which was given as:

$$\big(D_{11}\big) \wedge \qquad C_1$$

$$\big(D_{21} \vee D_{22} \vee D_{23} \vee D_{24}\big) \wedge \qquad C_2$$

$$\big(D_{31} \vee D_{32} \vee D_{33} \vee D_{34}\big) \wedge \qquad C_3$$

$$\big(D_{41} \vee D_{42} \vee D_{43} \vee D_{44} \vee D_{45}\big) \wedge \qquad C_4$$

$$\big(D_{51} \vee D_{52} \vee D_{53} \vee D_{54} \vee D_{55} \vee D_{56} \vee D_{57}\big) \qquad C_5$$

is rewritten as:

$$\big(D_{11} \wedge D_{21} \wedge D_{31} \wedge D_{41} \wedge D_{51}\big) \vee$$

$$\big(D_{11} \wedge D_{21} \wedge D_{31} \wedge D_{41} \wedge D_{52}\big) \vee$$

$$\dots$$

$$\big(D_{11} \wedge D_{22} \wedge D_{31} \wedge D_{41} \wedge D_{51}\big) \vee$$

$$\big(D_{11} \wedge D_{22} \wedge D_{31} \wedge D_{41} \wedge D_{52}\big) \vee$$

$$\dots$$

$$\big(D_{11} \wedge D_{24} \wedge D_{34} \wedge D_{45} \wedge D_{57}\big)$$

by distributing conjunctions over disjunctions, where each D_{ij} represents the j^{th} disjunct of the i^{th} conjunct in the given invariant. Since there is one disjunct in C_1, four disjuncts in C_2, four disjuncts in C_3, five disjuncts in C_4 and seven disjuncts in C_5, the re-written invariant will be a big disjunction of $1 \times 4 \times 4 \times 5 \times 7 = 560$ disjuncts (where each disjunct in itself is a conjunction).

We check stability by applying *post* on each of the disjuncts and checking if the resulting state is already in the invariant or not. But we know that *post* will be applicable on the states that satisfy the condition set by the transition. Therefore, during checking stability of the invariant with respect to a given transition, we must first filter those states *post* will be applicable.

1. $PW \to PM$
The transition $PW \to PM$ can be represented as $\rho(v, v') = (pc_1 = PW \wedge pc_1' = PM \wedge buf' = buf[in \ mod \ N \mapsto x] \wedge x' = x \wedge y' = y \wedge full' = full \wedge empty' = empty \wedge pc_2' = pc_2 \wedge in' = in \wedge out' = out)$.
post will not be applicable on disjuncts which contain D_{22}, D_{24}, D_{31}, D_{32}, D_{41}, D_{42}, D_{44} and D_{45} since $pc_1 \neq PW$ in such disjuncts reducing the candidates to from 560 to 28. In addition, some disjuncts are simply unsatisfiability together which further reduces the number of candidates. For example, although D_{21} and D_{34} satisfy $pc_1 = PW$, there is no value for pc_2 that satisfies $D_{21} \wedge D_{34}$ which makes *post* inapplicable over disjuncts that contain both D_{21} and D_{34}. This leaves us with only 8 disjuncts that *post* is applicable to $\rho(v, v')$, which are given below:

$$\begin{gathered}
(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{51}) \vee \\
(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{52}) \vee \\
(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{56}) \vee \\
(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{57}) \vee \\
(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{43} \wedge D_{54}) \vee \\
(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{43} \wedge D_{55}) \vee \\
(D_{11} \wedge D_{23} \wedge D_{34} \wedge D_{43} \wedge D_{52}) \vee \\
(D_{11} \wedge D_{23} \wedge D_{34} \wedge D_{43} \wedge D_{53})
\end{gathered}$$

Let us now apply *post* on each of these disjuncts and check if the resulting state is already in the invariant or not.

(a) $post(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{51}, \rho)$

$$\begin{aligned}
&= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge \\
&(pc_2 \leq CA \vee pc_2 \geq CL) \wedge (full = in - out) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge \\
&PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CS \wedge out = 0 \wedge \forall k \in [out, in) : buf[k \ mod \ N] = g(A[k]), \rho(v, v')) \\
&= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CS \wedge \\
&(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out = 0 \wedge \\
&(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])), \rho(v, v')) \\
&= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CS \wedge \\
&(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out = 0 \wedge \\
&(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge buf[in \ mod \ N] = x) \\
&\models D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{44} \wedge D_{51}
\end{aligned}$$

(b) $post(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{52}, \rho)$

$$\begin{aligned}
&= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge \\
&(pc_2 \leq CA \vee pc_2 \geq CL) \wedge (full = in - out) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge \\
&PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge CA \leq pc_2 \leq CR \wedge out < M \wedge \\
&(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])), \rho(v, v')) \\
&= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CA \wedge \\
&(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge \\
&(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])), \rho(v, v')) \\
&= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CA \wedge \\
&(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge \\
&(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge buf[in \ mod \ N] = x) \\
&\models D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{44} \wedge D_{52}
\end{aligned}$$

(c) $post(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{56}, \rho)$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(pc_2 \leq CA \vee pc_2 \geq CL) \wedge (full = in - out) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CL \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])), \rho(v, v'))$
$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])), \rho(v, v'))$
$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge buf[in \ mod \ N] = x)$
$\models D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{44} \wedge D_{56}$

(d) $post(D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{57}, \rho)$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(pc_2 \leq CA \vee pc_2 \geq CL) \wedge (full = in - out) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CF \wedge out = M, \rho(v, v'))$
$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CF \wedge$
$(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out = M), \rho(v, v'))$
$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CF \wedge$
$(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out = M \wedge buf[in \ mod \ N] = x)$
$\models D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{44} \wedge D_{57}$

To avoid over-writing of some buffer content during the transition, $in - out < N$ should be satisfied. This is justified for the above four cases since from $full = in - out$ and $full + empty = N - 1$, we get $in - out = N - 1 - empty$.

(e) $post(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{43} \wedge D_{54}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CW \wedge out < M \wedge (\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge$
$y = g(A[out]), \rho(v, v'))$
$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CW \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge y = g(A[out]), \rho(v, v'))$
$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CW \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge y = g(A[out]) \wedge buf[in \ mod \ N] = x)$
$\models D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{44} \wedge D_{54}$

(f) $post(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{43} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CI \wedge out < M \wedge (\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge$
$B[out] = f(g(A[out])), \rho(v, v'))$
$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])), \rho(v, v'))$
$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge (\forall k \in (out, in) :$
$buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])) \wedge buf[in \ mod \ N] = x)$
$\models D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{44} \wedge D_{55}$

To avoid over-writing of some buffer content during the transition, $in - (out + 1) < N$ should be satisfied. This is justified for the above two cases since from $full = in - out - 1$ and $full + empty = N - 1$, we get $in - out - 1 = N - 1 - empty$.

(g) $post(D_{11} \wedge D_{23} \wedge D_{34} \wedge D_{43} \wedge D_{52}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (PW \leq pc_1 \leq PM) \wedge (CR \leq pc_2 \leq CM) \wedge (empty + full = N - 2) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge CA \leq pc_2 \leq CR \wedge out < M \wedge$
$\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k]), \rho(v, v'))$
$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CR \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 2) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])), \rho(v, v'))$
$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CR \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 2) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge buf[in \ mod \ N] = x)$
$\models D_{11} \wedge D_{23} \wedge D_{34} \wedge D_{44} \wedge D_{52}$

(h) $post(D_{11} \wedge D_{23} \wedge D_{34} \wedge D_{43} \wedge D_{53}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (PW \leq pc_1 \leq PM) \wedge (CR \leq pc_2 \leq CM) \wedge (empty + full = N - 2) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CM \wedge out < M \wedge$
$\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k]) \wedge y = g(A[out]), \rho(v, v'))$
$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CM \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 2) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge y = g(A[out]), \rho(v, v'))$
$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CM \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 2) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge y = g(A[out]) \wedge buf[in \ mod \ N] = x)$
$\models D_{11} \wedge D_{23} \wedge D_{34} \wedge D_{44} \wedge D_{53}$

To avoid over-writing of some buffer content during the transition, $in - out < N$ should be satisfied. This is justified for the above two cases since from $full = in - out - 1$ and $full + empty = N - 2$, we get $in - out = N - 1 - empty$.

Therefore, we can say that the invariant is stable under the transition $PW \rightarrow PM$ since applying the transition on the invariant results only in states that are already in the invariant.

2. $CI \rightarrow CL$
The transition $CI \rightarrow CL$ can be represented as $\rho(v, v') = (pc_2 = CI \wedge pc_2' = CL \wedge out' = out + 1 \wedge x' = x \wedge y' = y \wedge full' = full \wedge empty' = empty \wedge pc_1' = pc_1 \wedge in' = in \wedge buf' = buf)$.
Like the case for the first question, we identify the applicable disjuncts. *post* will not be applicable on disjuncts which contain D_{21}, D_{24}, D_{32}, D_{34}, D_{51}, D_{52}, D_{53}, D_{54}, D_{56} and D_{57} since $pc_2 \neq CI$ in such disjuncts reducing the candidates to from 448 to 20. In addition, some disjuncts are simply unsatisfiability together. For example, although D_{22} and D_{33} satisfy $pc_2 = CI$, there is no value for pc_1 that satisfies $D_{22} \wedge D_{33}$ which makes *post* inapplicable over disjuncts that contain both D_{22} and D_{33}. This leaves us with only 7 disjuncts that *post* is applicable with respect to $\rho(v, v')$, which are given below:

$$\big(D_{11} \wedge D_{22} \wedge D_{31} \wedge D_{44} \wedge D_{55}\big) \vee$$

$$\big(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{41} \wedge D_{55}\big) \vee$$

$$\big(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{42} \wedge D_{55}\big) \vee$$

$$\big(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{43} \wedge D_{55}\big) \vee$$

$$\big(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{45} \wedge D_{55}\big) \vee$$

$$\big(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{43} \wedge D_{55}\big) \vee$$

$$\big(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{44} \wedge D_{55}\big)$$

Let us now apply *post* on each of these disjuncts and check if the resulting state is already in the invariant or not.

(a) $post(D_{11} \wedge D_{22} \wedge D_{31} \wedge D_{44} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 = PI) \wedge (CR \leq pc_2 \leq CI) \wedge$
$(full = in - out) \wedge (pc_1 \leq PA \vee pc_1 \geq PI) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N) \wedge PM \leq pc_1 \leq PI \wedge$
$in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PI \wedge pc_2 = CI \wedge$
$full = in - out \wedge empty + full = N \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PI \wedge pc_2 = CL \wedge full = in - out + 1 \wedge$
$empty + full = N \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge$
$B[out - 1] = f(g(A[out - 1])))$

$\models D_{11} \wedge D_{24} \wedge D_{31} \wedge D_{44} \wedge D_{56}$

(b) $post(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{41} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (pc_1 \leq PA \vee pc_1 \geq PI) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge$
$(empty + full = N) \wedge pc_1 = PS \wedge in = 0 \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PS \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N) \wedge in = 0 \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out - 1 : B[k] = f(g(A[k]))) \wedge pc_1 = PS \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge in = 0 \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out - 1] = f(g(A[out - 1])))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PS \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge in = 0 \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])))$

$\models D_{11} \wedge D_{21} \wedge D_{31} \wedge D_{41} \wedge D_{56}$

(c) $post(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{42} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (pc_1 \leq PA \vee pc_1 \geq PI) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge$
$(empty + full = N) \wedge pc_1 = PR \wedge in < M \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PR \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N) \wedge in < M \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out - 1 : B[k] = f(g(A[k]))) \wedge pc_1 = PR \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge in < M \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out - 1] = f(g(A[out - 1])))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PR \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge in < M \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])))$

$\models D_{11} \wedge D_{21} \wedge D_{31} \wedge D_{42} \wedge D_{56}$

(d) $post(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{43} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (pc_1 \leq PA \vee pc_1 \geq PI) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge$
$(empty + full = N) \wedge PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PA \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out - 1 : B[k] = f(g(A[k]))) \wedge pc_1 = PA \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge in < M \wedge x = g(A[in]) \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out - 1] = f(g(A[out - 1])))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PA \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge in < M \wedge x = g(A[in]) \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])))$

$\models D_{11} \wedge D_{21} \wedge D_{31} \wedge D_{43} \wedge D_{56}$

(e) $post(D_{11} \wedge D_{23} \wedge D_{31} \wedge D_{45} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (pc_1 \leq PA \vee pc_1 \geq PI) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge$
$(empty + full = N) \wedge PL \leq pc_1 \leq PF \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge PL \leq pc_1 \leq PF \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out]))), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out - 1 : B[k] = f(g(A[k]))) \wedge PL \leq pc_1 \leq PF \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out - 1] = f(g(A[out - 1])))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge PL \leq pc_1 \leq PF \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N) \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])))$

$\models D_{11} \wedge D_{21} \wedge D_{31} \wedge D_{45} \wedge D_{56}$

(f) $post(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{43} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge$
$PA \leq pc_1 \leq PW \wedge in < M \wedge x = g(A[in]) \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out - 1 : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out - 1] = f(g(A[out - 1])))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PW \wedge pc_2 = CL \wedge full = in - out \wedge$
$(empty + full = N - 1) \wedge in < M \wedge x = g(A[in]) \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])))$

$\models D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{43} \wedge D_{56}$

(g) $post(D_{11} \wedge D_{23} \wedge D_{33} \wedge D_{44} \wedge D_{55}, \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge (pc_1 \leq PM \vee pc_1 \geq PL) \wedge$
$(CR \leq pc_2 \leq CI) \wedge (full = in - out - 1) \wedge (PW \leq pc_1 \leq PM) \wedge (pc_2 \leq CA \vee pc_2 \geq CW) \wedge (empty + full = N - 1) \wedge$
$PM \leq pc_1 \leq PI \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge pc_2 = CI \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])), \rho(v, v'))$

$= post(0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 0 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CI \wedge$
$(full = in - out - 1) \wedge (empty + full = N - 1) \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge out < M \wedge$
$(\forall k \in (out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out] = f(g(A[out])), \rho(v, v'))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out - 1 : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CL \wedge$
$(full = in - out) \wedge (empty + full = N - 1) \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge out \leq M \wedge$
$(\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])) \wedge B[out - 1] = f(g(A[out - 1])))$

$= (0 \leq empty \wedge 0 \leq full \wedge 0 \leq in \wedge 1 \leq out \wedge (\forall k < out : B[k] = f(g(A[k]))) \wedge pc_1 = PM \wedge pc_2 = CL \wedge full = in - out \wedge$
$(empty + full = N - 1) \wedge in < M \wedge buf[in \ mod \ N] = g(A[in]) \wedge out \leq M \wedge (\forall k \in [out, in) : buf[k \ mod \ N] = g(A[k])))$

$\models D_{11} \wedge D_{21} \wedge D_{33} \wedge D_{44} \wedge D_{56}$

Therefore, we can say that the invariant is stable under the transition $CI \to CL$ since applying the transition on the invariant results only in states that are already in the invariant.

Question 2 Consider the Dijkstra's two-threaded algorithm. Prove or refute that the inductive invariant given in class is the strongest one, i.e. that every invariant that implies the given one is already equivalent to the given one.

The given inductive invariant is:

$$\begin{pmatrix} (pc_1 \in \{S, L_2\} \wedge \neg req_1) \vee \\ (pc_1 \in \{L_1, C\} \wedge req_1) \end{pmatrix} \wedge \begin{pmatrix} (pc_2 \in \{S, L_2\} \wedge \neg req_2) \vee \\ (pc_2 \in \{L_1, C\} \wedge req_2) \end{pmatrix} \wedge \neg(pc_1 = pc_2 = C) \tag{3}$$

One way of proving (or refuting) is to compute the strongest inductive invariant and compare it with the given inductive invariant. The computed strongest invariant is:

$(pc_1 = S \wedge \neg req_1 \wedge pc_2 = S \wedge \neg req_2) \vee (pc_1 = L_1 \wedge req_1 \wedge pc_2 = S \wedge \neg req_2) \vee (pc_1 = S \wedge \neg req_1 \wedge pc_2 = L_1 \wedge req_2) \vee$
$(pc_1 = L_1 \wedge req_1 \wedge pc_2 = L_1 \wedge req_2) \vee (pc_1 = L_1 \wedge req_1 \wedge pc_2 = C \wedge req_2) \vee (pc_1 = L_1 \wedge req_1 \wedge pc_2 = L_2 \wedge \neg req_2) \vee$
$(pc_1 = S \wedge \neg req_1 \wedge pc_2 = L_2 \wedge \neg req_2) \vee (pc_1 = C \wedge req_1 \wedge pc_2 = L_2 \wedge \neg req_2) \vee (pc_1 = C \wedge req_1 \wedge pc_2 = L_1 \wedge req_2) \vee$
$(pc_1 = C \wedge req_1 \wedge pc_2 = S \wedge \neg req_2) \vee (pc_1 = S \wedge \neg req_1 \wedge pc_2 = C \wedge req_2) \vee (pc_1 = L_2 \wedge \neg req_1 \wedge pc_2 = L_1 \wedge req_2) \vee$
$(pc_1 = L_2 \wedge \neg req_1 \wedge pc_2 = C \wedge req_2) \vee (pc_1 = L_2 \wedge \neg req_1 \wedge pc_2 = S \wedge \neg req_2)$

The given invariant contians the state satisfying $(pc_1 = L_2 \wedge \neg req_1 \wedge pc_2 = L_2 \wedge \neg req_2)$ which is not in the strongest inductive invariant. Therefore, the given inductive invariant is not the strongest one.

Question 3 The following mutual exclusion algorithm for 2 threads is suggested:

initially turn$\in \{1,2\} \wedge Q_1 = Q_2 = false$

// Thread 1:	// Thread 2:
while(true) {	while(true) {
// noncritical section	// noncritical section
A: Q_1:=true	A: Q_2:=true
B: turn:=1	B: turn:=2
C: $\langle$await $\neg Q_2 \vee$ turn=2$\rangle$	C: $\langle$await $\neg Q_1 \vee$ turn=1$\rangle$
// critical section	// critical section
D: Q_1:=false	D: Q_2:=false
// noncritical section	// noncritical section
}	}

Prove or refute the mutual exclusion property, which here says that in any state reachable from the initial ones the two threads are not simultaneously at the critical locations D. You may assume that the threads start at locations A and the transitions between each pair of labels is atomic.
One way to prove (or refute) mutual exclusiveness is to compute the strongest inductive invariant by staring from the inital state and applying the possible transitions from both threads until all computed states are already reached.

The strongest inductive invariant is:

$$
\begin{aligned}
I = \quad & (PC_1 = A \wedge PC_2 = A \wedge \neg Q_1 \wedge \neg Q_2) && \vee (PC_1 = B \wedge PC_2 = A \wedge Q_1 \wedge \neg Q_2) \vee \\
& (PC_1 = A \wedge PC_2 = B \wedge \neg Q_1 \wedge Q_2) && \vee (PC_1 = C \wedge PC_2 = A \wedge Q_1 \wedge \neg Q_2 \wedge turn = 1) \vee \\
& (PC_1 = B \wedge PC_2 = B \wedge Q_1 \wedge Q_2) && \vee (PC_1 = A \wedge PC_2 = C \wedge \neg Q_1 \wedge Q_2 \wedge turn = 2) \vee \\
& (PC_1 = D \wedge PC_2 = A \wedge Q_1 \wedge \neg Q_2 \wedge turn = 1) && \vee (PC_1 = C \wedge PC_2 = B \wedge Q_1 \wedge Q_2 \wedge turn = 1) \vee \\
& (PC_1 = B \wedge PC_2 = C \wedge Q_1 \wedge Q_2 \wedge turn = 2) && \vee (PC_1 = A \wedge PC_2 = D \wedge \neg Q_1 \wedge Q_2 \wedge turn = 2) \vee \\
& (PC_1 = D \wedge PC_2 = B \wedge Q_1 \wedge Q_2 \wedge turn = 1) && \vee (PC_1 = C \wedge PC_2 = C \wedge Q_1 \wedge Q_2 \wedge turn = 2) \vee \\
& (PC_1 = C \wedge PC_2 = C \wedge Q_1 \wedge Q_2 \wedge turn = 1) && \vee (PC_1 = B \wedge PC_2 = D \wedge Q_1 \wedge Q_2 \wedge turn = 2) \vee \\
& (PC_1 = D \wedge PC_2 = C \wedge Q_1 \wedge Q_2 \wedge turn = 2) && \vee (PC_1 = C \wedge PC_2 = D \wedge Q_1 \wedge Q_2 \wedge turn = 1) \vee
\end{aligned}
$$

and, we can see that there is no reachable state that satisfies $(PC_1 = D \wedge PC_2 = D)$.

Homework 8

Question 1 In class a formula I for the Szymanski's mutual exclusion protocol was given as:

$$\begin{aligned}
&Let\\
&L_j = \{t \in Tid \mid pc_t = l_j\} forall j : 1 \leq j \leq 12,\\
&L_{j1,j2,\ldots,jm} = L_{j1} \cup L_{j2} \cup \ldots \cup L_{jm} = \bigcup_{i=1}^{m} L_{ji},\\
&F_k = \{t \in Tid \mid flag[t] = k\} forall k : 0 \leq k \leq 4,\\
&F_{k1,k2,\ldots,km} = F_{k1} \cup F_{k2} \cup \ldots \cup F_{km} = \bigcup_{i=1}^{m} F_{ki},\\
&IF = \begin{pmatrix} F_0 = L_{1,2} \wedge F_1 = L_{3,4} \wedge F_2 \subseteq L_{7,8} \wedge F_3 = L_{5,6,8} \wedge \\ F_4 = L_{9,\ldots,12} \wedge Tid \subseteq F_{0,\ldots,4} \end{pmatrix},\\
&A_0 = \left(L_{8,\ldots,12} \neq \emptyset \rightarrow L_4 = \emptyset\right),\\
&A_1 = \left(L_{8,\ldots,12} \neq \emptyset \rightarrow L_{8,\ldots,12} \cap F_{3,4} \neq \emptyset\right),\\
&A_2 = \left(\forall t \in L_{10,11,12}, : \forall k < t : k \notin L_{5,\ldots,12}\right),\\
&A_3 = \left(L_{12} \neq \emptyset \rightarrow L_{5,\ldots,12} \subseteq F_4\right),\\
&and\\
&I = IF \wedge A_0 \wedge A_1 \wedge A_2 \wedge A_3.
\end{aligned}$$

1. Prove that I is stable under each transition at locations l_5 till l_{12} of each thread.
(************** to be added soon ********************)

2. Show that I implies the mutual exclusion property, namely, that $\forall i, j \in Tid : (l_i = 10 = l_j) \rightarrow (i = j)$.
We can show the property by assuming the contrary and reaching a contradiction.

   Let us assume $\exists i, j \in Tid : (l_i = 10 = l_j) \wedge (i \neq j)$.
   Case $i < j$: by A_2 we get $i \notin L_{5,\ldots,12}$ which is a contradiction to $l_i = 10$.
   Case $j < i$: similarly, by A_2 we get $j \notin L_{5,\ldots,12}$ which is a contradiction to $l_j = 10$.

3. Assume that the transition at location l_{11} is replaced by a no-operation (which changes just the program counter of the executing thread, while the remaining variables retain their values). Is the mutual exclusion property still satisfied?

   Yes, the mutual exclusion property is still satisfied. The conjunct A_3 does not hold anymore, and the new inductive invariant will be $I = IF \wedge A_0 \wedge A_1 \wedge A_2$, which still contains A_2 that ensures mutual exclusion.
   However, since some thread may fail to close the door, the next batch of threads may come in and access the critical section even before that thread. This may happen infinitely often so that this thread may never actually get access the critical section. So, the algorithm will not be fair anymore.

Question 2 (An optional task with an increased difficulty level.) Let $(L, \leq)$ be a complete lattice (i.e. a partial order in which for every set $A \subseteq L$, the least upper bound $\sup A$ and the greatest lower bound $\inf A$ exist). Let $f : L \rightarrow L$.

1. If f is monotone, then the least fixpoint of f, written $\mathit{lfp}(f)$, exists and is equal to $\inf\{x \in L | f(x) = x\} = \inf\{x \in L | f(x) \leq x\}$.

   There are three proofs to be done here:

   - show that $\mathit{lfp}(f)$ exists **(let's call it P_1)**,
   - show that $\mathit{lfp}(f) = \inf\{x | f(x) = x\}$ **(let's call it P_2)**, and
   - show that $\mathit{lfp}(f) = \inf\{x | f(x) \leq x\}$ **(let's call it P_3)**.

   (a) Let $l = \inf\{x | f(x) \leq x\}$, i.e. l is the greatest lower bound of $\{x | f(x) \leq x\}$.
   (b) We get $f(l) \leq l$.
   (c) $\forall : y\ f(y) \leq y \rightarrow y \geq l$.
   (d) since f is monotone, we have $f(f(l)) \leq f(l)$.
   (e) By (c), we have $f(l) \geq l$.
   (f) $f(l) = l$, i.e. l is a fixpoint, from (b) and (e) **proving P_1**.
   (g) l is the least fixpoint by (a) and (f) **proving P_3**.
   (h) $l \in \{x | f(x) = x\}$ by (f).
   (i) $l = \inf\{x | f(x) = x\}$ by (a) since $\{x | f(x) = x\} \subseteq \{x | f(x) \leq x\}$ **proving P_2**.

2. For all nonempty chains $C \subseteq L$, if we have $\sup f(C) = f(\sup C)$, then $\mathit{lfp}(f) = \sup\{f^i(\inf L) | i \in \mathbb{N}_0\}$.

   Let $f(B) = B$ be any fixpoint, we first show that $\sup\{f^i(\inf L) | i \in \mathbb{N}_0\} \leq B$, and then that it is a fixpoint. We will prove by induction that $\forall i \in \mathbb{N}_0 : f^i(\inf L) \leq B$.

   (a) **base case:** $\mathbf{i = 0}$. $f^i(\inf L) = f^0(\inf L) = \inf L \leq B$.
   (b) **induction step:** we assume $f^{i-1}(\inf L) \leq B$, and then we try to show $f^i(\inf L) \leq B$.
   (c) $f^i(\inf L) = f(f^{i-1}(\inf L))$.
   (d) $f(f^{i-1}(\inf L)) \leq \sup\{f(f^{i-1}(\inf L)), f(B)\}$.
   (e) $f(f^{i-1}(\inf L)) \leq f(\sup\{f^{i-1}(\inf L), B\})$ by the assumption for non-empty chains.
   (f) $f(f^{i-1}(\inf L)) \leq f(B)$ by inductive hypothesis.
   (g) $f^i(\inf L) \leq B$ from $f(B) = B$, and this shows $\sup\{f^i(\inf L) | i \in \mathbb{N}_0\} \leq \mathit{lfp}(f)$.
   (h) $f(\sup\{f^i(\inf L) | i \in \mathbb{N}_0\}) = \sup\{f^i(\inf L) | i \in \mathbb{N}^+\}$.
   (i) $f(\sup\{f^i(\inf L) | i \in \mathbb{N}_0\}) = \sup\{\{f^i(\inf L) | i \in \mathbb{N}^+\} \cup \{\inf L\}\} = \sup\{f^i(\inf L) | i \in \mathbb{N}_0\}$ since adding $\inf L$ to the set will not affect the value of sup for the set; i.e. $\sup\{f^i(\inf L) | i \in \mathbb{N}_0\}$ is a fixpoint.
   (j) By (g) and (i), $\mathit{lfp}(f) = \sup\{f^i(\inf L) | i \in \mathbb{N}_0\}$.

Homework 9

1. Infer the type in the empty typing environment:

```
let fun f x y =
    if x $>$ y then true
    else  false
in f 0
end
```

```
T1 := {f : int -> int -> bool, x: = int, y := int}

T1 |- > : int -> int -> bool
T1 |- x : int
T1 |- y : int
-------------------       T1 |- true : bool
T1 |-  x > y : bool       T1 |- false : bool
--------------------------------------------
{f : int -> int -> bool, x: = int, y := int}  {f : int -> int -> bool} |- f : int -> int -> bool
  |-  if x > y then true else false : bool    {f : int -> int -> bool} |- 0 : int
------------------------------------------    ------------------------------------------------
{} |> fun f x y =                             {f : int -> int -> bool} |- f 0: int -> bool
   if x > y then true else false :
   {f : int -> int -> bool}
------------------------------------------------------------------------------------
{} |- let fun f x y = if x > y then true else false in f 0 end : int -> bool
```

2. Which typing environment is obtained by typing the following declarations in the empty typing environment:

1. val t = 3$\{t : int\}$

2. fun fib n = if n < 3 then 1 else fib (n-1) + fib(n-2)$\{fib : int \rightarrow int\}$

3. fun square r = let fun exp y x = if y = 0 then 1 else if y < 0 then 0 else x * (exp (y-1) x) in exp 2 r end$\{square : int \rightarrow int\}$

3. Which sequence of value environments is obtained by evaluating the following program?
fun f x = if x then 1 else 0;
val x = 5*7;
fun g z = f (z < x) < x;
val x = g 5;
val k = let fun h x = x * x in h end;

Note: In the script the type does not need to be specified for functions, so it is left out here as well (unlike during the exercises and the lecture).

[f := (fun f x = if x then 1 else 0, [])]

[f := (fun f x = if x then 1 else 0, []), x := 35]

[f := (fun f x = if x then 1 else 0, []), x := 35,
g:= (fun g z = f (z < x) < x, [x := 35 , f := (fun f x = if x then 1 else 0, [])])]

[f:= (fun f x = if x then 1 else 0, []),
g:= (fun g z = f (z < x) < x, [x := 35 , f := (fun f x = if x then 1 else 0, [])]), x := true]

[f:= (fun f x = if x then 1 else 0, []),
g:= (fun g z = f (z < x) < x, [x := 35 , f := (fun f x = if x then 1 else 0, [])]), x := true,
k:= (fun h x = x * x, [])]

4. Evaluate the following expression: let fun square r = let fun exp y x = if y = 0 then 1 else if y ¡ 0 then 0 else x * (exp (y-1) x) in exp 2 r end in square 5 end

```
let fun square r =
  E
  in
  square 5
end

E :=
let fun exp y x  =
  F
  in
  exp 2 r
end

F :=
if y = 0 then
    1
  else
    G

G :=
if y < 0 then
  0
else
  x * (exp (y-1) x))


V1 := [square := (fun square r = E, [])]
V2 := [square := (fun square r = E, []), r := 5, exp := (fun exp y x = F, [])]
V3 := [exp := (fun exp y x = F, []), y := 2, x := r]
V4 := [exp := (fun exp y x = F, []), y := 1, x := r]
V4 := [exp := (fun exp y x = F, []), y := 0, x := r]
```

```
                                                                 V4(y) = 1
                                                                 -----------
                                                                 V4 |= y : 1      V4 |= 1 : 1
V5 |= y = 0 : true      V5 |= 1 : 1                              ----------------------------
-----------------------------------                 V4 |= x : r      V4 |= y-1 : 0
                         V5 |= F: 1                 V4 |= exp : (fun exp y x= F, [])
                         ------------------------------------------------
```

```
                                             V4 |= x : r      V4 |= exp (y-1) x : 1
                                             --------------------------------------                                                 V3(V) = 2
                    V4 |= y < 0 : false      V4 |= x * (exp (y-1) x) : r                                                            -----------
                    ----------------------------------------------------                                                            V3 |= y : 2    V3 |- 1 : 1
                                                             V4 |= G : r      V4 |= y = 0 : false                                   --------------------------
                                                             ------------------------------------    V3 |= x : r    V3 |= y-1 : 1
                                                                                          V4 |= F: r     V3 |= exp : (fun exp y x= F, [])
                                                                                          -----------------------------------------------
                                                                      V3 |= x : r      V3 |= exp (y-1) x : r
                                                                      --------------------------------------
                                             V3 |= y < 0 : false      V3 |= x * (exp (y-1) x) : r^2
                                             ------------------------------------------------------
                    V3 |= y = 0 : false      V3 |= G : r^2
                    --------------------------------------    V2 |= 2 : 2      V2 |= r : 5
                                              V3 |= F: r^2      V2 |= exp : (fun exp y x= F, [])      FI(fun exp x y = F) = []
                                              --------------------------------------------------      ------------------------------------
                                                                              V2 |= exp 2 r : 25      V1+ [r := 5]|>> fun exp y x = F : V2
                                  V1 |= 5 : 5                                 ------------------------------------------------------------
FI(fun square r = E) = []         V1 |= square :  (fun square r = E, [])      V1+[r := 5] |= E : 25
----------------------------      -----------------------------------------------------------------
[] |>> fun square r = E : V1      V1 |= square 5 : 25
-----------------------------------------------------
[] |= let fun square r = E in square 5 end : 25
```

5. Formalize as a refinement type: the value of x is a negative integer that is greater then the sum of values of y and z. $x : \{v : int | v < 0 \wedge v > y + z\}$

6. Formalize as a refinement type: the value of f is a function that takes as input a positive integer and returns the doubled value. $f : (x : \{v : int | v > 0\} \rightarrow \{v : int | v = 2n\})$

Homework 10

Part I - Refinement types Provide refinement type derivation for the following functions (as shown on Slides 16.4 and 16.5).

1. The fibonacci sequence:

```
fun fib n =
    if n < 3 then 1
    else
        let val m = fib (n - 1) in
        m + fib(n - 2)
```

```
R1 := {fib : (n : r1 -> r2, n : r1}
R2 := R1,  n >= 3, m : r3
r1 = {v : int | P1(v,..)}
r2 = {v : int | P2(v,..)}

r3 = r2[n-1/n]
r4 = r2[n-2/n]
```

```
                    formula(R1, n >= 3) |= P1[n-1/v]        formula(R2) |= P1[n-2/v]
                    ---------------------------------       -----------------------
                    R1, n >= 3 |- n - 1 : r1                R2 |- n - 2 : r1
                    R1, n >= 3 |- fib : (n : r1 -> r2)      R2 |- fib  : (n : r1 -> r2)
                    -----------------------------------     -------------------------------------
                    R1, n >= 3 |- fib (n - 1) : r3          R2|- m : r3     R2|- fib (n - 2) : r4
                    ---------------------------------------  -------------------------------------
formula(R1,n < 3)   R1, n >= 3 |> val m = fib (n - 1) : R2   R2 |- m + fib (n - 2) : r2
|= P2[1/v]
-------------------   -------------------------------------------------------------------
R1, n < 3 |- 1 : r2   R1, n >= 3 |-  let val m = fib (n - 1) in m + fib(n - 2) end : r2
-----------------------------------------------------------------
R1 |- if n < 3 then 1 else let val m = fib (n - 1) in m + fib(n - 2) end : r2
-----------------------------------------------------------------------------------
{} |> fib : (n : r1 -> r2) : R1

possible solutions: P2 = (v > 0), P2 = true
```

2. The maximum of two numbers:

```
fun max x y =
    if x > y then x
    else y
```

```
R1 := {max : (x : r1 -> y:  r2 -> r3, x : r1, y : r2}
r3 = {v : int | P3(v,..)}
```

```
formula(R1, x > y) |= P3[x/v]    formula(R1, x <= y) = P3[y/v]
-----------------------------    -----------------------------
R1, x > y |- x : r3              R1, x <= y |- y : r3
-----------------------------------------------------
R1 |- if x > y then x else y : r3
----------------------------------------------------------------------------------
{} |> max x y : (x : r1 -> y : r2 -> r3) : R1

possible solutions: P3 = (v > y \or x <= v), P3 = true
```

3. Factorial of an integer n:

```
fun fact n =
    if n < 1 then 1
    else
        let val m = fact (n - 1) in
        n * m
```

Part II - LTL

A. Let $AP = \{green, yellow, red\}$ and $\sigma = (\{green\}\{green\}\{green\}\{yellow\}\{red\}\{red\}\{red\}\{yellow, red\})^\omega$. Does σ satisfy the following properties?

1. $\bigcirc green \vee yellow$
   Yes.

2. $\neg green \,\mathsf{U}\, red$
   No.

3. $\neg(green \,\mathsf{U}\, red)$
   Yes.

B. Let $AP = \{green, yellow, red\}$. Write the following properties as LTL formulas (derived operators are allowed).

1. Red and yellow occur together infinitely often.
   $\Box\Diamond(red \wedge yellow)$.

2. From some time point onward red and green never occur together.
   $\Diamond\Box(\neg(red \wedge green))$.

3. Whenever green turns on, green continues for at least two consecutive time units.
   $\Box(\neg green \wedge \bigcirc green \rightarrow \bigcirc\bigcirc green \wedge \bigcirc\bigcirc\bigcirc green)$.

C. Consider a program P with State$= \{s_0, s_1, s_2\}$, init$= \{s_0\}$, transitions $s_1 \rightarrow s_0 \rightarrow s_2 \rightarrow s_1$. Let $AP = \{a, b\}$. Let a label just s_1 and b label just s_2. Do the following formulas hold for P?

1. $\Diamond a \wedge \Diamond b$.
   Yes.

2. $\Diamond(a \wedge b)$.
   No.

3. $\Diamond a \,\mathsf{U}\, \Box\neg(a \wedge b)$.
   Yes.

4. $\Box\Diamond b$.
   Yes.

D. Let AP be a set of atomic propositions and φ, ψ be LTL formulas over AP. Show the following properties about distributivity, negation propagation, and expansion of temporal connectives:

1. $\bigcirc(\varphi \wedge \psi) \equiv \bigcirc\varphi \wedge \bigcirc\psi$.
   We have to show that $\sigma \models \bigcirc(\varphi \wedge \psi)$ if and only if $\sigma \models \bigcirc\varphi \wedge \bigcirc\psi$.

   " $\Rightarrow$ " :

   (a) assume $\sigma \models \bigcirc(\varphi \wedge \psi)$. We have $\sigma[0..] \models \bigcirc(\varphi \wedge \psi)$.
   (b) $\sigma[1..] \models \varphi \wedge \psi$ by applying the definition of $\bigcirc$.
   (c) $\sigma[1..] \models \varphi$ and $\sigma[1..] \models \psi$ by eliminating $\wedge$.
   (d) $\sigma \models \bigcirc\varphi$ and $\sigma \models \bigcirc\psi$ by reduction to $\bigcirc$ using its definition.
   (e) $\sigma \models \bigcirc\varphi \wedge \bigcirc\psi$ by introducing $\wedge$.

   " $\Leftarrow$ " :

   (a) Assume $\sigma \models \bigcirc\varphi \wedge \bigcirc\psi$.

(b) We get $\sigma \models \bigcirc\varphi$ and $\sigma \models \bigcirc\psi$ by eliminating $\wedge$.

(c) By applying definition of $\bigcirc$, we get $\sigma[1..] \models \varphi$ and $\sigma[1..] \models \psi$.

(d) We then get $\sigma[1..] \models \varphi \wedge \psi$ by introducing $\wedge$.

(e) $\sigma \models \bigcirc(\varphi \wedge \psi)$ by reducing to $\bigcirc$ using its definition.

2. $\bigcirc(\varphi \,\mathsf{U}\, \psi) \equiv \bigcirc\varphi \,\mathsf{U} \bigcirc\psi$.

3. $\neg\Box\varphi \equiv \Diamond\neg\varphi$.

4. $\neg(\varphi \,\mathsf{U}\, \psi) \equiv \neg\varphi \,\mathsf{R}\, \neg\psi$.

   We have to show that $\neg(\varphi \,\mathsf{U}\, \psi)$ if and only if $(\neg\varphi \,\mathsf{R}\, \neg\psi)$.

   " $\Rightarrow$ " :
   $\varphi \,\mathsf{U}\, \psi$ is defined as $\exists j \geq 0(\sigma[j..] \models \psi \wedge \forall i < j\sigma[i..] \models \varphi)$, and its negation $\neg(\varphi \,\mathsf{U}\, \psi)$ is $\neg(\exists_{j\geq 0}(\sigma[j..] \models \psi \wedge \forall_{i<j}\sigma[i..] \models \varphi))$ which is equivalent to $\forall j \geq 0(\sigma[j..] \models \neg\psi \vee \exists i < j\sigma[i..] \models \neg\varphi)$. But this defines $\neg\varphi \,\mathsf{R}\, \neg\psi$.

   " $\Leftarrow$ " :
   This is done by exactly doing the reverse of the " $\Rightarrow$ " proof. We know that $\neg\varphi \,\mathsf{R}\, \neg\psi \equiv \neg\neg(\neg\varphi \,\mathsf{R}\, \neg\psi)$. By applying the double negation on the definition of $\neg\varphi \,\mathsf{R}\, \neg\psi$, we get $\neg\neg(\forall_{j\geq 0}(\sigma[j..] \models \neg\psi \vee \exists_{i<j}\sigma[i..] \models \neg\varphi))$ which is equivalent to $\neg(\exists_{j\geq 0}(\sigma[j..] \models \psi \wedge \forall_{i<j}\sigma[i..] \models \varphi))$. But the one inside the negation defines $\varphi \,\mathsf{U}\, \psi$, and hence the whole formula will be $\neg(\varphi \,\mathsf{U}\, \psi)$.

5. $\neg(\varphi \,\mathsf{W}\, \psi) \equiv (\varphi \wedge \neg\psi) \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$.

   We have to show that $\sigma \models \neg(\varphi \,\mathsf{W}\, \psi)$ if and only if $\sigma \models (\varphi \wedge \neg\psi) \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$.

   " $\Rightarrow$ " :
   We have $\Diamond\neg\varphi \wedge \neg\varphi \,\mathsf{R}\, \neg\psi$. Let $j \geq 0$ be the first state that $\neg\varphi$ holds, i.e. $\sigma[j..] \models \neg\varphi$ and $\forall i < j : \sigma[i..] \models \varphi$. From $\neg\varphi \,\mathsf{R} \neg\psi$, we have $\Box\neg\psi$ or $\neg\psi \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$.
   Case $\Box\neg\psi$: then, $\exists j : \sigma[j..] \models \neg\varphi \wedge \neg\psi$ and $\forall i < j : \sigma[i..] \models \varphi \wedge \neg\psi$. Therefore, $(\varphi \wedge \neg\psi) \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$.
   Case $\neg\psi \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$: then, $\forall i < j : \sigma[i..] \not\models \neg\varphi \wedge \neg\psi$, and hence $\forall i < j : \sigma[i..] \models \neg\psi$ and $\sigma[j..] \models \neg\psi$. Thus, $\exists j : \sigma[j..] \models \neg\varphi \wedge \neg\psi$ and $\forall i < j : \sigma[i..] \models \varphi \wedge \neg\psi$. Therefore, $(\varphi \wedge \neg\psi) \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$.
   " $\Leftarrow$ " :
   From $\varphi \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$, we get $\Diamond\neg\varphi$, and from $\psi \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$, we get $\Box\neg\psi \vee \psi \,\mathsf{U}\, (\neg\varphi \wedge \neg\psi)$ which implies $\neg\varphi \,\mathsf{R}\, \neg\psi$. $\Diamond\neg\varphi$ and $\neg\varphi \,\mathsf{R}\, \neg\psi$ together imply $\neg(\Box\varphi \vee \varphi \,\mathsf{R}\, \psi)$, i.e. $\neg(\varphi \,\mathsf{W}\, \psi)$.

6. $\neg(\varphi \,\mathsf{R}\, \psi) \equiv (\neg\varphi \,\mathsf{U}\, \neg\psi)$.

7. $\varphi \,\mathsf{U}\, \psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$.

   We have to show that $\sigma \models \varphi \,\mathsf{U}\, \psi$ if and only if $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$.

   " $\Rightarrow$ " :
   There is $j \geq 0$ such that $\sigma[j..] \models \psi$ and $\forall_{i<j}\sigma[i..] \models \varphi$.
   Case j=0: then $\sigma \models \psi$, therefore $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$.
   Case j>0: then $\sigma[0..] \models \varphi$, therefore $\sigma \models \varphi$. Also, $\forall i < j-1 : \sigma[i+1..] \models \varphi$, i.e. $\forall i < j-1 : \sigma[1..][i..] \models \varphi$. In addition, $\sigma[1..][j-1..] \models \psi$. Thus, $\sigma[1..] \models \varphi \,\mathsf{U}\, \psi$. Therfore, $\sigma \models \varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi)$ which implies that $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$.

   " $\Leftarrow$ " :
   Case $\sigma \models \psi$: then $\sigma[0..] \models \psi$ and there is no i such that $i < 0$. Therefore, $\sigma \models \varphi \,\mathsf{U}\, \psi$.
   Case $\sigma \models \varphi \wedge \bigcirc(\varphi \mathsf{U} \psi)$: then, $\sigma[1..] \models \varphi \mathsf{U} \psi$. There is $j \geq 0$ such that $\sigma[1..][j..] \models \psi$ and $\forall i < j : \sigma[1..][i..] \models \varphi$. Therefore, $\forall i < j : \sigma[i+1..] \models \varphi$ and $\sigma[j+1..] \models \psi$. Since $\sigma[0..] = \sigma \models \varphi$, we have $\forall i < j+1 : \sigma[i..] \models \varphi$. Thus, $\sigma \models \varphi \,\mathsf{U}\, \psi$.

8. $\varphi \,\mathsf{W}\, \psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{W}\, \psi))$.

   We have to show that $\sigma \models \varphi \,\mathsf{W}\, \psi$ if and only if $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{W}\, \psi))$. We use the definition $\varphi \,\mathsf{W}\, \psi \equiv \Box\varphi \vee \varphi \,\mathsf{U}\, \psi$.

   " $\Rightarrow$ " :
   $\sigma \models \varphi \,\mathsf{W}\, \psi$ implies $\sigma \models \Box\varphi \vee \varphi \,\mathsf{U}\, \psi$, i.e. $\sigma \models \Box\varphi$ or $\sigma \models \varphi \,\mathsf{U}\, \psi$.
   Case $\sigma \models \Box\varphi$: then, $\sigma \models \varphi$ and $\sigma[1..] \models \Box\varphi$. Therefore, $\sigma[1..] \models \varphi \,\mathsf{W}\, \psi$ which is equivalent with $\sigma \models \bigcirc(\varphi \,\mathsf{W}\, \psi)$. Thus, $\sigma \models \varphi \wedge \bigcirc(\varphi \,\mathsf{W}\, \psi)$.
   Case $\sigma \models \varphi \,\mathsf{U}\, \psi$: then, $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$ as it was proven in (7) above. This implies $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\Box\varphi \vee \varphi \,\mathsf{U}\, \psi))$. Thus, $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{W}\, \psi))$.

   " $\Leftarrow$ " :
   By the definition of W, we have $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\Box\varphi \vee \varphi \,\mathsf{U}\, \psi))$, which can be reduced to $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi)) \vee \varphi \wedge \bigcirc\Box\varphi$ i.e. $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$ or $\sigma \models \varphi \wedge \bigcirc\Box\varphi$. But, $\sigma \models \psi \vee (\varphi \wedge \bigcirc(\varphi \,\mathsf{U}\, \psi))$ implies $\sigma \models \varphi \,\mathsf{U}\, \psi$, and hence $\sigma \models \varphi \,\mathsf{W}\, \psi$, as it was proven in (7) above. For $\sigma \models \varphi \wedge \bigcirc\Box\varphi$, which is equivalent to $\sigma \models \Box\varphi$, we have $\sigma \models \Box\varphi \vee \varphi \,\mathsf{U}\, \psi$ which is equivalent with $\sigma \models \varphi \,\mathsf{W}\, \psi$.

E. Let $AP = \{green, yellow, red\}$. Convert the following formulas into positive LTL:

1. $\neg((yellow \mathsf{U}\, green) \mathsf{U}\, red)$.
   $(\neg\mathbf{yellow}\ \mathsf{R}\ \neg\mathbf{green})\ \mathsf{R}\ \neg\mathbf{red}$.

2. $\neg(green \mathsf{W} (red \mathsf{U}\, green))$.
   $\equiv \neg(\Box green \vee (green\ \mathsf{U}\ (red\ \mathsf{U}\ green)))$
   $\Diamond\neg\mathbf{green} \wedge \neg\mathbf{green}\ \mathsf{R}\ (\neg\mathbf{red}\ \mathsf{R}\ \neg\mathbf{green})$.

3. $\neg((yellow \mathsf{U}\, green) \mathsf{R} (red \mathsf{U}\, green))$.
   $(\neg\mathbf{yellow}\ \mathsf{R}\ \neg\mathbf{green})\ \mathsf{U}\ (\neg\mathbf{red}\ \mathsf{R}\ \neg\mathbf{green})$.

F. (A task with an increased level of difficulty, **.) Show that weak until is "the greatest solution of the expansion law". More formally, show that for all LTL formulas φ, ψ over a set of atomic propositions AP,

1. $words(\varphi \mathsf{W} \psi)$ is a fixpoint of the map $\lambda S \in \mathfrak{P}(\mathbb{N}_0 \to \mathfrak{P}(AP)).words(\psi) \cup \{\sigma \in words(\varphi) | \sigma[1..] \in S\}$.

   Let $f(S) = words(\psi) \cup \{\sigma \in words(\varphi) | \sigma[1..] \in S\}$. We will show $words(\varphi \mathsf{W} \psi)$ is a fixpoint of f; i.e. $f(words(\varphi \mathsf{W} \psi)) = words(\varphi \mathsf{W} \psi)$.

   " $\subseteq$ "
   Let $\sigma \in f(words(\varphi \mathsf{W} \psi))$.
   Case $\sigma \in words(\psi)$: then, $\sigma \models \psi$. So, $\sigma \models \varphi \mathsf{U} \psi$, and hence $\sigma \models \varphi \mathsf{W} \psi$. Therefore, $\sigma \in words(\varphi \mathsf{W} \psi)$.
   Case $\sigma \in words(\varphi)$ **and** $\sigma[1..] \in word(\varphi \mathsf{W} \psi)$: then, $\sigma \models \varphi$, and $(\sigma[1..] \models \Box\varphi$ or $\sigma[1..] \models \varphi \mathsf{U} \psi)$.

   - **sub-case** $\sigma[1..] \models \Box\varphi$: then, $\sigma \models \Box\varphi$. Therefore, $\sigma \models \varphi \mathsf{W} \psi$, i.e. $\sigma \in words(\varphi \mathsf{W} \psi)$.
   - **sub-case** $\sigma[1..] \models \varphi \mathsf{U} \psi$: then, there is $j \geq 0$ such that $\sigma[1..][j..] \models \psi$ and $\forall i < j : \sigma[1..][i..] \models \varphi$. This results in $\sigma[j+1..] \models \psi$ and $\forall 0 < i < j+1 : \sigma[i..] \models \varphi$. Since $\sigma \models \varphi$, we get $\forall i < j+1 : \sigma[i..] \models \varphi$. Thus, $\sigma \models \varphi \mathsf{U} \psi$, and hence, $\sigma \models \varphi \mathsf{W} \psi$, i.e. $\sigma \in words(\varphi \mathsf{W} \psi)$.

   " $\supseteq$ "
   Let $\sigma \in words(\varphi \mathsf{W} \psi)$. Then, $\sigma \models \Box\varphi$ or $\sigma \models \varphi \mathsf{U} \psi$.
   Case $\sigma \models \Box\varphi$: then, $\sigma \models \varphi$ and $\sigma[1..] \models \Box\varphi$. Thus $\sigma \in words(\varphi)$ and $\sigma[1..] \in \varphi \mathsf{W} \psi$. Therefore, $\sigma \in f(\varphi \mathsf{W} \psi)$.
   Case $\sigma \models \varphi \mathsf{U} \psi$: then, there is $j \geq 0$ such that $\sigma[j..] \models \psi$ and $\forall i < j : \sigma[i..] \models \varphi$.

   - **sub-case j=0**: then, $\sigma \models \psi$, therefore $\sigma \in words(\psi) \subseteq f(words(\varphi \mathsf{W} \psi))$.
   - **sub-case j>0**: then, for $k = j - 1$ we have $(\sigma[1..][k..] \models \psi$ and $\forall i < k : \sigma[1..][i..] \models \varphi)$ which gives $\sigma[1..] \models \varphi \mathsf{U} \psi$. Since $\sigma \models \varphi$, we have $\sigma \in \{\hat{\sigma} \in words(\varphi) | \hat{\sigma}[1..] \models \varphi \mathsf{W} \psi\}$. Therefore, $\sigma \in f(\varphi \mathsf{W} \psi)$.

2. and, that it is the greatest of all such fixpoints.
   Let $f(S) = S$. We will show that $S \subseteq words(\varphi \mathsf{W} \psi)$. Let $\sigma \in S$. Assume for the purpose of contradiction that $\sigma \not\models \varphi \mathsf{W} \psi$, i.e. $\sigma \models \neg(\varphi \mathsf{W} \psi)$, i.e. $\sigma \models (\varphi \wedge \neg\psi) \mathsf{U} (\neg\varphi \wedge \neg\psi)$. Then, there is $j \geq 0$ such that $\sigma[j..] \models \neg\varphi \wedge \neg\psi$ and forall $i < j$ we have $\sigma[i..] \models \varphi \wedge \neg\psi$. We will show by backward induction that $\sigma[k..] \notin S$ for all $k < j$.

   - **Case k=j**: $\sigma[j..] \not\models \psi$ and $\sigma[j..] \not\models \varphi$, so $\sigma[j..] \notin f(S)$, and hence, $\sigma[j..] \notin S$.
   - **Case k<j**: Assume by induction hypothesis that $\sigma[k+1..] \notin S$. Notice that $\sigma[k..] \models \neg\psi$, so $\sigma[k..] \notin words(\psi)$. In addition, $(\sigma[k..][1..] \notin S)$. Thus, $\sigma[k..] \notin f(S) = S$.

   By induction, $\forall k \leq j : \sigma[k..] \notin S$. In particular, $\sigma \notin S$.