Verification of Concurrent Programs

Proof of ME of third attempt

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Proof of ME of third attempt

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Notation: $p3..5$ means $p3 \vee p4 \vee p5$, which means statement $p3$ or $p4$ or $p5$ is the statement that will be executed next when process p is chosen.

Proof of ME of third attempt

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Lemma 4.3

$A: \ p3..5 \equiv want\,p$ is invariant.

Proof of ME of third attempt

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Lemma 4.3

$A: \ p3..5 \equiv want\,p$ is invariant.

Proof is by mathematical induction on the states.

The base case is the initial state.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$p3..5 \equiv want\,p$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$p3..5 \equiv wantp$$

$$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$p3..5 \equiv wantp$$

$$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$$

$$false \vee false \vee false \equiv false$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$p3..5 \equiv wantp$$

$$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$$

$$false \vee false \vee false \equiv false$$

$$= \quad \langle (3.26) \text{ Idempotency of } \vee,\ p \vee p \equiv p \rangle$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$p3..5 \equiv wantp$$

$$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$$

$$false \vee false \vee false \equiv false$$

$$= \quad \langle (3.26) \text{ Idempotency of } \vee,\ p \vee p \equiv p \rangle$$

$$false \equiv false$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$
\begin{aligned}
& p3..5 \equiv want\,p \\
= \quad & \langle \text{The initial state is } (p1, q1, false, false) \rangle \\
& false \vee false \vee false \equiv false \\
= \quad & \langle (3.26) \text{ Idempotency of } \vee,\ p \vee p \equiv p \rangle \\
& false \equiv false \\
= \quad & \langle (3.5)\ p \equiv p \rangle
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$
\begin{aligned}
& p3..5 \equiv want\,p \\
= \quad & \langle \text{The initial state is } (p1, q1, false, false) \rangle \\
& false \vee false \vee false \equiv false \\
= \quad & \langle (3.26) \text{ Idempotency of } \vee,\ p \vee p \equiv p \rangle \\
& false \equiv false \\
= \quad & \langle (3.5)\ p \equiv p \rangle \\
& true
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume A is true in *any* state.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume A is true in *any* state.

Prove that A is true in any *subsequent* state.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume A is true in *any* state.

Prove that A is true in any *subsequent* state.

Question: Execution of which statements can change

$$A: \ p3..5 \equiv want\,p?$$

Algorithm 4.1: Third attempt

boolean wantp ← false, wantq ← false

p		q	
	loop forever		loop forever
p1:	non-critical section	q1:	non-critical section
p2:	wantp ← true	q2:	wantq ← true
p3:	await wantq = false	q3:	await wantp = false
p4:	critical section	q4:	critical section
p5:	wantp ← false	q5:	wantq ← false

Induction case

Inductive hypothesis: Assume A is true in *any* state.

Prove that A is true in any *subsequent* state.

Question: Execution of which statements can change

$A: \ p3..5 \equiv want\,p$?

Answer: Only $p2$ and $p5$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p2$:
Execution of $p2$ makes $p3..5$ $true$, but it also makes $want\,p$ true.
So, A becomes $true \equiv true$, which remains $true$.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false 	q5: wantq ← false

Case $p2$:
Execution of $p2$ makes $p3..5$ $true$, but it also makes $want p$ true.
So, A becomes $true \equiv true$, which remains $true$.

Case $p5$:
Execution of $p5$ makes $p3..5$ $false$, but it also makes $want p$ false.
So, A becomes $false \equiv false$, which remains $true$. //

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Lemma 4.3
$p3..5 \equiv want\,p$ is invariant.

Similarly,
$q3..5 \equiv want\,q$ is invariant.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Theorem 4.4

$\neg(p4 \wedge q4)$ is invariant.

That is, Algorithm 4.1 ensures mutual exclusion.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Theorem 4.4
$\neg(p4 \wedge q4)$ is invariant.
That is, Algorithm 4.1 ensures mutual exclusion.

Proof by mathematical induction.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$$\neg(p4 \wedge q4)$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$\neg(p4 \wedge q4)$

= $\langle$The initial state is $(p1, q1, false, false)\rangle$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$\neg(p4 \wedge q4)$

= $\langle$The initial state is $(p1, q1, false, false)\rangle$

$\neg(false \wedge false)$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$\neg(p4 \wedge q4)$

= $\langle$The initial state is $(p1, q1, false, false)\rangle$

$\neg(false \wedge false)$

= $\langle$(3.26) Idempotency of $\wedge$, $p \wedge p \equiv p\rangle$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$\neg(p4 \wedge q4)$

$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$

$\neg(false \wedge false)$

$= \quad \langle (3.26) \text{ Idempotency of } \wedge,\ p \wedge p \equiv p \rangle$

$\neg false$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$\neg(p4 \wedge q4)$

$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$

$\neg(false \wedge false)$

$= \quad \langle (3.26) \text{ Idempotency of } \wedge,\ p \wedge p \equiv p \rangle$

$\neg false$

$= \quad \langle (3.13) \rangle$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Base case

$\neg(p4 \wedge q4)$

$= \quad \langle \text{The initial state is } (p1, q1, false, false) \rangle$

$\neg(false \wedge false)$

$= \quad \langle (3.26) \text{ Idempotency of } \wedge,\ p \wedge p \equiv p \rangle$

$\neg false$

$= \quad \langle (3.13) \rangle$

$true$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume $\neg(p4 \wedge q4)$ is true in *any* state.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume $\neg(p4 \wedge q4)$ is true in *any* state.

Prove that $\neg(p4 \wedge q4)$ is true in any *subsequent* state.

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume $\neg(p4 \wedge q4)$ is true in *any* state.

Prove that $\neg(p4 \wedge q4)$ is true in any *subsequent* state.

Question: Execution of which statements can make

$\neg(p4 \wedge q4)$ false?

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Induction case

Inductive hypothesis: Assume $\neg(p4 \wedge q4)$ is true in *any* state.

Prove that $\neg(p4 \wedge q4)$ is true in any *subsequent* state.

Question: Execution of which statements can make

$\neg(p4 \wedge q4)$ false?

Answer: Only $p3$ and $q3$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$p3$ executes

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$p3 \text{ executes}$$
$$\Rightarrow \quad \langle \text{Code inspection} \rangle$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$p3 \text{ executes}$$
$$\Rightarrow \quad \langle \text{Code inspection} \rangle$$
$$\neg wantq$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& p3 \text{ executes} \\
\Rightarrow \quad & \langle \text{Code inspection} \rangle \\
& \neg wantq \\
= \quad & \langle \text{Lemma } 4.3, q3..5 \equiv wantq \rangle
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& p3 \text{ executes} \\
\Rightarrow \quad & \langle \text{Code inspection} \rangle \\
& \neg wantq \\
= \quad & \langle \text{Lemma 4.3}, q3..5 \equiv wantq \rangle \\
& \neg q3..5
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& p3 \text{ executes} \\
\Rightarrow \quad & \langle \text{Code inspection} \rangle \\
& \neg wantq \\
= \quad & \langle \text{Lemma 4.3}, q3..5 \equiv wantq \rangle \\
& \neg q3..5 \\
= \quad & \langle (3.47\text{b}) \text{ De Morgan}, \neg(p \vee q) \equiv \neg p \wedge \neg q \rangle
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& p3 \text{ executes} \\
\Rightarrow \quad & \langle \text{Code inspection} \rangle \\
& \neg wantq \\
= \quad & \langle \text{Lemma 4.3}, q3..5 \equiv wantq \rangle \\
& \neg q3..5 \\
= \quad & \langle \text{(3.47b) De Morgan}, \neg(p \vee q) \equiv \neg p \wedge \neg q \rangle \\
& \neg q3 \wedge \neg q4 \wedge \neg q5
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$\neg q3 \wedge \neg q4 \wedge \neg q5$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$\neg q3 \wedge \neg q4 \wedge \neg q5$$

$$\Rightarrow \quad \langle (3.76\text{b}) \text{ strengthening}, p \wedge q \Rightarrow p \rangle$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$\begin{aligned} & \neg q3 \wedge \neg q4 \wedge \neg q5 \\ \Rightarrow \quad & \langle (3.76\text{b}) \text{ strengthening}, p \wedge q \Rightarrow p \rangle \\ & \neg q4 \end{aligned}$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$\begin{aligned} & \neg q3 \wedge \neg q4 \wedge \neg q5 \\ \Rightarrow \quad & \langle (3.76\text{b}) \text{ strengthening}, p \wedge q \Rightarrow p \rangle \\ & \neg q4 \\ \Rightarrow \quad & \langle (3.76\text{a}) \text{ weakening}, p \Rightarrow p \vee q \rangle \end{aligned}$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& \neg q3 \wedge \neg q4 \wedge \neg q5 \\
\Rightarrow \quad & \langle \text{(3.76b) strengthening}, p \wedge q \Rightarrow p \rangle \\
& \neg q4 \\
\Rightarrow \quad & \langle \text{(3.76a) weakening}, p \Rightarrow p \vee q \rangle \\
& \neg p4 \vee \neg q4
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& \neg q3 \wedge \neg q4 \wedge \neg q5 \\
\Rightarrow \quad & \langle (3.76\text{b}) \text{ strengthening}, p \wedge q \Rightarrow p \rangle \\
& \neg q4 \\
\Rightarrow \quad & \langle (3.76\text{a}) \text{ weakening}, p \Rightarrow p \vee q \rangle \\
& \neg p4 \vee \neg q4 \\
= \quad & \langle (3.47\text{a}) \text{ De Morgan } \neg(p \wedge q) \equiv \neg p \vee \neg q \rangle
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $p3$:

$$
\begin{aligned}
& \neg q3 \wedge \neg q4 \wedge \neg q5 \\
\Rightarrow \quad & \langle (3.76\text{b}) \text{ strengthening}, p \wedge q \Rightarrow p \rangle \\
& \neg q4 \\
\Rightarrow \quad & \langle (3.76\text{a}) \text{ weakening}, p \Rightarrow p \vee q \rangle \\
& \neg p4 \vee \neg q4 \\
= \quad & \langle (3.47\text{a}) \text{ De Morgan } \neg(p \wedge q) \equiv \neg p \vee \neg q \rangle \\
& \neg(p4 \wedge q4)
\end{aligned}
$$

Algorithm 4.1: Third attempt	
boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: await wantq = false	q3: await wantp = false
p4: critical section	q4: critical section
p5: wantp ← false	q5: wantq ← false

Case $q3$ is similar.

Conclusion: $\neg(p4 \wedge q4)$ is invariant
and Algorithm 4.1 ensures mutual exclusion.

A Calculational Deductive System for Linear Temporal Logic

J. STANLEY WARFORD, Pepperdine University, USA
DAVID VEGA, The Aerospace Corporation, USA
SCOTT M. STALEY, Ford Motor Company Research Labs (retired), USA

This article surveys the linear temporal logic (LTL) literature and presents all the LTL theorems from the survey, plus many new ones, in a calculational deductive system. Calculational deductive systems, developed by Dijkstra and Scholten and extended by Gries and Schneider, are based on only four inference rules—Substitution, Leibniz, Equanimity, and Transitivity. Inference rules in the older Hilbert-style systems, notably modus ponens, appear as theorems in this calculational deductive system. This article extends the calculational deductive system of Gries and Schneider to LTL, using only the same four inference rules. Although space limitations preclude giving a proof of every theorem in this article, every theorem has been proved with calculational logic.

CCS Concepts: • **Theory of computation → Modal and temporal logics**;

Additional Key Words and Phrases: Calculational logic, equational logic, linear temporal logic

ACM Reference format:
J. Stanley Warford, David Vega, and Scott M. Staley. 2020. A Calculational Deductive System for Linear Temporal Logic. *ACM Comput. Surv.* 53, 3, Article 53 (June 2020), 38 pages.
https://doi.org/10.1145/3387109