Bounded Partial-Order Reduction

Proof Companion Source Material

Katherine Coons∗ Madanlal Musuvathi† Kathryn McKinley∗†
University of Texas, Austin∗ Microsoft Research†

This companion source material contains the proofs for all the theorems in the main paper. For completeness, it repeats the definitions, theorems, and lemmas.

1. Definitions

Definition 1.1. Traces [1].
Equivalence classes of \( \equiv_\Lambda \) are traces over \( \Lambda \). The term \( [\omega] \) denotes the trace that contains the sequence of transitions \( \omega \).

Definition 1.2. Prefix([\omega]) [1].
Prefix([\omega]) returns the set containing all prefixes of all sequences in the Mazurkiewicz trace defined by \( \omega \).

Definition 1.3. Local sufficient.
A nonempty set \( T \subseteq T \) of transitions enabled in a state \( s \) in \( A_{G(Br,c)} \) is local sufficient in \( s \) if and only if for all sequences \( \omega \) of transitions from \( s \) in \( A_{G(Br,c)} \), there exists a sequence \( \omega' \) from \( s \) in \( A_{G(Br,c)} \) such that \( \omega \in \text{Prefix}([\omega']) \) and \( \omega' \in T \).

Definition 1.4. ext(s,t).
Given a state \( s = \text{final}(S) \) and a transition \( t \in \text{enabled}(s) \), ext(s,t) returns the unique sequence of transitions \( \beta \) from \( s \) such that
1. \( \forall i \in \text{dom}(\beta) : \beta_i.tid = t.tid \)
2. \( t.tid \notin \text{enabled}(\text{final}(S,\beta)) \)

1.1 Preemption-bounded search

Definition 1.5. Preemption bound [2].

\[
Pb(t) = \begin{cases} 
0 & \text{if } t.tid = 0 \\
Pb(S) + 1 & \text{if } t.tid \neq \text{last}(S).tid \text{ and } \text{last}(S).tid \in \text{enabled}\left(\text{final}(S)\right) \\
Pb(S) & \text{otherwise}
\end{cases}
\]

Definition 1.6. Preemption-bound persistent sets.
A set \( T \subseteq T \) of transitions enabled in a state \( s = \text{final}(S) \) is preemption-bound persistent in \( s \) if and only if for all nonempty sequences \( \alpha \) of transitions from \( s \) in \( A_{G(Pb,c)} \) such that \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \) and for all \( t \in T \),
1. \( Pb(S,t) \leq Pb(S,\alpha_1) \)
2. if \( Pb(S,t) < Pb(S,\alpha_1) \), then \( t \leftrightarrow \text{last}(\alpha) \) and \( t \leftrightarrow \text{next}\left(\text{final}(S,\alpha),\text{last}(\alpha).tid\right) \)
3. if \( Pb(S,t) = Pb(S,\alpha_1) \), then \( \text{ext}(s,t) \leftrightarrow \text{last}(\alpha) \) and \( \text{ext}(s,t) \leftrightarrow \text{next}\left(\text{final}(S,\alpha),\text{last}(\alpha).tid\right) \)

Definition 1.7. PC for Explore(S) - Preemption bound.
\( \forall \omega \exists s.t : \text{if } Pb(S,\omega) \leq c \text{ then } \text{Post}(S,\omega,\text{len}(S),u) \)

Definition 1.8. Post(S,k,u) - Preemption bound.
\( \forall \omega : \text{if } i = \max\{\{ i \in \text{dom}(S) | S_i \leftrightarrow \text{next}\left(\text{final}(S,\omega),u\right) \text{ and } S_i.tid = v \} \} \text{ then } \)
1. if \( i \leq k \) then
   if \( u \in \text{enabled}(\text{pre}(S,i)) \) then \( u \in \text{backtrack}(\text{pre}(S,i)) \)
   else \( \text{backtrack}(\text{pre}(S,i)) = \text{enabled}(\text{pre}(S,i)) \)
2. if \( j = \max\{\{ j \in \text{dom}(S) | j = 0 \text{ or } S_{j-1}.tid \neq S_j.tid \text{ and } j < i \}\} \) and \( j < k \) then
   if \( u \in \text{enabled}(\text{pre}(S,j)) \) then \( u \in \text{backtrack}(\text{pre}(S,j)) \)
   else \( \text{backtrack}(\text{pre}(S,j)) = \text{enabled}(\text{pre}(S,j)) \)

1.2 Fair-bounded search

Definition 1.9. Fair bound (Fb).
Let \( Y(S,u) \) return Thread \( u \)’s yield count in \( \text{final}(S) \).
\[
Fb(t) = \begin{cases} 
0 & \text{if } t.tid = 0 \\
Fb(S,t) = \max(Fb(S), \max_{u \in \text{enabled}(\text{final}(S))}(Y(S,t.tid) - Y(S,u))) & \text{otherwise}
\end{cases}
\]

Definition 1.10. Fair-bound persistent sets.
A set \( T \subseteq T \) of transitions enabled in a state \( s = \text{final}(S) \) is fair-bound persistent in \( s \) if and only if for all nonempty sequences \( \alpha \) of transitions from \( s \) in \( A_{G(Fb,c)} \) such that
\( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \) and for all \( t \in T \),
1. \( Fb(S,t) \leq c \)
2. if \( t \) is a release operation, then \( \forall u \in \text{enabled}(s) : \text{next}(s,u) \in T \)
3. \( t \leftrightarrow \text{last}(\alpha) \)

Definition 1.11. PC for Explore(S) - Fair bound.
\( \forall \omega : \text{if } Fb(S,\omega) \leq c \text{ and } \text{len}(S,\omega) \leq \text{MAX} \text{ then } \text{Post}(S,\omega,\text{len}(S),u) \)
Definition 1.12. Post \((S, k, u)\) - Fair bound.
\[\forall v : \text{if } i = \max\{i \in \text{dom}(S) \mid S_i \leftrightarrow \text{next(final}(S), u) \text{ and } S_i.tid = v\} \text{ and } i \leq k \text{ then} \]
\[\text{if } u \in \text{enabled}(pre(S, i)) \text{ and } S_i \text{ is not a release then } u \in \text{backtrack(pre(S, i))} \]
\[\text{else backtrack(pre(S, i))} = \text{enabled(pre(S, i))}\]

2. Proofs

Let \(A_{R_{(Bv,c)}}\) be the reduced state space explored by a selective search that explores a nonempty local sufficient set in each state.

Theorem 1. Let \(s\) be a state in \(A_{R_{(Bv,c)}}\) and let \(l\) be a local state reachable from \(s\) in \(A_{G_{(Bv,c)}}\) by a sequence \(\omega\) of transitions. Then, \(l\) is also reachable from \(s\) in \(A_{R_{(Bv,c)}}\).

Proof. The proof is by induction on the length of the longest sequence of transitions that leads to \(l\) from \(s\) in \(A_{G_{(Bv,c)}}\).

Case 1.1. Base Case.
For \(\text{len}(\omega) = 0\) the result is immediate.

Case 1.2. Inductive case.
Let \(l\) be a local state such that the longest sequence of transitions \(\omega\) from \(s\) to \(l\) has length \(n + 1\). Let \(u\) be a thread such that \(l = \text{local(final}(S, \omega), u)\). Let \(T\) be the nonempty local sufficient set explored from \(s\) in \(A_{R_{(Bv,c)}}\).

By Definition 1.3 of local sufficient sets, there exists a sequence \(\omega'\) of transitions from \(s\) in \(A_{G_{(Bv,c)}}\) such that \(\omega' \in \text{Prefix}(\omega)\). Thus, by Definition 1.2 of the prefix function, there exists a sequence \(\beta\) of transitions from final \((S, \omega)\) such that \(\omega, \beta \in [\omega']\). Assume that none of the transitions in \(\omega\) are by \(u\). Then, by definition of local states,

\[\text{local(final}(S, \omega), u) = \text{local(final}(S), u)\]

and the result is immediate.

Assume that a transition in \(\omega\) is by \(u\). Let \(i \in \text{dom}(\omega)\) be the maximum value of \(i\) such that \(\omega.i.tid = u\). Because \(\omega, \beta \in [\omega']\), there must exist \(j \in \text{dom}(\omega')\) such that \(\omega_j = \omega_i\). Let \(\omega' = \alpha.t.\gamma\) such that \(t = \omega'_j\). Because \(\omega, \beta \in [\omega']\),

\[\text{local(final}(S, \omega), u) = \text{local(final}(S, \alpha.t), u)\]

Thus, \(\omega'\) leads to \(l\). Because \(\omega'_i\) is in \(T\), it is explored from \(s\) and the state final \((S, \omega'_i)\) is reachable in \(A_{R_{(Bv,c)}}\). Because \(\omega\) is the longest sequence of transitions that leads to \(l\) in \(A_{G_{(Bv,c)}}\), len \((S, \alpha.t)\) \(\leq\) len \((\omega)\). Thus, from final \((S, \omega'_i), l\) is reachable via a sequence of transitions of length \(n\). By the inductive hypothesis, \(l\) is also reachable from \(s\) in \(A_{R_{(Bv,c)}}\).

\[\square\]

2.1 Preemption-bounded search

Let \(A_{R_{(Pb,c)}}\) be the reduced state space for a selective search that explores a preemption-bound persistent set in each state.

We provide two lemmas to manage the bound, and a theorem stating that a nonempty preemption-bound persistent set is local sufficient.

Lemma 2. Let \(\alpha\) and \(\beta\) be nonempty sequences of transitions from \(s = \text{final}(S)\) in \(A_{G_{(Pb,c)}}\) such that
1. \(\beta \leftrightarrow \alpha\)
2. \(Pb(S, \beta_1) \leq Pb(S, \alpha_1)\)
3. \(\forall i \in \text{dom}(\beta) : \beta_i.tid = \beta_1.tid\)
4. \(\beta \leftrightarrow \text{next(final}(S, \alpha_1 \ldots \alpha_i), \alpha_i.tid), 1 \leq i < \text{len}(\alpha)\)
5. if \(Pb(S, \beta_1) = Pb(S, \alpha_1)\), then \(\beta_1.tid \notin \text{enabled(final}(S, \beta))\)

Then, \(\beta, \alpha\) is a sequence of transitions from \(s\) in \(A_{G_{(Pb,c)}}\).

Proof. By Assumption 1, \(\beta, \alpha\) is a sequence of transitions from \(s\) to \(A_{G}\). For each preemption in \(S, \beta, \alpha\), from left to right, show that there exists a unique preemption in \(S, \alpha\). Assume that \(\beta_1\) requires a preemption from final \((S)\). By Assumption 2, \(\alpha_1\) also requires a preemption from final \((S)\). By Assumption 3, no transition in \(\beta\) after \(\beta_1\) requires a preemption.

Assume that \(\alpha_1\) requires a preemption from final \((S, \beta)\). Then,

\[\beta_1.tid \in \text{enabled(final}(S, \beta))\]

and thus by Assumptions 2 and 5, \(Pb(S, \beta_1) < Pb(S, \alpha_1)\). Thus, \(\alpha_1\) requires a preemption from final \((S)\) and \(\beta_1\) does not, so this preemption is unique. Assume that a transition \(\alpha_i, 2 \leq i \leq \text{len}(\alpha)\), requires a preemption in \(S, \beta, \alpha\). By Assumption 4, \(\alpha_i\) also requires a preemption in \(S, \alpha\). Thus, for each preemption in \(S, \beta, \alpha\) there exists a unique preemption in \(S, \alpha\) and

\[Pb(S, \beta_1) \leq Pb(S, \alpha) \leq c\]

Thus, \(\beta, \alpha\) is a sequence of transitions from \(s\) in \(A_{G_{(Pb,c)}}\).

\[\square\]

Lemma 3. Let \(T\) be a nonempty preemption-bound persistent set in a state \(s = \text{final}(S)\) in \(A_{R_{(Pb,c)}}\) and let \(\alpha, \beta, \gamma\) be a sequence of transitions from \(s\) in \(A_{G_{(Pb,c)}}\) such that \(\alpha\) and \(\beta\) are nonempty and
1. \(\forall i \in \text{dom}(\alpha) : \alpha_i \notin T\)
2. \(\beta_1 \in T\)
3. \(\forall i \in \text{dom}(\beta) : \beta_i.tid = \beta_1.tid\)
4. if \(Pb(S, \beta_1) < Pb(S, \alpha_1)\) then \(\text{len}(\beta) = 1\)
5. if \(Pb(S, \beta_1) = Pb(S, \alpha_1)\) and \(\gamma\) is empty, then \(\beta_1.tid \notin \text{enabled(final}(S, \beta))\)
6. if \(Pb(S, \beta_1) = Pb(S, \alpha_1)\) and \(\gamma\) is nonempty, then \(\gamma_1.tid \neq \beta_1.tid\)

Then, \(\beta, \alpha, \gamma\) is a sequence of transitions from \(s\) in \(A_{G_{(Pb,c)}}\).

Proof. By Assumptions 1-4 and by Requirements 2 and 3 of Definition 1.6 of preemption-bound persistent sets, \(\beta \leftrightarrow \alpha\).
Thus, $\beta, \alpha, \gamma$ is a sequence of transitions from $s$ in $A_G$. For each preemption in $S , \beta, \alpha, \gamma$, from left to right, show that there exists a unique preemption in $S, \alpha, \beta, \gamma$. Assume that $\beta_1$ requires a preemption from $\text{final}(S)$. Then, by Requirement 1 of Definition 1.6 of preemption-bound persistent sets, $\alpha_1$ also requires a preemption from $\text{final}(S)$. By Assumption 3, no transition in $\beta$ after $\beta_1$ requires a preemption.

Assume that $\alpha_1$ requires a preemption from $\text{final}(S, \beta)$. If $Pb(S, \beta_1) < Pb(S, \alpha_1)$, then $\alpha_1$ requires a preemption from $\text{final}(S)$ and $\beta_1$ does not, so this preemption is unique. Otherwise, by Requirement 1 of Definition 1.6 of preemption-bound persistent sets, $Pb(S, \beta_1) = Pb(S, \alpha_1)$. Because $\alpha_1$ requires a preemption from $\text{final}(S, \beta)$,

$$\beta_1.tid \in \text{enabled}(\text{final}(S, \beta))$$

By Assumption 5, $\gamma$ is nonempty, and by Assumption 6 $\gamma_1.tid \neq \beta_1.tid$. By Equation 2 and Requirement 3 of Definition 1.6 of preemption-bound persistent sets,

$$\beta_1.tid \in \text{enabled}(\text{final}(S, \alpha, \beta))$$

Thus, $\gamma_1$ requires a preemption from $\text{final}(S, \alpha, \beta)$. Assume that a transition $\alpha_i, 2 \leq i \leq \text{len}(\alpha)$, requires a preemption in $S, \beta, \alpha, \gamma$. By Equation 1, $\alpha_i$ also requires a preemption in $S, \alpha, \beta, \gamma$.

Assume that $\gamma_1$ requires a preemption from $\text{final}(S, \beta, \alpha)$. Then,

$$\text{last}(\alpha).tid \in \text{enabled}(\text{final}(S, \beta, \alpha))$$

By Equation 1,

$$\text{last}(\alpha).tid \in \text{enabled}(\text{final}(S, \alpha))$$

Because $\beta \leftrightarrow \alpha, \beta_1.tid \neq \text{last}(\alpha).tid$. Thus, $\beta_1$ requires a preemption from $\text{final}(S, \alpha)$. Assume that a transition $\gamma_i, 2 \leq i \leq \text{len}(\gamma)$, requires a preemption in $S, \beta, \alpha, \gamma$. Because $\beta \leftrightarrow \alpha, \text{final}(S, \beta, \alpha, \gamma_1) = \text{final}(S, \beta, \alpha, \gamma_1)$. Thus, by Definition 1.5 of the preemption bound, $\gamma_i$ also requires a preemption in $S, \alpha, \beta, \gamma$. Thus, for each preemption in $S, \beta, \alpha, \gamma$ there exists a unique preemption in $S, \alpha, \beta, \gamma$ and

$$Pb(S, \beta, \alpha, \gamma) \leq Pb(S, \alpha, \beta, \gamma) \leq c$$

Thus, $\beta, \alpha, \gamma$ is a sequence of transitions from $s$ in $A_G(Pb, e)$.

**Theorem 4.** If $T$ is a nonempty preemption-bound persistent set in a state $s$ in $A_{R(Pb, c)}$, then $T$ is local sufficient in $s$.

**Proof.** Let $s$ be a state in $A_{R(Pb, c)}$ and let $l$ be a local state reachable from $s$ in $A_G(Pb, e)$ via a nonempty sequence $\omega$ of transitions.

Case 4.1. $\forall i \in \text{dom}(\omega): \omega_i \notin T$.

Let $t$ be any transition in $T$. By Requirement 1 of Definition 1.6 of preemption-bound persistent sets, $Pb(S, t) \leq Pb(S, \omega)$.

Let $\beta = t$ if $Pb(S, t) < Pb(S, \omega)$, and let $\beta = \text{ext}(s, t)$ otherwise. Consider the sequence $\omega' = \beta, \omega$.

By Requirements 2 and 3 of Definition 1.6 of preemption-bound persistent sets, $\beta \leftrightarrow \omega$ and $\forall i \in \text{dom}(\omega): \beta \leftrightarrow \text{next}(\text{final}(S, \omega_1 \ldots \omega_i), \omega_i.tid)$. Thus, by Lemma 2 $\beta, \omega$ is a sequence of transitions from $s$ in $A_G(Pb, c)$ and by Definition 1.1 of a trace, $\omega, \beta \in [\omega']$. By Definition 1.2 of the prefix function, $\omega \in \text{Prefix}([\omega'])$. Thus, $T$ is local sufficient in $s$.

Case 4.2. $\exists i \in \text{dom}(\omega): \omega_i \in T$.

Let $\omega = \alpha, \beta, \gamma$ such that

1. $\forall i \in \text{dom}(\omega): \omega_i \notin T$
2. $\beta_i \in T$
3. $\forall i \in \text{dom}(\beta): \beta_i.tid = \beta_1.tid$
4. if $Pb(S, \beta_1) < Pb(S, \alpha_1)$ then $\text{len}(\beta) = 1$
5. if $Pb(S, \beta_1) = Pb(S, \alpha_1)$ and $\gamma$ is nonempty, then $\gamma_1.tid \neq \beta_1.tid$

Assume that $\alpha$ is empty. Then, $T$ is local sufficient in $s$ because $\omega_1 \in T$ and $l$ is reachable via $\omega$. Assume that $\alpha$ is nonempty. By Requirement 1 of Definition 1.6 of preemption-bound persistent sets, $Pb(S, \beta_1) \leq Pb(S, \alpha_1)$.

Case 4.2a. $\gamma$ is nonempty, or $\gamma$ is empty and $\beta_1.tid \notin \text{enabled}(\text{final}(S, \beta)), \text{or } Pb(S, \beta_1) < Pb(S, \alpha_1)$.

Consider the sequence $\omega' = \beta, \alpha, \gamma$, i.e., $\omega$ with $\beta$ moved to the beginning. By Requirements 2 and 3 of Definition 1.6 of preemption-bound persistent sets, $\beta \leftrightarrow \alpha$ and $\forall i \in \text{dom}(\alpha): \beta \leftrightarrow \text{next}(\text{final}(S, \alpha_1 \ldots \alpha_i), \alpha_i.tid)$. Thus, by Lemma 3 $\omega'$ is a sequence of transitions from $s$ in $A_G(Pb, c)$ and by Definition 1.1 of a trace $\omega' \in [\omega]$. By Definition 1.2 of the prefix function $\omega \in \text{Prefix}([\omega'])$, so $T$ is local sufficient in $s$.

Case 4.2b. $\gamma$ is empty, $\beta_1.tid \in \text{enabled}(\text{final}(S, \beta)), \text{and } Pb(S, \beta_1) = Pb(S, \alpha_1)$.

Let $\beta' = \text{ext}(s, \beta_1)$. Consider the sequence $\omega' = \beta', \alpha$.

By Requirement 3 of Definition 1.6 of preemption-bound persistent sets, $\beta' \leftrightarrow \alpha$ and $\forall i \in \text{dom}(\alpha): \beta' \leftrightarrow \text{next}(\text{final}(S, \alpha_1 \ldots \alpha_i), \alpha_i.tid)$. Thus, by Lemma 2 $\beta', \omega$ is a sequence of transitions from $s$ in $A_G(Pb, c)$ and by Definition 1.1 of a trace $\omega, \beta' \in [\omega']$. By Definition 1.2 of the prefix function $\omega \in \text{Prefix}([\omega'])$, so $T$ is local sufficient in $s$.

**Lemma 5.** Whenever a state $s = \text{final}(S)$ is backtracked by Algorithm 1, the set $T$ of transitions explored from $s$ is preemption-bound persistent in $s$, provided that postcondition $PC$ holds for every recursive call $\text{Explore}(S, t)$ for all $t \in T$. 

\[\square\]
Thus, by Line 3 of Algorithm 2,
bound
T
Proceed by contradiction. Assume that there exist transitions
Let
Add
10:  
13:  
11:  
12:  
13:  

Algorithm 1 BPOR with bound function \( Bv \) and bound \( c \)

1:  Initially, Explore\((e)\) from \( s_0 \)
2:  procedure Explore\((S)\) begin
3:    Let \( s = \text{final}(S) \)
4:      # Add backtrack points
5:      for all \((u \in Tid)\) do
6:        for all \((v \in Tid \mid v \neq u)\) do
7:          # Find most recent dependent transition
8:            if \((3i = \max\{i \in \text{dom}(S) \mid S_i \leftrightarrow \text{next}(s, u) \text{ and } S_i.tid = v\})\) then
9:              Backtrack\((S, i, u)\)
10: # Continue the search by exploring successor states
11:    Initialize\((S)\)
12:    while \((\exists u \in (\text{enabled}(s) \cap \text{backtrack}(s) \setminus \text{visited}))\) do
13:      add \( u \) to \( \text{visited} \)
14:    if \((Bv(S.next(s, u)) \leq c)\) then
15:      Explore\((S.next(s, u))\)

3. \( t \) is dependent with last(\( \alpha \)) or with next\( (\text{final}(S.\alpha), u) \)

Case 5.2. \( T \) violates Requirement 2.
Proceed by contradiction. Assume that there exists a nonempty sequence \( \alpha \) of transitions from \( s \) in \( A_{G(p,c)} \) and a transition \( t \in T \) such that, if we let \( u = \text{last}(\alpha).\text{tid} \):
1. \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \)
2. \( Pb(S.t) < Pb(S.\alpha_1) \)
3. \( t \) is dependent with last(\( \alpha \)) or with next\( (\text{final}(S.\alpha), u) \)

Let \( n = \text{len}(\alpha) \) and let \( \omega = \alpha_1 \ldots \alpha_{n-1} \), i.e., \( \alpha \) with its last transition removed. Let there be no prefixes of \( \alpha \) that also meet the criteria above, and thus
4. \( t \leftrightarrow \omega \) and \( \forall i \in \text{dom}(\omega) : \omega_i \leftrightarrow \text{next}(\text{final}(S.\omega), u) \)

Assume that \( t.\text{tid} = u \). Because \( t \leftrightarrow \omega \),

Thus, last(\( \alpha \)) is in \( T \) and we have a contradiction.

Assume that \( t.\text{tid} \neq u \). Let \( \omega' = \omega \) if \( t \) is dependent with last(\( \alpha \)) and let \( \omega' = \alpha \) if \( t \leftrightarrow \alpha \) and \( t \) is dependent with next\( (\text{final}(S.\alpha), u) \). Consider the postcondition

for the recursive call Explore\((S,t)\). By Lemma 2, \( t.\omega' \) is a sequence of transitions from \( s \) in \( A_{G(p,c)} \). Because \( t \leftrightarrow \omega' \), \( t \) is the most recent transition by \( t.\text{tid} \) that is dependent with next\( (\text{final}(S.\omega'), u) \). Thus, by Definition 1.8 of Post, either \( u \in \text{backtrack}(s) \), or \( \text{backtrack}(s) = \text{enabled}(s) \) and thus \( \alpha_1 \in T \). In either case, we have a contradiction.

Case 5.3. \( T \) violates Requirement 3.
Proceed by contradiction. Assume that there exists a nonempty sequence \( \alpha \) of transitions from \( s \) in \( A_{G(p,c)} \) and a transition \( t \in T \) such that, if we let \( u = \text{last}(\alpha).\text{tid} \) and let \( \beta = \text{ext}(s, t) \):
1. \( Pb(S.t) = Pb(S.\alpha_1) \)
2. \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \)
3. a transition in \( \beta \) is dependent with last(\( \alpha \)) or with next\( (\text{final}(S.\alpha), u) \)

Let \( n = \text{len}(\alpha) \), and let \( \omega = \alpha_1 \ldots \alpha_{n-1} \), i.e., \( \alpha \) with its last transition removed. Let there be no prefixes of \( \alpha \) that also meet the criteria above, and thus
4. \( \beta \leftrightarrow \omega \) and \( \forall i \in \text{dom}(\omega) : \beta_i \leftrightarrow \text{next}(\text{final}(S.\omega_1 \ldots \omega_i), \omega_i.\text{tid}) \)

Assume that \( \beta_1.\text{tid} = u \). Because \( \beta \leftrightarrow \omega \),

Thus, last(\( \alpha \)) is in \( T \) and we have a contradiction.

Assume that \( \beta_1.\text{tid} \neq u \). Let \( \beta_k \) be the last transition in \( \beta \) that is dependent with last(\( \alpha \)) or with next\( (\text{final}(S.\alpha), u) \).

Let \( \omega' = \omega \) if \( \beta_k \) is dependent with last(\( \alpha \)) and let \( \omega' = \alpha \) if \( \beta \leftrightarrow \alpha \) and \( \beta_k \) is dependent with next\( (\text{final}(S.\alpha), u) \).

Proof. Let \( T = \text{next}(s, u) \mid u \in \text{backtrack}(s) \). Show that if \( T \) violates any requirement in Definition 1.6 of preemption-bound persistent sets, then we have a contradiction.

Case 5.1. \( T \) violates Requirement 1.
Procede by contradiction. Assume that there exist transitions \( t \in T \) and \( t' \notin T \) such that \( t \) and \( t' \) are both enabled in \( s \) and \( Pb(S.t') < Pb(S.t) \). By Definition 1.5 of the preemption bound

Thus, by Line 3 of Algorithm 2, \( t'.\text{tid} \in \text{backtrack}(s) \) and thus \( t' \in T \), and we have a contradiction.
By Lemma 2, $\beta.\omega'$ is a sequence of transitions from $s$ in $A_{G(p,c)}$. Consider the postcondition
\[ \text{Post}(S,\beta.\omega', \text{len}(S) + 1, u) \]
for the recursive call $\text{Explore}(S,\beta_1)$. Because $\beta \leftrightarrow \omega'$, $\beta_k$ is the most recent transition by $\beta_1,\text{tid}$ that is dependent with $\text{next}((\text{final}(S,\beta.\omega'), u))$. Because $\text{Pb}(S,\beta_1) = \text{Pb}(S,\alpha_1)$, by Definition 1.5 of the preemption bound either $\beta_1,\text{tid} \neq \text{last}(S).\text{tid}$, or $S$ is empty. Because all transitions in $\beta$ are by the same thread, $\beta_1$ is the most recent such location to $\beta_k$. Thus, by Requirement 2 of Definition 1.8 of postcondition $\text{Post}$, either $u \in \text{backtrack}(s)$, or $\text{backtrack}(s) = \text{enabled}(s)$ and thus $\alpha_1 \in T$. In either case, we have a contradiction.

Thus, if postcondition $\text{PC}$ holds in each state $s$ that Algorithm 1 explores with the Backtrack procedure from Algorithm 2, then the set of transitions Algorithm 1 explores from $s$ is preemption-bound persistent in $s$.

Next, we prove that postcondition $\text{PC}$ holds in each state $s$ that Algorithm 1 explores. First, we prove a lemma that simplifies the inductive step. Lemma 6 differs from the similar lemma used in depth-bounded and context-bounded search because it must account for the more complex postcondition that preemption-bound search requires.

**Lemma 6.** Let $s = \text{final}(S)$ be a state in $A_{R(p,c)}$, let $\omega$ and $\omega'$ be nonempty sequences of transitions from $s$ in $A_{C(p,c)}$ such that $\text{Pb}(S,\omega') \leq \text{Pb}(S,\omega)$, and let $u$ be a thread such that

1. $\exists \beta : \omega,\beta \in [\omega']$ and $\beta \leftrightarrow \text{next}(\text{final}(S,\omega), u)$, or
2. $\exists \beta : \omega',\beta \in [\omega']$ and $\beta \leftrightarrow \text{next}(\text{final}(S,\omega), u)$

Then, $\text{Post}(S,\omega', \text{len}(S) + 1, u) \implies \text{Post}(S,\omega, \text{len}(S), u)$.

**Proof.** Because $\beta \leftrightarrow \text{next}(\text{final}(S,\omega), u)$,
\[
\text{next}(\text{final}(S,\omega), u) = \text{next}(\text{final}(S,\omega'), u)
\]
Assume that in Definition 1.8 of postcondition $\text{Post}$, $i \leq k$ for $\text{Post}(S,\omega, \text{len}(S), u)$. Then, $i$ and $j$ have the same values in $\text{Post}(S,\omega', \text{len}(S), u)$ that they have in $\text{Post}(S,\omega, \text{len}(S), u)$ because $\beta \leftrightarrow \text{next}(\text{final}(S,\omega), u)$.

Assume that $i > k$ for $\text{Post}(S,\omega, \text{len}(S), u)$. Because $\text{Pb}(S,\omega') \leq \text{Pb}(S,\omega_1)$, by Definition 1.5 of the preemption bound either $S$ is empty or $\omega_1.\text{tid} \neq \text{last}(S).\text{tid}$. Thus, $j \geq k$ for $\text{Post}(S,\omega, \text{len}(S), u)$, so Definition 1.8 of $\text{Post}$ does not require any backpointers in either case,
\[
\text{Post}(S,\omega', \text{len}(S), u) \implies \text{Post}(S,\omega, \text{len}(S), u) \quad (3)
\]
Because Requirement 1 of Definition 1.8 of $\text{Post}$ requires that $i \leq k$ and Requirement 2 of Definition 1.8 of $\text{Post}$ requires that $j < k$,
\[
\text{Post}(S,\omega', \text{len}(S) + 1, u) \implies \text{Post}(S,\omega', \text{len}(S), u)
\]
Thus, by Equation 3,
\[
\text{Post}(S,\omega', \text{len}(S) + 1, u) \implies \text{Post}(S,\omega, \text{len}(S), u)
\]

**Theorem 7.** Whenever a state $s = \text{final}(S)$ is backtracked during the search performed by Algorithm 1 in an acyclic state space, the postcondition $\text{Post}$ for $\text{Explore}(S)$ is satisfied, and the set $T$ of transitions explored from $s$ is preemption-bound persistent in $s$.

**Proof.** The proof is by induction on the order in which states are backtracked.

**Base case.**
Because the search is acyclic, is performed in depth-first order, and the preemption bound provides a zero-cost transition in each state, the first backtracked state must be a deadlock state in which no transition is enabled. Thus, the postcondition for the first backtracked state is
\[
\forall u : \text{Post}(S, \text{len}(S), u)
\]
and is directly established by Lines 4-7 in Algorithm 1.

**Inductive case.**
Assume that each recursive call to $\text{Explore}(S,t)$ satisfies its postcondition. By Lemma 5, $T$ is preemption-bound persistent in $s$. Show that $\text{Explore}(S)$ satisfies its postcondition for any sequence $\omega$ of transitions from $s$ in $A_{C(p,c)}$ and for any thread $u$.

**Case 7.1.** $\forall i \in \text{dom}(\omega) : \omega_i \notin T$ and $u \in \text{backtrack}(s)$.
Because $u \in \text{backtrack}(s)$, $\text{next}(s,u) \in T$. By Definition 1.5 of preemption-bound persistent sets, $\text{next}(s,u) \leftrightarrow \omega$, and thus
\[
\text{next}(\text{final}(S,\omega), u) = \text{next}(s,u)
\]
Thus, $\text{next}(\text{final}(S,\omega), u) \leftrightarrow \omega$, and $\text{Post}(S,\omega, \text{len}(S), u)$ if $\text{Post}(S,\text{len}(S), u)$. The latter is directly established by Lines 4-7 in Algorithm 1.

**Case 7.2.** $\forall i \in \text{dom}(\omega) : \omega_i \notin T$ and $u \notin \text{backtrack}(s)$.
Because $u \notin \text{backtrack}(s)$, $\text{next}(s,u) \notin T$. Let $t$ be any transition in $T$, and thus $t.\text{tid} \neq u$. Let $\beta = t$ if $\text{Pb}(S,t) < \text{Pb}(S,\omega)$, and let $\beta = \text{ext}(s,t)$ otherwise. Consider the sequence $\omega' = \beta.\omega$. By Definition 1.6 of preemption-bound persistent sets,

1. $\text{Pb}(S,t) \leq \text{Pb}(S,\omega_1)$
2. $\beta \leftrightarrow \omega$
3. $\forall i \in \text{dom}(\omega) : \beta \leftrightarrow \text{next}(\text{final}(S,\omega_1 \ldots \omega_i), \omega_i,\text{tid})$

By Lemma 2, $\omega'$ is a sequence of transitions from $s$ in $A_{C(p,c)}$. Because $\beta \leftrightarrow \omega, \beta, \omega \in [\omega']$. By the inductive hypothesis for the recursive call $\text{Explore}(S,t)$,
\[
\text{Post}(S,\omega', \text{len}(S) + 1, u)
\]

5

2013/8/19
Assume that next(final(S,ω′), u) is dependent with a transition in β. Because β ↔ ω, the most recent dependent transition to next(final(S,ω′), u) by β.tid must be in β. If βi is the most recent dependent transition, then by Requirement 1 of Definition 1.8 of Post either u ∈ backtrack(s), or backtrack(s) = enabled(s) and thus ωi ∈ T. If the most recent dependent transition is another transition in β, then Pb(S,βi) = Pb(S,ωi) because otherwise β would contain only a single transition, and thus either S is empty or last(S).tid ≠ βi.tid. Thus, j must be len(S) in Definition 1.8, and thus either u ∈ backtrack(s), or backtrack(s) = enabled(s) and thus ω1 ∈ T. In either case, we have a contradiction.

Assume that β ↔ next(final(S,ω′), u). Because βi.tid ≠ u, next(final(S,ω), u) = next(final(S,ω′), u) and

\[ β ↔ next(final(S,ω), u) \]

Thus, by Lemma 6 where ω,β ∈ [ω′]

Next(final(S,ω), u)

Case 7.3. \( \exists i \in dom(α) : α_i ∈ T \).

Let \( ω = α_i β_i γ_i \) such that

1. \( ∀ i \in dom(α) : α_i \notin T \)
2. \( β_i ∈ T \)
3. \( ∀ i \in dom(β) : β_i.tid = β_i.tid \)
4. if \( Pb(S,β_i) < Pb(S,α_i) \) then len(β) = 1
5. if \( Pb(S,β_i) = Pb(S,α_i) \) and \( γ_i.tid ≠ β_i.tid \)

Assume that \( α \) is empty. Then, \( ω_i ∈ T \) and by the inductive hypothesis,

\[ Post(S,ω, len(S) + 1, u) \]

Because Requirement 1 of Definition 1.8 of Post requires that \( i ≤ k \) and Requirement 2 of Definition 1.8 of Post requires that \( j < k \),

\[ Post(S,ω, len(S), u) \]

as required.

Assume that \( α \) is nonempty. By Requirement 1 of Definition 1.6 of preemption-bound persistent sets, \( Pb(S,β_i) ≤ Pb(S,α_i) \).

Case 7.3a. \( γ_i \) is nonempty, or \( γ_i \) is empty and \( β_i.tid \notin enabled(final(S,β_i)) \), or \( Pb(S,β_i) < Pb(S,α_i) \).

Consider the sequence \( ω = β_α γ_i \), i.e., \( ω \) with \( β \) moved to the beginning. By Requirements 2 and 3 of Definition 1.6 of preemption-bound persistent sets, \( β ↔ α \) and \( ∀ i \in dom(α) : β ↔ next(final(S,α_1 \ldots α_i), α_i.tid) \). Thus, by Definition 1.1 of a trace, \( ω \)′ is a sequence of transitions from \( s \) in \( A_{G(Pb,c)} \). By the inductive hypothesis for the recursive call \( Explore(S,β_i) \).

\[ Post(S,ω′, len(S) + 1, u) \]

and thus by Lemma 6 where \( β \) is empty and \( ω′ ∈ [ω] \),

\[ Post(S,ω, len(S), u) \]

Case 7.3b. \( γ = empty, β_i.tid ∈ enabled(final(S,β_i)), Pb(S,β_i) = Pb(S,α_i), and u ∈ backtrack(s) \).

Because \( γ \) is empty, \( ω = α, β_i \). Consider the sequence \( ω' = ω' \).

Assume that \( β_i.tid = u \), then next(final(S,ω), u) is a transition in \( ext(s,β_i) \) and by Requirement 3 of Definition 1.6 of preemption-bound persistent sets next(final(S,ω), u) ↔ α. In either case,

\[ next(final(S,ω), u) ↔ α \]

Because \( Pb(S,β_i) = Pb(S,α_i) \) and all transitions in \( β \) are by the same thread and thus do not require a preemption, \( ω' \) is a sequence of transitions from \( s \) in \( A_{G(Pb,c)} \). By the inductive hypothesis for the recursive call \( Explore(S,β_i) \).

\[ Post(S,ω′, len(S) + 1, u) \]

and thus by Lemma 6 where \( β = α \) and \( ω'.α ∈ ω \),

\[ Post(S,ω, len(S), u) \]

Case 7.3c. \( γ \) is empty, \( β_i.tid ∈ enabled(final(S,β_i)), Pb(S,β_i) = Pb(S,α_i), and u ∈ backtrack(s) \).

Because \( γ \) is empty, \( ω = α, β_i \). Let \( β' \) be the unique, nonempty sequence of transitions from final(S,β) such that \( β',β' ∈ ext(s,β_i) \). Consider the sequence \( ω = γ, β_α γ_i \).

By Requirement 3 of Definition 1.6 of preemption-bound persistent sets, \( β,β' ↔ α \) and \( ∀ i \in dom(α) : β,β' ↔ next(final(S,α_1 \ldots α_i), α_i.tid) \). Thus, by Lemma 2, \( ω' = ω' \) is a sequence of transitions from \( s \) in \( A_{G(Pb,c)} \). Because \( β,β' ↔ α \),

\[ ω',β' ∈ [ω′] \]

By the inductive hypothesis for \( Explore(S,β_i) \).

\[ Post(S,ω′, len(S) + 1, u) \]

Assume that next(final(S,ω′), u) is dependent with a transition in \( β' \). Then, because \( β,β' ↔ α \), the most recent dependent transition to next(final(S,ω′), u) by \( β_i.tid \) is in \( β' \).

Thus, by Definition 1.8 of Post, either \( u ∈ backtrack(s) \) or backtrack(s) = enabled(s) and thus \( ω_i ∈ T \). In either case, we have a contradiction.

Assume that \( β' ↔ next(final(S,ω′), u) \). Because \( β_i ∈ T \) and \( u ∈ backtrack(s) \), \( β_i.tid ≠ u \). Thus, it must be the case that \( next(final(S,ω), u) = next(final(S,ω′), u) \), and

\[ β' ↔ next(final(S,ω′), u) \]

Thus, by Lemma 6 where \( β = β' \) and \( ω,β' ∈ [ω′] \),

\[ Post(S,ω, len(S), u) \]
2.2 Fair-bounded search

Let \( A_{\overline{R}(F_b, c)} \) be the reduced state space explored by a selective search that explores a fair-bound persistent set in each state. We provide two lemmas to manage the bound, and a theorem stating that a nonempty fair-bound persistent set is local sufficient.

**Lemma 8.** Let \( \alpha \) be a nonempty sequence of transitions from \( s = \text{final}(S) \) in \( A_{G(F_b, c)} \) and let \( t \) be a transition enabled in \( s \) such that
1. \( F_b(S.t) \leq c \)
2. \( t \) is not a release operation
3. \( t \leftrightarrow \alpha \)

Then, \( t.\alpha \) is a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \).

**Proof.** Because \( t \leftrightarrow \alpha \), \( t.\alpha \) is a sequence of transitions from \( s \) in \( A_G \). Because \( t \) is not a release operation,

\[
\forall i \in \text{dom}(\alpha) : \text{enabled}(\text{final}(S.t.\alpha_1 \ldots \alpha_i)) \subseteq \text{enabled}(\text{final}(S.\alpha_1 \ldots \alpha_i))
\]

Thus, by Definition 1.9 of the fair bound, the transitions in \( \alpha \) cost no more in \( S.\alpha \) than they do in \( S.\alpha \). By Assumption 1, \( t \) is within the bound from \( s \). Thus, by Definition 1.9 of the fair bound,

\[
F_b(S.t.\alpha) \leq c
\]

and \( t.\alpha \) is a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \).

**Lemma 9.** Let \( T \) be a nonempty fair-bound persistent set in a state \( s = \text{final}(S) \) in \( A_{\overline{R}(F_b, c)} \) and let \( \alpha.t.\gamma \) be a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \) such that \( \alpha \) is nonempty, \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \), and \( t \in T \). Then, \( t.\alpha.\gamma \) is a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \).

**Proof.** By Requirement 3 of Definition 1.10 of fair-bound persistent sets, \( t \leftrightarrow \alpha \). Thus, \( t.\alpha.\gamma \) is a sequence of transitions from \( s \) in \( A_G \). By Requirements 1 and 2 of Definition 1.10 of fair-bound persistent sets, \( F_b(S.t) \leq c \) and \( t \) is not a release operation. Thus, by Lemma 8,

\[
F_b(S.t.\alpha) \leq F_b(S.\alpha)
\]

Assume that \( \gamma_1 \) exceeds the bound from \( \text{final}(S.t.\alpha) \), yet \( t \) does not exceed the bound from \( \text{final}(S.\alpha) \) and \( \gamma_1 \) does not exceed the bound from \( \text{final}(S.\alpha.t) \). Then, \( t \) must be a release operation that enables a transition \( t' \) such that \( t'.\text{tid} \) has a lower yield count than \( \gamma_1.\text{tid} \) has in \( \text{final}(S.t.\alpha) \), because otherwise \( \gamma_1 \) would also exceed the bound from \( \text{final}(S.\alpha) \). Because \( t \) is not a release operation, we have a contradiction. Thus,

\[
F_b(S.t.\alpha.\gamma_1) \leq c
\]

Because \( t \leftrightarrow \alpha \), \( \text{final}(S.t.\alpha.\gamma_1) = \text{final}(S.\alpha.t.\gamma_1) \) and thus each transition in \( \gamma \) executes from exactly the same state in

**Algorithm 3** BPOR procedures for fair-bounded search

1. **procedure Initialize** \((S)\) begin
   2. if \((\text{len}(S) > \text{MAX})\) then
      3. report livelock and exit
   4. **Backtrack** \((S, \text{len}(S), u)\) where \( u \) is a lowest cost enabled thread in \( \text{final}(S) \)
5. **procedure Backtrack** \((S, i, u)\) begin
   6. if \((u \in \text{enabled}(\text{pre}(S, i)) \) and \( \text{next}(\text{pre}(S, i, u) \) is not a release operation)\) then
      7. add \( u \) to **backtrack**\((\text{pre}(S, i))\)
   8. else
   9. **backtrack**\((\text{pre}(S, i))\) = enabled\((\text{pre}(S, i))\)

\( S.\alpha.\gamma \) as it does in \( S.\alpha.t.\gamma \). Thus, by Definition 1.9 of the fair bound,

\[
F_b(S.\alpha.\gamma) \leq c
\]

Thus, \( t.\alpha.\gamma \) is a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \).

**Theorem 10.** If \( T \) is a nonempty fair-bound persistent set in a state \( s \) in \( A_{\overline{R}(F_b, c)} \), then \( T \) is local sufficient in \( s \).

**Proof.** Let \( s \) be a state in \( A_{\overline{R}(F_b, c)} \) and let \( l \) be a local state reachable from \( s \) in \( A_{G(F_b, c)} \) via a nonempty sequence \( \omega \) of transitions.

**Case 10.1.** \( \forall i \in \text{dom}(\omega) : \omega_i \notin T \).

Let \( t \) be any transition in \( T \). Consider the sequence \( \omega' = t.\omega \).

By Requirement 3 of Definition 1.10 of fair-bound persistent sets, \( t \leftrightarrow \omega \). Thus, \( \omega.t \in [\omega'] \) and \( \omega \in \text{Prefix}(\omega') \).

By Requirements 1 and 2 of Definition 1.10 of fair-bound persistent sets, \( F_b(S.t) \leq c \) and \( t \) is not a release operation. Thus, by Lemma 8, \( t.\omega \) is a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \) and \( T \) is local sufficient in \( s \).

**Case 10.2.** \( \exists i \in \text{dom}(\omega) : \omega_i \notin T \).

Let \( \omega = \alpha.t.\gamma \) such that \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \) and \( t \in T \). Assume that \( \alpha \) is empty. Then, \( T \) is local sufficient in \( s \) because \( \omega_1 \in T \) and \( l \) is reachable via \( \omega \).

Assume that \( \alpha \) is nonempty. Consider the sequence \( \omega' = t.\omega \gamma, \i.e., \omega \) with \( t \) moved to the first position. By Requirement 3 of Definition 1.10 of fair-bound persistent sets, \( t \leftrightarrow \alpha \). Thus, \( \omega'.t \in [\omega] \) and \( \omega \in \text{Prefix}(\omega') \). By Lemma 9, \( t.\alpha.\gamma \) is a sequence of transitions from \( s \) in \( A_{G(F_b, c)} \), and \( T \) is local sufficient in \( s \).

**Lemma 11.** Whenever Algorithm 1 backtracks a state \( s = \text{final}(S) \), the set \( T \) of transitions explored from \( s \) is fair-bound persistent in \( s \), provided that postcondition PC holds for every recursive call **Explore**\((S.t)\) for all \( t \in T \).
Proof. Let $T = \text{next}(s, u)$ | $u \in \text{backtrack}(s)$. Show that if $T$ violates any requirement in Definition 1.10 of fair-bound persistent sets, then we have a contradiction.

Proceed by contradiction. Assume that for some $t \in T$, $Fb(S.t) > c$. By Line 12 in Algorithm 1, the search explores only transitions that do not exceed the bound from $s$. Thus, we have a contradiction.

Case 11.2. $T$ violates Requirement 2.
Proceed by contradiction. Assume that there exists a transition $t \in T$ such that $t$ is a release operation and a thread $u \in enabled(s)$ such that $\text{next}(s, u) \not\in T$. Because $t$ is a release operation Line 9 in Algorithm 3 must add it to backtrack$(s)$. Because $u \in enabled(s)$, Line 9 also adds $u$ to backtrack$(s)$ and thus $\text{next}(s, u) \in T$ and we have a contradiction.

Case 11.3. $T$ violates Requirement 3.
Proceed by contradiction. Assume that there exists a nonempty sequence $\alpha$ of transitions from $s$ in $A_{G(f_b, c)}$ such that $\forall i \in \text{dom}(\alpha): \alpha_i \not\in T$, and a transition $t \in T$ such that
1. $Fb(S.t) \leq c$
2. $t$ is not a release operation
3. $t$ is dependent with last($\alpha$)

Let $n = \text{len}(\alpha)$ and let $\omega = \alpha_1 \ldots \alpha_n$, i.e., $\alpha$ with its last transition removed. Let there be no prefixes of $\alpha$ that also meet the criteria above, and thus

3. $t \leftrightarrow \omega$

Let $u = \text{last}(\alpha).\text{tid}$. Assume that $t.\text{tid} = u$. Because $t \leftrightarrow \omega$,

$t = \text{next}(\text{final}(S, u)) = \text{next}(\text{final}(S.\omega, u)) = \text{last}(\alpha)$

Thus, last($\alpha$) $\in T$ and we have a contradiction.

Assume that $t.\text{tid} \neq u$. Consider the postcondition

$\text{Post}(S.t.\omega, \text{len}(S) + 1, u)$

for the recursive call $\text{Explore}(S.t)$. By Lemma 8, $t.\omega$ is a sequence of transitions from $s$ in $A_{G(f_b, c)}$. Because $t \leftrightarrow \omega$, $t$ is the most recent transition by $t.\text{tid}$ that is dependent with next($S.t.\omega$), $u$). Thus, by Definition 1.12 of Post, $u \in \text{backtrack}(s)$ and thus a transition in $\alpha$ must be in $T$ so we have a contradiction.

Therefore, if postcondition $\text{PC}$ holds in each state $s$ explored by Algorithm 1 with the $\text{Backtrack}$ procedure from Algorithm 3, then the set of transitions explored from $s$ is fair-bound persistent in $s$. Next, we prove that postcondition $\text{PC}$ holds in each state $s$ explored by Algorithm 1. First, we prove a lemma to simplify the inductive step.

Lemma 12. Let $s = \text{final}(S)$ be a state in $A_{R(f_b, c)}$, let $\omega$ and $\omega'$ be nonempty sequences of transitions from $s$ in $A_{G(f_b, c)}$, and let $u$ be a thread such that
1. $\exists \beta: \omega, \beta \in [\omega']$ and $\beta \leftrightarrow \text{next}(\text{final}(S.\omega), u)$, or
2. $\exists \beta': \omega', \beta \in [\omega]$ and $\beta \leftrightarrow \text{next}(\text{final}(S.\omega), u)$

Then, $\text{Post}(S.\omega', \text{len}(S) + 1, u) \implies \text{Post}(S.\omega, \text{len}(S), u)$.

Proof. Because $\beta \leftrightarrow \text{next}(\text{final}(S.\omega), u)$,

$\text{next}(\text{final}(S.\omega), u) = \text{next}(\text{final}(S.\omega'), u)$

Assume that in Definition 1.12 of Post($S.\omega, \text{len}(S), u$) for some thread $v$, $i > k$. Then, Post does not require any backtrack points for $v$.

Assume that for some thread $v$ in Definition 1.12 of Post($S.\omega, \text{len}(S), u$), $i \leq k$. Then, $i$ is the same for thread $v$ in Post($S.\omega', \text{len}(S), u$) because $\beta \leftrightarrow \text{next}(\text{final}(S.\omega), u)$. Because $i \leq \text{len}(S)$, the yield counts for all threads are the same in pre($S, i$), as well. Thus, by Definition 1.12 of Post,

$\text{Post}(S.\omega, \text{len}(S), u) \iff \text{Post}(S.\omega', \text{len}(S), u)$ (4)

Because Definition 1.12 of Post requires that $i$ be less than or equal to $k$,

$\text{Post}(S.\omega', \text{len}(S) + 1, u) \implies \text{Post}(S.\omega', \text{len}(S), u)$

Thus, by Equation 4,

$\text{Post}(S.\omega', \text{len}(S) + 1, u) \implies \text{Post}(S.\omega, \text{len}(S), u)$

Theorem 13. Whenever a state $s = \text{final}(S)$ is backtracked during the search performed by Algorithm 1, the postcondition Post for $\text{Explore}(S)$ is satisfied, and the set $T$ of transitions explored from $s$ is fair-bound persistent in $s$.

Proof. The proof is by induction on the order in which states are backtracked.

Base case.
If the stack depth exceeds MAX, then the search terminates and reports a livelock. Thus, the state space that the search may explore without reporting a livelock is a subset of the cyclic state space. Assume that the test does not contain a livelock. Because the search is performed in depth-first order, and the fair bound always provides a zero-cost transition, the first backtracked state must be a deadlock state in which no transition is enabled. Thus, the postcondition for the first backtracked state is

$\forall u: \text{Post}(S, \text{len}(S), u)$

and is directly established by Lines 4-7 in Algorithm 1.

Inductive case.
Assume that each call to $\text{Explore}(S,t)$ satisfies its postcondition. By Lemma 11, $T$ is fair-bound persistent in $s$. Show that $\text{Explore}(S)$ satisfies its postcondition for any sequence $\omega$ of transitions from $s$ in $A_G(Fb,c)$ and for any thread $u$. If $\omega$ is empty then the postcondition is directly established by Lines 4-7 in Algorithm 1, so assume that $\omega$ is nonempty.

**Case 13.1.** $\forall i \in \text{dom}(\omega) : \omega_i \notin T$ and $u \in \text{backtrack}(s)$. Because $u \in \text{backtrack}(s)$, $\text{next}(s,u) \in T$. Thus, by Requirement 3 of Definition 1.10 of fair-bound persistent sets, $\text{next}(s,u) \leftrightarrow \omega$, and thus

$$\text{next}(\text{final}(S,\omega), u) = \text{next}(s,u)$$

Thus, $\text{next}(\text{final}(S,\omega), u) \leftrightarrow \omega$, and thus $\text{Post}(S,\omega, \text{len}(S), u)$ iff $\text{Post}(S,\omega, \text{len}(S), u)$, the latter is directly established by Lines 4-7 in Algorithm 1.

**Case 13.2.** $\forall i \in \text{dom}(\omega) : \omega_i \notin T$ and $u \notin \text{backtrack}(s)$. Let $t$ be any transition in $T$. By Definition 1.10 of fair-bound persistent sets, $Fb(S,t) \leq c$ and $t \leftrightarrow \omega$. Because $\omega$ is nonempty and $\omega \notin T$, by Requirement 2 of Definition 1.10 of fair-bound persistent sets, $t$ is not a release operation. Thus, by Lemma 8, $\omega'$ is a sequence of transitions from $s$ in $A_G(Fb,c)$. Because $t \leftrightarrow \omega$,

$$\omega.t \in [\omega']$$

By the inductive hypothesis for $\text{Explore}(S,t)$,

$$\text{Post}(S,\omega', \text{len}(S) + 1, u)$$

If $t$ is dependent with $\text{next}(\text{final}(S,\omega'), u)$, then because $t \leftrightarrow \omega$, $\omega'$ must be the most recent dependent transition to $\text{next}(\text{final}(S,\omega'), u)$ by $t$.tid. Thus, by Definition 1.12 of $\text{Post}$, either $u \in \text{backtrack}(s)$ or $\text{backtrack}(s) = \text{enabled}(s)$, in which case $\omega \in T$. In either case, we have a contradiction. Thus, $t \leftrightarrow \text{next}(\text{final}(S,\omega'), u)$ and additionally, $t \leftrightarrow \text{next}(\text{final}(S,\omega), u)$. Thus, by Lemma 12 where $\beta = t$ and $\omega.t \in [\omega']$,

$$\text{Post}(S,\omega, \text{len}(S), u)$$

**Case 13.3.** $\exists i \in \text{dom}(\omega) : \omega_i \in T$.

Let $\omega = \alpha.t.\gamma$ such that

1. $\forall i \in \text{dom}(\alpha) : \alpha_i \notin T$
2. $t \in T$

Assume that $\alpha$ is empty. Then, $\omega_i \in T$, and by the inductive hypothesis

$$\text{Post}(S,\omega, \text{len}(S) + 1, u)$$

Thus, because Definition 1.12 of $\text{Post}$ requires that $i \leq k$,

$$\text{Post}(S,\omega, \text{len}(S), u)$$

as required.

Assume that $\alpha$ is nonempty. Consider the sequence $\omega' = t.\alpha.\gamma$, i.e., $\omega$ with $t$ moved to the beginning. By Definition 1.10 of fair-bound persistent sets, $Fb(S,t) \leq c$ and $t \leftrightarrow \alpha$. Thus, by Definition 1.1 of a trace,

$$\omega' \in [\omega]$$

By Lemma 9, $\omega'$ is a sequence of transitions from $s$ in $A_G(Fb,c)$. By the inductive hypothesis for the recursive call $\text{Explore}(S,t)$,

$$\text{Post}(S,\omega', \text{len}(S) + 1, u)$$

and thus by Lemma 12 where $\beta$ is empty and $\omega' \in [\omega]$,

$$\text{Post}(S,\omega, \text{len}(S), u)$$

□

**References**
