Bounded Partial-Order Reduction

Katherine Coons† Madanlal Musuvathi‡ Kathryn McKinley‡

University of Texas, Austin* Microsoft Research‡

1. Definitions

Definition 1.1. Traces [1].
Equivalence classes of \(\equiv_{\Lambda}\) are called 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 \(T \subseteq \mathcal{T}\) of transitions enabled in a state \(s\) in \(A_{G(Bv,c)}\) is local sufficient in \(s\) if and only if for all sequences \(\omega\) of transitions from \(s\) in \(A_{G(Bv,c)}\), there exists a sequence \(\omega'\) from \(s\) in \(A_{G(Bv,c)}\) such that \(\omega \in Prefix([\omega'])\) and \(\omega'_1 \in T\).

Definition 1.4. ext(\(s,t\)).
Given a state \(s = final(S)\) and a transition \(t \in enabled(s)\), ext(\(s,t\)) returns the unique sequence of transitions \(\beta\) from \(s\) such that

1. \(\forall i \in dom(\beta) : \beta_i.tid = t.tid\)
2. \(t.tid \notin enabled(final(S,\beta))\)

1.1 Preemption-bounded search

Definition 1.5. Preemption bound [2].

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

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

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

Definition 1.8. Post(\(S,k,u\)) - Preemption bound.
\(\forall v : \text{if } i = \max\{i \in dom(S) \mid S_i \leftrightarrow next(final(S),u) \text{ and } S_i.tid = v\} \text{ then }\)

1. if \(i \leq k\) then
   if \(u \in enabled(pre(S,i))\) then \(u \in backtrack(pre(S,i))\)
   else \(backtrack(pre(S,i)) = enabled(pre(S,i))\)
2. if \(j = \max\{j \in dom(S) \mid j = 0 \text{ or } S_{j-1}.tid \neq S_j.tid \text{ and } j \leq i\}\) and \(j \leq k\) then
   if \(u \in enabled(pre(S,j))\) then \(u \in backtrack(pre(S,j))\)
   else \(backtrack(pre(S,j)) = enabled(pre(S,j))\)

1.2 Fair-bounded search

Definition 1.9. Fair bound.

\[
Fb(S) = \max\{|Y(pre(S,i),S_i.tid) - Y(pre(S,i),v)| \mid i \in dom(S) \text{ and } v \in enabled(pre(S,i))\}
\]

Definition 1.10. Fair-bound persistent sets.
A set \(T \subseteq \mathcal{T}\) of transitions enabled in a state \(s = final(S)\) is fair-bound persistent in \(s\) iff for all threads \(u\) such that \(next(s,u) \notin T\),

1. \(Fb(S,t) \leq c\)
2. if \(T \neq enabled(s)\), then for all sequences \(\alpha\) of transitions from \(s\) in \(A_{G(Bv,c)}\) such that \(\forall i \in dom(\alpha), \alpha_i \notin T\) and for all transitions \(t \in T\), \(t \leftrightarrow next(final(S,\alpha),u)\)

Definition 1.11. PC for Explore(\(S\)) - Fair bound.
\(\forall u \forall \omega : \text{if } Fb(S,\omega) \leq c \text{ then } Post(S,\omega,\text{len}(S),u)\)

Definition 1.12. Post(\(S,k,u\)) - Fair bound.
\(\forall v : \text{if } i = \max\{i \in dom(S) \mid S_i \leftrightarrow next(final(S),u) \text{ and } S_i.tid = v\} \text{ and } i \leq k\) then

1. if \(u \in enabled(pre(S,i))\) then \(u \in backtrack(pre(S,i))\)
   else \(backtrack(pre(S,i)) = enabled(pre(S,i))\)

2. Proofs

Theorem 1. Let \(s\) be a state in \(A_{G(Bv,c)}\), 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 \(A_{G(Bv,c)}\).
Algorithm 1 Bounded selective search.

1: Initially, Explore(\{}\)
2: procedure Explore(S) begin
3: \[T = \text{Sufficient set}(\text{final}(S))\]
4: for all (t ∈ T) do
5: if (Bv(S,t) \leq c) then
6: Explore(S,t)

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.
Assume that the theorem holds for all sequences of transitions of length less than or equal to \(n > 0\), and prove that it holds for sequences of length \(n + 1\). Assume that \(l\) is reachable from \(s\) via a nonempty sequence \(\omega\) of transitions of length \(n + 1\) in \(A_{G(Bv,c)}\). Let \(\omega\) be the longest such sequence of transitions, and let \(l = \text{local}(\text{final}(S,\omega), u)\) for some thread \(u\).

Let \(T\) be the nonempty local sufficient set selected in \(s\) by Algorithm 1, i.e., the set of transitions explored from \(s\) in \(A_{G(Bv,c)}\). By Definition 1.3 of local sufficient sets, there exists a sequence \(\omega'\) of transitions from \(s\) in \(A_{G(Bv,c)}\) such that \(\omega' \in T\) and \(\omega \in \text{Prefix}([\omega'])\). Thus, by Definition 1.2 of the prefix function, there exists a sequence \(\beta\) of transitions from \(\text{final}(S,\omega)\) in \(A_{G(Bv,c)}\) such that \(\omega, \beta \in [\omega']\).

Assume that \(\beta\) is empty. Then, \(\omega \in [\omega']\) and

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

Thus, \(\omega'\) leads to \(l\). Because \(\omega' \in T\), it is explored from \(s\) and the state \(\text{final}(S,\omega')\) is reachable in \(A_{G(Bv,c)}\). From \(\text{final}(S,\omega')\), the longest path that leads to \(l\) in \(A_{G(Bv,c)}\) has length \(n\). Thus, by the inductive hypothesis, \(l\) is also reachable from \(s\) in \(A_{G(Bv,c)}\).

Assume that \(\beta\) is nonempty. Then, \(\beta_1.tid = u\), because otherwise \(\omega, \beta_1\) would also lead to \(l\) and be longer than \(\omega\). Because \(\omega, \beta \in [\omega']\), \(\beta_1\) is also a transition in \(\omega'\). Let \(\omega' = \alpha.\gamma\) such that \(\gamma_1 = \beta_1\). Because \(\omega, \beta \in [\alpha.\gamma]\), \(\gamma_1 = \beta_1\), and \(\beta_1.tid = u\),

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

Additionally, because \(\omega\) is the longest sequence that leads to \(l\),

\[\text{len}(\alpha) \leq \text{len}(\omega)\]

Because \(\alpha_1 \in T\), \(\alpha_1\) is explored from \(s\) and the state \(\text{final}(S,\alpha_1)\) is reachable in \(A_{G(Bv,c)}\). From \(\text{final}(S,\alpha_1)\), the longest path that leads to \(l\) in \(A_{G(Bv,c)}\) has length less than or equal to \(n\). Thus, by the inductive hypothesis, \(l\) is also reachable from \(s\) in \(A_{G(Bv,c)}\).

2.1 Preemption-bounded search
Assume that in each state of the reduced state space \(A_{R(Pb,c)}\), Algorithm 1 returns a preemption-bound persistent set. 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. \(\beta \leftrightarrow \text{next}(\text{final}(S,\alpha_1 \ldots \alpha_i), \alpha_i, \text{tid}), 1 \leq i \leq \text{len}(\alpha) - 1\)
5. if \(Pb(S,\beta_1) = Pb(S,\alpha_1)\), then \(\beta_1.tid \notin \text{enabled}(\text{final}(S,\beta))\)

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 \(s\) in \(A_G\). For each preemption in \(S, \beta, \alpha\), from left to right, show that there exists a unique preemption in \(S\).

Assume that \(\beta_1\) requires a preemption from \(\text{final}(S)\). By Assumption 2, \(\alpha_1\) 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 \(\text{final}(S,\beta)\).

Then,

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

and thus by Assumptions 2 and 5, \(Pb(S,\beta_1) < Pb(S,\alpha_1)\). Thus, \(\alpha_1\) requires a preemption from \(\text{final}(S)\) 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 \(S\).

Thus, for each preemption in \(S, \beta, \alpha\), there exists a unique preemption in \(S, \alpha\) and

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

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

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 \(\beta_1.tid \notin \text{enabled}(\text{final}(S,\beta))\)
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 $A_{G(P_b,c)}$.

Proof. By Assumptions 1-4 and by Requirements 2 and 3 of Definition 1.6 of preemption-bound persistent sets, $\beta \leftrightarrow \alpha$ and

$$\forall i \in \text{dom}(\alpha) : \beta \leftrightarrow \text{next}(final(S,\alpha_1 \ldots \alpha_i),\alpha_i.tid) \quad (1)$$

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 $\text{final}(S.\beta)$,

$$\beta_1.tid \in \text{enabled}(\text{final}(S.\beta)) \quad (2)$$

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 $\beta \leftrightarrow \alpha$, $\text{final}(S.\alpha.\beta.\gamma_1) = \text{final}(S.\beta.\alpha.\gamma_1)$. 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 $A_{G(P_b,c)}$.

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

Proof. Let $s$ be a state in $A_{R(P_b,c)}$ and let $l$ be a local state reachable from $s$ in $A_{G(P_b,c)}$ 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, $Pb(S,t) \leq Pb(S,\omega_1)$. Let $\beta = t$ if $Pb(S,t) < Pb(S,\omega_1)$, 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 $\forall i \in \text{dom}(\omega) : \beta \leftrightarrow \text{next}(final(S.\omega_1 \ldots \omega_i),\omega_i.tid)$. Thus, $\omega,\beta,\gamma \in [\omega']$, and by Lemma 2, $\beta.\omega$ is a sequence of transitions from $s$ in $A_{G(P_b,c)}$. 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. $\beta_1 \in T$
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 nonempty, then $\gamma_1.tid \neq \beta_1.tid$

Assume that $\alpha$ is empty. Then, $T$ is local sufficient in $s$ because $\omega_1 \subset T$ and $l$ is reachable via $S$. 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 $\beta_1.tid \notin \text{enabled}(\text{final}(S.\beta))$, 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 $\forall i \in \text{dom}(\alpha) : \beta \leftrightarrow \text{next}(final(S.\alpha_1 \ldots \alpha_i),\alpha_i.tid)$. Thus, by Definition 1.1 of a trace, $\omega' \in [\omega]$. By Lemma 3, $\omega'$ is a sequence of transitions from $s$ in $A_{G(P_b,c)}$, and $T$ is local sufficient in $s$.

Case 4.2b. $\gamma$ is empty, $\beta_1.tid \in \text{enabled}(\text{final}(S.\beta))$, 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 $\forall i \in \text{dom}(\alpha) : \beta' \leftrightarrow \text{next}(final(S.\alpha_1 \ldots \alpha_i),\alpha_i.tid)$. Thus, $\omega',\beta' \in [\omega']$, $\omega' \in \text{Prefix}([\omega'])$, and by Lemma 2, $\beta'.\omega'$ is a sequence of transitions from $s$ in $A_{G(P_b,c)}$. Thus, $T$ is local sufficient in $s$.

Lemma 5. Whenever a state $s = \text{final}(S)$ is backtracked by Algorithm 2, 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 $t \in T$. 

3 2013/8/28
Thus, by Line 3 of Algorithm 3, \( t'.tid \in \text{backtrack}(s) \) and thus \( t' \in T \), and we have a contradiction.

Case 5.2. \( T \) violates Requirement 2.
Proceed by contradiction. Assume that there exists a nonempty sequence \( \alpha \) of transitions from \( s \) in \( A_{G(P_b,c)} \) such that \( u = \text{last}(\alpha).tid \), and a transition \( t \in T \) such that:
1. \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \)
2. \( P_b(S,t) < P_b(S,\omega_1) \)
3. \( t \) is dependent with last(\( \alpha \)) or with \( \text{next}(\text{final}(S,\alpha),u) \)

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 \( \forall i \in \text{dom}(\omega) : t \leftrightarrow \text{next}(\text{final}(S,\omega_1 \ldots \omega_i),\omega_i.tid) \)
Assume that \( t.tid = u \). 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 \( t.tid \neq u \). Let \( \omega' = \omega \) if \( t \) is dependent with last(\( \alpha \)), and let \( \omega' = \alpha \) if \( t \leftrightarrow \alpha \) and \( t \) is dependent with \( \text{next}(\text{final}(S,\alpha),u) \).
Consider the postcondition
\[
\text{Post}(S.t.\omega',\text{len}(S) + 1,u)
\]
for the recursive call \( \text{Explore}(S,t) \). By Lemma 2, \( t.\omega' \) is a sequence of transitions from \( s \) in \( A_{G(P_b,c)} \). Because \( t \leftrightarrow \omega' \), \( t \) is the most recent transition by \( t.tid \) that is dependent with \( \text{next}(\text{final}(S.t.\omega'),u) \). 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 \( A_{G(P_b,c)} \) where \( u = \text{last}(\alpha).tid \), and a transition \( t \in T \) where \( \beta = \text{ext}(s,t) \) such that:
1. \( P_b(S,t) = P_b(S,\omega_1) \)
2. \( \forall i \in \text{dom}(\alpha) : \alpha_i \notin T \)
3. a transition in \( \beta \) is dependent with last(\( \alpha \)) or with \( \text{next}(\text{final}(S,\alpha),u) \)

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 \( \forall i \in \text{dom}(\omega) : t \leftrightarrow \text{next}(\text{final}(S,\omega_1 \ldots \omega_i),\omega_i.tid) \)
Assume that \( \beta_1.tid = u \). Because \( t \leftrightarrow \omega \),
\[
  \beta_1 = \text{next}(\text{final}(S),u) = \text{next}(\text{final}(S,\omega),u) = \text{last}(\alpha)
\]
Thus, last(\( \alpha \)) \( \in T \) and we have a contradiction. Assume that \( \beta_1.tid \neq u \). Let \( \beta_k \) be the last transition in \( \beta \) that is dependent with last(\( \alpha \)) or with \( \text{next}(\text{final}(S,\alpha),u) \). 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 \( \text{next}(\text{final}(S,\alpha),u) \).

---

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

Algorithm 3 BPOR procedures for preemption-bounded search.
1: procedure Initialize(S) begin
2:   if \((\text{last}(S).tid \in \text{enabled}(\text{final}(S)))\) then
3:     add \( \text{last}(S).tid \) to \( \text{backtrack}(\text{final}(S)) \)
4: else
5:     add any \( u \in \text{enabled}(\text{final}(S)) \) to \( \text{backtrack}(\text{final}(S)) \).
6: procedure Backtrack(S,i,u) begin
7:   AddBacktrackPoint(S,i,u)
8:   if \((j = \max\{j \in \text{dom}(S) \mid j = 0 \text{ or } S_{j-1}.tid \neq S_j.tid \text{ and } j \leq i\})\) then
9:     AddBacktrackPoint(S,i,u)
10:    procedure AddBacktrackPoint(S,i,u) begin
11:       if \((u \in \text{enabled}(pre(S,i)))\) then
12:         add \( u \) to \( \text{backtrack}(pre(S,i)) \)
13:       else
14:         \( \text{backtrack}(pre(S,i)) = \text{enabled}(pre(S,i)) \)

Proof. Let \( T = \text{next}(s,u) \) \( | u \in \text{backtrack}(s) \). 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.
Proceed 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 \( P_b(S,t') < P_b(S,t) \). By Definition 1.5 of the preemption bound
\[
t'.tid = \text{last}(S).tid
\]
Thus, by Line 3 of Algorithm 3, \( t'.tid \in \text{backtrack}(s) \) 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

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

for the recursive call Explore$(S, \beta_1)$. Because $\beta \leftrightarrow \omega'$, $\beta_k$ is the most recent transition by $\beta_1 \text{tid}$ that is dependent with next$(\text{final}(S, \beta, \omega'), u)$, 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, $\beta_i$ is the most recent such location to $\beta_k$. Thus, by Requirement 2 of Definition 1.8 of postcondition 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 PC holds in each state $s$ that Algorithm 2 explores with the Backtrack procedure from Algorithm 3, then the set of transitions Algorithm 2 explores from $s$ is preemption-bound persistent in $s$. Next, we prove that postcondition PC holds in each state $s$ that Algorithm 2 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-bounded search requires.

**Lemma 6.** Let $s = \text{final}(S)$ be a state in $A_G(p,c)$, let $\omega$ and $\omega'$ be nonempty sequences of transitions from $s$ in $A_G(p,c)$ such that $Pb(S, \omega_1') \leq Pb(S, \omega_1)$, 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, $Post(S, \omega', \text{len}(S) + 1, u) \implies Post(S, \omega, \text{len}(S), u)$.

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

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

Additionally, $\omega$ contains a transition that is dependent with $\text{next}(\text{final}(S, \omega), u)$ if and only if $\omega'$ contains that same transition. Thus, the most recent transition that is dependent with $\text{next}(\text{final}(S, \omega), u)$ by each thread is the same for both $S, \omega$ and $S, \omega'$, and has the same location relative to $\text{len}(S)$.

Assume that $i > k$ in $Post(S, \omega, \text{len}(S), u)$. Because $Pb(S, \omega_1') \leq Pb(S, \omega_1)$, $\omega_1 \text{tid} \neq \text{last}(S) \text{tid}$. Thus, $j > k$, and Requirement 2 of Definition 1.8 of Post does not require any backtrack points. Assume that $i \leq k$. Then, $j$ is the same for both $S, \omega$ and $S, \omega'$. Thus,

$$Post(S, \omega', \text{len}(S), u) \implies Post(S, \omega, \text{len}(S), u) \quad (3)$$

Because Requirements 1 and 2 of Definition 1.8 of Post require that $i$ and $j$, respectively, are less than or equal to $k$,

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

Thus, by Equation 3,

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

\[ \Box \]

**Theorem 7.** Whenever a state $s = \text{final}(S)$ is backtracked during the search performed by Algorithm 2 in an acyclic state space, the postcondition Post for 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 is extensible, the first backtracked state must be a deadlock state in which no transition is enabled. Thus, the postcondition for the first backtracked state is

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

and is directly established by Lines 4-7 in Algorithm 2.

**Inductive case.**

Assume that each recursive call to Explore$(S,t)$ satisfies its postcondition. By Lemma 5, $T$ is preemption-bound persistent in $s$. Show that Explore$(S,t)$ satisfies its postcondition for any sequence $\omega$ of transitions from $s$ in $A_G(p,c)$ 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 $u \in \text{backtrack}(s)$, next$(s, u) \in T$. By Definition 1.5 of preemption-bound persistent sets, 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 therefore $Post(S, \omega, \text{len}(S), u)$ iff $Post(S, \text{len}(S), u)$. The latter is directly established by Lines 4-7 in Algorithm 2.

**Case 7.2.** $\forall i \in \text{dom}(\omega) : \omega_i \notin T$ and $u \notin \text{backtrack}(s)$.

Because $u \notin \text{backtrack}(s)$, next$(s, u) \notin T$. Let $t$ be any transition in $T$, and thus $\text{ttid} \neq u$. Let $\beta = t$ if $Pb(S,t) < Pb(S,\omega_1)$, and let $\beta = \text{ext}(s,t)$ otherwise. Consider the sequence $\omega' = \beta, \omega$. By Definition 1.6 of preemption-bound persistent sets,

1. $Pb(S, \omega_1') \leq Pb(S, \omega_1)$
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 $A_G(p,c)$. Because $\beta \leftrightarrow \omega$,

$$\omega, \beta \in [\omega']$$

\[ \Box \]
By the inductive hypothesis for the recursive call $\text{Explore}(S.\omega')$, $\beta$. By Requirement 3 of Definition 1.6 of preemption-bound persistent sets, $\beta \leftrightarrow \alpha$ and thus $w'.\alpha \in [\omega]$

Because $u \in \text{backtrack}(s)$, $\text{next}(s, u) \in T$ and $\text{next}(s, u) \leftrightarrow \alpha$. If $\beta_1.tid = u$, then $\text{next}(S.\omega, u)$ is a transition in $\text{ext}(s, \beta_1)$ and by Requirement 3 of Definition 1.6 of preemption-bound persistent sets $\text{next}(S.\omega, u) \leftrightarrow \alpha$. If $\beta_1.tid \neq u$, then $\text{next}(s, u) = \text{next}(S.\omega, u)$. In either case,

$\text{next}(S.\omega, u) \leftrightarrow \alpha$

Because $Pb(S.\beta_1) = Pb(S.\alpha_1)$ and all transitions in $\beta$ are by the same thread and thus do not require a preemption, $\omega'$ is a sequence of transitions from $s$ in $A_{G(p, e)}$. By the inductive hypothesis for the recursive call $\text{Explore}(S.\omega')$,

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

and thus by Lemma 6,

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

Case 7.3. $\exists \iota \in \text{dom}(\omega) : \omega_1 \in T$.

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

1. $\forall i \in \text{dom}(\alpha) : \alpha_i \notin T$
2. $\beta_1 \in T$
3. $\exists i \in \text{dom}(\beta) : \beta_i.tid = \beta_1.tid$
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, $\omega_1 \in T$ and by the inductive hypothesis,

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

Because Requirements 1 and 2 of Definition 1.8 of Post require that $i$ and $j$, respectively, are less than or equal to $k$,

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

as required. 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 7.3a. $\gamma$ is nonempty, or $\gamma$ is empty and $\beta_1.tid \notin \text{enabled}(\text{final}(S.\beta))$, 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 Definition 1.1 of a trace, $\omega' \in [\omega]$. By Lemma 3, $\omega'$ is a sequence of transitions from $s$ in $A_{G(p, e)}$. By the inductive hypothesis for the recursive call $\text{Explore}(S.\omega')$,

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

and thus by Lemma 6,

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

Case 7.3b. $\gamma$ is empty, $\beta_1.tid \in \text{enabled}(\text{final}(S.\beta))$, $Pb(S.\beta_1) = Pb(S.\alpha_1)$, and $u \in \text{backtrack}(s)$.

Because $\gamma$ is empty, $\omega = \alpha.\beta$. Consider the sequence $\omega' = \beta.\alpha.\gamma$. By Requirement 3 of Definition 1.6 of preemption-bound persistent sets, $\beta \leftrightarrow \alpha$ and thus $w'.\alpha \in [\omega]$

Because $u \in \text{backtrack}(s)$, $\text{next}(s, u) \in T$ and $\text{next}(s, u) \leftrightarrow \alpha$. If $\beta_1.tid = u$, then $\text{next}(S.\omega, u)$ is a transition in $\text{ext}(s, \beta_1)$ and by Requirement 3 of Definition 1.6 of preemption-bound persistent sets $\text{next}(S.\omega, u) \leftrightarrow \alpha$. If $\beta_1.tid \neq u$, then $\text{next}(s, u) = \text{next}(S.\omega, u)$. In either case,

$\text{next}(S.\omega, u) \leftrightarrow \alpha$

Because $Pb(S.\beta_1) = Pb(S.\alpha_1)$ and all transitions in $\beta$ are by the same thread and thus do not require a preemption, $\omega'$ is a sequence of transitions from $s$ in $A_{G(p, e)}$. By the inductive hypothesis for the recursive call $\text{Explore}(S.\omega')$,

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

and thus by Lemma 6,

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

Case 7.3c. $\gamma$ is empty, $\beta_1.tid \in \text{enabled}(\text{final}(S.\beta))$, $Pb(S.\beta_1) = Pb(S.\alpha_1)$, and $u \notin \text{backtrack}(s)$.

Because $\gamma$ is empty, $\omega = \alpha.\beta$. Let $\beta'$ be the unique, nonempty sequence of transitions from $\text{final}(S.\beta)$ such that $\beta.\beta' = \text{ext}(s, \beta_1)$. Consider the sequence $\omega' = \beta.\beta'.\alpha$. By Requirement 3 of Definition 1.6 of preemption-bound persistent sets, $\beta.\beta' \leftrightarrow \alpha$ and $\forall i \in \text{dom}(\alpha) : \beta.\beta' \leftrightarrow \text{next}(\text{final}(S.\alpha_1 \ldots \alpha_i), \alpha_i.tid)$. Thus, by Lemma 2, $\omega'$ is a sequence of transitions from $s$ in $A_{G(p, e)}$. Because $\beta.\beta' \leftrightarrow \alpha$,

$\omega'.\alpha \in [\omega]$

By the inductive hypothesis for the recursive call $\text{Explore}(S.\omega')$,

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

Assume that a transition in $\beta'$ is dependent with $\text{next}(\text{final}(S.\omega'), u)$. Then, by Definition 1.8 of Post, either $u \in \text{backtrack}(s)$ or $\text{backtrack}(s) = \text{enabled}(s)$ and thus $\omega_1 \in T$. In either case, we have a contradiction. Assume that $\beta' \leftrightarrow \text{next}(\text{final}(S.\omega'), u)$. Because $\beta_1 \in T$ and $u \notin \text{backtrack}(s)$, $\beta_1.tid \neq u$. Thus, $\text{next}(\text{final}(S.\omega), u) = \text{next}(\text{final}(S.\omega'), u)$, and

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

Thus, by Lemma 6,

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

□
2.2 Fair-bounded search

Assume that in each state of the reduced state space $A_{R(Fb,c)}$, Algorithm 1 returns a fair-bound persistent set. 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 $T$ be a nonempty fair-bound persistent set in a state $s = \text{final}(S)$, let $\alpha$ be a nonempty sequence of transitions from $s$ in $A_{G(Fb,c)}$ such that $\forall i \in \text{dom}(\alpha) : \alpha_i \notin T$, and let $t \in T$. Then, $t.\alpha$ is a sequence of transitions from $s$ in $A_{G(Fb,c)}$.

**Proof.** By Requirement 2 of Definition 1.10, $t \leftrightarrow \alpha$, and thus $t.\alpha$ is a sequence of transitions from $s$ in $A_G$. Assume that $t$ is a release operation that enables a transition $t'$ with lower yield count than $\alpha_i$, $i \in \text{dom}(\alpha)$ in $\text{final}(S.t.\alpha_1 \ldots \alpha_{i-1})$. Then,

$\begin{align*}
  t'.\text{tid} \notin \text{enabled}(\text{final}(S.\alpha_1 \ldots \alpha_{i-1})) \\
  t'.\text{tid} \in \text{enabled}(\text{final}(S.t.\alpha_1 \ldots \alpha_{i-1}))
\end{align*}$

Assume that $\text{next}(s,t'.\text{tid}) \in T$. By Requirement 2 of Definition 1.10 of fair-bound persistent sets, $\text{next}(s,t'.\text{tid}) \leftrightarrow \alpha$. Thus, $t'.\text{tid} \in \text{enabled}(\text{final}(S.\alpha_1 \ldots \alpha_{i-1}))$, and we have a contradiction. Assume that $\text{next}(s,t'.\text{tid}) \notin T$. By Requirement 2 of Definition 1.10 of fair-bound persistent sets, $t \leftrightarrow \text{next}(\text{final}(S.\alpha_1 \ldots \alpha_{i-1}),t'.\text{tid})$ and thus $t \leftrightarrow t'$, so $t$ cannot enable $t'$ and we have a contradiction. Thus, because $Fb(S.t) \leq c$ and $t$ cannot increase the cost of any of the transitions in $\alpha$, by Definition 1.9 of the fair bound,

$Fb(S.t.\alpha) \leq Fb(S.\alpha) \leq c$

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

**Lemma 9.** Let $T$ be a nonempty fair-bound persistent set in a state $s = \text{final}(S)$ in $A_{R(Fb,c)}$ and let $\alpha.t.\gamma$ be a sequence of transitions from $s$ in $A_{G(Fb,c)}$ 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 $A_{G(Fb,c)}$.

**Proof.** By Requirement 2 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 Lemma 8,

$Fb(S.t.\alpha) \leq Fb(S.\alpha)$

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 $\text{final}(S.\alpha)$. Thus,

$\begin{align*}
  t'.\text{tid} \notin \text{enabled}(\text{final}(S.\alpha)) \\
  t'.\text{tid} \in \text{enabled}(\text{final}(S.t.\alpha))
\end{align*}$

Assume that $\text{next}(s,t'.\text{tid}) \in T$. By Requirement 2 of Definition 1.10 of fair-bound persistent sets, $\text{next}(s,t'.\text{tid}) \leftrightarrow \alpha$. Thus, $t'.\text{tid} \in \text{enabled}(\text{final}(S.\alpha))$, and we have a contradiction.

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

**Proof.** Let $s$ be a state in $A_{R(Fb,c)}$ and let $l$ be a local state reachable from $s$ in $A_{G(Fb,c)}$ 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 $T$. By Requirement 1 of Definition 1.10 of fair-bound persistent sets, $Fb(S.t) \leq c$. Consider the sequence $\omega' = t.\omega$. By Requirement 2 of Definition 1.10 of fair-bound persistent sets, $t \leftrightarrow \omega$. Thus, $\omega.t \in [\omega']$, and $\omega \in \text{Prefix}([\omega'])$. By Lemma 8, $t.\omega$ is a sequence of transitions from $s$ in $A_{G(Fb,c)}$ and $T$ is local sufficient in $s$.

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

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 $\omega' = t.\alpha.\gamma$, i.e., $\omega$ with $t$ moved to the first position. By Requirement 2 of Definition 1.10 of fair-bound persistent sets, $t \leftrightarrow \alpha$. Thus, $\omega' \in \omega$ and $\omega \in \text{Prefix}(\omega')$. By Lemma 9, $t.\alpha.\gamma$ is a sequence of transitions from $s$ in $A_{G(Fb,c)}$, and $T$ is local sufficient in $s$.

**Lemma 11.** Whenever Algorithm 2 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 $\text{Explore}(S.t)$ for all $t \in T$.

**Proof.** Let $T = \text{next}(s,u) \mid u \in \text{backtrack}(s)$. Show that if $T$ violates any requirement in Definition 1.10 of fair-bound persistent sets, then we have a contradiction.
Algorithm 4 BPOR procedures for fair-bounded search.

1: procedure Initialize($S$) begin
2: add any minimum cost enabled thread $u$ to backtrack(final($S$))
3: procedure Backtrack($S$, $i$, $u$) begin
4: if (IsRelease($S$, $i$) or $u$ $\notin$ enabled(pre($S$, $i$))) then
5: backtrack(pre($S$, $i$)) = enabled(pre($S$, $i$))
6: else
7: add $u$ to backtrack(pre($S$, $i$))

Proceed by contradiction. Assume that $Fb(t)$ $> c$ and that Algorithm 2 explores $t$ from final($S$). By Line 12, Algorithm 2 explores each transition $t$ only if $t$ does not exceed the bound from final($S$). Thus, we have a contradiction.

Case 11.2. $T$ violates Requirement 2.
Proceed by contradiction. Assume that there exists a thread $u$ such that next($s$, $u$) $\notin T$, a sequence $\alpha$ of transitions from $S$ in $A_{G(Fb,c)}$ such that $\forall i \in dom(\alpha) : \alpha_i \notin T$, and a transition $t \in T$ such that
1. $Fb(t) \leq c$
2. $T \neq enabled(s)$ and $t$ is dependent with next(final($S$.\alpha), $u$)

Let $\alpha$ be the shortest sequence of transitions that meets these criteria, and thus
3. $\forall i \in dom(\alpha)$, for all threads $v$ such that next($s$, $v$) $\notin T$
   and for all $t' \in T$,
   if $T \neq enabled(s)$ then $t' \leftrightarrow$ next(final($S$.\alpha $\ldots \alpha_{i-1}$), $v$)

Assume that $t$.tid = $u$. Because $t \leftrightarrow \alpha$,
$$t = next(final(S), u) = next(final(S.\alpha), u)$$

Thus, next(final($S$.\alpha), $u$) $\in T$ and we have a contradiction. Assume that $t$.tid $\neq u$. Consider the postcondition
$$Post(S.t.\alpha, len(S) + 1, u)$$
for the recursive call Explore($S$, $t$). Assume that $t$ is a release operation that enables a transition $t'$ with lower yield count than $\alpha_i$, $i \in dom(\alpha)$, in final($S$.t.\alpha $\ldots \alpha_{i-1}$). Then,
$$t'.tid \notin enabled(final(S.\alpha_1 \ldots \alpha_{i-1}))$$
$$t'.tid \in enabled(final(S.\alpha_1 \ldots \alpha_{i-1}))$$

Because $t$ is dependent with $t'$, by Assumption 3 either next($s$, $t$.tid) $\notin T$, or $T = enabled(s)$. If $T = enabled(s)$ then we have a contradiction. If next($s$, $t$.tid) $\in T$ then by Assumption 3, next($s$, $t'.tid$) $\leftrightarrow \alpha$. Thus,
$$next(s, t'.tid) = next(final(S.\alpha), t'.tid) = t'$$
and thus $t' \in T$ and $t'.tid \in enabled(s)$. Because $t' \leftrightarrow \alpha$, however, and $t'.tid \notin enabled(final(S.\alpha))$, $t'.tid$ cannot be enabled in $s$, and we have a contradiction. Thus, $t$ cannot increase the cost of any of the transitions in $\alpha$, and thus $t.\alpha$ is a sequence of transitions from $s$ in $A_{G(Fb,c)}$. Because $t \leftrightarrow \alpha$, $t$ is the most recent transition by $t$.tid that is dependent with next(final($S$.t.\alpha), $u$). Thus, by Definition 1.12 of Post, either $u \in backtrack(s)$ and thus next($s$, $u$) $\in T$, or backtrack($s$) $= enabled(s)$. In either case, we have a contradiction.

Thus, if postcondition $PC$ holds in each state $s$ explored by Algorithm 2 with the Backtrack procedure from Algorithm 4, then the set of transitions explored from $s$ is fair-bound persistent in $s$. Next, we prove that postcondition $PC$ holds in each state $s$ explored by Algorithm 2. First, we provide a lemma to simplify the inductive step.

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

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

Proof. Because $\beta \leftrightarrow$ next(final($S$.\omega), $u$),
$$next(final(S.\omega), u) = next(final(S.\omega'), u)$$

Additionally, $\omega$ contains a transition that is dependent with next(final($S$.\omega), $u$) if and only if $\omega'$ contains that same transition. Thus, the most recent transition that is dependent with next(final($S$.\omega), $u$) by each thread is the same for both $S.\omega$ and $S.\omega'$, and has the same location relative to len($S$). Thus,
$$Post(S.\omega, len(S), u) \iff Post(S.\omega', len(S), u) \quad (4)$$

Because Definition 1.12 of Post requires that $i$ be less than or equal to $k$,
$$Post(S.\omega', len(S) + 1, u) \implies Post(S.\omega', len(S), u)$$

Thus, by Equation 4,
$$Post(S.\omega', len(S) + 1, u) \implies Post(S.\omega, len(S), u)$$

Theorem 13. Whenever a state $s$ = final($S$) is backtracked during the search performed by Algorithm 2 in an acyclic state space, the postcondition Post for 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.
Because the search is acyclic, is performed in depth-first order, and the fair bound is extensible, 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 2.

**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 \(u\).

**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 Definition 1.10 of fair-bound persistent sets, \(\text{next}(s,u) \leftrightarrow \omega\), and
\[
\text{next(final}(S,\omega), u) = \text{next}(s,u)
\]
Thus, \(\text{next(final}(S,\omega), u) \leftrightarrow \omega\), and therefore \(\text{Post}(S,\omega, \text{len}(S), u)\) iff \(\text{Post}(S,\text{len}(S), u)\). The latter is directly established by Lines 4-7 in Algorithm 2.

**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 \(T\). Consider the sequence \(\omega' = t.\omega\). By Definition 1.10 of fair-bound persistent sets, \(\text{Fb}(S,t) < \text{Fb}(S,\omega_1)\) and \(t \leftrightarrow \omega\). 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 the recursive call \text{Explore}(S,\omega'_1),
\[
\text{Post}(S,\omega', \text{len}(S) + 1, u)
\]
By Requirement 2 of Definition 1.10 of fair-bound persistent sets,
\[
t \leftrightarrow \text{next(final}(S,\omega), u)
\]
Thus, by Lemma 12,
\[
\text{Post}(S,\omega, \text{len}(S), u)
\]

**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, \(\omega_1 \in T\), 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, \(\text{Fb}(S,t) < \text{Fb}(S,\alpha_1)\) 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,\omega'_1),
\[
\text{Post}(S,\omega', \text{len}(S) + 1, u)
\]
and thus by Lemma 12,
\[
\text{Post}(S,\omega, \text{len}(S), u)
\]