In this work, we extend the context-aware idea of Boolean programs running by a fixed, finite number of threads and the ability to perform on-the-fly exploration and efficient SAT solvers.

Our main contributions include:
1. performing BWRA on-the-fly by operating directly on Boolean programs;
2. avoiding local state explosion with the aid of on-the-fly exploration and efficient SAT solvers;
3. optimizations to limit the size of obtained covering pre-images.

**Algorithm On-the-fly Backward Exploration**

**Input:** $B$: A Boolean program with the set of initial thread states $I$; $T_{fin}$: the set of target thread states $G$: An edge clause of the CFG constructed from $B$.

**Output:** $\Phi$ such that $T_{fin}$ reachable?

1. $\Phi := \emptyset$
2. for each $e\in\{1,\ldots,k\}$ do:
   1. $\Phi := \emptyset$
   2. $\Phi := \emptyset$
   3. $\Phi := \emptyset$
   4. $\Phi := \emptyset$
3. return $\Phi$

**Procedure Cov-Predecessors($\tau$)**

1. $\Phi := \emptyset$
2. for each $e\in\{1,\ldots,k\}$ do:
   1. $\Phi := \emptyset$
   2. $\Phi := \emptyset$
   3. $\Phi := \emptyset$
   4. $\Phi := \emptyset$
3. return $\Phi$

**Procedure Update-Counters($\ell, \ell'$)**

1. $Z := \emptyset$
2. $Z := \emptyset$
3. $Z := \emptyset$
4. $Z := \emptyset$
5. $Z := \emptyset$
6. $Z := \emptyset$
7. return $Z$