Overapproximating Program Paths using FOL Formula

Jan Strejček and Marek Trtík
Motivation

Our heuristic

Z3 performance

Experimental Results
Motivation

(1) Relax exact interleaving of paths through a loop.
(2) Express variables as functions of path counters.
Motivation

(1) Relax exact interleaving of paths through a loop.
(2) Express variables as functions of path counters.

> $2^{2^{32}}$ paths
Motivation

(1) Relax exact interleaving of paths through a loop.
(2) Express variables as functions of path counters.

> $2^{2^{32}}$ paths
(1) Relax exact interleaving of paths through a loop.
(1) Relax exact interleaving of paths through a loop.
(2) Express variables as functions of path counters.
Our heuristic

4

i < n

A[i] == 1

A[i] != 1

++a

++i

κ_1

κ_1 + i

κ_2

κ_2 + i

κ_1 + a

κ_1 + κ_2 + i

κ_1 + a

ϕ ⃗κ ≡ ∀τ_1 (0 ≤ τ_1 < κ_1 → ∃τ_2 (0 ≤ τ_2 ≤ κ_2 ∧ τ_1 + τ_2 + i < n ∧ A(τ_1 + τ_2 + i) = 1)) ∧ ∀τ_2 (0 ≤ τ_2 < κ_2 → ∃τ_1 (0 ≤ τ_1 ≤ κ_1 ∧ τ_1 + τ_2 + i < n ∧ A(τ_1 + τ_2 + i) ≠ 1))
Our heuristic

\[
i < n \\
A[i] == 1 \\
A[i] != 1 \\
++i \\
++a \\
i \rightarrow i \;\rightsquigarrow \; i \rightarrow i + 1 \\
a \rightarrow a \;\rightsquigarrow \; a \rightarrow a + 1
\]
Our heuristic

\[ i \rightarrow i \sim i \rightarrow i + 1 \]
\[ a \rightarrow a \sim a \rightarrow a + 1 \]

\[ i(\kappa_1) = \kappa_1 + i \]
Our heuristic

\[
i \rightarrow i \leadsto i \rightarrow i + 1
\]
\[
a \rightarrow a \leadsto a \rightarrow a + 1
\]
\[
i(\kappa_1) = \kappa_1 + i
\]
\[
a(\kappa_1) = \kappa_1 + a
\]
Our heuristic

\[
\begin{align*}
A[i] &= 1 & \Rightarrow & \quad i \rightarrow i \sim i \rightarrow i + 1 \\
A[i] &\neq 1 & \Rightarrow & \quad a \rightarrow a \sim a \rightarrow a \\
\end{align*}
\]

\[
\begin{align*}
i(\kappa_1) &= \kappa_1 + i & i(\kappa_2) &= \kappa_2 + i \\
a(\kappa_1) &= \kappa_1 + a & a(\kappa_2) &= a \\
\end{align*}
\]
Our heuristic

Merging counter functions.

\[ i(\kappa_1) = \kappa_1 + i \quad i(\kappa_2) = \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \quad a(\kappa_2) = a \]
\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]
Our heuristic

Merging counter functions.

\[
\begin{align*}
i(\kappa_1) &= \kappa_1 + i \\
i(\kappa_2) &= \kappa_2 + i \\
a(\kappa_1) &= \kappa_1 + a \\
a(\kappa_2) &= a \\
i(\kappa_1, \kappa_2) &= \kappa_1 + \kappa_2 + i \\
a(\kappa_1) &= \kappa_1 + a
\end{align*}
\]
Our heuristic

\[ i < n \]

\[ \text{A}[i] == 1 \]

\[ \text{A}[i] != 1 \]

\[ ++a \]

\[ ++i \]

\[ \kappa_1 = \kappa_1 + i \]

\[ \kappa_2 = \kappa_2 + i \]

\[ a(\kappa_1) = \kappa_1 + a \]

\[ a(\kappa_2) = a \]

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]

\[ a(\kappa_1) = \kappa_1 + a \]

\[ \varphi^\kappa \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 + i < n \land \text{A}(\tau_1 + \tau_2 + i) = 1)) \land \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 + i < n \land \text{A}(\tau_1 + \tau_2 + i) \neq 1)) \]
Our heuristic

\[ A[i] \neq 1 \]

\[ i < n \]

\[ A[i] = 1 \]

\[ ++i \]

\[ ++a \]

\[ i(\kappa_1) = \kappa_1 + i \quad i(\kappa_2) = \kappa_2 + i \]

\[ a(\kappa_1) = \kappa_1 + a \quad a(\kappa_2) = a \]

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]

\[ a(\kappa_1) = \kappa_1 + a \]

\[ \varphi^\kappa \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) = 1)) \land \]

\[ \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) \neq 1)) \]
Our heuristic

\[ i(\kappa_1) = \kappa_1 + i \quad i(\kappa_2) = \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \quad a(\kappa_2) = a \]

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \]

\[ \varphi^k \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) = 1)) \land \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) \neq 1)) \]
Our heuristic

\[
\begin{align*}
\phi^k & \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \\
& \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \\
& \quad \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) = 1)) \land \\
\forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \\
& \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \\
& \quad \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) \neq 1))
\end{align*}
\]

\[
\begin{align*}
i(\kappa_1) & = \kappa_1 + i \\
a(\kappa_1) & = \kappa_1 + a
\end{align*}
\]

\[
\begin{align*}
i(\kappa_2) & = \kappa_2 + i \\
a(\kappa_2) & = a
\end{align*}
\]

\[
\begin{align*}
i(\kappa_1, \kappa_2) & = \kappa_1 + \kappa_2 + i \\
a(\kappa_1) & = \kappa_1 + a
\end{align*}
\]
Our heuristic

\[ i < n \]

\[ A[i] == 1 \]

\[ A[i] != 1 \]

\[ ++a \]

\[ ++i \]

\[ i(\kappa_1) = \kappa_1 + i \]

\[ a(\kappa_1) = \kappa_1 + a \]

\[ i(\kappa_2) = \kappa_2 + i \]

\[ a(\kappa_2) = a \]

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]

\[ a(\kappa_1) = \kappa_1 + a \]

\[ \varphi^\vec{\kappa} \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \wedge \tau_1 + \tau_2 + i < n \wedge A(\tau_1 + \tau_2 + i) = 1)) \wedge \]

\[ \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \wedge \tau_1 + \tau_2 + i < n \wedge A(\tau_1 + \tau_2 + i) \neq 1)) \]
Our heuristic

\[ i(\kappa_1) = \kappa_1 + i \quad i(\kappa_2) = \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \quad a(\kappa_2) = a \]
\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \]

\[ \varphi^\kappa \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) = 1)) \land \]
\[ \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) \neq 1)) \]
Our heuristic

\[ i(\kappa_1) = \kappa_1 + i \]
\[ a(\kappa_1) = \kappa_1 + a \]
\[ a(\kappa_2) = a \]

\[ i(\kappa_2) = \kappa_2 + i \]

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \]

\[ \varphi^\widetilde{\kappa} \equiv \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) = 1)) \land \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 + i < n \land A(\tau_1 + \tau_2 + i) \neq 1)) \]
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
7. \( a > 12 \)
8. \( a > 12 \)
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
7. \( a > 12 \)

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \]
\[ \varphi \tilde{\kappa} \]
Our heuristic

1. $a = 0$
2. $i = 0$
3. $i \geq n$

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 + i \]
\[ a(\kappa_1) = \kappa_1 + a \]
\[ \varphi_{\vec{\kappa}} \]

7. $a > 12$
8.
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
7. \( a > 12 \)

\[
i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2
\]

\[
a(\kappa_1) = \kappa_1 + a
\]

\[
\phi \vec{\kappa}
\]
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
7. \( a > 12 \)

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 \]
\[ a(\kappa_1) = \kappa_1 + a \]
\[ \varphi \bar{\kappa} \]
Our heuristic

\[
i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2
\]

\[
a(\kappa_1) = \kappa_1
\]

\[
\varphi(\vec{\kappa})
\]
Our heuristic

1. $a = 0$
2. $i = 0$
3. $i \geq n$
7. $a > 12$

$i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2$

$a(\kappa_1) = \kappa_1$

$\varphi[\kappa[i/0, a/0]]$
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
7. \( a > 12 \)

\[
i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2
\]
\[
a(\kappa_1) = \kappa_1
\]
\[
\varphi(\kappa_1) = \kappa_1
\]
\[
\varphi[i/0, a/0]
\]

\[
\phi \equiv \exists \kappa_1 \ (\kappa_1 \geq 0 \land \exists \kappa_2 \ (\kappa_2 \geq 0 \land \varphi[i/0, a/0] \land \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12))
\]
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
7. \( a > 12 \)

\[ i(κ_1, κ_2) = κ_1 + κ_2 \]
\[ a(κ_1) = κ_1 \]
\[ ϕ^{κ}[i/0, a/0] \]

\[ \hat{ϕ} \equiv \exists κ_1 (κ_1 \geq 0 \land \exists κ_2 (κ_2 \geq 0 \land \varphi^{κ}[i/0, a/0] \land κ_1 + κ_2 \geq n \land κ_1 > 12)) \]
Our heuristic

1. $a = 0$
2. $i = 0$
3. $i \geq n$
4. $a > 12$

$i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2$

$a(\kappa_1) = \kappa_1$

$\varphi^\kappa[i/0, a/0]$

$\hat{\varphi} \equiv \exists \kappa_1 \ (\kappa_1 \geq 0 \land \exists \kappa_2 \ (\kappa_2 \geq 0 \land \varphi^\kappa[i/0, a/0] \land \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12))$
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i >= n \)
7. \( a > 12 \)

\[ i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2 \]
\[ a(\kappa_1) = \kappa_1 \]
\[ \varphi[\kappa[i/0, a/0] \]

\[ \hat{\varphi} \equiv \exists \kappa_1 (\kappa_1 \geq 0 \land \exists \kappa_2 (\kappa_2 \geq 0 \land \varphi[\kappa[i/0, a/0] \land \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12)) \]}
Our heuristic

1. \( a = 0 \)
2. \( i = 0 \)
3. \( i \geq n \)
4. \( a > 12 \)

\[
i(\kappa_1, \kappa_2) = \kappa_1 + \kappa_2
\]
\[
a(\kappa_1) = \kappa_1
\]
\[
\varphi[\kappa_i/0, a/0]
\]

\[
\hat{\phi} \equiv \exists \kappa_1 \ (\kappa_1 \geq 0 \land \exists \kappa_2 \ (\kappa_2 \geq 0 \land
\varphi[\kappa_i/0, a/0] \land
\kappa_1 + \kappa_2 \geq n \land
\kappa_1 > 12))
\]
Usage of the heuristic

- **Symbolic execution:**
  - $\text{PC} \leftarrow \text{true}$
Usage of the heuristic

- Symbolic execution:
  - $\text{PC} \leftarrow \hat{\phi}$
Usage of the heuristic

- **Symbolic execution:**
  - $PC \leftarrow \hat{\phi}$

- **DART:**
  - Initialization $= \text{random}$
Usage of the heuristic

- **Symbolic execution:**
  - \( PC \leftarrow \hat{\phi} \)

- **DART:**
  - Initialization = model of \( \hat{\phi} \)
Usage of the heuristic

- Symbolic execution:
  - PC $\leftarrow \hat{\varphi}$

- DART:
  - Initialization $=$ model of $\hat{\varphi}$
  - Next input $=$ model of $\varphi$
Usage of the heuristic

- **Symbolic execution:**
  - $\text{PC} \leftarrow \hat{\varphi}$

- **DART:**
  - Initialization = model of $\hat{\varphi}$
  - Next input = model of $\varphi \land \hat{\varphi}$
Usage of the heuristic

- **Symbolic execution:**
  - $\text{PC} \leftarrow \hat{\phi}$

- **DART:**
  - Initialization = model of $\hat{\phi}$
  - Next input = model of $\varphi \land \hat{\phi}$

- **Tools:**
  - **KLEE, EXE, PEX, SAGE**
$\exists \kappa_1 \ (\kappa_1 \geq 0 \land \exists \kappa_2 \ (\kappa_2 \geq 0 \land \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) = 1)) \land \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) \neq 1)) \land \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12)$
\begin{align*}
\exists \kappa_1 \ (\kappa_1 & \geq 0 \land \\
\exists \kappa_2 \ (\kappa_2 & \geq 0 \land \\
\forall \tau_1 \ (0 & \leq \tau_1 < \kappa_1 \rightarrow \\
& \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) = 1)) \land \\
\forall \tau_2 \ (0 & \leq \tau_2 < \kappa_2 \rightarrow \\
& \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) \neq 1)) \land \\
\kappa_1 + \kappa_2 & \geq n \land \kappa_1 > 12))
\end{align*}

(1) \( \hat{\varphi} \rightarrow true \mid \varphi(\emptyset) \lor \hat{\varphi} \)

(2) \( \varphi(V) \rightarrow \gamma(V) \mid \exists x \ (0 \leq x \land \psi(V \cup \{x\}) \land \varphi(V \cup \{x\})) \)

(3) \( \psi(V \cup \{y\}) \rightarrow true \mid \forall x \ (0 \leq x < y \rightarrow \rho(V \cup \{x, y\})) \land \psi(V \cup \{y\}) \)

(4) \( \rho(V \cup \{y\}) \rightarrow \varphi(V \cup \{y\}) \mid \exists x \ (0 \leq x \leq y \land \rho(V \cup \{x, y\})) \)
\[ \exists \kappa_1 \left( \kappa_1 \geq 0 \land \exists \kappa_2 \left( \kappa_2 \geq 0 \land \forall \tau_1 \left( 0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \left( 0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) = 1 \right) \right) \right) \land \forall \tau_2 \left( 0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \left( 0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) \neq 1 \right) \right) \land \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12 \) \]

(1) \( \hat{\varphi} \rightarrow true \mid \varphi(\emptyset) \lor \hat{\varphi} \)

(2) \( \varphi(V) \rightarrow \gamma(V) \mid \exists x \left( 0 \leq x \land \psi(V \cup \{x\}) \land \varphi(V \cup \{x\}) \right) \)

(3) \( \psi(V \cup \{y\}) \rightarrow true \mid \forall x \left( 0 \leq x < y \rightarrow \rho(V \cup \{x, y\}) \right) \land \psi(V \cup \{y\}) \)

(4) \( \rho(V \cup \{y\}) \rightarrow \varphi(V \cup \{y\}) \mid \exists x \left( 0 \leq x \leq y \land \rho(V \cup \{x, y\}) \right) \)
\[ \exists \kappa_1 \ (\kappa_1 \geq 0 \land \\
\exists \kappa_2 \ (\kappa_2 \geq 0 \land \\
\forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \\
\exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) = 1)) \land \\
\forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \\
\exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) \neq 1)) \land \\
\kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12) \]
\[ \exists \kappa_1 \ (\kappa_1 \geq 0 \land \exists \kappa_2 \ (\kappa_2 \geq 0 \land \forall \tau_1 \ (0 \leq \tau_1 < \kappa_1 \rightarrow \exists \tau_2 \ (0 \leq \tau_2 \leq \kappa_2 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) = 1)) \land \forall \tau_2 \ (0 \leq \tau_2 < \kappa_2 \rightarrow \exists \tau_1 \ (0 \leq \tau_1 \leq \kappa_1 \land \tau_1 + \tau_2 < n \land A(\tau_1 + \tau_2) \neq 1)) \land \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12) \]

(1) \( \hat{\phi} \longrightarrow true \mid \varphi(\emptyset) \lor \hat{\phi} \)

(2) \( \varphi(V) \longrightarrow \gamma(V) \mid \exists x \ (0 \leq x \land \psi(V \cup \{x\}) \land \varphi(V \cup \{x\})) \)

(3) \( \psi(V \cup \{y\}) \longrightarrow true \mid \forall x \ (0 \leq x < y \rightarrow \rho(V \cup \{x, y\})) \land \psi(V \cup \{y\}) \)

(4) \( \rho(V \cup \{y\}) \longrightarrow \varphi(V \cup \{y\}) \mid \exists x \ (0 \leq x \leq y \land \rho(V \cup \{x, y\})) \)
\[(\kappa_1 \geq 0 \land \kappa_1 \leq 25 \land \\
(\kappa_2 \geq 0 \land \kappa_2 \leq 25 \land \\
(0 \leq 0 < \kappa_1) \to \\
(0 \leq \tau_{2,0} \leq \kappa_2 \land 0 + \tau_{2,0} < n \land A(0 + \tau_{2,0}) = 1)) \land \\
\ldots \\
(0 \leq 24 < \kappa_1) \to \\
(0 \leq \tau_{2,24} \leq \kappa_2 \land 24 + \tau_{2,24} < n \land A(24 + \tau_{2,24}) = 1)) \land \\
(0 \leq 0 < \kappa_2) \to \\
(0 \leq \tau_{1,0} \leq \kappa_1 \land \tau_{1,0} + 0 < n \land A(\tau_{1,0} + 0) \neq 1)) \land \\
\ldots \\
(0 \leq 24 < \kappa_2) \to \\
(0 \leq \tau_{1,24} \leq \kappa_1 \land \tau_{1,24} + 24 < n \land A(\tau_{1,24} + 24) \neq 1)) \land \\
\kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12))\]
(\kappa_1 \geq 0 \land \kappa_1 \leq 25 \land \\
(\kappa_2 \geq 0 \land \kappa_2 \leq 25 \land \\

(0 \leq 0 < \kappa_1) \rightarrow \\
(0 \leq \tau_{2,0} \leq \kappa_2 \land 0 + \tau_{2,0} < n \land A(0 + \tau_{2,0}) = 1)) \land \\
\ldots \\

(0 \leq 24 < \kappa_1) \rightarrow \\
(0 \leq \tau_{2,24} \leq \kappa_2 \land 24 + \tau_{2,24} < n \land A(24 + \tau_{2,24}) = 1)) \land \\

(0 \leq 0 < \kappa_2) \rightarrow \\
(0 \leq \tau_{1,0} \leq \kappa_1 \land \tau_{1,0} + 0 < n \land A(\tau_{1,0} + 0) \neq 1)) \land \\
\ldots \\

(0 \leq 24 < \kappa_2) \rightarrow \\
(0 \leq \tau_{1,24} \leq \kappa_1 \land \tau_{1,24} + 24 < n \land A(\tau_{1,24} + 24) \neq 1)) \land \\

\kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12))
\( (\kappa_1 \geq 0 \land \kappa_1 \leq 25 \land \kappa_2 \geq 0 \land \kappa_2 \leq 25 \land \)

\[
(0 \leq 0 < \kappa_1) \rightarrow (0 \leq \tau_{2,0} \leq \kappa_2 \land 0 + \tau_{2,0} < n \land A(0 + \tau_{2,0}) = 1)) \land \\
\ldots \\
(0 \leq 24 < \kappa_1) \rightarrow (0 \leq \tau_{2,24} \leq \kappa_2 \land 24 + \tau_{2,24} < n \land A(24 + \tau_{2,24}) = 1)) \land \\
\]

\[
(0 \leq 0 < \kappa_2) \rightarrow (0 \leq \tau_{1,0} \leq \kappa_1 \land \tau_{1,0} + 0 < n \land A(\tau_{1,0} + 0) \neq 1)) \land \\
\ldots \\
(0 \leq 24 < \kappa_2) \rightarrow (0 \leq \tau_{1,24} \leq \kappa_1 \land \tau_{1,24} + 24 < n \land A(\tau_{1,24} + 24) \neq 1)) \land \\
\]

\( \kappa_1 + \kappa_2 \geq n \land \kappa_1 > 12)\)
### Experimental Results

<table>
<thead>
<tr>
<th>Test</th>
<th>PEX</th>
<th>Total</th>
<th>Build</th>
<th>Full</th>
<th>QF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>5.257</td>
<td>0.614</td>
<td>0.091</td>
<td>0.433</td>
<td>0.09</td>
</tr>
<tr>
<td>HW</td>
<td>25.05</td>
<td>1.608</td>
<td>0.400</td>
<td>0.998</td>
<td>0.21</td>
</tr>
<tr>
<td>HWM</td>
<td>fail</td>
<td>11.00</td>
<td>7.338</td>
<td>2.748</td>
<td>0.92</td>
</tr>
<tr>
<td>MatrIR</td>
<td>95.00</td>
<td>1.435</td>
<td>0.105</td>
<td>1.330</td>
<td>-</td>
</tr>
<tr>
<td>WinDriver</td>
<td>35.53</td>
<td>0.382</td>
<td>0.089</td>
<td>0.143</td>
<td>0.150</td>
</tr>
</tbody>
</table>

- Intel® Core™ i7 CPU 920 @ 2.67GHz, 6GB RAM, Windows 7 Professional 64-bit
- MS PEX 0.92.50603.1, MS Moles 1.0.0.0, MS Visual Studio 2008, MS .NET Framework v3.5 SP1
- MS Z3 SMT solver v3.2, and boost v1.42.0.
The heuristic computes a formula that is a necessary condition for reaching the target. We build the formula according to the following two relaxations:

- We relax an exact interleaving of paths through a loop.
- And we express variables as functions of path counters.
Conclusion

- The heuristic computes a formula that is a necessary condition for reaching the target. We build the formula according to the following two relaxations:
  - We relax an exact interleaving of paths through a loop.
  - And we express variables as functions of path counters.

- Computed formulae belong to a fragment of FOL expressible by a simple grammar. In this fragment each universally quantified variable is bound to an interval with a path counter as the upper bound.
The heuristic computes a formula that is a necessary condition for reaching the target. We build the formula according to the following two relaxations:

- We relax an exact interleaving of paths through a loop.
- And we express variables as functions of path counters.

Computed formulae belong to a fragment of FOL expressible by a simple grammar. In this fragment each universally quantified variable is bound to an interval with a path counter as the upper bound.

Z3 often performs purely on computed formulae, because of quantifiers. But structure of formulae allows to generate bounded quantifier free formulae, where Z3 performs very well.
Conclusion

- The heuristic computes a formula that is a necessary condition for reaching the target. We build the formula according to the following two relaxations:
  - We relax an exact interleaving of paths through a loop.
  - And we express variables as functions of path counters.

- Computed formulae belong to a fragment of FOL expressible by a simple grammar. In this fragment each universally quantified variable is bound to an interval with a path counter as the upper bound.

- Z3 often performs purely on computed formulae, because of quantifiers. But structure of formulae allows to generate bounded quantifier free formulae, where Z3 performs very well.

- We showed that results of the heuristic can be easily and directly used in tools based on either original symbolic execution or DART algorithm. And experimental results show a potential to improve performance of such tools.