Optimization-based Modeling and Analysis Techniques for Safety-Critical Software Verification

Mardavij Roozbehani Eric Feron

Laboratory for Information and Decision Systems
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

Workshop on Critical Research Areas in Aerospace Software,
Tuesday August 9^{th}, 2005, MIT

Outline

- Introduction
- Basic principles of automated software analysis
  —Computer programs as dynamical systems
  —Lyapunov functions as behavior certificates
- System specific models of computer programs
  —Mixed integer/linear systems
  —Linear systems with conditional switching
- Convex optimization of Lyapunov certificates
- Recursive search for system invariants to improve analysis
- Conclusion

Introduction and motivation

- Safety-critical software is becoming pervasive in aerospace systems, medical technology and embedded systems in general
- It is crucial to verify reliability and correctness of the embedded software. The very least to require is that the software must be free of run-time errors.
- Reliable software properties include but are not limited to:
  —Absence of overflow
  —Absence of 'array index out-of-bounds' errors
  —Termination in finite-time
- The above problems are undecidable but efficient algorithms that work reasonably well in practice can be developed.

Basic principles of automated software analysis

- A computer program can be viewed as a rule for iterative modification of operating memory, possibly in response to real-time inputs

- Dynamical systems representation of computer programs:
  - State space X with selected subsets:
  - $X_0 \subset X$ (initial states)
  - $X_\infty \subset X$ (terminal states)
  - Set-valued function $f : X \to 2^X$ is s.t. $f(x) \subseteq X_\infty, \forall x \in X_\infty$.
  - A Program/dynamical system $\mathcal{S}{=}\mathcal{S}(X, f, X_0, X_\infty)$ is the set of all sequences $\mathcal{X} = (x(0), x(1), ..., x(t), ...)$ of elements of X, satisfying $x(0) \in X_0$, $x(t+1) \in f(x(t))$, $\forall t \in \mathbb{Z}^+$.

Example

- Consider the program:

program $\mathcal{P}$:

$x_1 \geq 50; x_2 \leq 0;$
while $x_1 > x_2,$
  $x_1 = x_1 - 1;$
  $x_2 = x_2 + 1;$
end

$$\Longrightarrow \begin{cases} \mathcal{P}=\mathcal{S}\left(X, f, X_0, X_\infty\right) \\ X = \mathbb{R}^2 \\ X_0 = \left\{x | x \in \mathbb{R}^2, Hx \leq b\right\} \text{ where:} \\ H = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } b = \begin{bmatrix} -50 \\ 0 \end{bmatrix} \\ X_\infty = \left\{x | x \in \mathbb{R}^2, Lx \leq 0\right\}, \ L = \begin{bmatrix} 1 & -1 \end{bmatrix}; \\ f(x) = \begin{cases} x + B, & Lx > 0 \\ x, & Lx \leq 0 \end{cases}, \ B = \begin{bmatrix} -1 \\ +1 \end{bmatrix} \end{cases}$$

- For instance $\mathcal{X} = \left(\begin{bmatrix} 60 \\ -10 \end{bmatrix}, \begin{bmatrix} 59 \\ -9 \end{bmatrix}, \ldots, \begin{bmatrix} 25 \\ 25 \end{bmatrix}, \begin{bmatrix} 25 \\ 25 \end{bmatrix}, \ldots \right) \in \mathcal{P}$

Basic principles of automated software analysis

- **Definition 1:** A computer program represented by a dynamical system $\mathcal{S} = \mathcal{S}(X, f, X_0, X_\infty)$ is said to terminate in finite time if every solution $\mathcal{X} = x(t)$ of $\mathcal{S}$ satisfies $x(t) \in X_\infty$ for some $t \in Z_+$.
- **Definition 2:** The computer program $\mathcal{S}$ is said to run without variable overflow if $x(t)$ does not belong to a certain (unsafe) subset X_- of X for every solution $\mathcal{X} = x(t)$ of $\mathcal{S}$.
- **Definition 3:** A Lyapunov function for system $\mathcal{S}$ is defined to be a real valued function $V : X \to \mathbb{R}$, which will strictly monotonically decrease along the trajectories of $\mathcal{S}$ until they reach a terminal state, i.e.

$$V(\overline{x}) < V(x) \quad \forall x \in X,\ \overline{x} \in f(x) : x \notin X_\infty.$$

Lyapunov functions as behavior certificates

- Termination in finite-time:
  Lemma 1: If the state space X is finite and if there exists a function V satisfying
  $$V(\overline{x}) < V(x) \quad \forall x \in X,\ \overline{x} \in f(x) : x \notin X_\infty.$$
  then a terminal state X_∞ will be reached in a finite number of steps.

  Lemma 2: If there exists a bounded function $V : X \mapsto R^-$, and a constant $\theta > 1$ satisfying
  $$V(\overline{x}) < \theta V(x) \quad \forall x \in X,\ \overline{x} \in f(x) : x \notin X_\infty.$$
  then a terminal state X_∞ will be reached in a finite number of steps.

Lyapunov functions as behavior certificates

- Absence of overflow:

  Lemma 3: Consider the system $\mathcal{S}$ and let $\mathcal{V}$ denote the space of all Lyapunov functions for this system. An unsafe subset X_- of the state space X can never be reached along all the trajectories of $\mathcal{S}$ if there exists $V \in \mathcal{V}$ satisfying

$$\inf_{x \in X_-} V(x) \geq \sup_{x \in X_0} V(x)$$

Certifying boundedness and/or finite-time termination

Proposition 1: Consider the program $\mathcal{P} = \mathcal{S}(X, f, X_0, X_\infty)$ defined as before and assume that there exists a function $V : X \mapsto R$ s.t.

$$V(\overline{x}) < \theta V(x) \quad \forall x \in X,\ \overline{x} \in f(x) : x \notin X_\infty.$$
$$V(x) < 0 \quad \forall x \in X_0.$$
$$V(x) > \left\| \frac{x}{M} \right\|^2 - 1 \quad \forall x \in X.$$

where θ is a positive constant. Then, every solution $\mathcal{X} = x(t)$ of $\mathcal{P}$ remains bounded in the (safe) region defined by $\|x\| < M$. Moreover, if $\theta > 1$, every solution $\mathcal{X} = x(t)$ reaches a terminal state X_∞ in finite time.

Remark: As mentioned, proofs of absence of 'array index out-of-bounds' errors are important in software verification. This property too, can be verified by employing this Proposition.

System specific models

- Mixed integer/linear models: This system model has state space $X = \mathbb{R}^n$, and the set of initial conditions $X_0 \subset \mathbb{R}^n$. Its state transition function $f : X \mapsto 2^X$ is defined by two matrices F and H, according to

$$
\begin{aligned}
f(x) \;=\; & \left\{ [F_x \; F_w \; F_v \; 1] \begin{bmatrix} x \\ w \\ v \\ 1 \end{bmatrix} : \right. \\
& \left. \exists (w,v) \in [-1,1]^q \times \{-1,1\}^r \text{ s.t. } [H_x \; H_w \; H_v \; 1] \begin{bmatrix} x \\ w \\ v \\ 1 \end{bmatrix} = 0 \right\}
\end{aligned}
$$

Naturally, X_∞ is defined by

$$
\begin{aligned}
X_\infty \;=\; & \{x \mid x \in \mathbb{R}^n, \\
& \forall (w,v) \in [-1,1]^q \times \{-1,1\}^r, \; H[x; w; v; 1] \neq 0\}
\end{aligned}
$$

Convex optimization of Lyapunov certificates:

Our method of automated code analysis is based on using convex optimization in the search for the proposed Lyapunov functions.

1. Let the certificate function $V(.) : X \rightarrow \mathbb{R}$ take an appropriate form, e.g. V can be a linear, piecewise linear, quadratic, piecewise quadratic, or polynomial function of $x \in X$.

2. Various versions of the convex relaxation methods, including sums of squares in positivity verification, S-procedure and semi-definite relaxations of combinatorial problems can be used to formulate a convex optimization problem.

3. The resulting convex optimization problem is an LP, MILP or an LMI problem.

4. The appropriate choice of $V(.)$, and the optimization method are influenced by:

—Availability of efficient relaxation techniques

—Compatibility with a particular numerical engine for convex optimization

—Computational costs and complexity growth with the size of the problem

Numerical Example

Consider the following program:

$$
\begin{array}{l}
x(0) = x_0; c(0) = c_0; \\
\text{While } x < 500 \\
\quad \text{if } x \leq 480 \text{ and } c = 1 \\
\quad\quad x = x + a; \\
\quad\quad c = 1; \\
\quad \text{else if } x > -450 \\
\quad\quad x = x + b; \\
\quad\quad c = -1; \\
\quad \text{end} \\
\quad \text{if } x \leq -450 \\
\quad\quad c = 1; \\
\quad \text{end} \\
\text{end;}
\end{array}
$$

where $a \in [25, 45]$ and $b \in [-15, -2]$, are uncertain input parameters, $x_0 \in [-450, 480]$ and $c_0 \in \{-1, 1\}$ are uncertain initial conditions. Bounded-ness and finite-time termination of this program are not trivial.

The mixed integer/linear model of this program is defined with matrices F, and H, given by:

$$F = \begin{bmatrix} 1 & 0_{1\times 7} & \frac{a}{2} & 0_{1\times 3} & \frac{b}{2} & 0 & 0 & \frac{a+b}{2} \\ 0_{1\times 6} & \frac{1}{2} & 0_{1\times 3} & \frac{1}{2} & 0 & -\frac{1}{2} & 0_{1\times 2} & \frac{1}{2} \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & -u_m & 0_{1\times 2} & u_m & 0_{1\times 7} & 0 & 0 & -u \\ 1 & 0_{1\times 2} & l_m & 0_{1\times 2} & -l_m & 0_{1\times 6} & 0 & 0 & -l \\ 1 & 0_{1\times 3} & r_m & 0_{1\times 6} & 0 & 0 & 0 & 0 & r_{\overline{m}} \\ 0 & 1 & 0_{1\times 5} & -1 & 0_{1\times 4} & 0 & 0 & 0 & 0 \\ 0_{1\times 5} & 1 & -1 & 0_{1\times 4} & 0 & 0 & 0 & -1 & -1 \\ 0_{1\times 5} & \frac{-1}{2} & 0 & \frac{-1}{2} & 1 & \frac{1}{2} & 0_{1\times 4} & 0 & \frac{1}{2} \\ 0_{1\times 6} & \frac{1}{2} & 0 & \frac{-1}{2} & 0 & 1 & \frac{1}{2} & 0_{1\times 3} & \frac{1}{2} \\ 0_{1\times 6} & \frac{1}{2} & 0 & \frac{1}{2} & 0_{1\times 3} & 1 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

where $u_m = \frac{u+M}{2}$, $l_m = \frac{l-M}{2}$, $r_m = \frac{R+M}{2}$, $r_{\overline{m}} = \frac{R-M}{2}$, $R = 500$, $u = 480$, $l = -450$.

The quadratic Lyapunov certificate

$$V(x, c) = \left(10^{-3}\right) \times \begin{bmatrix} \frac{x}{M} \\ \frac{c}{M} \\ 1 \end{bmatrix}^T \begin{bmatrix} 0.126 & 1.502 & -1.455 \\ 1.502 & 7660.045 & -1.061 \\ -1.455 & -1.061 & -2.119 \end{bmatrix} \begin{bmatrix} \frac{x}{M} \\ \frac{c}{M} \\ 1 \end{bmatrix}$$

is the certificate for finite-time termination and bounded-ness of x, for $M = 750$, i.e. $|x| \leq 750$.

Recursive Invariant Search for Software Systems

Consider the following program:

$$x_1(0) = 1; x_2(0) = 3; x_3(0) = 0; y(0) = x_1(0);$$

While $x_3 < x_1$

$\quad y = x_1;$

$\quad x_1 = 2.5 * x_1 + x_2;$

$\quad x_2 = -0.5 * y + x_2;$

$\quad x_3 = 8 * x_3 + 1;$

end

- Assume that the overflow bound is $M = 1000$.
- Finite-time termination and boundedness are not trivial for this example
- The initial attempt to prove the desired properties via Lyapunov-like functions that were introduced comes unsuccessful!
- System invariants that prove neither finite-time termination nor boundedness but give additional information about the system behavior can be extracted.

- Affine functions are good (computationally very cheap) candidates for such assisting invariants.
- For instance, consider Lyapunov-like functions:

$$V : X \to \mathbb{R},\ V(x) = Lx$$
$$V(x(k+1)) \leq \theta V(x(k)) + \alpha$$

- Finding such linear invariants is a mixed integer linear program.
- For the previous example, $V(x) = -x_3$ is found. which provides the invariant:

$$x_3(k) \geq 0$$

- This additional information, is used via convex relaxation methods to find the appropriate Lyapunov function.

- The quadratic function

$$V\left(x\left(k\right)\right)=10^{-6}\times \begin{bmatrix} x_1\left(k\right) \\ x_2\left(k\right) \\ x_3\left(k\right) \\ 1 \end{bmatrix}^T \begin{bmatrix} 2610 & 4569 & 0.2045 & -12904 \\ 4569 & 31559 & 0.8954 & -97174 \\ 0.2045 & 0.8954 & 82 & -34904 \\ -12904 & -97174 & -34904 & 0.2948 \end{bmatrix} \begin{bmatrix} x_1\left(k\right) \\ x_2\left(k\right) \\ x_3\left(k\right) \\ 1 \end{bmatrix}$$

  Proves boundedness and finite-time termination, with overflow bound $M = 1000$.
- Additional linear invariants could be found to assist the analysis, for instance

$$\begin{aligned} x_1 + x_2 - 8x_3 &\leq 4 \\ x_1 + 2x_2 - 10x_3 &\leq 7 \end{aligned}$$

  are other such linear invariants.
- With this additional information, reachability analysis can be improved even more. In this case, boundedness with overflow bound $M = 450$ is proven!

Conclusions

- An approach towards safety analysis of software was introduced.
- The novelty of this approach is in the transfer of fundamental concepts (Lyapunov invariants) and associated computational techniques from the control systems analysis arena to software engineering
- It was shown that software, as a rule for iterative modification of computer memory, can be modeled as a dynamical system.
- Specific models carrying this task were also suggested. These include mixed integer/linear models and linear systems with conditional switching.

- System invariants, found by Lagrangian relaxations and convex optimization of certain Lyapunov-like functions prove the desired properties of the dynamical system/software.

- The properties include bounded-ness of all variables within acceptable ranges and finite time termination of the program in most cases.

- Scalability of the technique needs to be improved for applications to large computer programs with thousands of lines of code.

mardavij@mit.edu

feron@mit.edu