INVARIANT RELATIONS: A CONCEPT FOR ANALYZING WHILE LOOPS

Ali Mili, NJIT
NII, Tokyo, Japan
December 20, 2011
**Plan**

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
MOTIVATION

Research community:
- new theories, methods, tools.

Software Industrial Practice:
- Increasing large, complex applications (ULS),
  - Verification is difficult.
- Increasingly critical applications,
  - Verification is crucial.

Maintaining a technological gap between needs and capabilities.
**Motivation**

Research community: Rallying around a global/international initiative, the *Verified Software Initiative*.

Three goals:
- Developing a comprehensive theory of programming,
- Evolving an Industrial Strength Tool set,
- Deploying Verified Software in Selected Applications.
**Motivation**

Multiple Partners, multiple profiles.
- Academia (Oxford, York).
- Industry (Microsoft UK, Microsoft WA).
- Research Labs (SRI, SEI).

How I bumped into this...
- 2005: SEI’s FX project: automated tools to compute program functions, most notably loop functions.
- Broad effort to generate loop invariants (WING).
**Motivation**

VSI... but why loops?

- Despite the emergence of several programming languages/ programming paradigms (functional, logical, object oriented, aspect oriented, etc), most code under development/ operation/ maintenance is written in C-like languages.

- C-like languages: iteration is main locus of complexity, main source of faults, main focus of testing and verification.

- OOP shifts the locus of complexity towards data organization / coordination vs function calculation, but iteration remains important even there.
Motivation

Premises:
- Mills’ logic of programming using functions and relations (vs Hoare’s logic, using logic predicates).
  - Semantics as program functions (vs predicate transformer semantics).
- Relational algebra (vs predicate calculus).
- Bottom Up Analysis, composing program functions (vs top down analysis, decomposing verification conditions).
Plan

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
RELATIONAL MATHEMATICS

Three Steps.
- Definitions
- Operations on Relations
- Properties of Relations
RELATIONAL MATHEMATICS

Relation $R$, on $S$: subset of $S \times S$.

Universal Relation, $L = S \times S$. 
RELATIONAL MATHEMATICS

Identity relation on $S$.

Empty Relation on $S$. 

$S'$

$I$

$S$

$S'$

$\phi$

$S$
Relational Mathematics

Operations on Relations:
Union, Intersection, Complement, Converse.

Complement of $R$: $\bar{R}$.

Converse of $R$: $\hat{R}$.
**Relational Mathematics**

Operations on Relations:

Operations on Relations: Domain, Range.

Domain of $R$: $\text{dom}(R)$.

Range of $R$: $\text{rng}(R)$.
**Relational Mathematics**

Operations on Relations:

**Product:** $R \circ R'$ or $RR'$:
Relational Mathematics

Operations on Relations:

Vector: $C \subseteq S \rightarrow V = C \times S$.

Invector: $C \subseteq S \rightarrow V = S \times C$.

We use vectors as relational representation of sets.
RELATIONAL MATHEMATICS

Operations on Relations:

Two sets we want to represent as vector: \( \text{dom}(R) \) and \( \text{rng}(R) \).

\[ S' \]

\[ \text{dom}(R) \]

\[ \text{rng}(R) \]

\[ S \]

\[ S' \]

\[ \text{dom}(R) \]

\[ \text{rng}(R) \]

\[ S \]

\( \text{dom}(R) \) as RL

\( \text{rng}(R) \) as LR
Operations on Relations:

Pre-restriction, postrestriction of relation $R$ to subset $C$ of $S$: vector $V = C \times S$. 

$R \cap V$

$R \cap \hat{V}$
RELATIONAL MATHEMATICS

Properties of Relations:

$R$ is total: $RL = L$.

$R$ is deterministic: $\hat{R}R \subseteq I$.
RELATIONAL MATHEMATICS

Properties of Relations: Symmetry and Reflexivity.

$R$ is symmetric: $\hat{R} = R$.

$R$ is reflexive: $I \subseteq R$.

Transitive: $RR \subseteq R$. 
PLAN

• Motivation
• Relational Mathematics
• Invariant Relations
• Invariant Relations and Loop Functions
• Invariant Relations and Invariant Assertions
• Invariant Relations and Invariant Functions
• Generating Invariant Relations
• Conclusion
INVARIANT RELATIONS

Program Semantics: We consider a program $g$ on variables $x$: \textit{Xtype}, $y$: \textit{Ytype}, $z$: \textit{Ztype}.

\textit{Space} of program $g$: aggregates of values that variables $x$, $y$, $z$ may take, i.e. $S=X\times Y\times Z$.

\textit{State} of program $g$: $s=\langle x,y,z \rangle$. We address the components of $s$ as $x(s)$, $y(s)$, $z(s)$.

\textit{Function} of program $g$:

$G=\{(s,s') | \text{ if program } g \text{ starts execution in state } s \text{ then it terminates in state } s'\}$.

Whence,

$\text{dom}(G) = \{(s,s') | \text{ if program } g \text{ starts execution in state } s \text{ then it terminates}\}$. 
INVARIANT RELATIONS

We consider a while loop on space $S$ of the form

\[ w: \text{while } t \text{ do } b. \]

We let $B$ be the function of $b$, $T$ be the vector defined by $t$ (i.e. $T=\{(s,s')| t(s)\}$), and $W$ be the function of $w$. We assume that $W$ is total (the while loop terminates for all states in $S$).

Definition. An invariant relation of $w$ is a reflexive transitive superset of $(T \cap B)$. 
**Invariant Relations**

Intuitively: An invariant relation is a relation that holds between $s$ and $s'$ such that $s'$ can be obtained from $s$ by an arbitrary number (0 or more) of iterations.

How do we compute/characterize invariant relations:

- If $x$ is incremented in $b$: $\{(s,s') \mid x \leq x'\}$.
- If $x$ is decremented in $b$: $\{(s,s') \mid x \geq x'\}$.
- If $x$ is not modified in $b$: $\{(s,s') \mid x' = x\}$.
- If some function $F$ is preserved by $b$: $\{(s,s') \mid F(s) = F(s')\}$.
**Invariant Relations**

Examples, on space $S$ defined by natural variables $x, y, z$.

- **while** $(x \not= 0) \{ x = x - 1 \}$;
  - $R = \{ (s, s') \mid \}$

- **while** $(x \not= 0) \{ x = x - 1; y = y + 1 \}$;
  - $R = \{ (s, s') \mid \}$
  - $R' = \{ (s, s') \mid \}$

- **while** $(x \geq 5) \{ x = x - 5 \}$;
  - $R = \{ (s, s') \mid \}$
  - $R' = \{ (s, s') \mid \}$

- **while** $(x \geq 5) \{ x = x - 5; y = y + 1 \}$;
  - $R = \{ (s, s') \mid \}$
  - $R' = \{ (s, s') \mid \}$
**Invariant Relations**

Examples, on space $S$ defined by natural variables $x, y, z$.

- $\text{while } (x \neq 0) \{ x = x - 1 \}$;
  - $R = \{ (s, s') | x \geq x' \}$

- $\text{while } (x \neq 0) \{ x = x - 1; y = y + 1 \}$;
  - $R = \{ (s, s') | x \geq x' \}$
  - $R' = \{ (s, s') | \}$

- $\text{while } (x \geq 5) \{ x = x - 5 \}$;
  - $R = \{ (s, s') | x \geq x' \}$
  - $R' = \{ (s, s') | \}$

- $\text{while } (x \geq 5) \{ x = x - 5; y = y + 1 \}$;
  - $R = \{ (s, s') | y \leq y' \}$
  - $R' = \{ (s, s') | \}$
INVARIANT RELATIONS

Examples, on space $S$ defined by natural variables $x, y, z$.

- **while** $(x\neq 0)$ \{ $x=x-1$ \};
  - $R = \{(s,s') \mid x \geq x'\}$

- **while** $(x\neq 0)$ \{ $x=x-1; y=y+1$ \};
  - $R = \{(s,s') \mid x \geq x'\}$
  - $R' = \{(s,s') \mid x+y=x'+y'\}$

- **while** $(x\geq 5)$ \{ $x=x-5$ \};
  - $R = \{(s,s') \mid x \geq x'\}$
  - $R' = \{(s,s') \mid x \mod 5 = x' \mod 5\}$

- **while** $(x\geq 5)$ \{ $x=x-5; y=y+1$ \};
  - $R = \{(s,s') \mid y \leq y'\}$
  - $R' = \{(s,s') \mid x+5y=x'+5y'\}$
**Invariant Relations**

Example on space $S$ defined by natural variables $n, f, k$, such that $1 \leq k \leq n+1$.

- `while (k!=n+1) {f=f*k; k=k+1};`

- $R = \{(s,s')\}$
- $R = \{(s,s')\}$
- $R = \{(s,s')\}$
- $R = \{(s,s')\}$
**Invariant Relations**

Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

- **while** $(k \neq n+1)$ \{ $f = f \times k$; $k = k+1$ \};
  - $R = \{(s,s') | k \leq k'\}$
  - $R = \{(s,s') | n = n'\}$
  - $R = \{(s,s') | \}$
Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

- $\text{while } (k \neq n+1) \{ f = f \times k; \ k = k+1 \};$
  - $R = \{(s, s') | k \leq k'\}$
  - $R = \{(s, s') | \ n = n'\}$
  - $R = \{(s, s') | \ f/(k-1)! = f'/(k'-1)!\}$
**Invariant Relations**

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.  
- $\text{while } (k\neq N+1) \{x = x + a[k]; k = k+1\}$;  
  - $R = \{(s,s') | \ k \leq k'\}$  
  - $R = \{(s,s') | \}$  
- $\text{while } (k\neq0) \{x = x + a[k]; k = k-1\}$;  
  - $R = \{(s,s') | \ k \geq k'\}$  
  - $R' = \{(s,s') | \}$

Space $S$ defined by list variables $l$, $m$.  
- $\text{while } (\text{not empty}(l)) \{m = m + \text{head}(l); l = \text{tail}(l)\}$;  
  - $R = \{(s,s') | \ (l,l') \text{ in tail}*\}$  
  - $R' = \{(s,s') | \}$
Invariant Relations

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.

- while $(k \neq N+1)$ \{ $x = x + a[k]$; $k = k + 1$ \}
  
  - $R = \{(s, s')| \ k \leq k'\}$
  
  - $R = \{(s, s')| \ x + \sum_{i=k}^{N} a[i] = x' + \sum_{i=k'}^{N} a[i] \land a' = a\}$

- while $(k \neq 0)$ \{ $x = x + a[k]$; $k = k - 1$ \}
  
  - $R = \{(s, s')| \ k \geq k'\}$
  
  - $R' = \{(s, s')| \ x + \sum_{i=1}^{k} a[i] = x' + \sum_{i=1}^{k'} a[i] \land a' = a\}$

Space $S$ defined by list variables $l$, $m$.

- while (not empty($l$)) \{ $m = m + \text{head}(l)$; $l = \text{tail}(l)$ \}
  
  - $R = \{(s, s')| \ (l, l') \text{ in tail}^*\}$
  
  - $R' = \{(s, s')| \ l + m = l' + m'\}$
Plan

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
INVARIANT RELATIONS AND LOOP FUNCTIONS

Examples, on space $S$ defined by natural variables $x, y, z$.

- $\text{while } (x \neq 0) \{ x = x - 1 \}$;
  - $W = \{(s,s') \mid \}$

- $\text{while } (x \neq 0) \{ x = x - 1; y = y + 1 \}$;
  - $W = \{(s,s') \mid \}$

- $\text{while } (x \geq 5) \{ x = x - 5 \}$;
  - $W = \{(s,s') \mid \}$

- $\text{while } (x \geq 5) \{ x = x - 5; y = y + 1 \}$;
  - $W = \{(s,s') \mid \}$
INVARIANT RELATIONS AND LOOP FUNCTIONS

Examples, on space $S$ defined by natural variables $x, y, z$.

- **while** $(x \neq 0)$ \{ $x = x - 1$; \}
  - $W = \{(s, s') | x' = 0 \land y' = y \land z' = z\}$

- **while** $(x \neq 0)$ \{ $x = x - 1$; $y = y + 1$; \}
  - $W = \{(s, s') | x' = 0 \land y' = y + x \land z' = z\}$

- **while** $(x \geq 5)$ \{ $x = x - 5$; \}
  - $W = \{(s, s') | x' = x \mod 5 \land y' = y \land z' = z\}$

- **while** $(x \geq 5)$ \{ $x = x - 5$; $y = y + 1$; \}
  - $W = \{(s, s') | x' = x \mod 5 \land y' = y + x / 5 \land z' = z\}$
IN Variant RELATIONS AND Loop Functions

Example on space $S$ defined by natural variables $n, f, k$, such that $1 \leq k \leq n+1$.

- $\textbf{while } (k \neq n+1) \{ f = f \cdot k; \ k = k + 1 \};$
  - $W = \{ (s, s') \mid \}$
Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

- $\textbf{while} \ (k \neq n+1) \ \{ f = f \cdot k; \ k = k + 1 \};$
  - $W = \{ (s, s') \mid n' = n \land k' = n + 1 \land f' = f \cdot n!/(k-1)! \}$
INVARIANT RELATIONS AND LOOP FUNCTIONS

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.

- $\textbf{while } (k \neq N+1) \{x=x+a[k]; \ k=k+1\}$;
  - $W = \{(s,s') | \}$.

- $\textbf{while } (k \neq 0) \{x=x+a[k]; \ k=k-1\}$;
  - $W = \{(s,s') | \}$.

Space $S$ defined by list variables $l$, $m$.

- $\textbf{while } (\text{not empty}(l)) \{m=m+\text{head}(l); \ l=\text{tail}(l)\}$;
  - $W = \{(s,s') | \}$. 
INVARIANT RELATIONS AND LOOP FUNCTIONS

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.

- $\text{while } (k \neq N+1) \{ x = x + a[k]; \; k = k+1 \};$
  - $W = \{(s,s') | k' = N+1 \land a' = a \land x' = x + \sum_{i=k}^{N} a[i] \}$
- $\text{while } (k \neq 0) \{ x = x + a[k]; \; k = k-1 \};$
  - $W = \{(s,s') | k' = 0 \land a' = a \land x' = x + \sum_{i=1}^{k} a[i] \}.$

Space $S$ defined by list variables $l$, $m$.

- $\text{while } (\text{not empty}(l)) \{ m = m + \text{head}(l); \; l = \text{tail}(l) \};$
  - $W = \{(s,s') | l' = () \land m' = m + 1 \}.$
IN Variant Relations and Loop Functions

```cpp
#include <iostream> #include <math.h>
using namespace std; // header
int fact (int n); // factorial function
int main () {const int cN, ca;
  int i,j,fb,nc, np; float x,x1,x2,x3;
  while (j!=cN) {j=j+i; nc=fb; fb=np+nc;
    np=nc; x2=x2+pow((x-ca),i)/fact(i);
    x3=x3+pow(x,i)/fact(i);
    x1=x1+pow(x,j)/fact(j); i=i+1; j=j-i;}
```
INVARIANT RELATIONS AND LOOP FUNCTIONS

\[ W = \{ (s, s') | j = cN \land s' = s \} \cup \]

\[
(s, s') | j \geq cN \land x' = x \land j' = cN \land \\
x_1' = \frac{x_1 \Gamma(1+cN) \Gamma(1+j) - e^x \Gamma(1+j) \Gamma(1+cN,x)}{\Gamma(1+j) \Gamma(1+cN)} \\
+ e^x \Gamma(1+cN) \Gamma(1+j, x) \land \\
x_2' = \frac{e^{-ca} (e^{ca} x_2 \Gamma(i) \Gamma(i+j-cN))}{\Gamma(i) \Gamma(i+j-cN)} \\
+ \frac{-e^x \Gamma(i+j-cN) \Gamma(i,x-ca) + e^x \Gamma(i) \Gamma(i+j-ca,x-ca)}{\Gamma(i) \Gamma(i+j-cN)} \land \\
x_3' = \frac{x_3 \Gamma(i) \Gamma(i+j-cN) - e^x \Gamma(i+j-cN) \Gamma(i, x)}{\Gamma(i) \Gamma(i+j-cN)} \\
+ \frac{e^x \Gamma(i) \Gamma(i+j-cN,x)}{\Gamma(i) \Gamma(i+j-cN,x)} \land \\
f_{b'} = np \times F(j - cN) + fb \times F(j + 1 - cN) \land \\
n_{c'} = np \times F(j - cN - 1) + fb \times F(j - cN) \land \\
n_{p'} = np \times F(j - cN - 1) + fb \times F(j - cN) \land \\
i' = i + j - cN\]
**Invariant Relations and Loop Functions**

Proposition. Given an invariant relation $R$ of $w$, we can write:

$$ R \cap \hat{T} \supseteq W. $$

We can convert any invariant relation into a superset of $W$.

If we can find many invariant relations, take their intersection (the intersection of invariant relations is an invariant relation), then the post-restriction of the result to $\neg t$, then we may be able to find $W$, the function of the loop.
INVARIANT RELATIONS AND LOOP FUNCTIONS

Examples, on space $S$ defined by natural variables $x, y, z$.

- **while** $(x\neq 0) \{x=x-1\}$;
  - $R/\neg t = \{(s,s') | \; x \geq x' \land x'=0\}$

- **while** $(x\neq 0) \{x=x-1; \; y=y+1\}$;
  - $R/\neg t = \{(s,s') | \; x \geq x' \land x+y=x'+y' \land x'=0\}$

- **while** $(x\geq 5) \{x=x-5\}$;
  - $R/\neg t = \{(s,s') | \; x \geq x' \land x \mod 5 = x' \mod 5 \land x'<5\}$

- **while** $(x\geq 5) \{x=x-5; \; y=y+1\}$;
  - $R/\neg t = \{(s,s') | \; y \leq y' \land x+5y=x'+5y' \land x'<5\}$
Examples, on space $S$ defined by natural variables $x, y, z$.

- $\textbf{while } (x \neq 0) \{ x = x - 1 \}$;  
  $R/\neg t = \{(s, s') | x \geq 0 \land x' = 0\}$

- $\textbf{while } (x \neq 0) \{ x = x - 1; y = y + 1 \}$;  
  $R/\neg t = \{(s, s') | x \geq 0 \land y' = x + y \land x' = 0\}$

- $\textbf{while } (x \geq 5) \{ x = x - 5 \}$;  
  $R/\neg t = \{(s, s') | x' = x \mod 5 \}$

- $\textbf{while } (x \geq 5) \{ x = x - 5; y = y + 1 \}$;  
  $R/\neg t = \{(s, s') | x' = x \mod 5 \land y' = y + (x / 5)\}$
INARIANT FUNCTIONS AND LOOP FUNCTIONS

Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

- $\text{while } (k \neq n+1) \{ f = f \times k; \ k = k + 1 \}$;

  - $R /\rightarrow t$

  $= \{(s,s') | k \leq k' \land n = n' \land f/(k-1)! = f'/(k'-1)! \land k' = n' + 1\}$
**Invariant Functions and Loop Functions**

Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

- **while** $(k \neq n+1)$ \{ $f = f \times k$; $k = k+1$ \};
  - $R \models = \{(s,s') | n = n' \land f = n! \times f/(k-1)! \land k' = n'+1\}$. 
Invariant Functions and Loop Functions

Space S defined real variable $x$, index variable $k$, real array $a[1..N]$.

- $\text{while } (k \neq N+1) \{ x = x + a[k]; k = k+1 \};$

- $R/→t$

  $= \{(s,s')/ k \leq k' \land x + \sum_{i=k}^{N} a[i] = x' + \sum_{i=k'}^{N} a[i] \land a' = a \lor k'=N+1 \}$

- $\text{while } (k \neq 0) \{ x = x + a[k]; k = k-1 \};$

- $R/→t = \{(s,s')/ k \geq k' \land x + \sum_{i=1}^{k} a[i] = x' + \sum_{i=1}^{k'} a[i] \land a' = a \land k'=0 \}$

Space S defined by list variables $l$, $m$.

- $\text{while } (\text{not empty}(l)) \{ m = m + \text{head}(l); l = \text{tail}(l) \};$

- $R/→t = \{(s,s')/ (l,l') \text{ in } \text{tail}^* \land l + m = l' + m' \land l' = () \}$
**Invariant Functions and Loop Functions**

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.

- **while** $(k \neq N+1)$ \{ $x = x + a[k]$; $k = k + 1$; \}

- $R/-t = \{(s,s')/ x + \sum_{i=k}^{N} a[i] = x' \land a' = a \land k'=N+1\}$

- **while** $(k \neq 0)$ \{ $x = x + a[k]$; $k = k - 1$; \}

- $R/-t = \{(s,s')/ x + \sum_{i=1}^{k} a[i] = x' \land a' = a \land k'=0\}$

Space $S$ defined by list variables $l$, $m$.

- **while** (not empty($l$)) \{ $m = m + \text{head}(l)$; $l = \text{tail}(l)$; \}

- $R/-t = \{(s,s')/ l' = l + m \land l' = ()\}$
PLAN

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
Invariant Relations and Invariant Assertions

Invariant assertion for a Hoare formula:
\[ \{p\} \text{ while } t \text{ do } b \{q\} \]

Predicate \( a \) such that
- \( p \Rightarrow a \)
- \( \{a \land t\} \ b \ \{a\} \)
- \( a \land \neg t \Rightarrow q \)

Relational Form: Invariant Assertion \( A \) for while loop \( \text{while } t \text{ do } b \) with respect to \( P, Q \) (all vectors):
- \( P \subseteq A \),
- \( A \cap (T \cap B) \subseteq \hat{A} \),
- \( A \cap \bar{T} \subseteq Q. \)
INVARIANT RELATIONS AND INVARIANT ASSERTIONS

Examples, on space $S$ defined by natural variables $x$, $y$, $z$.  

- **while** $(x \neq 0)$ \{ $x = x - 1$; \}
  - $P = \{ (s,s') | \ x = 10 \}$.
  - $A = \{ (s,s') | \}$.

- **while** $(x \neq 0)$ \{ $x = x - 1; \ y = y + 1$; \}
  - $P = \{ (s,s') | \ x = 10 \land y = 8 \}$
  - $A = \{ (s,s') | \}$

- **while** $(x \geq 5)$ \{ $x = x - 5$; \}
  - $P = \{ (s,s') | \ x = 28 \}$
  - $A = \{ (s,s') | \}$

- **while** $(x \geq 5)$ \{ $x = x - 5; \ y = y + 1$; \}
  - $P = \{ (s,s') | \ x = 28 \land y = 8 \}$
  - $A = \{ (s,s') | \}$
INVARIANT RELATIONS AND INVARIANT ASSERTIONS

Examples, on space $S$ defined by natural variables $x$, $y$, $z$.

- **while** $(x \neq 0)$ \{ $x = x - 1$ \};
  - $P = \{(s, s') \mid x = 10\}$.
  - $A = \{(s, s') \mid x \leq 10\}$.

- **while** $(x \neq 0)$ \{ $x = x - 1; y = y + 1$ \};
  - $P = \{(s, s') \mid x = 10 \land y = 8\}$
  - $A = \{(s, s') \mid x + y = 18\}$

- **while** $(x \geq 5)$ \{ $x = x - 5$ \};
  - $P = \{(s, s') \mid x = 28\}$
  - $A = \{(s, s') \mid x \mod 5 = 3\}$

- **while** $(x \geq 5)$ \{ $x = x - 5; y = y + 1$ \};
  - $P = \{(s, s') \mid x = 28 \land y = 8\}$
  - $A = \{(s, s') \mid x + 5y = 68 \land y \geq 8\}$
Invariant Relations and Invariant Assertions

Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

- while $(k \neq n+1)$ \{ $f = f \times k$; $k = k + 1$ \};
  - $P =\{(s,s') | f = 1 \land k = 1\}$
  - $A =\{(s,s') | \}
Invariant Relations and Invariant Assertions

Example on space \( S \) defined by natural variables \( n, f, k \), such that \( 1 \leq k \leq n+1 \).

\[ \text{while } (k \neq n+1) \{ f = f \times k; k = k+1 \}; \]

- \( P = \{(s, s') \mid f = 1 \land k = 1 \land n = 12\} \)
- \( A = \{(s, s') \mid f = (k-1)! \land n = 12\} \)
Invariant Relations and Invariant Assertions

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.

- **while** $(k \neq N+1)$ \( \{ x=x+a[k]; \ k=k+1 \}$;
  - $P = \{(s,s') | \ x=0 \land k=1 \land a = [1,2,3,4,5,6]\}$
  - $A = \{(s,s') | \}$
- **while** $(k \neq 0)$ \( \{ x=x+a[k]; \ k=k-1 \}$;
  - $P = \{(s,s') | \ x=0 \land k=1 \land a = [1,2,3,4,5,6]\}$
  - $A = \{(s,s') | \}$

Space $S$ defined by list variables $l$, $m$.

- **while** \( (\text{not empty}(l)) \) \( \{ m=m+\text{head}(l); \ l=\text{tail}(l) \}$;
  - $P = \{(s,s') | \ l=(3,4,5) \land m=(0,1,2)\}$
  - $R' = \{(s,s') | \}$
Invariant Relations and Invariant Assertions

Space $S$ defined real variable $x$, index variable $k$, real array $a[1..N]$.

- While ($k \neq N+1$) \{ $x=x+a[k]; \ k=k+1$ \};
  - $P = \{(s,s') | \ x=0 \land k=1 \land a = [1,2,3,4,5,6]\}$
  - $A = \{(s,s') | \ x+\text{sum}(a,k,N)=21\}$

- While ($k \neq 0$) \{ $x=x+a[k]; \ k=k-1$ \};
  - $P = \{(s,s') | \ x=0 \land k=1 \land a = [1,2,3,4,5,6]\}$
  - $A = \{(s,s') | \ x+\text{sum}(a,1,k)=21\}$

Space $S$ defined by list variables $l$, $m$.

- While (not empty($l$)) \{ $m=m+\text{head}(l); \ l=\text{tail}(l)$ \};
  - $P = \{(s,s') | \ l=(3,4,5) \land m=(0,1,2)\}$
  - $R' = \{(s,s') | \ m+l=(0,1,2,3,4,5)\}$
Invariant Relations and Invariant Assertions

Proposition: Given an invariant relation $R$ of the loop, $w$: \textit{while} $t$ \textit{do} $b$, and a precondition $P$, the vector $A = \hat{R}P$ is an invariant assertion for $w$.

Furthermore... All invariant assertions have this form, i.e. stem from an invariant relation.

Two implications:

- The search for invariant assertions is amenable to the search for invariant relations (a more general concept).
- Structure of invariant assertion: product of loop-dependent term, and context-dependent term.
Examples, on space $S$ defined by natural variables $x$, $y$, $z$.

- $\textbf{while } (x \neq 0) \{ x = x - 1 \};$
  - $P = \{(s,s') | x = 10\}$, $R = \{(s,s') | x' \leq x\}$
  - $A = \{(s,s') | \}$

- $\textbf{while } (x \neq 0) \{ x = x - 1; y = y + 1 \};$
  - $P = \{(s,s') | x = 10 \land y = 8\}$, $R = \{(s,s') | x + y = x' + y'\}$
  - $A = \{(s,s') | \}$

- $\textbf{while } (x \geq 5) \{ x = x - 5 \};$
  - $P = \{(s,s') | x = 28\}$, $R = \{(s,s') | x \mod 5 = x' \mod 5\}$
  - $A = \{(s,s') | \}$

- $\textbf{while } (x \geq 5) \{ x = x - 5; y = y + 1 \};$
  - $P = \{(s,s') | x = 28 \land y = 8\}$, $R = \{(s,s') | x + 5y = x' + 5y'\}$
  - $A = \{(s,s') | \}$
INvariants Relations and Invariant Assertions

Examples, on space $S$ defined by natural variables $x$, $y$, $z$.

- **while** $(x\neq 0)$ $\{x=x-1\}$;
  - $P = \{(s,s')| x=10\}$, $R = \{(s,s')| x'\leq x\}$
  - $A = \{(s,s')| \exists s'': x\leq x'' \land x''=10\}$

- **while** $(x\neq 0)$ $\{x=x-1; y=y+1\}$;
  - $P = \{(s,s')| x=10 \land y=8\}$, $R = \{(s,s')| x+y=x'+y'\}$
  - $A = \{(s,s')| \exists s'': x+y=x''+y'' \land x''=10 \land y''=8\}$

- **while** $(x\geq 5)$ $\{x=x-5\}$;
  - $P = \{(s,s')| x=28\}$, $R = \{(s,s')| x \mod 5 = x' \mod 5\}$
  - $A = \{(s,s')| \exists s'': x \mod 5 = x'' \mod 5 \land x''=28\}$

- **while** $(x\geq 5)$ $\{x=x-5; y=y+1\}$;
  - $P = \{(s,s')| x=28 \land y=8\}$, $R = \{(s,s')| x+5y=x'+5y'\}$
  - $A = \{(s,s')| \exists s'': x+5y=x''+5y'' \land x''=28 \land y''=8\}$
Invariant Relations and Invariant Assertions

Examples, on space $S$ defined by natural variables $x$, $y$, $z$.

- **while** ($x \neq 0$) {$x=x-1$};
  - $P = \{(s,s') | x=10\}$, $R = \{(s,s') | x' \leq x\}$
  - $A = \{(s,s') | x \leq 10\}$

- **while** ($x \neq 0$) {$x=x-1; y=y+1$};
  - $P = \{(s,s') | x=10 \land y=8\}$, $R = \{(s,s') | x+y=x'+y'\}$
  - $A = \{(s,s') | x+y=18\}$

- **while** ($x \geq 5$) {$x=x-5$};
  - $P = \{(s,s') | x=28\}$, $R = \{(s,s') | x \text{ mod } 5 = x' \text{ mod } 5\}$
  - $A = \{(s,s') | x \text{ mod } 5 = 3\}$

- **while** ($x \geq 5$) {$x=x-5; y=y+1$};
  - $P = \{(s,s') | x=28 \land y=8\}$, $R = \{(s,s') | x+5y=x'+5y'\}$
  - $A = \{(s,s') | x+5y=68\}$
Invariant Relations and Invariant Assertions

Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

\[ \text{while } (k \neq n+1) \{ f = f \times k; k = k+1 \}; \]

- \[ P = \{(s, s') | k = 1 \land f = 1\} \]
- \[ R = \{(s, s') | f/(k-1)! = f'/(k'-1)!\} \]
- \[ A = \{(s, s') | \exists s'': f/(k-1)! = f''/(k''-1)! \land k'' = 1 \land f'' = 1\} \]
Example on space $S$ defined by natural variables $n$, $f$, $k$, such that $1 \leq k \leq n+1$.

*while* $(k \neq n+1)$ { $f = f \times k$; $k = k+1$; }

- $P = \{(s, s') | k = 1 \land f = 1\}$
- $R = \{(s, s') | f/(k-1)! = f'/(k'-1)!\}$
- $A = \{(s, s') | f = (k-1)!\}$
INVARIANT RELATIONS AND INVARIANT ASSERTIONS

while loop:

```c
#include <iostream> #include <math.h> using namespace std;
int main () {const int e,g,cN; const float a,m,b,c,d; int h,i,j,k,l,n,p,xx,yy;
float x, y, z, u, v, w, q, r,s, t;  int aa[cN]; int ab[cN]; float f (float z);
while (k>=7) {j=j+aa[i];  i=i+n;  l=l+ab[n];  n=n-k;  k=k-7;  yy=xx*yy;  x=a+m*x;
y=a*pow(y,m);  t=t+b*w;  n=n+k+6;  s=s+c*z;  z=z+b;  u=a*u+m;  v=v-c;  w=d*w;  q=q+r;
r=f(r);  h=h+g*p;  p=p/e;  i=i-n;}}
```

Initialization (P):

```c
const int e=2, g=1; cN=21; const float a=2.,m=2.,b=1.,c=1.,d=2.;
int i=1, n=20, j=0, k=150, p=20, h=0, l=0, xx=5, yy=2;
float x=0., y=2., z=1., v=5., w=1., s=0., t=0., u=0., q=0., r=1.;
float f (float z) {return z+1;}; int aa[cN] = 
{2,8,10,38,15,0,3,6,23,90,57,14,46,175,23,19,0,16,22,17,72};
int ab[cN] = {12,50,4,9,6,3,0,22,19,12,15,2,0,0,8,1,42,12,5,3,0};
```
INVARIANT RELATIONS AND INVARIANT ASSERTIONS

Invariant Assertion:

\[
A = \left\{ \begin{array}{l}
\{ i \geq 1 \land n \leq 20 \land \phi(1.4427\log(p)) = 0.321928 \land xx = 5 \land h + 2p = 40 \land i + n = 21 \land \\
abla \begin{array}{l}
aa = [2, 8, 10, 15, 0, 3, 6, 23, 90, 57, 14, 46, 175, 23, 19, 0, 16, 22, 17, 72] \land 2^i = x + 2 \land \\
ab = [12, 50, 4, 9, 6, 3, 0, 22, 19, 12, 15, 2, 0, 0, 8, 1, 42, 12, 5, 3, 0] \land l + \Sigma_{H = 1}^n ab[H] = 225 \\
\end{array} \\
\begin{array}{l}
q + \Sigma_{H = 1}^n f^H(r) = \Sigma_{H = 1}^{20} f^H(1) \land i = z \land 1.4427n\log(xx) + 1.4427\log(yy) = 47.4386 \land \\
7i + k = 157 \land 2^i = 1 + 1.4427\log(y) \land 2^i = u + 2 \land j + \Sigma_{H = 1}^{21} aa[H] = 656 \land i + v = 6 \\
\end{array} \\
\begin{array}{l}
w = 0.52^i \land f^{21-i}(r) = f^{20}(1) \land i + \left[ \frac{\log(p)}{\log(2)} \right] = 5 \land s - 0.5(z - 1)z = 0 \land t - w = -1 \\
\end{array}
\end{array} \right\}.
\]
INVARIANT RELATIONS AND INVARIANT ASSERTIONS

Aligator (Linz, Zurich):

\[ c \times q + r \times v = 5 \times r \land b \times v + c \times z = 5 \times b + c \land b \times q + r = r \times z \]
\[ \land 2 \times s + (v - 5) \times (1 + z) = b \times (v - 5). \]

Daikon (MIT, Boston):

\[ i + n = 21 \land 7 \times i + k = 157 \land 7 \times n + 10 = k \land z + v = 6 \land w - t = 1 \land \]
\[ 2 \times w = x + 2 \land 2 \times t = x \land z = r \land s = q \land x = u \land x \%a = 0 \land y y \%e = 0 \land e \in a a]. \]

Loopfrog (Lugano, Oxford):

\[ z z \leq r \land k \leq n \land r = z \land t = w \land u = x. \]
Plan

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
**INVARIANT RELATIONS AND INVARIANT FUNCTIONS**

*Invariant Function* of a while loop $w$: `while t do b`, function $V$ on $S$ such that: $T \cap V = T \cap BV$.

Rewriting as: $t(s) \Rightarrow V(s) = V(B(s))$. 
INVARIANT RELATIONS AND INVARIANT FUNCTIONS

Examples, on space $S$ defined by natural variables $x$, $y$, $z$.

- **while** $(x\neq 0)$ \{ $x=x-1$ \};
  - $V(s)=0$.
  - $V(B(s))=V(x-1)=0$.

- **while** $(x\neq 0)$ \{ $x=x-1$; $y=y+1$ \};
  - $V(s)=x+y$.
  - $V(B(s))=V(x-1,y+1)=x-1+y+1=x+y=V(s)$.

- **while** $(x \geq 5)$ \{ $x=x-5$ \};
  - $V(s)=x \mod 5$.
  - $x \geq 5 \Rightarrow V(B(s))=V(x-5)=(x-5) \mod 5 = x \mod 5 = V(s)$.

- **while** $(x \geq 5)$ \{ $x=x-5$; $y=y+1$ \};
  - $V(s)=x+5y$.
  - $x \geq 5 \Rightarrow V(B(s))=V(x-5,y+1)=x-5+5(y+1)=x+5y=V(s)$.
Example on space $S$ defined by natural variables $n, f, k$, such that $1 \leq k \leq n+1$.

- **while** $(k \neq n+1)$ \{ $f = f \times k$; $k = k+1$ \};

  - $V(s) = f / (k-1)!$.
  - $V(B(s)) = V(f \times k, k+1)$
    
    $= f \times k / (k+1-1)!$
    
    $= f \times k / k!$
    
    $= f / (k-1)!$
    
    $= V(s)$. 
Proposition: If $V$ is an invariant function then $R=\{(s,s') \mid V(s)=V(s')\}$ is an invariant relation.

All Symmetric invariant relations stem from invariant functions.
INVARIANT RELATIONS AND INVARIANT FUNCTIONS

Examples, on space \( S \) defined by natural variables \( x, y, z \).

- **while** \((x \neq 0) \) \{\(x = x - 1\)\};
  - \( R = \{(s, s') \mid x \geq x'\} \).
  - *Not symmetric.*

- **while** \((x \neq 0) \) \{\(x = x - 1; y = y + 1\)\};
  - \( R = \{(s, s') \mid x + y = x' + y'\} \).
  - \( V(s) = x + y \).

- **while** \((x \geq 5) \) \{\(x = x - 5\)\};
  - \( R = \{(s, s') \mid x \mod 5 = x' \mod 5\} \).
  - \( V(s) = x \mod 5 \).

- **while** \((x \geq 5) \) \{\(x = x - 5; y = y + 1\)\};
  - \( R = \{(s, s') \mid x + 5y = x' + 5y'\} \).
  - \( V(s) = x + 5y \).
INVARIANT RELATIONS AND INVARIANT FUNCTIONS

Example on space \( S \) defined by natural variables \( n, f, k, \) such that \( 1 \leq k \leq n+1. \)

- **while** \( (k \neq n+1) \) \{ \( f = f \times k; \ k = k + 1 \); \}
  - \( R = \{(s,s') \mid f/(k-1)! = f'/(k'-1)!\}. \)
  - \( V(s) = f/(k-1)! \).
Plan

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
Generating Invariant Relations

Given a while loop of the form, \( w: \text{while to do } b \), the following is an invariant relation: \( I \cup T(T \cap B) \).

The only invariant relation we get for free. For all others, we have to work hard.

How hard? We have to match source code against known code patterns for which we know an invariant relation.
GENERATING INVARIANT RELATIONS

Crucial Divide-and-Conquer strategy: Given that an invariant relation is a superset of \((T \cap B)\), we rewrite \((T \cap B)\) as an intersection of terms:

\[
(T \cap B) = B_1 \cap B_2 \cap B_3 \cap \ldots \cap B_n.
\]

Then,
- Any superset of \(B_1\) is a superset of \((T \cap B)\),
- Any superset of \(B_1 \cap B_2\) is a superset of \((T \cap B)\),
- Any superset of \(B_1 \cap B_2 \cap B_3\) is a superset of \((T \cap B)\).

We stop at 3 to keep combinatorics under control. May, and likely will, change.
GENERATING INVARIANT RELATIONS

Stored Patterns (recognizers):
- Variable declarations,
- Formal code pattern,
- Formal invariant relation form.

Three classes, depending on number of terms.
- 1-recognizers,
- 2-recognizers,
- 3-recognizers.

Currently: about 70, mostly 2-recognizers.
## Generating Invariant Relations

<table>
<thead>
<tr>
<th>ID</th>
<th>State Space</th>
<th>Code Pattern (cca)</th>
<th>Invariant Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1R1</td>
<td>int x;</td>
<td>x=x+c</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td>const int c&gt;0;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1R2</td>
<td>int x;</td>
<td>x=x+c</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td>const int c&gt;0;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2R1</td>
<td>int x, y;</td>
<td>x=x+a, y=y+b</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td>const int a, b;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2R2</td>
<td>int x, y;</td>
<td>x=x*a, y=y+x</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td>const int a, b;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2R3</td>
<td>int x, y;</td>
<td>x=x+a, y=y*b</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td>const int a, b;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2R4</td>
<td>int i; function f; sometype x;</td>
<td>i=i-1, x=f(x)</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3R1</td>
<td>int x, i; int a[N];</td>
<td>i=i+1, x=x+a[i],</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a=a</td>
<td></td>
</tr>
<tr>
<td>3R2</td>
<td>int x, i; int a[N];</td>
<td>i=i-1, x=x+a[i],</td>
<td>{(s, s')</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a=a</td>
<td></td>
</tr>
</tbody>
</table>
Generating Invariant Relations

How are recognizers generated:

- Vast majority through invariant functions.
  - \( R = \{(s, s') \mid V(s) = V(s')\} \).

- When the cca2mat step fails to find a recognizer to match a combination of statements, it notifies the user, with an indication of what pattern needs a match.

- Mostly numeric recognizers, some non numeric.

- Can be adapted/ specialized as needed.
Generating Invariant Relations

Three Step Analysis Process:

- **cpp2cca**: Mapping source code to an internal notation. Two goals:
  - isolate language specific step,
  - write loop as intersection.

- **cca2mat**: Generate invariants in Mathematica format (© Wolfram Research) using recognizer database.

- **mat2nb**: Use invariant relation to compute feature of interest.
  - Function: \( R \cap \hat{T} \).
  - Invariant Assertion: \( \hat{RP} \).
Generating Invariant Relations

But we do not always get to choose how the function of the loop body can be written:

- If the loop body has if-then-else statements, its outer structures is a union.
  - In that case, we compute a reflexive transitive superset for each branch of the union.
  - But the union of transitive relations is not transitive: We have a program, written in Mathematica, that tries to find a joint superset that is reflexive and transitive.
Plan

- Motivation
- Relational Mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
CONCLUSION

Introduced Invariant Relations and shown how they can be used to analyze loops, and how they can be generated.

Shown how invariant relations can be used to generate or approximate loop functions, and loop invariants (invariant assertions).

Shown how invariant relations can be generated automatically from recognizers and discussed how recognizers can themselves be generated (off-line, by hand) from invariant functions.
**Conclusion**

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Invariant Assertion</th>
<th>Invariant Function</th>
<th>Invariant Relation</th>
<th>Loop Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invariant Assertion</td>
<td>Is</td>
<td>No relation was found</td>
<td>$R = \overline{A} \cup \hat{A}$</td>
<td>$W \supseteq (\overline{A} \cup \hat{A}) \cap \hat{T}$</td>
</tr>
<tr>
<td>Invariant Function</td>
<td>$A = {(s, s')</td>
<td>V(s) \in C}$</td>
<td>Is</td>
<td>$R = V\hat{V}$</td>
</tr>
<tr>
<td>Invariant Relation</td>
<td>$A = \widehat{R}w$</td>
<td>If $R = \widehat{R}$</td>
<td>Is</td>
<td>$W \supseteq R \cap \hat{T}$</td>
</tr>
<tr>
<td>Loop Function</td>
<td>$A = {(s, s')</td>
<td>W(s) \in C}$</td>
<td>IsA</td>
<td>$R = W\hat{W}$</td>
</tr>
</tbody>
</table>
CONCLUSION

Prospects:

- Migrate the invariant generation program from syntactic matching to semantic matching.
- Populate the recognizer database with recognizers that capture relevant programming knowledge.
- Populate the recognizer database with recognizers that capture relevant domain knowledge for selected application domains.
- Consolidate post IR-generation steps with domain knowledge to enable domain-aware analysis.
Next week: How invariant relations can also be used for other analysis steps, such as:

- Computing the termination condition of a loop.
- Computing the weakest precondition of a loop for a given post-condition.
- Computing the strongest post-condition of a loop for a given precondition.
- Computing a necessary condition of correctness (a sufficient condition of non-correctness).
- Computing a sufficient condition of correctness (where correctness can be proven before the function of the loop is computed).
Plan

- Motivation
- Relational mathematics
- Invariant Relations
- Invariant Relations and Loop Functions
- Invariant Relations and Invariant Assertions
- Invariant Relations and Invariant Functions
- Generating Invariant Relations
- Conclusion
To be continued, next week