Karlsruhe Institute of Technology

Entscheidungsverfahren mit Anwendungen in der Softwareverifikation

X: Modulare Arithmetik

Carsten Sinz
Institut für Theoretische Informatik

09.01.2019

"Program Arithmetic"

```
unsigned int
square_check(unsigned int x)
{
    unsigned int y = x * x;
    if (y == 33) { error(); }
    return y;
}
```

Is `error()` reachable?

Has $x^2 \equiv 33 \mod 2^{32}$ a solution?

Yes!
4 Solutions, e.g. 663169809

Algebraic Properties

Karlsruhe Institute of Technology

Mathematical Integers vs. Signed vs. Unsigned

Addition

Property	$\mathbb{Z}$	signed int (if defined)	unsigned int
Closure	yes	yes	yes
Associativity `a+(b+c) = (a+b)+c`	yes	yes	yes
Commutativity `a+b = b+a`	yes	yes	yes
Ex. of identity `a+0 = a`	yes	yes	yes
Ex. of inverse `a+(-a) = 0`	yes	yes	**no**

Multiplication

Property	$\mathbb{Z}$	signed int (if defined)	unsigned int
Closure	yes	yes	yes
Associativity `a*(b*c) = (a*b)*c`	yes	yes	yes
Commutativity `a*b = b*a`	yes	yes	yes
Ex. of identity `a*1 = a`	yes	yes	yes
Ex. of inverse $a*(a^{-1}) = 1$	only 1 and -1	only 1 and -1	**all odd numbers**

- $\mathbb{Z}$: commutative ring with unity; integral domain (no zero divisors); Euclidian domain (division with remainder)
- $\mathbb{Z}/2^k\mathbb{Z}$: also commutative ring with unity, but no integral domain (for k>1)

Arithmetic in $\mathbb{Z}/2^k\mathbb{Z}$

- Definition:

  $\mathbb{Z}/n\mathbb{Z} = \{\bar{a}_n \,|\, a \in \mathbb{Z}\}$ with $\bar{a} = \{\ldots, a-n, a, a+n, \ldots\}$
- As usual, we identify $\bar{a}$ with a, where $0 \leq a < n$, thus

  $\mathbb{Z}/2^k\mathbb{Z} = \{0,\ldots,2^k - 1\}$
- Examples of arithmetic in $\mathbb{Z}/2^k\mathbb{Z}$:
  - When has the equation $a \cdot x = b$ a solution? Is it unique?
  - Has the equation $x^2 = 33$ a solution in $\mathbb{Z}/2^8\mathbb{Z}$? Is it unique?
- Basic facts:
  - $\sum_{i=1}^{n} a_i x_i \equiv b \pmod{m}$ is solvable for the unknowns x_i, iff the greatest common divisor of $\{a_1, \ldots, a_n, m\}$ divides b.
  - a has a multiplicative inverse $\bmod\ m$, iff $\gcd(a, m) = 1$.
  - a^{-1} can be computed using the extended Euclidian algorithm or using Euler's theorem, $a^{-1} \equiv a^{\phi(m)-1} \pmod{m}$. For $m = 2^k$, $\phi(m) = \phi(2^k) = 2^{k-1}$, and thus $a^{-1} \equiv a^{2^{k-1}-1} \pmod{2^k}$.

Solving Equations in $\mathbb{Z}/2^k\mathbb{Z}$

- **Given:** Polynomial $p(x)$
- **Goal:** Solutions of $p(x) \equiv 0 \mod 2^k$
- First, consider the linear case: $p(x) = a \cdot x - b$, i.e. solving the equation $a \cdot x = b$ modulo $m = 2^k$.
- If a is invertible, then $x = b \cdot a^{-1}$ is the (unique) solution. (This is the case, if a is odd.)
- Otherwise, $a \cdot x = b$ has solutions, iff $\gcd(a, 2^k) \mid b$. The solution is not unique, but a particular solution is given by $x = b/a$.
- **Theorem:** *The congruence ax ≡ b (mod m) is soluble in integers if, and only if, gcd(a, m) | b. The number of incongruent solutions modulo m is gcd(a, m).*
- How can we find all solutions?
- For all solutions x, the following holds: $\exists t \,.\, ax + tm = b$. Having a first solution x_0, all solutions are given by $x_k = x_0 + k \cdot (m/\gcd(a, m))$ for $0 \leq k < \gcd(a, m)$.

Solving Systems of Linear Congruences

Karlsruhe Institute of Technology

- Given a system $S = \{E_j\}$ of linear congruences (mod m = 2^k) over n variables, with

$$E_j : \sum_{i=1}^{n} a_{ji} x_i \equiv b_j \mod 2^k \quad ,$$

  find its solution set.

- Algorithm [Ganesh, 2007]:
  - If there is an odd coefficient a_{ji}, solve equation E_j for x_i and substitute x_i in all other equations. If E_j cannot be solved for x_i, i.e. if $\gcd\{a_{j1}, \ldots, a_{jn}, m\} \nmid b_j$, then there is no solution to S.
  - If all coefficients a_{ji} are even, divide all a_{ji}, b_j by two and decrease k by one.
  - Repeat the algorithm with the resulting system of congruences and stop with "success" if there is only one solved equation left.
- Properties:
  - The algorithm is a sound and complete decision procedure for linear congruences.

Solving Systems of Linear Congruences

- Example: Solve the following system of congruences modulo 8:

$$
\begin{aligned}
3x + 4y + 2z &= 0 \\
2x + 2y &= 6 \\
4y + 2x + 2z &= 0
\end{aligned}
$$

- Note:
  - Ganesh considers the unknowns as bit-vectors of length k; when the system is divided by 2, the highest bit in each bit-vector is dropped (i.e. left unconstrained)
- Question:
  - How can the set of all solutions of S be determined after the algorithm finished?

Solving Non-Linear Congruences

- **Task:** Given a polynomial p(x), find all solutions of $p(x) \equiv 0 \mod 2^k$.
- **Hensel lifting algorithm** (special case for m = 2^k):
  1. [k=1] Check, whether $p(x) \equiv 0 \mod 2$ has a solution. If not, exit with "no solution".
  2. [k→k+1] Let {x_i} be the set of solutions for $p(x) \equiv 0 \mod 2^k$. We distinguish two cases to lift each x_i from k to k+1:
     A. If $p'(x_i) \equiv 0 \mod 2$: [0 or 2 lifted solutions]
        1. If $p(x_i) \not\equiv 0 \mod 2^{k+1}$, x_i cannot be lifted
        2. Otherwise there are two lifted solutions $x_i^* = x_i + t \cdot 2^k,\ t \in \{0,1\}$
     B. If $p'(x_i) \not\equiv 0 \mod 2$: [unique lifting]
        $x_i^* = x_i - p(x_i) \mod 2^{k+1}$
- **Note:** Hensel-lifting also works for multivariate polynomials. However, already the base case (k=1) is NP-complete. (Why?)

Solving Non-Linear Congruences

- **Example:** $x^2 \equiv 33 \mod 2^4$
- $p(x) = x^2 - 33, \quad p'(x) = 2x$
- **[k=1, mod 2]:** x^2=1 mod 2 has solution x*=1
- **[k=2, mod 4]:** Try to lift x*=1: p'(x*)=0 mod 2, thus 0 or 2 lifted solutions p(x*)=0 mod 4, thus 2 liftings: x*'= x*+2t = {1, 3}
- **[k=3, mod 8]:**
  - Lifting x*=1: 0 or 2 lifted solutions, p(x*)=0 mod 8, x*' = { 1, 5 }
  - Lifting x*=3: 0 or 2 lifted solutions, p(x*)=0 mod 8, x*' = { 3, 7 }
- **[k=4, mod 16]:**
  - Lifting x*=1: p(x*)=0 mod 16, x*' = { 1, 9 }
  - Lifting x*=3: p(x*)=8 mod 16, no lifting
  - Lifting x*=5: p(x*)=8 mod 16, no lifting
  - Lifting x*=7: p(x*)=0 mod 16, x*' = { 7, 15 }

Hensel's Lemma

- **Theorem:** Let f(x) be a polynomial with integer coefficients, k ≥ m > 0, r an integer with $f(r) \equiv 0 \mod p^k$. Then if $f'(r) \not\equiv 0 \mod p$, there is an integer s such that $f(s) \equiv 0 \mod p^{k+m}$ and $s \equiv r \mod p^k$. So s is a „lifting" of r to a root mod p^{m+k}. Moreover, s is unique mod p^{m+k}.
- **Proof:** Consider the Taylor series expansion of f:

$$f(r + p^k t) = f(r) + f'(r)p^k t + \frac{f''(r)}{2!} p^{2k} t^2 + \ldots$$

Since m ≤ k, all terms but the first two vanish mod p^{k+m}, so

$$f(r + p^k t) = f(r) + f'(r)p^k t \pmod{p^{k+m}}$$

Setting $f(r + p^k t) \equiv 0$, we can solve for t:

$$f(r) + f'(r)p^k t \equiv 0 \pmod{p^{k+m}}$$

$$p^k t \equiv -\frac{f(r)}{f'(r)} \pmod{p^{k+m}}$$

$$t \equiv -\frac{\frac{f(r)}{p^k}}{f'(r)} \pmod{p^m}$$

Notes to Hensel's Lemma

- $f(r)/p^k$ is an integer by the lemma's assumption $f(r) \equiv 0 \mod p^k$.
- $f'(r)$ has a multiplicative inverse mod p^m, as $f'(r) \not\equiv 0 \mod p$.
- The solution s unique mod p^{k+m} is given by $s = r + p^k t = r - f(r)/a$, where $a \equiv f'(r)^{-1} \pmod{p^m}$.

- Case without a unique lifting (restricted to the case m=1 in the Lemma):
  - Assume $f(r) \equiv 0 \mod p^k$ and $f'(r) \equiv 0 \mod p$. Then $s \equiv r \mod p^k$ implies $f(r) \equiv f(s) \mod p^{k+1}$ by the Taylor expansion, i.e. $f(r + tp^k) \equiv f(r) \mod p^{k+1}$ for all integers t. We thus have two cases:
    - $f(r) \not\equiv 0 \mod p^{k+1}$: Then there is no lifting from k to k+1.
    - $f(r) \equiv 0 \mod p^{k+1}$: Then every lifting of r from k to k+1 is a root mod p^{k+1}, i.e. $s = r + tp^k$ is a solution for each $t \in \{0,\ldots,p-1\}$.