Solving Universe Level Constraints

Marc Bezem
Department of Informatics
University of Bergen

August 2022

Thierry: thanks for all the collaboration!

- 14 joint papers spanning three decades
- many, many interesting ideas and discussions
- a lot of friendship

The talk will be about the latest joint paper:
[BC22] Loop-checking and the uniform word problem for join-semilattices with an inflationary endomorphism, TCS, 2022

Crash course: Solving max-constraints in the integers

- Example: $a \geq b + 9$ and $\max(b, x) \geq x + 1$
- Simple algorithm (may not terminate):
  1. Start in, e.g., the all-zero state $a = b = x = 0$
  2. Evaluate the lhs's in the state
  3. Adjust the rhs's to become $\leq$ the (old) lhs's
  4. Repeat from point 2 with the new state until stable
- Example:

a	0	0	0	...	0	0
b	0	-9	-9	...	-9	-9
x	0	-1	-2	...	-9	-10

Remarks on the ‘max-atom problem’ MAP [BNR0x]

- Easy example of a *loop*: $\max(x, y) \geq x + 1, \max(x, y) \geq y + 1$
- MAP can be encoded in Presburger Arithmetic. Better:
- There is a ‘small-model property’: we know when to stop the simple algorithm and conclude unsatisfiability
- The simple algorithm is *weakly* polynomial, but not *strongly*
- MAP has many (polynomial) equivalents: in game theory, AND/OR scheduling, shortest hyperpaths, ...
- MAP is in NP and in co-NP (short certificates)
- Big open problem: is MAP (strongly) PTIME solvable?
  - Yes: ButkovicZimmermannDAM2006
  - Open: BezemNieuwenhuisRodriguezDAM2008
  - Withdrawn by authors: LahlouTruffetArxiv2021

Connection with type theory

- $A : \mathrm{U}_i,\ B : \mathrm{U}_j \vdash A \to B : \mathrm{U}_{\max(i,j)}$, similarly for $\Pi, \Sigma, \ldots$
- $\vdash \mathrm{U}_i : \mathrm{U}_{i+1}$
- Type inference involves constraint solving for universes
- Pioneers: Huet, Harper, Pollack (implicit universe levels, with cumulativity, no $\max$)
- Even with cumulativity, $\max$ can be useful (Herbelin)
- Other important influences:
  - Courant (explicit universe levels)
  - Sozeau, Tabareau (universe polymorphism in Coq)
  - Voevodsky (universe polymorphism, explicit constraints)
- Later, p.12: application to Coq, experiments by Sozeau

Our minimalistic proposal [BC22]

- Design consideration: a weak theory allows many possible interpretations
- Algebraic theory of universe levels: join-semilattice with inflationary endomorphism $\_^+$,

$$x \vee x^+ = x^+ \qquad (x \vee y)^+ = x^+ \vee y^+,$$

  plus associativity, commutativity and idempotency of $\vee$.
- We use $x \leq y$ for $x \vee y = y$, so $x \leq x^+$ means $\_^+$ inflationary
- Given a set C of constraints, a *loop* is a term t such that $t^+ \leq t$ can be inferred from C.
- THM: if there is no loop, then there is a model of C in $(\mathbb{N}, \max, \mathrm{succ})$
- Important for type inference: loop-checking and the uniform word problem

From constraints to Horn clauses (Lorenzen + $\_^+$)

- Atoms $v + k$, where $v + 0 := v$, $v + (k + 1) := (v + k)^+$
- Recall the example: $a \geq b + 9$ and $\max(b, x) \geq x + 1$
- Horn clauses: $a \to b + 9$ and $b, x \to x + 1$
- The endo-axiom + congruence lead to $a + k \to b + k + 9$ and $b + k, x + k \to x + k + 1$, for all $k \in \mathbb{N}$
- Also, $v^+ \geq v$ leads to $v + k + 1 \to v + k$ for all $k \in \mathbb{N}$, $v \in V$
- Now we can infer from a, b, x the atoms $b + 9, \ldots, b + 1$ and $x + 1$, so $x + 2, \ldots, x + 9, x + 10$
- Details in [BC22]: the Horn clause approach is equivalent to the semilattice approach
- NB $a = 0,\ b = -9,\ x = -10$ mirrors $a + 0,\ b + 9,\ x + 10$ because of the duality in the translation from constraints to Horn clauses (disjunctive $\max(b, x)$ to conjunctive b, x).

Computation of the least model

- Let $\mathbb{N}^\infty$ be $\mathbb{N} \cup \{\infty\}$ and V a finite set of variables
- For $f : V \to \mathbb{N}^\infty$, let $\downarrow(f)$ be $\{v + k \mid v \in V,\ k \in \mathbb{N},\ k \leqslant f(v)\}$
- Let C be a set of Horn clauses coming from a set of constraints (as explained by example previously)
- Sufficient for all goals in [BC22]: given f, compute the least model of C including $\downarrow(f)$
- Possible: forward reasoning, setting $f(v) = \infty$ if $v + k$ exceeds the small-model bound
- However, often one needs just a few universe levels
- The proposal in [BC22] can be seen as forward reasoning with on-the-fly loop-checking
- [BC22] is constructive mathematics, the algorithm is implicit in Theorem 3.2 and Lemma 3.3

Algorithm

- Auxiliary functions:
  - $\mathit{forward}(V, C, f) := (W, f')$ with f' the (cumulative) result of one application of each Horn clause in C, with W the set of variables that have changed.
  - $\mathit{simplify}(V, C, f) := \mathit{None}$ if $f(v) < \infty$ for all $v \in V$, otherwise $\mathit{Some}(V', C', f')$ in which all v with $f(v) = \infty$ have been eliminated from (V, C, f)
  - (Skip on first reading) preconditions: $W \subset V$ and f finite on $V - W$ (for termination); notations: $C|W$ clauses over W, $C{\downarrow}W$ clauses with conclusion over W. Returns the least model of $C{\downarrow}W$ including ${\downarrow}(f)$.

    $\mathit{lem33}(V, W, C, f) :=$
    let $g = \mathit{thm32}(W, \emptyset, C|W, f)$ in
    let $(W', g') = \mathit{forward}(V, C{\downarrow}W, f \vee g)$ in
    if $W' = \emptyset$ then $f \vee g$ else $\mathit{lem33}(V, W, C, f \vee g')$
    endall

Main function related to Theorem 3.2 [BC22]

$U \subseteq V$ is the set of variables that have been increased since the main call $\mathit{thm32}(V, \emptyset, C, f)$. Primary induction on V, secondary induction on $V - U$. Returns the least model of C including $\downarrow(f)$.

$\mathit{thm32}(V, U, C, f) :=$ let $(W, f') = \mathit{forward}(V, C, f)$ in
  if $W = \emptyset$ then f
  elsif $U \cup W = V$ then $\lambda\_.\infty$
  else match $\mathit{simplify}(V, C, f')$ with
    $\mathit{Some}(V', C', g) \implies f' \vee \mathit{thm32}(V', (U \cup W) \cap V', C', g)$
    $\mathit{None} \implies$ (* all f'-values finite, so $\mathit{lem33}$ terminates *)
      let $g = \mathit{lem33}(V, U \cup W, C, f')$ in
        let $(W', g') = \mathit{forward}(V, C, g)$ in
          if $W' = \emptyset$ then g else $\mathit{thm32}(V, U \cup W \cup W', C, g')$
  endall

Example

- Let C consist of $b+3 \to x$ and $a \to b+9$ and $b, x \to x+1$
- Then $V = \{a, b, x\}$, subsets denoted bx, x
- For later use:
  - $\mathit{forward}(V, C, a0b0x0) = (bx, a0b9x1)$
  - $\mathit{forward}(bx, C|bx, a0b9x1) = (x, a0b9x7)$ cumulative
  - $\mathit{forward}(x, C|x, f) = (\emptyset, f)$ for all f, since $C|x = \emptyset$
  - $\mathit{forward}(x, C{\downarrow}x, a0b9xk) = (x, a0b9x(k+1))$ for all $k \le 9$
- $\mathit{thm32}(V, \emptyset, C, a0b0x0) \rightsquigarrow \mathit{lem33}(V, bx, C, a0b9x1) \rightsquigarrow$
  $\mathit{thm32}(bx, \emptyset, C|bx, a0b9x1) \rightsquigarrow \mathit{lem33}(bx, x, C|bx, a0b9x7) \rightsquigarrow$
  $\mathit{thm32}(x, \emptyset, C|x, a0b9x7) = a0b9x7$
  $\mathit{forward}(x, C{\downarrow}x, a0b9x7) = (x, a0b9x8)$
  $\ldots$ ‘pumping’
  $\mathit{thm32}(x, \emptyset, C|x, a0b9x10) = a0b9x10$ model
  $\ldots$ unwinding

Application to Coq

- Coq is currently not using $\max$ in the constraints, using instead extra variables and constraints, and cumulativity
- Example, Coq's standard library: ca. 3K variables, 11K constraints and only four universe levels needed
- [experiment] Universes loop checking with clauses #16022
- Discussion maps vs. hash tables: see realworldocampl
- Ideally, the trusted core of Coq is fully certified
- The problem that the algorithm solves is in NP and in co-NP, so we can get short certificates of either outcome
- Proposal: fastest possible implementation + certified validation of certificates
- NB in Coq the algorithm should be incremental and support backtracking