ZLive: A new Z Animator

The Community Z Tools Project
(CZT)
http://czt.sourceforge.net

Mark Utting, Petra Malik, Rohit Bansal

Agenda

- What is a Z animator?
- What use is a Z animator?
- Why build another Z animator?
- The Proposed Architecture
- Examples
- Questions and Comments

What is a Z animator?

- Goal: *Evaluate some Z expressions*

Search for one/all concrete solutions

Not all Z expressions are executable! Some are infinite…

ISO Z Standard (2002)

Set expressions, Predicates (∃∀…), Schemas (=Operations)

What use is a Z animator?

- Forward execution:
  - *Init ; Incr ; Add[in? := 3] ; Sqrt ; ...*
- Backwards/sideways evaluation.
  - *Sqrt[value' := 4]*
  - *[Sqrt | status! ≠ ok]*
- Testing an operation (provide inputs+outputs)
  - *Sqrt[value := 16, value' := 4]*
- Playing around (students)
  - *{ x,y : 0..N | x * y = N }*

Rapid Prototyping

Finite theorem proving
or model checking:
- Searching for a solution (∃)
- Looking at ALL solutions (∀)
- Searching for counter examples

Conclusion:

Validation
Verification
Experimentation

Why build another Z animator?

- Existing animators: limitations + unsupported
  - Evaluation time unpredictable and unbounded
  - Do not handle all Z constructs, especially Z Std.
- We want a Java-based, open-source animator that fits into CZT.
- Jaza was evaluated as best Z animator.
  - Goal: more *sophistication*, more *coverage*.
- It’s a lot of fun trying to build one!

The Proposed Architecture

1) Read Z Specification
2) Read target Expr
3) Unfold defns (schema ops, toolkit etc.)
4) Flatten to atomic predicates
5) Calculate modes of each predicate
6) Reorder predicates into cheapest order
7) Enumerate all solutions (lazily)

repeat

Example 1

1) Pop == [q,q':seq $\mathbb{N}$; out!:$\mathbb{N}$ | q = <out!> ⌢ q']
2) Pop[q := <10,11,12>]
3) { Pop
   | q = {(1,10),(2,11),(3,12)}
   • ≪ q'==q', out!==out! ≫ }

   Then unfold Pop and 'normalise'

   { q,q':$\mathbb{P}$($\mathbb{Z}$x$\mathbb{Z}$); out!:$\mathbb{Z}$
   | q∈seq $\mathbb{N}$ ∧ q'∈seq $\mathbb{N}$ ∧ out!∈$\mathbb{N}$
   ∧ q={(1,10),(2,11),(3,12)} ∧ q = {(1,out!)} ⌢ q'
   • ≪ q'==q', out!==out! ≫ }

Eg 1: Flatten

{ q,q':$\mathbb{P}(\mathbb{Z}x\mathbb{Z})$; out!:$\mathbb{Z}$
| q∈seq $\mathbb{N}$ ∧ q'∈seq $\mathbb{N}$ ∧ out!∈$\mathbb{N}$
∧ q={(1,10),(2,11),(3,12)} ∧ q = {(1,out!)} ⌢ q'
• ≪ q'==q', out!==out! ≫ }

After flattening expressions to one operator per predicate:

4) { q,q':$\mathbb{P}(\mathbb{Z}x\mathbb{Z})$; out!:$\mathbb{Z}$
| q∈seq $\mathbb{N}$ ∧ q'∈seq $\mathbb{N}$ ∧ out!∈$\mathbb{N}$
∧ q={v1,v2,v3} ∧ v1=(1,10) ∧ v2=(2,11) ∧ v3=(3,12)
∧ q = v4⌢q' ∧ v4={v5} ∧ v5=(1,out!)
• ≪ q'==q', out!==out! ≫ }

An aside: (5) modes

- Mode == {InputVars} → {OutputVars} [Num.of.Solns]
- Eg. *q={v1,v2,v3}* has modes:
  - {v1,v2,v3} → {q} [1]
  - {q} → {v1,v2,v3} [0.5]
  - {q,v1,v2,v3} → { } [0.5]
- Eg. *q=v4 ⌢q'* has modes:
  - {v4,q' } → {q} [1]
  - {q} → {v4,q' } [#q + 1]
  - {q,q'} → {v4} [1]
  - {q,v4} → {q} [1]
  - {q,v4,q' } → { } [0.5]
- Eg. *out!* $\in \mathbb{N}$ has modes:
  - {out!} → { } [0.5]
  - { } → {out!} [∞] (not usable, except for partial searches)

Eg 1: Reorder and Enumerate

4) { q,q':$\mathbb{P}(\mathbb{Z}x\mathbb{Z})$; out!:$\mathbb{Z}$
| q∈seq $\mathbb{N}$ ∧ q'∈seq $\mathbb{N}$ ∧ out!∈$\mathbb{N}$
∧ q={v1,v2,v3} ∧ v1=(1,10) ∧ v2=(2,11) ∧ v3=(3,12)
∧ q = v4⌢q' ∧ v4={v5} ∧ v5=(1,out!)
• ≪ q'==q', out!==out! ≫ }

6) After reordering (output vars in bold blue):

{ q,q':$\mathbb{P}(\mathbb{Z}x\mathbb{Z})$; out!:$\mathbb{Z}$
| v1=(1,10) ∧ v2=(2,11) ∧ v3=(3,12) ∧ q={v1,v2,v3}
∧ q = v4⌢q' ∧ v4={v5} ∧ v5=(1,out!)
∧ q∈seq $\mathbb{N}$ ∧ q'∈seq $\mathbb{N}$ ∧ out!∈$\mathbb{N}$
• ≪ q'==q', out!==out! ≫ }

7) Enumerate Solutions:
This has 4 solutions, but only one passes the next predicate.

Questions and Comments?

- How much unfolding?
  - Eg. $s \cup t == \{ x:T \mid x \in s \vee x \in t \}$ (unfold this toolkit defn?)
  - But can we deduce these modes for $r = s \cup t$?
    - $\{s,t\} \rightarrow \{r\}$ [1]
    - $\{r\} \rightarrow \{s,t\}$ $[2^{\#s} \times 2^{\#t}]$
    - $\{r,s\} \rightarrow \{t\}$ $[2^{\#s}]$
    - $\{r,t\} \rightarrow \{s\}$ $[2^{\#t}]$
- How best to handle disjuncts? DNF?
  - Too big and causes duplicate searching
  + More accurate mode analysis of each branch.