νZ – a wide-spectrum logic

University of Waikato

27 July 2004

Martin Henson

Some history ...

- *Program development in constructive set theories (–1997)*
  - $a : A$ – "Propositions as types"
- *Constructive Z (FMP 1998)*
  - Programs from (constructive) proofs
- *Z logics (FACJ, JLC, 1999–2000)*
  - $\mathcal{Z}_C$ and $\mathcal{Z}_C^\perp$ – Classical logics for Z
- *Classical "sets of implementations" (FACJ 2003)*
  - $f \Subset U$ – Classical, non-Z ...
- *Theories of refinement (JIGPAL 2003)*
  - total correctness refinement for partial relation semantics
- ν*Z– wide-spectrum logic*
  - $U_0 \sqsupseteq U_1$ – Classical, non-Z relational ...

What is νZ ...?

The framework νZ is a radical modification of the specification language Z. The differences are as follows:

- Z is based on a partial-correctness semantics; νZ is based on a total-correctness semantics.
- Z identifies *chaos* and *magic*; νZ distinguishes between these.
- Z schema operators are not monotonic; νZ schema operators are all monotonic.
- Z is based on *equality*; νZ is based on *refinement*.
- Z is a specification language; νZ is wide-spectrum, integrating a programming language.
- Z is relatively inflexible; νZ is extensible.
- Z is a *language*; νZ is a *logic*.

What is νZ ...?

Operation Specification:

$$
\begin{array}{lll}
U & ::= & \\
 & X & \text{– schema variable} \\
 & [T \mid P \mid Q] & \text{– atomic operation schemas} \\
 & U_0 \land U_1 & \text{– conjunction schemas} \\
 & U_0 \lor U_1 & \text{– disjunction schemas} \\
 & U_0 \Rightarrow U_1 & \text{– implication schemas} \\
 & U_0 \fatsemi U_1 & \text{– composition schemas} \\
 & \exists \mathsf{x} \bullet U_0 & \text{– existential hiding schemas} \\
 & \mu X \bullet U(X) & \text{– recursive schemas}
\end{array}
$$

What is νZ ...?

The language of νZ is interpreted within $\mathcal{Z}_{\mathcal{C}}^{\perp}$ a conservative extension of $\mathcal{Z}_{\mathcal{C}}$, the Z logic developed by Steve and me between 1997 and 2000.

Positive definitions:

$$\mathsf{OP}\left(\cdots x_i^{\mathbb{P}\,T_i}\cdots\right) =_{df} U\left(\cdots x_i\cdots\right)^{\mathbb{P}\,T} \qquad \text{where } T = \cdots \Upsilon\ T_i\ \Upsilon \cdots$$

where the types T_i are schema types and the variables x_i appear in the $\mathcal{Z}_{\mathcal{C}}^{\perp}$ expression $U \in W_T$ only in logically *positive* positions.

What is νZ ...?

Examples:

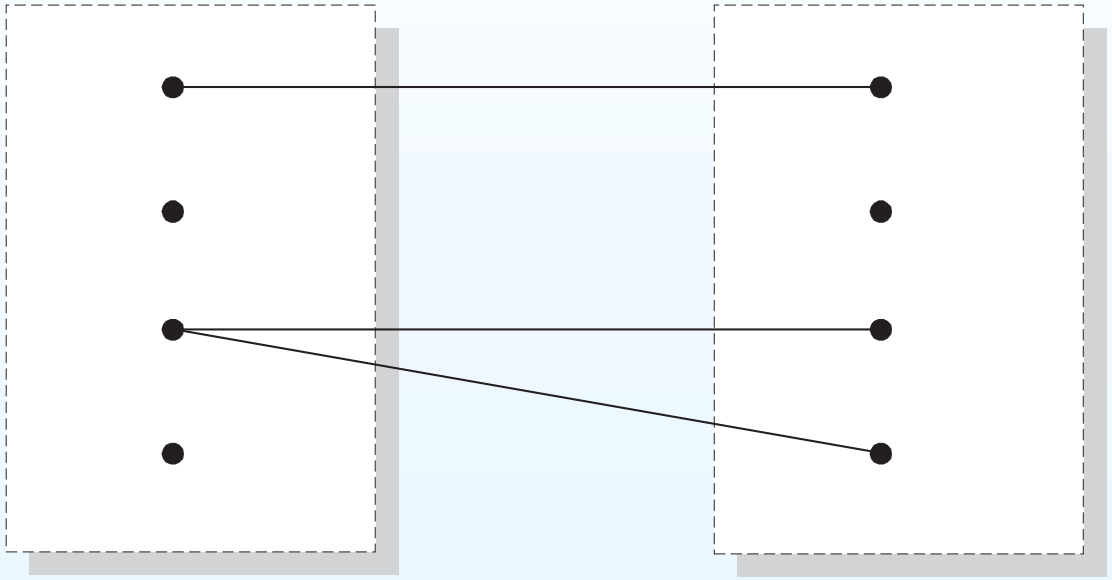
$$
\begin{array}{lll}
abort & =_{df} & [T \mid false \mid false] \quad \text{– abort} \\
chaos & =_{df} & [T \mid false \mid true] \quad \text{– chaos} \\
chaos_P & =_{df} & [T \mid \neg P \mid false] \quad \text{– P-chaos} \\
U[x \Rrightarrow E] & =_{df} & chaos_{x=E} \wedge U \quad \text{– schema application} \\
U \uparrow P & =_{df} & chaos_P \Rightarrow U \quad \text{– strengthened preconditions}
\end{array}
$$

Partial *versus* total correctness ...

An operation schema:

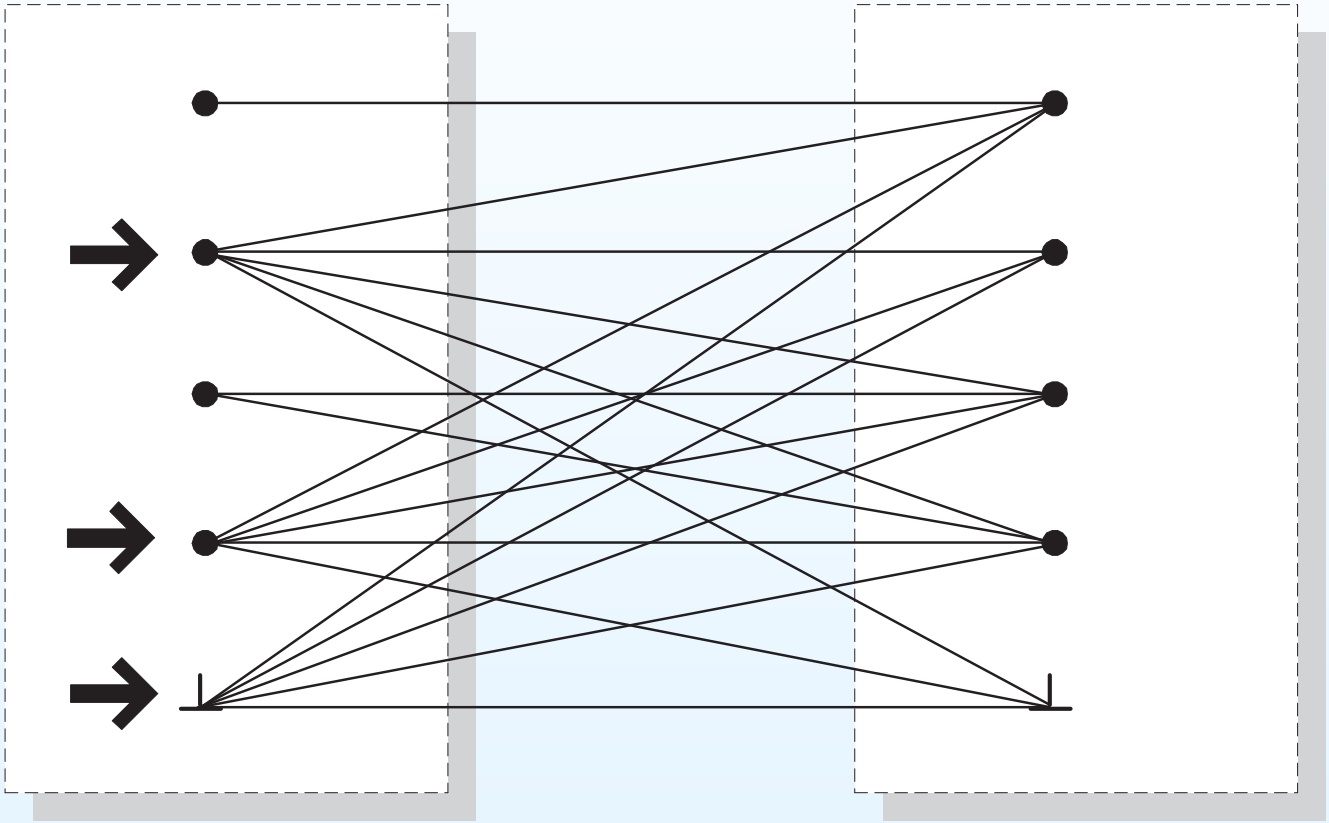
$$
\begin{array}{|l}
\hline
U \\
\mathsf{x}, \mathsf{x}' \in \{0, 1, 2, 3\} \\
\hline
(\mathsf{x} = 0 \land \mathsf{x}' = 0) \lor \\
(\mathsf{x} = 2 \land \mathsf{x}' = 2) \lor \\
(\mathsf{x} = 2 \land \mathsf{x}' = 3) \\
\hline
\end{array}
$$

Partial correctness ...



The dots on the left correspond to the possible input states, written $\langle\!|\ x \Rrightarrow n\ |\!\rangle$ for the various of n. The dots on the right to the possible output states, written $\langle\!|\ x' \Rrightarrow n\ |\!\rangle$.

Total correctness ...



What we have here is the *lifted-totalised completion* of the original relation.

What is νZ ...?

A programming language:

(i) $\mathtt{skip} =_{df} [\Delta T \mid true \mid \theta T = \theta' T]$

(ii) $\mathtt{x:=}\, E =_{df} [\Delta([x^T] \curlyvee T_E]) \mid true \mid x' = E \wedge \theta T_E = \theta' T_E]$

(iii) $\mathtt{if}\ D\ \mathtt{then}\ X_0\ \mathtt{else}\ X_1 =_{df} X_0 \uparrow D \wedge X_1 \uparrow \neg D$

(iv) $\mathtt{begin\ var\ x} : T_{\mathtt{x}}\mathtt{;}X\ \mathtt{end} =_{df} \exists\, \mathtt{x}^{T_{\mathtt{x}}}, \mathtt{x}'^{T_{\mathtt{x}}} \bullet X$

(v) $\mathtt{proc}\ f(\mathtt{x})\ \mathtt{cases\ x\ in}\ 0 : X_0;\ \mathtt{m}+1 : X_1(\mathtt{m}, f(\mathtt{m}))\ \mathtt{endcases} =_{df}$
$\mu X \bullet X_0 \uparrow \mathtt{x} = 0 \wedge \exists\, \mathtt{m} \bullet X_1(\mathtt{m}, X[\mathtt{x} \Rrightarrow \mathtt{m}]) \uparrow \mathtt{x} = \mathtt{m}+1$

(vi) $f(E) =_{df} f[\mathtt{x} \Rrightarrow E]$

What is a (wide-spectrum) logic ...?

Everything is characterised by introduction and elimination rules.
For example, atomic operation schemas:

$$\frac{z_0.P \vdash z_0.z_1'.Q}{z_0 \star z_1' \in [T \mid P \mid Q]}\ (U^+) \qquad \frac{z_0 \star z_1' \in [T \mid P \mid Q] \quad z_0.P}{z_1'.Q}\ (U^-)$$

For example, refinement:

$$\frac{z \in U_0 \vdash z \in U_1}{U_0 \sqsupseteq U_1}\ (\sqsupseteq^+) \qquad \frac{U_0 \sqsupseteq U_1 \quad z \in U_0}{z \in U_1}\ (\sqsupseteq^-)$$

For example, schema hiding:

$$\frac{z \in U}{z \mathrel{\dot\in} \exists\, \mathsf{x} : V \bullet U}\ (U^+_\exists) \qquad \frac{z \in \exists\, \mathsf{x} : V \bullet U \quad z \star \langle\!|\, \mathsf{x} \Rrightarrow y \,|\!\rangle \in U \vdash P}{P}\ (U^+_\exists)$$

Program logic ...

Simple example, conditionals:

$$\texttt{if } D \texttt{ then } U_0 \texttt{ else } U_1 =_{df} U_0 \uparrow D \wedge U_1 \uparrow \neg D$$

Rules:

$$\frac{z.D \vdash z \mathrel{\dot\in} U_0 \quad \neg z.D \vdash z \mathrel{\dot\in} U_1}{z \in \texttt{if } D \texttt{ then } U_0 \texttt{ else } U_1} \ (\texttt{if}^+)$$

$$\frac{z \in \texttt{if } D \texttt{ then } U_0 \texttt{ else } U_1 \quad z.D}{z \mathrel{\dot\in} U_0} \ (\texttt{if}^-_0)$$

$$\frac{z \in \texttt{if } D \texttt{ then } U_0 \texttt{ else } U_1 \quad \neg z.D}{z \mathrel{\dot\in} U_1} \ (\texttt{if}^-_1)$$

Refinement logic ...

$$\texttt{if } D \texttt{ then } [T \mid P \mid D \wedge Q] \texttt{ else } [T \mid P \mid \neg D \wedge Q] \sqsupseteq [T \mid P \mid Q]$$

Proof:

$$\dfrac{\dfrac{}{D \vee \neg D} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{z.P}\,(1) \quad \dfrac{z \in \texttt{if } \quad \dfrac{}{z.D}\,(2)}{z \in [T \mid P \mid D \wedge Q]}}{z.(D \wedge Q)}}{z.Q} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{z.P}\,(1) \quad \dfrac{z \in \texttt{if } \quad \dfrac{}{\neg z.D}\,(2)}{z \in [T \mid P \mid \neg D \wedge Q]}}{\neg z.(D \wedge Q)}}{z.Q}}{z.Q}\,(2)}{z \in [T \mid P \mid Q]}\,(1)$$

Proof-theoretic simplifications ...

νZ^+: reflexivity at the level of observations:

$$\frac{}{P \vdash_{\nu Z^+} x = x}$$

If:

$$z.\Gamma \vdash_{\nu Z} z.P$$

is a νZ derivation, then there exists a νZ^+ derivation whose conclusion is:

$$\Gamma \vdash_{\nu Z^+} P$$

The converse is also true.

Refinement logic ...

Preconditions and postconditions:

$$\frac{P_1 \vdash P_0}{[T \mid P_0 \mid Q] \sqsupseteq [T \mid P_1 \mid Q]}\ (\sqsupseteq^{+}_{pre}) \qquad \frac{Q_0 \vdash Q_1}{[T \mid P \mid Q_0] \sqsupseteq [T \mid P \mid Q_1]}\ (\sqsupseteq^{+}_{post})$$

Composition:

$$\frac{P_2 \vdash P_0 \wedge \forall \overline{v} \bullet Q_0[\overline{x'}/\overline{v}] \Rightarrow P_1[\overline{x'}/\overline{v}] \quad \exists \overline{u} \cdots \bullet Q_0[\overline{x'}/\overline{v}] \wedge Q_1[\overline{x'}/\overline{v}] \vdash Q_2}{[T_0 \mid P_0 \mid Q_0] \fatsemi [T_1 \mid P_1 \mid Q_1] \sqsupseteq [T_0 \curlyvee T_1 \mid P_2 \mid Q_2]}$$

Conjunction:

$$\frac{P_2 \vdash P_0 \wedge P_1 \quad Q_0 \vee Q_1 \vdash Q_2}{[T_0 \mid P_0 \mid Q_0] \wedge [T_1 \mid P_1 \mid Q_1] \sqsupseteq [T_0 \curlyvee T_1 \mid P_2 \mid Q_2]}\ (\sqsupseteq_{\wedge})$$

Refinement logic ...

Completely general transformation of conjunction to composition:

$$\frac{P_2 \vee P_3 \qquad \vdash P_0 \wedge \forall \overline{v} \bullet Q_0[\overline{x'}/\overline{v}] \Rightarrow P_1[\overline{x'}/\overline{v}]}{\exists \overline{u} \cdots \bullet Q_0[\overline{x'}/\overline{v}] \wedge Q_1[\overline{x'}/\overline{v}] \qquad \vdash Q_2 \wedge Q_3}$$

$$[T_0 \mid P_0 \mid Q_0] \fatsemi [T_1 \mid P_1 \mid Q_1] \sqsupseteq [T_2 \mid P_2 \mid Q_2] \wedge [T_3 \mid P_3 \mid Q_3]$$

Monotone inductive schemas ...

The $\mu X \bullet U(X)$ are *recursive schemas*.
The schema algebra is *monotonic* (but not ω-continuous). Thus:

$$[\![\mu X \bullet U(X)]\!] =_{df} \bigsqcap \{X \in W \mid [\![U(X)]\!] \sqsupseteq X\}$$

satisfies:

$$\mu X \bullet U(X) = U(\mu X \bullet U(X)) \qquad (\mu)$$

It can be characterised as follows:

$$\bigsqcup_{i<\kappa} U^i(\mathit{chaos}) = \mu X \bullet U(X)$$

for some suitably fuck-off-huge ordinal κ.

Primitive recursive procedures ...

It takes 5 schema operators to specify primitive recursion over the natural numbers:

$$\mathtt{proc}\ f(\mathsf{x})\ \mathtt{cases}\ \mathsf{x}\ \mathtt{in}\ 0 : U_0;\ \mathsf{m}+1 : U_1(f(\mathsf{m}))\ \mathtt{endcases} =_{df}$$
$$\mu X \bullet U_0 \uparrow \mathsf{x} = 0 \wedge \exists \mathsf{m} \bullet U_1(X[\mathsf{x} \Rrightarrow \mathsf{m}]) \uparrow \mathsf{x} = \mathsf{m}+1$$

Introduction rule:

$$\frac{z.\mathsf{x} = 0 \vdash z \mathrel{\dot\in} U_0 \quad z.\mathsf{x} = z.\mathsf{m}+1 \vdash z \mathrel{\dot\in} U_1(f(\mathsf{m}))}{z \mathrel{\dot\in} f}$$

Elimination rules:

$$\frac{z \mathrel{\dot\in} f \quad z.\mathsf{x} = 0}{z \mathrel{\dot\in} U_0} \qquad \frac{z \mathrel{\dot\in} f \quad z.\mathsf{x} = \mathsf{m}+1}{z \mathrel{\dot\in} U_1(f(\mathsf{m}))}$$

Primitive recursive procedures ...

Introduction rule:

$$\frac{z.\mathsf{x} = 0 \vdash z \mathrel{\dot\in} U_0 \quad z.\mathsf{x} = z.\mathsf{m} + 1 \vdash z \mathrel{\dot\in} U_1(f(\mathsf{m}))}{z \mathrel{\dot\in} f}$$

Proof:

$$\dfrac{\dfrac{\dfrac{}{z.\mathsf{x} = 0}\,(1)}{\vdots}\quad\dfrac{\dfrac{\dfrac{}{z.\mathsf{x} = z.\mathsf{m} + 1}\,(2)}{\vdots}}{\dfrac{z \mathrel{\dot\in} U_1[f(\mathsf{m})]}{z \mathrel{\dot\in} U_1[f(\mathsf{m})] \uparrow \mathsf{x} = \mathsf{m} + 1}\,(2)}}{\dfrac{z \mathrel{\dot\in} U_0}{z \mathrel{\dot\in} U_0 \uparrow \mathsf{x} = 0}\,(1) \qquad \dfrac{}{z \mathrel{\dot\in} \exists \mathsf{m} \bullet U_1[f(\mathsf{m})] \uparrow \mathsf{x} = \mathsf{m} + 1}}$$

$$\frac{z \mathrel{\dot\in} U_0 \uparrow \mathsf{x} = 0 \wedge \exists \mathsf{m} \bullet U_1[f(\mathsf{m})] \uparrow \mathsf{x} = \mathsf{m} + 1}{z \mathrel{\dot\in} f}\,(\mu)$$

Primitive recursive procedures ...

The rules for $\mathbb{N}$ are as follows:

$$\frac{}{0 \in \mathbb{N}}\ (\mathbb{N}_0^+) \qquad \frac{n \in \mathbb{N}}{n+1 \in \mathbb{N}}\ (\mathbb{N}_1^+) \qquad \frac{n \in \mathbb{N}}{0 \neq n+1}$$

$$\frac{n+1 = m+1}{n = m} \qquad \frac{P(0) \quad m \in \mathbb{N}, P(m) \vdash P(m+1)}{n \in \mathbb{N} \vdash P(n)}\ (\mathbb{N}^-)$$

Primitive recursive procedures ...

Characterising a procedure in terms of (all) its invocations:

$$\frac{n \in \mathbb{N} \vdash f(n) \sqsupseteq U[\![n]\!]}{f \sqsupseteq U}\ (inv_{\mathbb{N}})$$

Proof:

$$\dfrac{\dfrac{\dfrac{}{z \in f}\ (1) \quad \dfrac{}{z.\mathsf{x} = z.\mathsf{x}}}{z \in f(z.\mathsf{x})} \qquad \begin{array}{c} \dfrac{}{z.\mathsf{x} \in \mathbb{N}} \\ \vdots \\ f(z.\mathsf{x}) \sqsupseteq U[\![z.\mathsf{x}]\!] \end{array}}{\dfrac{z \in U[\![z.\mathsf{x}]\!]}{\dfrac{z \in U}{f \sqsupseteq U}\ (1)}}$$

Primitive recursive procedures ...

The rule for *recursive synthesis*:

$$\frac{U_0 \sqsupseteq U[0] \quad f(\mathtt{m}) \sqsupseteq U[\mathtt{m}] \vdash U_1(f(\mathtt{m})) \sqsupseteq U[\mathtt{m}+1]}{f \sqsupseteq U}$$

Proof:

$$\dfrac{\dfrac{\vdots}{f(0) \sqsupseteq U[0]} \quad \dfrac{\overline{f(\mathtt{m}) \sqsupseteq U[\mathtt{m}]}}{\vdots\ f(\mathtt{m}+1) \sqsupseteq U[\mathtt{m}+1]}\ \mathbb{N}^-}{\dfrac{f(n) \sqsupseteq U[n]}{f \sqsupseteq U}\ (inv_{\mathbb{N}})}$$

Primitive recursive procedures ...

Base case:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z \in f(0)}\,(2)}{z.\mathsf{x} = 0} \qquad \dfrac{\dfrac{}{z \in f(0)}\,(2)}{z \in f}}{z \in U_0} \qquad U_0 \sqsupseteq U[0]}{\dfrac{z \in U[0]}{f(0) \sqsupseteq U[0]}\,(2)}$$

Induction case:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{z \in f(\mathsf{m})}\,(3)}{z.\mathsf{x} = \mathsf{m}+1} \qquad \dfrac{\dfrac{}{z \in f(\mathsf{m})}\,(3)}{z \in f}}{z \in U_1(f(\mathsf{m}))} \qquad \begin{array}{c}\dfrac{}{f(\mathsf{m}) \sqsupseteq U[\mathsf{m}]}\,(0)\\ \vdots \\ U_1(f(\mathsf{m})) \sqsupseteq U[\mathsf{m}+1]\end{array}}{\dfrac{z \in U[\mathsf{m}+1]}{f(\mathsf{m}+1) \sqsupseteq U[\mathsf{m}+1]}\,(3)}$$

Other types ...

Lists:

$$\begin{array}{lll} \texttt{proc} & f(\texttt{x})\ \texttt{cases x in} & \\ & \texttt{Nil} & : U_0; \\ & \texttt{Cons m}_0\ \texttt{m}_1 & : U_1(f(\texttt{m}_1))\ \texttt{endcases} =_{df} \end{array}$$

$$\begin{array}{lll} \mu X \bullet & U_0 \uparrow \texttt{x} = \texttt{Nil} \wedge & \\ & \exists \texttt{m}_0, \texttt{m}_1 \bullet U_1(X[\texttt{x} \Rrightarrow \texttt{m}_1]) \uparrow \texttt{x} = \texttt{Cons m}_0\ \texttt{m}_1 & \end{array}$$

Other types ...

Trees:

$$\begin{array}{ll} \texttt{proc} & f(\mathtt{x})\ \texttt{cases}\ \mathtt{x}\ \texttt{in} \\ & \texttt{Leaf}\ \mathtt{m}_0 \qquad\qquad : U_0; \\ & \texttt{Node}\ \mathtt{m}_1\ \mathtt{m}_2 \qquad : U_1(f(\mathtt{m}))\ \texttt{endcases} =_{df} \end{array}$$

$$\begin{array}{ll} \mu X \bullet & \exists\, \mathtt{m}_0 \bullet U_0 \uparrow \mathtt{x} = \texttt{Leaf}\ \mathtt{m}_0 \wedge \\ & \exists\, \mathtt{m}_1, \mathtt{m}_2 \bullet U_1(X[\mathtt{x} \Rrightarrow \mathtt{m}_1], X[\mathtt{x} \Rrightarrow \mathtt{m}_1]) \uparrow \mathtt{x} = \texttt{Node}\ \mathtt{m}_1\ \mathtt{m}_2 \end{array}$$

The “frame problem” ...

At least four people in this room have worried about this issue in the past!

Dec

$$x, x' : \mathbb{N}$$

$$x > 0$$

$$x' = x - 1$$

Solutions:

– Refinement calculus: `skip` outside the frame.
– Henson-Reeves FACJ: *Chaos* outside the frame.
– νZ: *silent* outside the frame.

Conclusions ...