

Rely-Guarantee Frames

Ken Pierce

Newcastle University

2009-11-23

Overview

- Simplifying rely-guarantee in VDM
- Disjoint concurrency (static)
- Reduce rely-conditions
- Reduce proof effort

Rely-Guarantee Reasoning

- Shared variable concurrency
- Steps of *program* or *environment*
- Rely-condition — assumption
- Guarantee-condition — commitment

Problem: Whole-State Updates

- Must reference all variables (i.e. whole-state)
- Cluttered for large states
- Not all are required

Problem: Whole-State Updates

- Must reference all variables (i.e. whole-state)
- Cluttered for large states
- Not all are required

Problem: Whole-State Updates

- Must reference all variables (i.e. whole-state)
- Cluttered for large states
- Not all are required

Inspiration from RGSep

- Local and shared state
- Local state is not subject to interference
- Rely-guarantee for shared state
- Disjoint concurrency (dynamic)
- Reduces clauses in rely-guarantee

Disjoint Concurrency in Rely-Guarantee/VDM

- Exclusive write access
- No interference
- Static disjoint concurrency [Hoa72]
- Frames / externals clause

Disjoint Concurrency in Rely-Guarantee/VDM

- Exclusive write access
- No interference
- Static disjoint concurrency [Hoa72]
- Frames / externals clause

Disjoint Concurrency in Rely-Guarantee/VDM

- Exclusive write access
- No interference
- Static disjoint concurrency [Hoa72]
- Frames / externals clause

Disjoint Concurrency in Rely-Guarantee/VDM

- Exclusive write access
- No interference
- Static disjoint concurrency [Hoa72]
- Frames / externals clause

Externals Clause

OP

rd $x:\mathbb{N}$

wr $y:\mathbb{N}$

...

- **local** x — only this operation can read or write x .
- **owns wr** x — only this operation can write x , but other operations may read it.
- **wr** x — this operation can read or write x , but other operations may read or write it.
- **rd** x — this operation can read, but not write, x .

Externals Clause

OP

rd $x:\mathbb{N}$

wr $y:\mathbb{N}$

...

- **local** x — only this operation can read or write x .
- **owns wr** x — only this operation can write x , but other operations may read it.
- **wr** x — this operation can read or write x , but other operations may read or write it.
- **rd** x — this operation can read, but not write, x .

Externals Clause

OP

rd $x:\mathbb{N}$

wr $y:\mathbb{N}$

...

- **local** x — only this operation can read or write x .
- **owns wr** x — only this operation can write x , but other operations may read it.
- **wr** x — this operation can read or write x , but other operations may read or write it.
- **rd** x — this operation can read, but not write, x .

Proof Effort \rightarrow Static Checks

- Fewer rely-conditions
- Must check write-frames are disjoint
- Static checks instead of proof effort

Proof Effort \rightarrow Static Checks

- Fewer rely-conditions
- Must check write-frames are disjoint
- Static checks instead of proof effort

Proof Effort \rightarrow Static Checks

- Fewer rely-conditions
- Must check write-frames are disjoint
- Static checks instead of proof effort

Frame Access Functions

local-OP

ownswr-OP

wr-OP

rd-OP

writes-OP \triangleq *ownswr-OP* \cup *wr-OP*

frame-OP \triangleq *local-OP* \cup *ownswr-OP* \cup *wr-OP* \cup *rd-OP*

Frames Formally

Definition A framed operation, $\{P, R\} OP \{G, Q\}$, may have a rely-condition R which does not reference the variables in the set $ownswr-OP$.

Theorem 1 A framed operation, $\{P, R\} OP \{G, Q\}$, can be rewritten as a standard rely-guarantee specification, $\{P, R'\} OP' \{G, Q\}$, by selecting R' such that $\overline{R'} - \overline{R} = (ownswr-OP \triangleleft I)$.

Theorem 2 Two framed operations, $\{P_1, R_1\} OP_1 \{G_1, Q_1\}$ and $\{P_2, R_2\} OP_2 \{G_2, Q_2\}$, can be checked using standard rely-guarantee rules only when the write-frames are disjoint:

$$ownswr-S_1 \cap writes-S_2 = \{\} \wedge ownswr-S_2 \cap writes-S_1 = \{\}$$

Frames Formally

Definition A framed operation, $\{P, R\} OP \{G, Q\}$, may have a rely-condition R which does not reference the variables in the set $ownswr-OP$.

Theorem 1 A framed operation, $\{P, R\} OP \{G, Q\}$, can be rewritten as a standard rely-guarantee specification, $\{P, R'\} OP' \{G, Q\}$, by selecting R' such that $\overline{R'} - \overline{R} = (ownswr-OP \triangleleft I)$.

Theorem 2 Two framed operations, $\{P_1, R_1\} OP_1 \{G_1, Q_1\}$ and $\{P_2, R_2\} OP_2 \{G_2, Q_2\}$, can be checked using standard rely-guarantee rules only when the write-frames are disjoint:

$$ownswr-S_1 \cap writes-S_2 = \{\} \wedge ownswr-S_2 \cap writes-S_1 = \{\}$$

Frames Formally

Definition A framed operation, $\{P, R\} OP \{G, Q\}$, may have a rely-condition R which does not reference the variables in the set $ownswr-OP$.

Theorem 1 A framed operation, $\{P, R\} OP \{G, Q\}$, can be rewritten as a standard rely-guarantee specification, $\{P, R'\} OP' \{G, Q\}$, by selecting R' such that $\overline{R'} - \overline{R} = (ownswr-OP \triangleleft I)$.

Theorem 2 Two framed operations, $\{P_1, R_1\} OP_1 \{G_1, Q_1\}$ and $\{P_2, R_2\} OP_2 \{G_2, Q_2\}$, can be checked using standard rely-guarantee rules only when the write-frames are disjoint:

$$ownswr-S_1 \cap writes-S_2 = \{\} \wedge ownswr-S_2 \cap writes-S_1 = \{\}$$

Conclusions

- Exclusive write acces with **owns wr**
- Static disjoint concurrency
- Not RGSep etc., but *useful*
- Reduction of rely-conditions: simpler specifications
- Proof effort → static checks

Conclusions

- Exclusive write acces with **owns wr**
- Static disjoint concurrency
- Not RGSep etc., but *useful*
- Reduction of rely-conditions: simpler specifications
- Proof effort → static checks

Conclusions

- Exclusive write acces with **owns wr**
- Static disjoint concurrency
- Not RGSep etc., but *useful*
- Reduction of rely-conditions: simpler specifications
- Proof effort → static checks

Conclusions

- Exclusive write acces with **owns wr**
- Static disjoint concurrency
- Not RGSep etc., but *useful*
- Reduction of rely-conditions: simpler specifications
- Proof effort → static checks

Conclusions

- Exclusive write acces with **owns wr**
- Static disjoint concurrency
- Not RGSep etc., but *useful*
- Reduction of rely-conditions: simpler specifications
- Proof effort → static checks

Further Applications for Frames

- Disjoint frames, i.e. $frame-OP_1 \cap frame-OP_2 = \{ \}$
- Inheritance of frames
- Guide for designers
- ...

Further Applications for Frames

- Disjoint frames, i.e. $frame-OP_1 \cap frame-OP_2 = \{ \}$
- Inheritance of frames
- Guide for designers
- ...

Further Applications for Frames

- Disjoint frames, i.e. $frame-OP_1 \cap frame-OP_2 = \{ \}$
- Inheritance of frames
- Guide for designers
- ...

Further Applications for Frames

- Disjoint frames, i.e. $frame-OP_1 \cap frame-OP_2 = \{ \}$
- Inheritance of frames
- Guide for designers
- ...

Thanks for Listening

- Any questions?