

Non-Interference for Multi-Agent Workflows

Helmut Seidl

Joint work with:

Christian Müller, Bernd Finkbeiner

TU München + U. des Saarlandes

MOD 2019

Lecture 4:

Playing with NI

Leader Election Game

```
while (*) {  
  choose   $\text{Msg}(m, y) := \text{Msg}(m, y) \vee B(m, x)$   
  or {  
     $\text{Msg}(m, x) := \text{Msg}(m, x) \exists y.$   
     $\text{Next}(y, x) \wedge \text{Msg}(m, y) \wedge \text{Less}(y, m);$   
     $\text{Leader}(x) := \text{Leader}(x) \vee \text{Msg}(x, x);$   
  }  
}
```

// assuming appropriate axiomatization

// for Next , Less

Leader Election Game

```
while (*) {  
  choose  Msg(m, y) := Msg(m, y)  $\vee$  B(m, x)  
  or {  
    Msg(m, x) := Msg(m, x)  $\exists$  y.  
    Next(y, x)  $\wedge$  Msg(m, y)  $\wedge$  Less(y, m);  
    Leader(x) := Leader(x)  $\vee$  Msg(x, x);  
  }  
}
```

// assuming appropriate axiomatization

// for Next, Less

Synthesizing message contents??

Workflow Game

NoConflict(x, p) := \neg Author(x, p) \wedge $A_1(x, p)$;

// A_1 input predicate

Assign(x, p) := $B(x, p)$;

// strategy of pc chair

Report(x, p, r) := Assign(x, p) \wedge $A_2(x, p, r)$;

while (*) Discuss(x_1, x_2, p, d, r_1, r_2) := $\exists r_1 r_2$.

Report(x_1, p, r_1) \wedge

Report(x_2, p, r_2) \wedge

$A_3(x_1, x_2, p, d)$;

FO Safety Game

Reachability Player controls

FO Safety Game

Reachability Player controls

- ▶ universe and valuation

FO Safety Game

Reachability Player controls

- ▶ universe and valuation
- ▶ flow of control

FO Safety Game

Reachability Player controls

- ▶ universe and valuation
- ▶ flow of control
- ▶ subset of input predicates A

FO Safety Game

Reachability Player controls

- ▶ universe and valuation
- ▶ flow of control
- ▶ subset of input predicates A

Safety Player controls

- ▶ subset of input predicates B
- // e.g., for paper assignment

FO Safety Game

Reachability Player controls

- ▶ universe and valuation
- ▶ flow of control
- ▶ subset of input predicates A

Safety Player controls

- ▶ subset of input predicates B
- // e.g., for paper assignment

Initial Assumption Init, v_0

Objective

Invariant $I : \text{Nodes} \rightarrow \text{FOL}$, e.g.,

$$\forall x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Report}(x, p, r)$$

Games by Weakest Precondition

Given

- edges of reachability player E_A
- edges of safety player E_B
- invariant I
- initial condition I_{init} for v_0

Games by Weakest Precondition

Given edges of reachability player E_A
edges of safety player E_B
invariant I
initial condition Init for v_0

Compute

$$\Psi^{(0)}[u] = I[u]$$

$$\Psi^{(h)}[u] = \Psi^{(h-1)}[u] \wedge \bigwedge_{(u,\theta,v) \in E_A} \forall A. (\Psi^{(h-1)}[v]\theta)$$

Games by Weakest Precondition

Given edges of reachability player E_A
edges of safety player E_B
invariant I
initial condition Init for v_0

Compute

$$\Psi^{(0)}[u] = I[u]$$

$$\begin{aligned}\Psi^{(h)}[u] = & \Psi^{(h-1)}[u] \wedge \\ & \bigwedge_{(u,\theta,v) \in E_A} \forall A. (\Psi^{(h-1)}[v]\theta) \wedge \\ & \bigwedge_{(u,\theta,v) \in E_B} \exists B. (\Psi^{(h-1)}[v]\theta) \\ & \text{for } h > 0\end{aligned}$$

Games by Weakest Precondition

Given edges of reachability player E_A
edges of safety player E_B
invariant I
initial condition Init for v_0

Compute

$$\Psi^{(0)}[u] = I[u]$$

$$\begin{aligned}\Psi^{(h)}[u] = & \Psi^{(h-1)}[u] \wedge \\ & \bigwedge_{(u,\theta,v) \in E_A} \forall A. (\Psi^{(h-1)}[v]\theta) \wedge \\ & \bigwedge_{(u,\theta,v) \in E_B} \exists B. (\Psi^{(h-1)}[v]\theta) \\ & \text{for } h > 0\end{aligned}$$

Then safety player wins iff

$$\text{Init} \Rightarrow \Psi^{(h)}[v_0] \quad (h \geq 0)$$

Simplified HotCRP

$\text{Conflict}(x, p) := \text{Author}(x, p) \vee A_2(x, p);$

// A_2 input predicate

$\text{Assign}(x, p) := B(x, p);$

// strategy of pc chair

$\text{Report}(x, p, r) := \text{Assign}(x, p) \wedge A_3(x, p, r)$

Simplified HotCRP

Conflict(x, p) := Author(x, p) \vee $A_2(x, p)$;

// A_2 input predicate

Assign(x, p) := $B(x, p)$;

// strategy of pc chair

Report(x, p, r) := Assign(x, p) \wedge $A_3(x, p, r)$

Invariant

$$I[v_3] = \forall x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Report}(x, p, r)$$

Simplified HotCRP (cont.)

$$\psi^{(0)}[v_3] = \forall x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Report}(x, p, r)$$

$$\begin{aligned} \psi^{(1)}[v_2] &= \forall A_2, x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Assign}(x, p) \vee \neg A_2(x, p, r) \\ &= \forall x, p. \neg \text{Author}(x, p) \vee \neg \text{Assign}(x, p) \end{aligned}$$

$$\psi^{(2)}[v_1] = \exists B. \forall x, p. \neg \text{Author}(x, p) \vee \neg B(x, p)$$

Simplified HotCRP (cont.)

$$\psi^{(0)}[v_3] = \forall x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Report}(x, p, r)$$

$$\begin{aligned}\psi^{(1)}[v_2] &= \forall A_2, x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Assign}(x, p) \vee \neg A_2(x, p, r) \\ &= \forall x, p. \neg \text{Author}(x, p) \vee \neg \text{Assign}(x, p)\end{aligned}$$

$$\begin{aligned}\psi^{(2)}[v_1] &= \exists B. \forall x, p. \neg \text{Author}(x, p) \vee \neg B(x, p) \\ &= \top\end{aligned}$$

Simplified HotCRP (cont.)

$$\psi^{(0)}[v_3] = \forall x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Report}(x, p, r)$$

$$\begin{aligned}\psi^{(1)}[v_2] &= \forall A_2, x, p, r. \neg \text{Author}(x, p) \vee \neg \text{Assign}(x, p) \vee \neg A_2(x, p, r) \\ &= \forall x, p. \neg \text{Author}(x, p) \vee \neg \text{Assign}(x, p)\end{aligned}$$

$$\begin{aligned}\psi^{(2)}[v_1] &= \exists B. \forall x, p. \neg \text{Author}(x, p) \vee \neg B(x, p) \\ &= \top\end{aligned}$$

\implies pc chair has a winning strategy!

The Monadic Case

Theorem

Igor Walukiewicz

Safety for Monadic Safety Games (w/o $=$, \neq) is
undecidable.

The Monadic Case

Theorem

Igor Walukiewicz

Safety for Monadic Safety Games (w/o $=, \neq$) is
undecidable.

Proof $x \neq y \equiv \exists B. Bx \wedge \neg By$

The Monadic Case

Theorem

Igor Walukiewicz

Safety for Monadic Safety Games (w/o $=, \neq$) is
undecidable.

Proof $x \neq y \equiv \exists B. Bx \wedge \neg By$

Theorem

Safety remains decidable when there are B only.

The Monadic Case

Theorem

Igor Walukiewicz

Safety for Monadic Safety Games (w/o $=, \neq$) is undecidable.

Proof $x \neq y \equiv \exists B. Bx \wedge \neg By$

Theorem

Safety remains decidable when there are B only.

Proof

QE of $\exists B$ introduces $\neq \implies$
compute weakest strengthening without \neq

Example

Consider the game with single edge (u, θ, u) where

$$\theta = \begin{cases} Ry & \mapsto Ry \wedge \neg By, \\ \text{err} & \mapsto \text{err} \vee \forall x. \neg (Rx \wedge Bx) \end{cases}$$

Example

Consider the game with single edge (u, θ, u) where

$$\theta = \begin{cases} Ry & \mapsto Ry \wedge \neg By, \\ \text{err} & \mapsto \text{err} \vee \forall x. \neg(Rx \wedge Bx) \end{cases}$$

Claim

Invariant $\neg \text{err}$ cannot be guaranteed by safety player.

Example (cont.)

$$\psi^{(h)}[u] = \neg \text{err} \wedge \exists x_1 \dots x_h. \bigwedge_{i=1}^h R x_i \wedge \bigwedge_{i < j} (x_i \neq x_j)$$

Example (cont.)

$$\psi^{(h)}[u] = \neg \text{err} \wedge \exists x_1 \dots x_h. \bigwedge_{i=1}^h Rx_i \wedge \bigwedge_{i < j} (x_i \neq x_j)$$

Strengthening $(x \neq y)^\# = (\neg Rx \leftrightarrow Ry)$

Example (cont.)

$$\psi^{(h)}[u] = \neg \text{err} \wedge \exists x_1 \dots x_h. \bigwedge_{i=1}^h Rx_i \wedge \bigwedge_{i < j} (x_i \neq x_j)$$

Strengthening $(x \neq y)^\# = (\neg Rx \leftrightarrow Ry)$

$$\begin{aligned}(\psi^{(h)}[u])^\# &= \neg \text{err} \wedge \exists x_1 \dots x_h. \bigwedge_{i=1}^h Rx_i \wedge \bigwedge_{i < j} (\neg Rx_i \leftrightarrow Rx_j) \\ &= \neg \text{err} \wedge \perp \\ &= \perp \quad \text{for } h \geq 2\end{aligned}$$

Example (cont.)

$$\Psi^{(h)}[U] = \neg \text{err} \wedge \exists x_1 \dots x_h. \bigwedge_{i=1}^h Rx_i \wedge \bigwedge_{i < j} (x_i \neq x_j)$$

Strengthening $(x \neq y)^\# = (\neg Rx \leftrightarrow Ry)$

$$\begin{aligned} (\Psi^{(h)}[U])^\# &= \neg \text{err} \wedge \exists x_1 \dots x_h. \bigwedge_{i=1}^h Rx_i \wedge \bigwedge_{i < j} (\neg Rx_i \leftrightarrow Rx_j) \\ &= \neg \text{err} \wedge \perp \\ &= \perp \quad \text{for } h \geq 2 \end{aligned}$$

$(\dots)^\#$ allows strengthened fixpoint iteration — which necessarily terminates.

Games by SO Quantifier Elimination

Safety reduces to

fixpoint iteration

FO implication checking

SO quantifier elimination

Games by SO Quantifier Elimination

Safety reduces to

- ▶ accelerated fixpoint iteration
- FO implication checking
- SO quantifier elimination

Games by SO Quantifier Elimination

Safety reduces to

- ▶ accelerated fixpoint iteration
- ▶ approximative FO implication checking
SO quantifier elimination

Games by SO Quantifier Elimination

Safety reduces to

- ▶ accelerated fixpoint iteration
- ▶ approximative FO implication checking
- ▶ strengthened SO quantifier elimination

SO Hilbert Choice Operator

Given: $\exists B. \phi$

SO Hilbert Choice Operator

Given: $\exists B. \phi$

Wanted: ψ with

$$\exists B. \phi \equiv \phi[\psi/B]$$

SO Hilbert Choice Operator

Given: $\exists B. \phi$

Wanted: ψ with

$$\exists B. \phi \equiv \phi[\psi/B]$$

\implies synthesis of predicate B



SO Hilbert Choice Operator

Given: $\exists B. \phi$

Wanted: ψ with

$$\exists B. \phi \equiv \phi[\psi/B]$$

\implies synthesis of predicate B



Note: $\exists B. \phi \Leftarrow \phi[\psi/B]$ for any ψ .

SO Existential Quantifier Elimination

Ackermann's Lemma

SO Existential Quantifier Elimination

Ackermann's Lemma

$$\exists B. \forall y. (F \vee By) \wedge (G \vee \neg By)$$



SO Existential Quantifier Elimination

Ackermann's Lemma

$$\begin{aligned} \exists B. \forall y. (F \vee By) \wedge (G \vee \neg By) \\ \equiv \forall y. (F \vee G) \end{aligned}$$



SO Existential Quantifier Elimination

Ackermann's Lemma

$$\begin{aligned} \exists B. \forall y. (F \vee By) \wedge (G \vee \neg By) \\ \equiv \quad \forall y. (F \vee G) \end{aligned}$$

where $\neg F \Rightarrow By \Rightarrow G$



SO Existential Quantifier Elimination

Ackermann's Counterexample

$$\exists B. B(a) \wedge \neg B(b) \wedge \forall y, y'. (\neg B(y') \vee \neg R(y', y) \vee B(y))$$

SO Existential Quantifier Elimination

Ackermann's Counterexample

$$\exists B. B(a) \wedge \neg B(b) \wedge \forall y, y'. (\neg B(y') \vee \neg R(y', y) \vee B(y))$$

is not expressible in FO logic.

(Simplified) DLS* Algorithm

Andrzej Szalas et al., 1993, 1997



(Simplified) DLS* Algorithm

Andrzej Szalas et al., 1993, 1997

Assume

$$\begin{aligned}\psi \equiv & E \wedge \\ & \forall y. (F \vee By) \wedge \\ & \forall y'. (G \vee \neg By') \wedge \\ & \forall y, y'. (H \vee By \vee \neg By')\end{aligned}$$



(Simplified) DLS* Algorithm

Andrzej Szalas et al., 1993, 1997

Assume

$$\begin{aligned}\psi \equiv & E \wedge \\ & \forall y. (F \vee By) \wedge \\ & \forall y'. (G \vee \neg By') \wedge \\ & \forall y, y'. (H \vee By \vee \neg By')\end{aligned}$$

Define

$$H \circ H' \equiv \forall y_1. H[y_1/y'] \vee H'[y_1/y]$$



(Simplified) DLS* Algorithm

Andrzej Szalas et al., 1993, 1997

Assume

$$\begin{aligned}\psi \equiv & E \wedge \\ & \forall y. (F \vee B y) \wedge \\ & \forall y'. (G \vee \neg B y') \wedge \\ & \forall y, y'. (H \vee B y \vee \neg B y')\end{aligned}$$

Define $H \circ H' \equiv \forall y_1. H[y_1/y'] \vee H'[y_1/y]$

Then

$$\exists B. \psi \equiv E \wedge \bigwedge_{j \geq 0} G \circ H^j \circ F$$

— whenever $\exists B. \psi$ is FO definable



SO Hilbert Choice

Define

$$\gamma^* \equiv \neg E \vee \bigwedge_{j \geq 0} G \circ H^j[y/y'].$$

SO Hilbert Choice

Define

$$\gamma^* \equiv \neg E \vee \bigwedge_{j \geq 0} G \circ H^j[y/y'].$$

Then

$$\gamma^* \equiv \neg E \vee \bigwedge_{j=0}^k G \circ H^j[y/y']$$

— whenever conjunction is FO definable

SO Hilbert Choice

Define $\gamma^* \equiv \neg E \vee \bigwedge_{j \geq 0} G \circ H^j[y/y']$.

Then $\gamma^* \equiv \neg E \vee \bigwedge_{j=0}^k G \circ H^j[y/y']$

— whenever conjunction is FO definable

Moreover,

$$\exists B. \psi \equiv \psi[\gamma^*/B]$$

SO Hilbert Choice

Define $\gamma^* \equiv \neg E \vee \bigwedge_{j \geq 0} G \circ H^j[y/y']$.

Then $\gamma^* \equiv \neg E \vee \bigwedge_{j=0}^k G \circ H^j[y/y']$

— whenever conjunction is FO definable

Moreover,

$$\exists B. \psi \equiv \psi[\gamma^*/B]$$

Approximate Choice Operator

$$\gamma_k \equiv \neg E \vee \bigwedge_{j=0}^k G \circ H^j[y/y']$$

SO Hilbert Choice

Define $\gamma^* \equiv \neg E \vee \bigwedge_{j \geq 0} G \circ H^j[y/y']$.

Then $\gamma^* \equiv \neg E \vee \bigwedge_{j=0}^k G \circ H^j[y/y']$

— whenever conjunction is FO definable

Moreover,

$$\exists B. \psi \equiv \psi[\gamma^*/B]$$

Approximate Choice Operator

$$\gamma_k \equiv \neg E \vee \bigwedge_{j=0}^k G \circ H^j[y/y']$$

— precise for universes up to size h with $k \geq h'$

Disproving Safety

Symbolic Execution for Universes up to Size h

Disproving Safety

Symbolic Execution for Universes up to Size h

- ▶ Introduce FO variables $y_1 \dots y_h$;

Disproving Safety

Symbolic Execution for Universes up to Size h

- ▶ Introduce FO variables $y_1 \dots y_h$;
- ▶ Replace $\forall x.\phi$ with $\bigwedge_j \phi[y_j/x]$;
- ▶ Replace $\exists x.\phi$ with $\bigvee_j \phi[y_j/x]$;

Disproving Safety

Symbolic Execution for Universes up to Size h

- ▶ Introduce FO variables $y_1 \dots y_h$;
- ▶ Replace $\forall x.\phi$ with $\bigwedge_j \phi[y_j/x]$;
- ▶ Replace $\exists x.\phi$ with $\bigvee_j \phi[y_j/x]$;
- ▶ Iterate

Disproving Safety

Symbolic Execution for Universes up to Size h

- ▶ Introduce FO variables $y_1 \dots y_h$;
- ▶ Replace $\forall x.\phi$ with $\bigwedge_j \phi[y_j/x]$;
- ▶ Replace $\exists x.\phi$ with $\bigvee_j \phi[y_j/x]$;
- ▶ Iterate
- ▶ Thereby, **eliminate** $\forall A, \exists B \dots$

Disproving Safety

Symbolic Execution for Universes up to Size h

- ▶ Introduce FO variables $y_1 \dots y_h$;
- ▶ Replace $\forall x.\phi$ with $\bigwedge_j \phi[y_j/x]$;
- ▶ Replace $\exists x.\phi$ with $\bigvee_j \phi[y_j/x]$;
- ▶ Iterate
- ▶ Thereby, **eliminate** $\forall A, \exists B \dots$

\implies iteration terminates

\implies safety upto size h is decidable.

Proving Safety

- ▶ Keep universal FO quantifiers!
- ▶ Strengthen **existential** FO quantifiers!
// precise up to size h

Proving Safety

- ▶ Keep universal FO quantifiers!
- ▶ Strengthen **existential** FO quantifiers!
// precise up to size h
- ▶ Strengthen **existential** SO quantifiers!
// precise up to size h

Proving Safety

- ▶ Keep universal FO quantifiers!
- ▶ Strengthen **existential** FO quantifiers!
// precise up to size h
- ▶ Strengthen **existential** SO quantifiers!
// precise up to size h
- ▶ Iterate!

Proving Safety

- ▶ Keep universal FO quantifiers!
 - ▶ Strengthen **existential** FO quantifiers!
// precise up to size h
 - ▶ Strengthen **existential** SO quantifiers!
// precise up to size h
 - ▶ Iterate!
- ⇒ terminates — whenever result is **FO definable**
- ⇒ either proves safety or provides
(possibly spurious) counterexample of size $> h$

Proving Safety

- ▶ Keep universal FO quantifiers!
 - ▶ Strengthen **existential** FO quantifiers!
// precise up to size h
 - ▶ Strengthen **existential** SO quantifiers!
// precise up to size h
 - ▶ Iterate!
- ⇒ terminates — whenever result is **FO definable**
- ⇒ either proves safety or provides
(possibly spurious) counterexample of size $> h$
- ⇒ guaranteed CEGAR loop!

Experiments

Implementation in **SCALA** using **Z3** as backend

Experiments

Implementation in **SCALA** using **Z3** as backend

Conference, Safety

| | Mode | Size | Approx. | Invariant | #Str. | Max. inv. |
|--|-----------|------|---------|-----------|-------|-----------|
| | synthesis | 6 | no | inferred | 4 | 50 |

Leader Election

| | Mode | Size | Approx. | Invariant | #Str. | Max. inv. |
|--|-----------|------|---------|------------------|-------|-----------|
| | verify | 4 | no | proven inductive | 0 | 42 |
| | synthesis | 4 | no | proven inductive | 0 | 42 |

Conference, NI

| Model | Mode | Size | Approx. | Invariant | #Str. | Max. inv. |
|----------------|-----------|------|---------|----------------|-------|-----------|
| stubborn | verify | 6 | no | inferred | 4 | 850 |
| stubborn | synthesis | 6 | no | inferred | 4 | 850 |
| acyclic causal | synthesis | 8 | no | inferred | 4 | 137 |
| causal | synthesis | 11 | yes | inferred | 8 | 5090 |
| causal | verify | 11 | no | counterexample | 7 | - |

Summary

- ▶ We have introduced a robust concept of FO safety games that extend FO transition systems.

Summary

- ▶ We have introduced a robust concept of FO safety games that extend FO transition systems.
- ▶ Interesting **synthesis** problems can be formalized within that framework.

Summary

- ▶ We have introduced a robust concept of FO safety games that extend FO transition systems.
- ▶ Interesting **synthesis** problems can be formalized within that framework.
- ▶ For monadic safety games decidable subclasses could be identified.

Summary

- ▶ We have introduced a robust concept of FO safety games that extend FO transition systems.
- ▶ Interesting **synthesis** problems can be formalized within that framework.
- ▶ For monadic safety games decidable subclasses could be identified.
- ▶ In general, strengthening by **universal formulas** together with (approximate) **SO quantifier elimination** allowed to construct a CEGAR loop for proving/disproving safety.

Thank you!