Higher-Order Fixpoint Logic

Martin Lange

University of Kassel, Germany

NII Shonan Seminar on Higher-Order Model Checking
14/03/16 – 17/03/16

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus
2. Higher-Order Fixpoint Logic
3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness
4. Further Work

The Modal μ-Calculus

multi-modal logic + extremal fixpoint quantifiers

$$\varphi \quad ::= \quad q \mid X \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle a \rangle \varphi \mid \mu X.\varphi$$

usual abbreviations: $\varphi \wedge \psi$, $\varphi \to \psi$, $[a]\varphi := \neg\langle a \rangle \neg\varphi$, $\nu X.\varphi := \neg\mu X.\neg\varphi[\neg X/X]$

interpreted over transition system $\mathcal{T} = (S, \{\xrightarrow{a} \mid a \in A\}, L : S \to 2^P)$

semantics usually given as $[\![\varphi]\!]^{\mathcal{T}}_{\rho} \subseteq S$ with Knaster-Tarski

Examples

typical $\mathcal{L}_\mu$-definable properties:

- $\nu X.\langle a \rangle X$

There is an infinite a-path

Examples

typical $\mathcal{L}_\mu$-definable properties:

- $\nu X.\langle a \rangle X$
- $\mu X.p \vee (\Diamond \mathtt{tt} \wedge \Box X)$ ($\equiv \mathtt{AF}p$ in CTL)

on every path somewhere p

Examples

typical $\mathcal{L}_\mu$-definable properties:

- $\nu X.\langle a\rangle X$
- $\mu X.p \vee (\Diamond \mathtt{tt} \wedge \Box X)$ ($\equiv$ AFp in CTL)
- $\mu X.[a]X$

all a-paths are finite

Examples

typical $\mathcal{L}_\mu$-definable properties:

- $\nu X.\langle a\rangle X$
- $\mu X.p \vee (\Diamond \mathtt{tt} \wedge \Box X)$ ($\equiv \mathtt{AF}p$ in CTL)
- $\mu X.[a]X$
- $\nu X.\mu Y.(p \wedge \Diamond X) \vee \Diamond Y$

there is a path on which p holds infinitely often

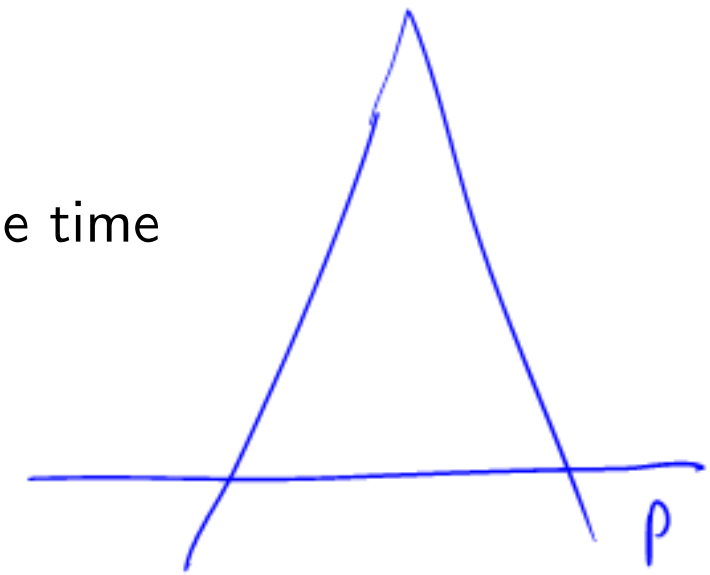
The Expressive Power of $\mathcal{L}_\mu$

Theorem 1 (Emerson/Jutla '88; Janin/Walukiewicz '96)

A bisimulation-invariant tree language is $\mathcal{L}_\mu$-definable iff it is regular

typical properties that are not $\mathcal{L}_\mu$-definable:

- uniform inevitability,
  something holds on all paths at the same time



The Expressive Power of $\mathcal{L}_\mu$

Theorem 1 (Emerson/Jutla '88; Janin/Walukiewicz '96)

A bisimulation-invariant tree language is $\mathcal{L}_\mu$-definable iff it is regular

typical properties that are not $\mathcal{L}_\mu$-definable:

- uniform inevitability,
  something holds on all paths at the same time
- unlimited counting like IO-buffer properties

Cmp. to CFL $\{w \in \{a,b\}^* \mid \forall v \underset{\text{(prefix)}}{\leq} w : |v|_a \geq |v|_b\}$

$S \to b \mid aSS$ generates minimal counterexamples

The Expressive Power of $\mathcal{L}_\mu$

Theorem 1 (Emerson/Jutla '88; Janin/Walukiewicz '96)

A bisimulation-invariant tree language is $\mathcal{L}_\mu$-definable iff it is regular

typical properties that are not $\mathcal{L}_\mu$-definable:

- uniform inevitability,
  something holds on all paths at the same time
- unlimited counting like IO-buffer properties
- repetitions of unbounded sequences of actions

cmp. to CSL $\{ww \mid w \in \{a,b\}^*\}$

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus

2. Higher-Order Fixpoint Logic

3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness

4. Further Work

Types

we need a simple type system with variances

$$\begin{array}{rcl} \tau & ::= & \mathsf{Pr} \mid \tau^{\nu} \to \tau \\ \nu & ::= & + \mid - \mid 0 \end{array}$$

because of right-associativity: $\tau = \tau_1^{\nu_1} \to \ldots \to \tau_m^{\nu_m} \to \mathsf{Pr}$

for a partial order $V = (M, \sqsubseteq)$ let

$$V^+ := (M, \sqsubseteq) \qquad V^- := (M, \sqsupseteq) \qquad V^0 := (M, =)$$

each type induces a complete lattice over transition system $\mathcal{T} = (\mathcal{S}, \to, L)$ using pointwise orderings $\sqsubseteq$

$$\begin{array}{rcl} [\![\mathsf{Pr}]\!] & := & (2^{\mathcal{S}}, \subseteq) \\ [\![\sigma^{\nu} \to \tau]\!] & := & ([\![\sigma]\!]^{\nu} \to_{\text{monotone}} [\![\tau]\!], \sqsubseteq) \end{array}$$

Formulas

HFL = modal μ-calculus + simply typed λ-calculus

[Viswanathan[2] '04]

$$\varphi \quad ::= \quad q \mid X \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle a \rangle \varphi \mid \mu(X : \tau).\varphi \mid \lambda(X^{\nu} : \tau).\varphi \mid \varphi\ \varphi$$

well-formedness condition given by type system

needed to exclude $\langle a \rangle q\ \langle b \rangle p$, $\mu X.\neg X$, etc.

Negation is Trickier

why not simple condition as in the modal μ-calculus

> *every fixpoint variable occurs under an even number of negation symbols in its defining fixpoint formula*

e.g. $\neg\mu X.\neg\mu Y.\langle a\rangle\neg X \vee \langle b\rangle Y$

λ-abstraction can shift negations into different branches of the syntax tree, e.g. $\mu X.(\lambda Y.\neg Y)\ X$

this formula is not well-formed

The Typing Rules

φ well-formed iff $\emptyset \vdash \varphi : \mathsf{Pr}$ is derivable

$$\frac{}{\Gamma \vdash q : \mathsf{Pr}} \qquad \frac{v \in \{0, +\}}{\Gamma, X^v : \tau \vdash X : \tau} \qquad \frac{\Gamma^- \vdash \varphi : \mathsf{Pr}}{\Gamma \vdash \neg\varphi : \mathsf{Pr}}$$

$$\frac{\Gamma \vdash \varphi : \mathsf{Pr} \quad \Gamma \vdash \psi : \mathsf{Pr}}{\Gamma \vdash \varphi \vee \psi : \mathsf{Pr}} \qquad \frac{\Gamma \vdash \varphi : \mathsf{Pr}}{\Gamma \vdash \langle a \rangle \varphi : \mathsf{Pr}} \qquad \frac{\Gamma, X^v : \sigma \vdash \varphi : \tau}{\Gamma \vdash \lambda(X^v : \sigma).\varphi : (\sigma^v \to \tau)}$$

$$\frac{\Gamma \vdash \varphi : (\sigma^+ \to \tau) \quad \Gamma \vdash \psi : \sigma}{\Gamma \vdash (\varphi\ \psi) : \tau} \qquad \frac{\Gamma \vdash \varphi : (\sigma^- \to \tau) \quad \Gamma^- \vdash \psi : \sigma}{\Gamma \vdash (\varphi\ \psi) : \tau}$$

$$\frac{\Gamma \vdash \varphi : (\sigma^0 \to \tau) \quad \Gamma \vdash \psi : \sigma \quad \Gamma^- \vdash \psi : \sigma}{\Gamma \vdash (\varphi\ \psi) : \tau} \qquad \frac{\Gamma, X^+ : \tau \vdash \varphi : \tau}{\Gamma \vdash \mu(X : \tau).\varphi : \tau}$$

Semantics of HFL

semantics of formula φ with $\emptyset \vdash \varphi : \tau$ is element of $[\![\tau]\!]$ over transition system $\mathcal{T} = (S, \to, L)$

$$
\begin{aligned}
[\![\Gamma \vdash q : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} &= \{s \in S \mid q \in L(s)\} \\
[\![\Gamma \vdash X : \tau]\!]^{\mathcal{T}}_{\eta} &= \eta(X) \\
[\![\Gamma \vdash \neg\varphi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} &= S \setminus [\![\Gamma^{-} \vdash \varphi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} \\
[\![\Gamma \vdash \neg\varphi : \sigma^{\nu} \to \tau]\!]^{\mathcal{T}}_{\eta} &= f \in [\![\sigma^{\nu} \to \tau]\!] \text{ s.t. } \bar{f} = [\![\Gamma^{-} \vdash \varphi : \sigma^{\nu} \to \tau]\!]^{\mathcal{T}}_{\eta} \\
[\![\Gamma \vdash \varphi \vee \psi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} &= [\![\Gamma \vdash \varphi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} \cup [\![\Gamma \vdash \psi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} \\
[\![\Gamma \vdash \langle a \rangle \varphi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta} &= \{s \in S \mid s \xrightarrow{a} t \text{ for some } t \in [\![\Gamma \vdash \varphi : \mathsf{Pr}]\!]^{\mathcal{T}}_{\eta}\} \\
[\![\Gamma \vdash \lambda(X^{\nu} : \sigma).\varphi : \sigma^{\nu} \to \tau]\!]^{\mathcal{T}}_{\eta} &= f \in [\![\sigma^{\nu} \to \tau]\!] \text{ s.t. } \forall x \in [\![\sigma]\!] \\
&\qquad f\ x = [\![\Gamma, X^{\nu} : \sigma \vdash \varphi : \tau]\!]^{\mathcal{T}}_{\eta[X \mapsto x]} \\
[\![\Gamma \vdash \varphi\ \psi : \tau]\!]^{\mathcal{T}}_{\eta} &= [\![\Gamma \vdash \varphi : \sigma^{\nu} \to \tau]\!]^{\mathcal{T}}_{\eta}\ [\![\Gamma' \vdash \psi : \sigma]\!]^{\mathcal{T}}_{\eta} \\
[\![\Gamma \vdash \mu(X : \tau)\varphi : \tau]\!]^{\mathcal{T}}_{\eta} &= \bigsqcap \{x \in [\![\tau]\!] \mid [\![\Gamma, X^{+} : \tau \vdash \varphi : \tau]\!]^{\mathcal{T}}_{\eta[X \mapsto x]} \sqsubseteq_{\tau} x\}
\end{aligned}
$$

Examples

what properties are expressed by the following formulas?

$\big(\mu F.\lambda X.X \vee \langle a\rangle(F\ \langle b\rangle X)\big)\ \mathtt{tt}$

$\langle a^* b^*\rangle \mathtt{tt}$

Examples

what properties are expressed by the following formulas?

$(\mu F.\lambda X.X \vee (F\ \Box X))\ \mathtt{ff}$

$\mathtt{ff} \vee \Box\mathtt{ff} \vee \Box\Box\mathtt{ff} \vee \ldots$ — i.e. finite — balanced trees

not the same as

$\mu X.\Box X$

Examples

what properties are expressed by the following formulas?

$$\varphi_{\text{word}} := \neg \bigvee_{a \neq b} \left(\mu F.\lambda X, Y.\, X \wedge Y \vee (F\ \Diamond X\ \Diamond Y)\right)\ \langle a\rangle \mathtt{tt}\ \langle b\rangle \mathtt{tt}$$

$$\neg \bigvee_{a \neq b} \left(\langle a \rangle \mathtt{tt} \wedge \langle b \rangle \mathtt{tt} \vee \Diamond\langle a \rangle \mathtt{tt} \wedge \Diamond\langle b \rangle \mathtt{tt} \vee \Diamond\Diamond\langle a \rangle \mathtt{tt} \wedge \Diamond\Diamond\langle b \rangle \mathtt{tt} \vee \ldots \right)$$

there are no $a \neq b$ s.t. both can be seen at the same distance, i.e. all paths are prefixes of one word

Examples

what properties are expressed by the following formulas?

$$\big(\mu F.\lambda g, g', g''.g \circ g' \circ g'' \vee (F\ g \circ \langle a \rangle\ g' \circ \langle b \rangle\ g'' \circ \langle c \rangle)\big)\ \mathit{id}\ \mathit{id}\ \mathit{id}\ \mathtt{tt}$$

where $\mathit{id} := \lambda X.X$, $\langle a \rangle := \lambda X.\langle a \rangle X$, and $f \circ g := \lambda X.f\ (g\ X)$

$\langle a^n b^n c^n \rangle \mathtt{tt}$

Examples

what properties are expressed by the following formulas?

$(\nu F.\lambda X.[b]X \wedge [a](F\ (F\ X)))\ \mathtt{ff}$

no buffer underflow (b = out, a = in)

cmp. to CFG $S \to b \mid aSS$

Examples

what properties are expressed by the following formulas?

$$\left(\mu F.\lambda g.g \circ g \vee \bigvee_{a \in \Sigma} (F\ g \circ \langle a \rangle)\right)\ id\ \mathrm{tt}$$

$$\bigvee_{w \in \Sigma^*} \langle w \rangle \mathrm{tt}$$

Examples

what properties are expressed by the following formulas?

$\psi_m\ \psi_{m-1}\ \ldots\ \psi_1\ \Diamond\ \Box\mathtt{ff}$ where $\psi_i := \lambda F.\lambda X.F\ (F\ X)$

Here is a maximal path of length $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}_{m}$

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus
2. Higher-Order Fixpoint Logic
3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness
4. Further Work

Goals

goal: understand HFL's model theory (and maybe even proof theory) as well as that of modal μ-calculus

- complexity of model checking / satisfiability checking
- an operational semantics / computational model for HFL, e.g. tree automata, model checking games
  - useful: negation normal form
- connection to model checking μ-calculus over higher-order recursion schemes
  - understanding fixpoint alternation
- expressive power

Fragments by Type Order

type order: $ord(\tau_1 \to \ldots \to \tau_m \to \mathsf{Pr}) = \max\{1 + ord(\tau_i)\}$

$\mathsf{HFL}^{k,m}$ = well-formed formulas using type annotations of order at most k and at most m arguments

recall examples above:

order 1: "balanced tree", "bisimilarity to a word"[1], all CFL path properties,[2] some CSL path properties

(1) "there is a path whose label belongs to CFL L" possible
"all paths belong to L" is not possible

(2) e.g. $\varphi_{\mathrm{word}} \wedge \langle a^n b^n c^* \rangle tt \wedge \langle a^* b^n c^n \rangle tt$
but does not just say $\langle a^n b^n c^n \rangle tt$!

Fragments by Type Order

type order: $ord(\tau_1 \to \ldots \to \tau_m \to \mathsf{Pr}) = \max\{1 + ord(\tau_i)\}$

$\mathsf{HFL}^{k,m}$ = well-formed formulas using type annotations of order at most k and at most m arguments

recall examples above:

order 1: "balanced tree", "bisimilarity to a word", all CFL path properties, some CSL path properties

order 2: (all?) CSL path properties this is probably wrong

there does not seem to be a "destructing" trick dual to the trick of building up formulas in arguments, see examples above

Fragments by Type Order

type order: $ord(\tau_1 \to \ldots \to \tau_m \to \mathsf{Pr}) = \max\{1 + ord(\tau_i)\}$

$\mathsf{HFL}^{k,m}$ = well-formed formulas using type annotations of order at most k and at most m arguments

recall examples above:

order 1: "balanced tree", "bisimilarity to a word", all CFL path properties, some CSL path properties

order 2: (all?) CSL path properties

order k: measure path lengths up to $2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}}$

Model Checking HFL

Theorem 2 (Axelsson/L./Somla '07)

For $k \geq 1, m \geq 0$: model checking $HFL^{k,m}$ is in k-EXPTIME.

PROOF: consider height of lattices $[\![\tau]\!]$:

$$height(\tau_1 \to \ldots \to \tau_m \to \mathsf{Pr}) \;=\; (n+1) \cdot \prod_{i=1}^{m} |[\![\tau_i]\!]|$$

with

$$|[\![\tau_1 \to \ldots \tau_m \to \mathsf{Pr}]\!]| \;=\; 2^{n \cdot \prod_{i=1}^{m} |[\![\tau_i]\!]|}$$

$\rightsquigarrow$ naïve bottom-up evaluation in time dominated by lattice height

□

Model Checking: Lower Bounds

Theorem 3 (Axelsson/L./Somla '07)

Model checking HFL^k is k-EXPTIME-hard (even its data complexity).

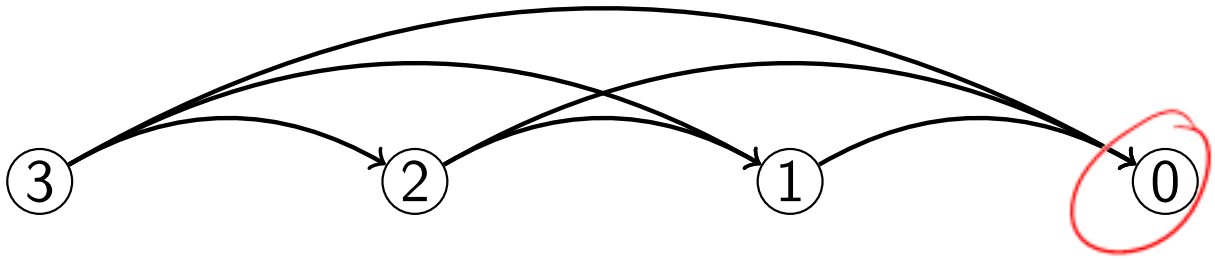
PROOF IDEA: reduction from the word problem for alternating $(k-1)$-EXPSPACE Turing machines

main ingredients:

- representation of large numbers by (lexicographically ordered) functions
- stepwise counting in HFL

Stepwise Counting in HFL

consider the following transition system

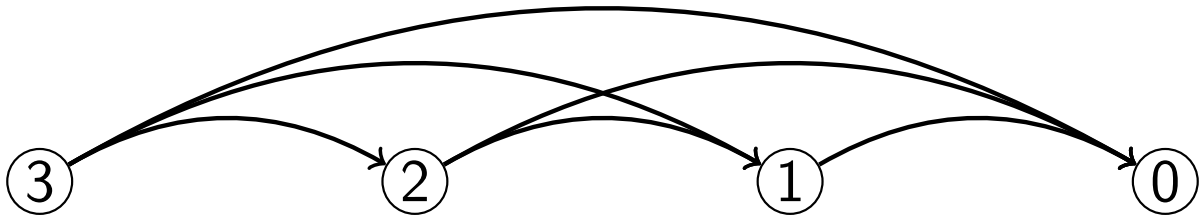


let $inc := \lambda(X : \mathrm{Pr}).X \leftrightarrow \Diamond\neg X$

what is $inc(\emptyset)$?

Stepwise Counting in HFL

consider the following transition system



let $inc := \lambda(X : \mathrm{Pr}).X \leftrightarrow \Diamond\neg X$

what is $inc(\emptyset)$, $inc^k(\emptyset)$ for $k > 1$?

creates 4-bit binary representation of 4

principle extendable to higher orders using tests for equality, less-than, greater-than

$\rightsquigarrow$ simulate run of space-bounded Turing machines

Undecidability of Satisfiability

Theorem 4

Satisfiability for HFL^1 is undecidable (at least Σ^1_1-hard)

follows from undecidability of Fixpoint Logic with Chop [Müller-Olm, '99] STACS '99 and embedding into HFL^1 [Viswanathan[2], '04] CONCUR 2004

undecidability not hard to see:

$$\varphi_{\text{word}} \wedge \bigvee_{w \in L(G_1)} \langle w \rangle \mathtt{tt} \wedge \bigvee_{w \in L(G_2)} \langle w \rangle \mathtt{tt}$$

expresses non-emptiness of intersection between CFGs G_1 and G_2

No Finite Model Property

decidability of model checking and Σ^1_1-hardness of satisfiability entails loss of finite model property

also possible to see directly:

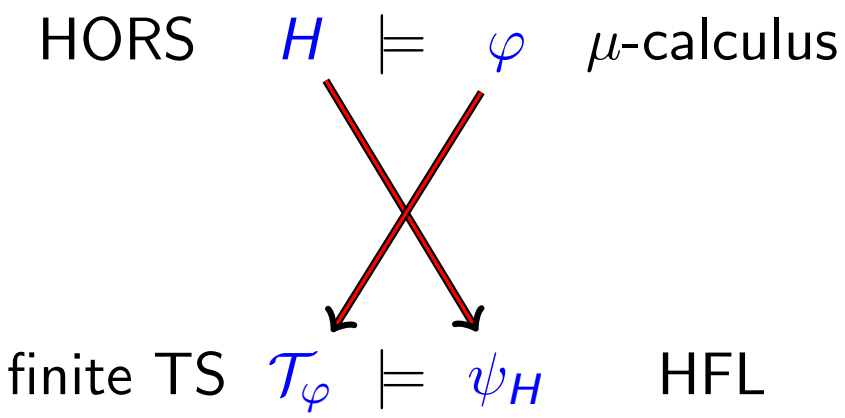
- $\langle a \rangle \mathtt{tt}$
- $\nu X.[a](\langle b \rangle \mathtt{tt} \wedge \langle a \rangle \mathtt{tt} \wedge X)$
- every state reachable by $a^n b^m$ with $m < n$ has a b-successor
- every state reachable by $a^n b^n$ is a dead-end

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus
2. Higher-Order Fixpoint Logic
3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness
4. Further Work

MCHO vs. HOMC

$\in$ [Conjecture,. . . ,Theorem] [Lozes]

Model checking higher-order recursion schemes is polynomially reducible to model checking HFL on finite transition systems.



other direction seems not to be possible: how to handle fixpoint alternation?

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus
2. Higher-Order Fixpoint Logic
3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness
4. Further Work

Negation Elimination

Def.: formulas in NNF use negation only in front of atomic propositions

Theorem 5 (Lozes '15) FICS 2015

Every $\mathrm{HFL}^{k,m}$ formula is equivalent to an $\mathrm{HFL}^{k,2m}$-formula (of quadratic size) in NNF.

Proof idea: monotonisation; double and inverse arguments

Ex.:

$$\big(\nu F.\lambda X.(X \leftrightarrow [a]X) \wedge (F\ [a]X)\big)\ p$$

becomes

$$\big(\nu F.\lambda X.\lambda \overline{X}.(\overline{X} \vee [a]X) \wedge (X \vee \langle a\rangle \overline{X}) \wedge (F\ [a]X\ \langle a\rangle \overline{X})\big)\ p\ \neg p$$

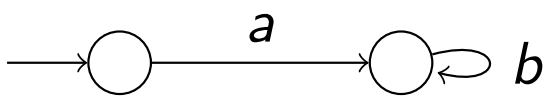
Parity vs. Stair-Parity Conditions

another difference to the μ-calculus: model checking is not a parity game

Ex.: consider

$$\left(\mu F.\lambda X.\langle b\rangle X \vee \langle a\rangle \nu G.F\ G\right)\ \mathrm{tt}$$

on



Def.: let $\rho = \Delta_0, \Delta_1, \Delta_2, \ldots \in ([k]^+)^\omega$ for some $k \in \mathbb{N}$ such that for all i:

- $\Delta_{i+1} = \Delta_i \cdot j$, or
- $\Delta_{i+1} \cdot j = \Delta_i$

for some j (i.e. an ω-word of priority stacks)
stair-parity condition is parity condition evaluated on $\lim \rho$ (if it exists)

An Operational Semantics for HFL

this part done by Florian Bruse

proposed automaton model: Alternating Parity Krivine Automata (APKA)

- alternation for Boolean and modal operators ($\vee, \wedge, \langle a \rangle, [b]$)
- (stair-)parity condition for fixpoints
- Krivine Abstract Machine for higher-order features

challenge: get acceptance condition right, i.e. synchronise parity condition with Krivine machine

Alternating Parity Krivine Automata

APKA of index m is $\mathcal{A} = (\mathcal{X}, \delta, I, \Lambda, (\tau_X)_{X \in \mathcal{X}})$ where

- finite set of (fixpoint) states $\mathcal{X} = \{X_1, \ldots, X_n\}$
- priority function $\Lambda \colon \mathcal{X} \to [1, m]$, resp. $[0, m-1]$
- transition function $\delta \colon X \mapsto \varphi_X$, generated from

$$\psi ::= P \mid \neg P \mid \psi \wedge \psi \mid \psi \vee \psi \mid \langle a \rangle \psi \mid [a] \psi \mid f_i^X \mid X' \mid (\psi\, \psi)$$

  where $f_i^X : \tau_i^X$ for $i \leq n_X$ and $\varphi_X : \tau_X$.
- assignment of argument and value types

$$\tau_X = \tau_1^X \to \cdots \to \tau_{n_X}^X \to \tau_{n_X+1}^X$$

- $I \in \mathcal{X}$ initial state with $\tau_I = \mathsf{Pr}$

state space is $\mathcal{Q} = \mathcal{X} \cup \bigcup_{X \in \mathcal{X}} \mathsf{sub}(\delta(X))$

Environments and Closures

environments handle variable lookup; assume unique names given by indices

$$e ::= e_0 \mid e_i = (f_1^X \mapsto (\psi_1, e_{i_1}), \ldots, f_{n_X}^X \mapsto (\psi_n, e_{i_n}), e')$$

where indices of e_j and e are strictly smaller than i; $\psi_i \in \mathcal{Q}$, $X \in \mathcal{X}$, $n = n_X$.

e' is parent environment

(ψ, e) called closure

difference to original Krivine Abstract Machine: assign all variables of a fixpoint at a time, no variable lookup in parent

variable lookup:

$$e_i(f) \;=\; \begin{cases} (\psi_j, e_j) & , \text{ if } f = f_j^X \\ \text{undefined} & , \text{ otherwise} \end{cases}$$

Configurations

APKA accept LTS; explained in terms of 2-player game on configurations of the form

$$C \;=\; (s, (\psi, e), e', \Gamma, \mathcal{E}, \Delta)$$

where

- s is current state in LTS
- (ψ, e) current closure with $\psi \in \mathcal{Q}$, $e \in \mathcal{E}$ environment
- e' distinguished environment (point of current computation)
- $\Gamma = (\psi_n, e_{i_n}), \ldots, (\psi_1, e_{i_1})$ stack of closures
- $\mathcal{E}$ set of environments
- Δ stack of priorities

only use well-formed configurations (all environments defined etc.)

Acceptance of APKA

run over $\mathcal{T}, s_0$ starts in $(s_0, (I, e_0), e_0, \epsilon, \{e_0\}, \epsilon)$

game played between **V** and **R**:

- players move as per the transition relation (below)
- automaton accepts structure if **V** wins
- player who gets stuck loses
- infinite plays $\rightsquigarrow$ stair-parity condition on sequence of priority stacks

transition from $(s, (\psi, e), e', \Gamma, \mathcal{E}, \Delta)$ depending on ψ

- $\psi = P$ or $\psi = \neg P$: **V** wins iff $\mathcal{T}, s \models \psi$
- $\psi = \psi_1 \vee \psi_2$: **V** chooses i, continue at $(s, (\psi_i, e), e', \Gamma, \mathcal{E}, \Delta)$
- $\psi = [a]\psi'$: **R** chooses $s \xrightarrow{a} t$, cont. at $(t, (\psi', e), e', \Gamma, \mathcal{E}, \Delta)$
- ...

More Game Moves

- $\psi = (\psi_1\, \psi_2)$: continue at $(s, (\psi_1, e), e', \Gamma \cdot (\psi_2, e), \mathcal{E}, \Delta)$
- $\psi = X$: continue $(s, (\delta(X), e''), e'', \Gamma', \mathcal{E} \cup \{e''\}, \Delta \cdot \Lambda(X))$
  where $\Gamma = \Gamma', C_1, \ldots, C_{n_X}$ and
  $e'' = (f_1^X \mapsto C_1, \ldots, f_{n_X}^X \mapsto C_{n_X}, e')$
- $\psi = f$ not of ground type: go to $(s, e(f), e', \Gamma, \mathcal{E}, \Delta)$
- $\psi = f$ of ground type and $e \neq e'$: go to $(s, (f, e), e'', \Gamma, \mathcal{E}, \Delta')$
  where e'' is parent of e', $\Delta = \Delta' \cdot i$ for some i
- $\psi = f$ of ground type and $e = e'$: go to $(s, e(f), e, \Gamma, \mathcal{E}, \Delta)$

special role for ground type variables: undo priorities one at a time until "caller" is reached

Example

$$\mathcal{X} = \{I, X, A, Y\}, \qquad \Lambda(I) = \Lambda(X) = \Lambda(A) = \Lambda(Y) = 1,$$
$$\tau_I = \mathsf{Pr}, \quad \tau_X = \tau_Y = \mathsf{Pr} \to \mathsf{Pr}, \quad \tau_A = (\mathsf{Pr} \to \mathsf{Pr}) \to \mathsf{Pr} \to \mathsf{Pr}$$

$$\begin{aligned}
\delta(I) &= \epsilon \mapsto (X\ Y) \\
\delta(X) &= f : \mathsf{Pr} \to \mathsf{Pr} \mapsto (f\ P) \vee (X\ (A\ f)) \\
\delta(A) &= g : \mathsf{Pr} \to \mathsf{Pr}, y : \mathsf{Pr} \mapsto \Box(g\ y) \\
\delta(Y) &= x : \mathsf{Pr} \mapsto x
\end{aligned}$$

APKA $\mathcal{A} = (\mathcal{X}, \Lambda, I, \delta, (\tau_X)_{X \in \mathcal{X}})$ defines uniform inevitability (in complicated way)

as HFL^2 formula: $\big(\mu X.\lambda f.(f\ P) \vee (X\ (\lambda z.\Box(f\ z)))\big)\ (\lambda x.x)$

expanded with extra fixpoint variables to name all subformulas:

$$\mu I.\big(\mu X.\lambda f.(f\ P) \vee (X\ ((\mu A.\lambda g.\lambda y.\Box(g\ y))\ f))\big)\ \mu Y.\lambda x.x)$$

Translation from HFL to APKA

1. mask λ-abstraction not directly after fixpoints: $\lambda x.\varphi$ becomes $\mu X.\lambda x.\varphi$
2. remove free λ-variables from fixpoints: $\sigma X.\varphi$ with $g \in \mathit{free}(\varphi)$ becomes $\big(\sigma X.\lambda g'.\varphi[g'/g, (X\ g')/X])\ , g\big)$
3. make types explicit: $\sigma X.\lambda f_1, \ldots, \lambda f_n.\varphi$ with $\varphi : \tau_1 \to \cdots \to \tau_n \to \mathsf{Pr}$ becomes $\sigma X.\lambda f_1, \ldots, \lambda f_n, \lambda g_1 \ldots, \lambda g_n.(\cdots(\varphi\, g_n)\cdots g_1)$ with $g_i : \tau_i$
4. choose priority function compatible with subformula relationship.

Fixpoint Alternation

higher-order does not conquer fixpoint alternation

Theorem 6 (Bruse, (subm.))

For every $k \geq 1$ and every $m \geq 2$ there is an APKA $\mathcal{A}_{k,m}$ or order 1 and index m that is not equivalent to any APKA of order k and index $< m$.

PROOF IDEA: see run of order-k and index-m APKA as tree again; tree language is definable; Banach's Fixpoint Theorem yields inexpressibility result, cmp. [Arnold, '99]

link to loss of small model property: fixpoint alternation hierarchy collapses over finite structures within HFL (but not necessarily within each HFL^k!)

$$\nu X.\varphi(X) \quad \equiv \quad \big(\mu F.\lambda X.(X \wedge \Box^*(X \to \varphi(X))) \vee (F\ \varphi(X))\big)\ \mathtt{tt}$$

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus
2. Higher-Order Fixpoint Logic
3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness
4. Further Work

Polyadic Model Logics

μ-calculus and HFL (etc.) are monadic: they define a set of states in each TS

polyadic modal logics are interpreted in tuples $\rightsquigarrow$ define relations of predetermined arity

syntactic solution: use tokens / names $1, 2, \ldots, r$

classic example [Andersen, '94; Otto, '99]

$$\nu X.(\bigwedge_{p \in P} p(1) \leftrightarrow p(2)) \wedge \bigwedge_{a \in \Sigma} [a]_1 \langle a \rangle_2 X \wedge [a]_2 \langle a \rangle_1 X$$

defines bisimilarity $\sim$

Declarative Complexity Theory

note: polyadic HFL^0 (PHFL^0) is polyadic μ-calculus

Theorem 7 (Otto, '99) TCS 224(1-2) : 237 - 265

$PHFL^0 \equiv PTIME/\sim$

Theorem 8 (L./Lozes, '14) IFIP TCS 2014

- $PHFL^1 \equiv EXPTIME/\sim$
- *tail-recursive* $PHFL^1 \equiv PSPACE/\sim$
- *tail-recursive* $PHFL^0 \equiv NLOGSPACE/\sim$ *on non-bisimilarity partially ordered structures*

conjecture: for $k \geq 1$

- $\mathrm{PHFL}^k \equiv k\text{-EXPTIME}/\sim$
- tail-recursive $\mathrm{PHFL}^k \equiv (k-1)\text{-EXPSPACE}/\sim$

1. Order-0 Fixpoint Logic, aka the Modal μ-Calculus
2. Higher-Order Fixpoint Logic
3. HFL's Model Theory
   - Complexity
   - Connection to Model Checking HORS
   - An Operational Semantics
   - Expressiveness
4. Further Work

Further Work

- how do fixpoint alternation and type order interact in detail?
- is there a linear procedure for negation elimination?
- is there a notion of guarded formulas for HFL; is it needed?
- does polyadic HFL^k capture k-EXPTIME/$\sim$?
- ...