On two notions of higher-order model checking

Naoki Kobayashi
University of Tokyo

Joint work with Etienne Lozes (ENS Cachan) and Florian Bruse (University of Kassel)

What's This Talk About?

- **Relationship between two higher-order extensions of model checking:**

	Models	Logic
finite state model checking	finite state systems	modal μ-calculus

What's This Talk About?

- Relationship between two higher-order extensions of model checking:
  - HORS model checking [Knapik+ 01; Ong 06]

	Models	Logic
finite state model checking	finite state systems	modal μ-calculus
HORS model checking	higher-order recursion schemes (HORS)	modal μ-calculus

What's This Talk About?

- Relationship between two higher-order extensions of model checking:
  - HORS model checking [Knapik+ 01; Ong 06]
  - HFL model checking [Viswanathan&Viswanathan 04]

	Models	Logic
finite state model checking	finite state systems	modal μ-calculus
HORS model checking	higher-order recursion schemes (HORS)	modal μ-calculus
HFL model checking	finite state systems	higher-order modal fixpoint logic (HFL)

What's This Talk About?

- Relationship between two higher-order extensions of model checking:
  - HORS model checking [Knapik+ 01; Ong 06]
  - HFL model checking [Viswanathan&Viswanathan 04]
- Type-based characterization of HFL model checking
  - $L \models \psi$ if and only if $\vdash_L \psi$

Outline

- Reviews of HORS model checking and HFL model checking
  - HORS model checking
  - HFL model checking
- From HORS to HFL model checking
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

Higher-Order Recursion Scheme (HORS)

- Grammar for generating an infinite tree

Order-1 HORS

$S \rightarrow A\ c$

$A\ x \rightarrow a\ x\ (A\ (b\ x))$

$S : o,\ A : o \rightarrow o$

Higher-Order Recursion Scheme (HORS)

- Grammar for generating an infinite tree

Tree whose paths are labeled by $a^{m+1}\ b^{m}\ c$

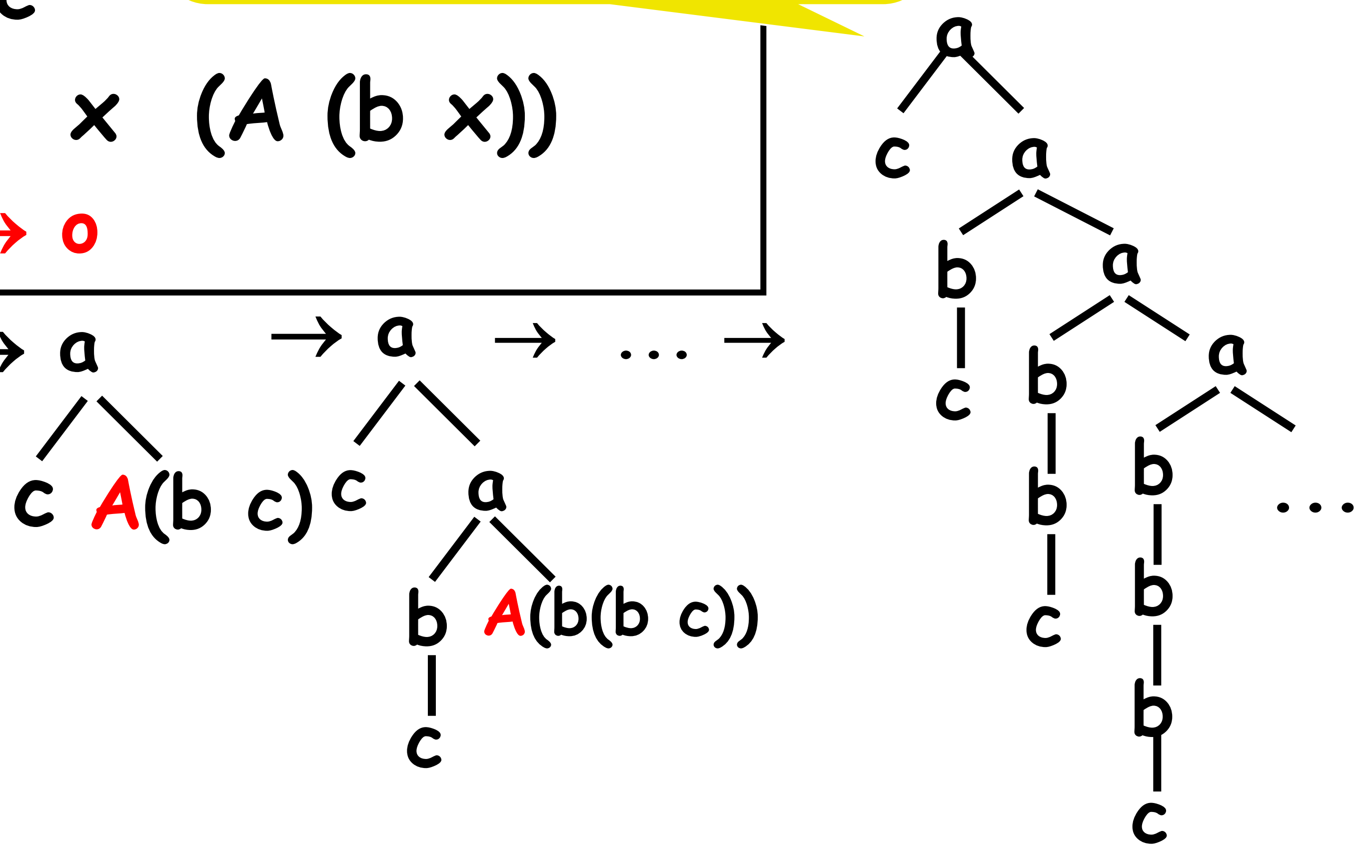
Order-1 HORS

$S \rightarrow A\ c$

$A\ x \rightarrow a\ x\ (A\ (b\ x))$

$S: o,\ A: o \rightarrow o$

$S \rightarrow A\ c \rightarrow$



Higher-Order Recursion Scheme (HORS)

- Grammar for generating an infinite tree

Order-1 HORS

$S \rightarrow A\ c$

$A\ x \rightarrow a\ x\ (A\ (b\ x))$

$S : o,\ A : o \rightarrow o$

HORS

≈

Call-by-name simply-typed λ-calculus

+

recursion, tree constructors

HORS Model Checking

Given
  G: HORS
  A: alternating parity tree automaton (APT)
   (a formula of modal μ-calculus or MSO),
does A accept Tree(G)?

e.g.
- Does every finite path end with "c"?
- Does "a" occur below "b"?

k-EXPTIME-complete [Ong, LICS06] $k \left\{ 2^{2^{\cdot^{\cdot^{2^{p(x)}}}}} \right.$
(for order-k HORS)

Outline

- Reviews of HORS model checking and HFL model checking
  - HORS model checking
  - HFL model checking
- From HORS to HFL model checking
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

Higher-Order Modal Fixpoint Logic (HFL) [Viswanathan&Viswanathan 04]

- Higher-order extension of the modal μ-calculus

φ ::= true

$\varphi_1 \wedge \varphi_2$	
$\varphi_1 \vee \varphi_2$	
$[a]\varphi$	φ must hold after a
$\langle a \rangle\varphi$	φ may hold after a
X	propositional variable
$\mu X.\varphi$	least fixpoint
$\nu X.\varphi$	greatest fixpoint

Higher-Order Modal Fixpoint Logic (HFL) [Viswanathan&Viswanathan 04]

- Higher-order extension of the modal μ-calculus

φ ::= true

$\varphi_1 \wedge \varphi_2$	
$\varphi_1 \vee \varphi_2$	
$[a]\varphi$	φ must hold after a
$\langle a \rangle \varphi$	φ may hold after a
X	predicate variable
$\mu X^{\kappa}.\varphi$	least fixpoint
$\nu X^{\kappa}.\varphi$	greatest fixpoint
$\lambda X^{\kappa}.\varphi$	(higher-order) predicate
$\varphi_1\ \varphi_2$	application

$\kappa ::= \bullet \mid \kappa_1 \rightarrow \kappa_2$

Selected Typing Rules for HFL

$$\Gamma \vdash \mathrm{true} : \bullet$$

$$\frac{\Gamma \vdash \varphi : \bullet}{\Gamma \vdash [a]\varphi : \bullet}$$

$$\frac{\Gamma \vdash \varphi : \bullet \quad \Gamma \vdash \psi : \bullet}{\Gamma \vdash \varphi \land \psi : \bullet}$$

$$\frac{\Gamma, X : \kappa_1 \vdash \varphi : \kappa_2}{\Gamma \vdash \lambda X . \varphi : \kappa_1 \to \kappa_2}$$

$$\Gamma, X : \kappa \vdash X : \kappa$$

$$\frac{\Gamma \vdash \varphi : \kappa_1 \to \kappa_2 \quad \Gamma \vdash \psi : \kappa_1}{\Gamma \vdash \varphi\ \psi : \kappa_2}$$

$$\frac{\Gamma, X : \kappa \vdash \varphi : \kappa}{\Gamma \vdash \mu X . \varphi : \kappa}$$

Semantics

$[\varphi]_I$: the set of states that satisfy φ

$L \models \varphi \Leftrightarrow s_{init} \in [\varphi]_\emptyset$ (s_{init}: initial state of L)

$[\mathrm{true}]_I = \mathrm{States}$

$[\varphi \wedge \psi]_I = [\varphi]_I \cap [\psi]_I$

$[\ [\alpha]\,\varphi\ ]_I = \{s \mid \forall t.(s$

$[\langle\alpha\rangle\,\varphi\ ]_I = \{s \mid \exists t.(s$

$[\bullet] = 2^{\mathrm{States}}$

$[\kappa_1 \to \kappa_2] = \{f \in [\kappa_1] \to [\kappa_2]$
$\mid f\text{: monotonic}\}$

$[X]_I = I(X)$

$[\mu X^\kappa.\varphi]_I = \mathrm{lfp}(\lambda x \in [\kappa].[\varphi]_{I\{X=x\}})$

$[\nu X^\kappa.\varphi]_I = \mathrm{gfp}(\lambda x \in [\kappa].[\varphi]_{I\{X=x\}})$

(Note: $\lambda x \in [\kappa].[\varphi]_{I\{X=x\}}$ is monotonic)

$[\lambda X^\kappa.\varphi]_I = \lambda x \in [\kappa].[\varphi]_{I\{X=x\}}$

$[\varphi\ \psi]_I = [\varphi]_I\ [\psi]_I$

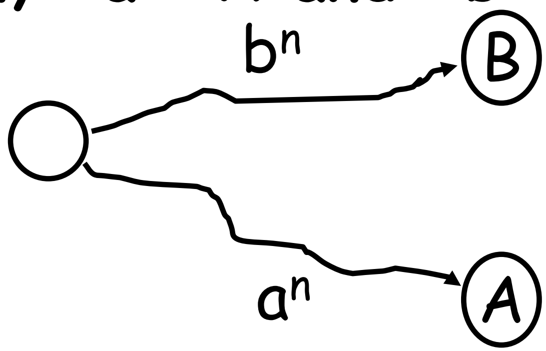
Example

$$\left(\mu F^{\bullet\to\bullet\to\bullet}.\lambda X.\lambda Y.\ (X\wedge Y) \vee F\ (\langle a\rangle X)\ (\langle b\rangle Y)\right)\ A\ B$$

$$= (A\wedge B) \vee \left(\mu F^{\bullet\to\bullet\to\bullet}.\lambda X.\lambda Y.\ (X\wedge Y) \vee F(\langle a\rangle X)(\langle b\rangle Y)\right)\ (\langle a\rangle A)(\langle b\rangle B)$$

$$= (A\wedge B) \vee (\langle a\rangle A \wedge \langle b\rangle B) \vee (\langle a\rangle\langle a\rangle A \wedge \langle b\rangle\langle b\rangle B) \vee \ldots$$

For some n, $\langle a\rangle^n\ A$ and $\langle b\rangle^n\ B$ hold

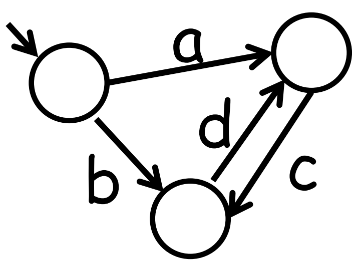


HFL Model Checking

Given
L: (finite-state) labeled transition system
φ: HFL formula,
does L satisfy φ?

e.g. $L \models \varphi$ for:

L:



$$\varphi: (\mu F.\lambda X.\lambda Y.\ (X \wedge Y) \vee F\ (\langle a\rangle X)\ (\langle b\rangle Y))\ (\langle c\rangle \mathrm{true})\ (\langle d\rangle \mathrm{true})$$

HORS vs HFL model checking

	Model	Spec.	complexity	Applications
HORS model checking	HORS	APT	k-EXPTIME complete (for order-k HORS)	Automated verification of functional programs [K 09][K+11]...
HFL model checking	LTS	HFL	k-EXPTIME complete (for order-k HFL)	Assume-guarantee reasoning [VV 04] Process equivalence checking [Lange+ 14]

APT: alternating parity tree automaton
LTS: finite-state labeled transition system

Hierarchical Equation Systems (HES)

$X_1 =_{\alpha 1} \varphi_1; \ \ldots; \ X_n =_{\alpha n} \varphi_n$

$(\alpha_i \in \{\mu, \nu\})$

toHFL($X =_{\alpha} \varphi$) = $\alpha X.\varphi$

toHFL(H; $X =_{\alpha} \varphi$) =

toHFL($[\alpha X.\varphi \,/\, X]H$)

Example:

HFL: $\nu X.\mu Y.(\langle a\rangle X \vee \langle b\rangle Y)$

(there exists a path $(b^*a)^{\omega}$)

HES: $X =_{\nu} Y; \ Y =_{\mu} \langle a\rangle X \vee \langle b\rangle Y$

Outline

- Reviews of HORS model checking and HFL model checking
  - HORS model checking
  - HFL model checking
- From HORS to HFL model checking
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

From HORS to HFL model checking

- Input:
  - HORS G
  - APT A (with largest priority p)
- Output:
  - LTS L_A
  - HFL formula $\varphi_{G,p}$

  such that $G \models A$ iff $L_A \models \varphi_{G,p}$

Intuition:

- L_A simulates the transitions of A
- $\varphi_{G,p}$ describes "L_A has transitions corresponding to an accepting run of A over Tree(G)"

Construction of L_A

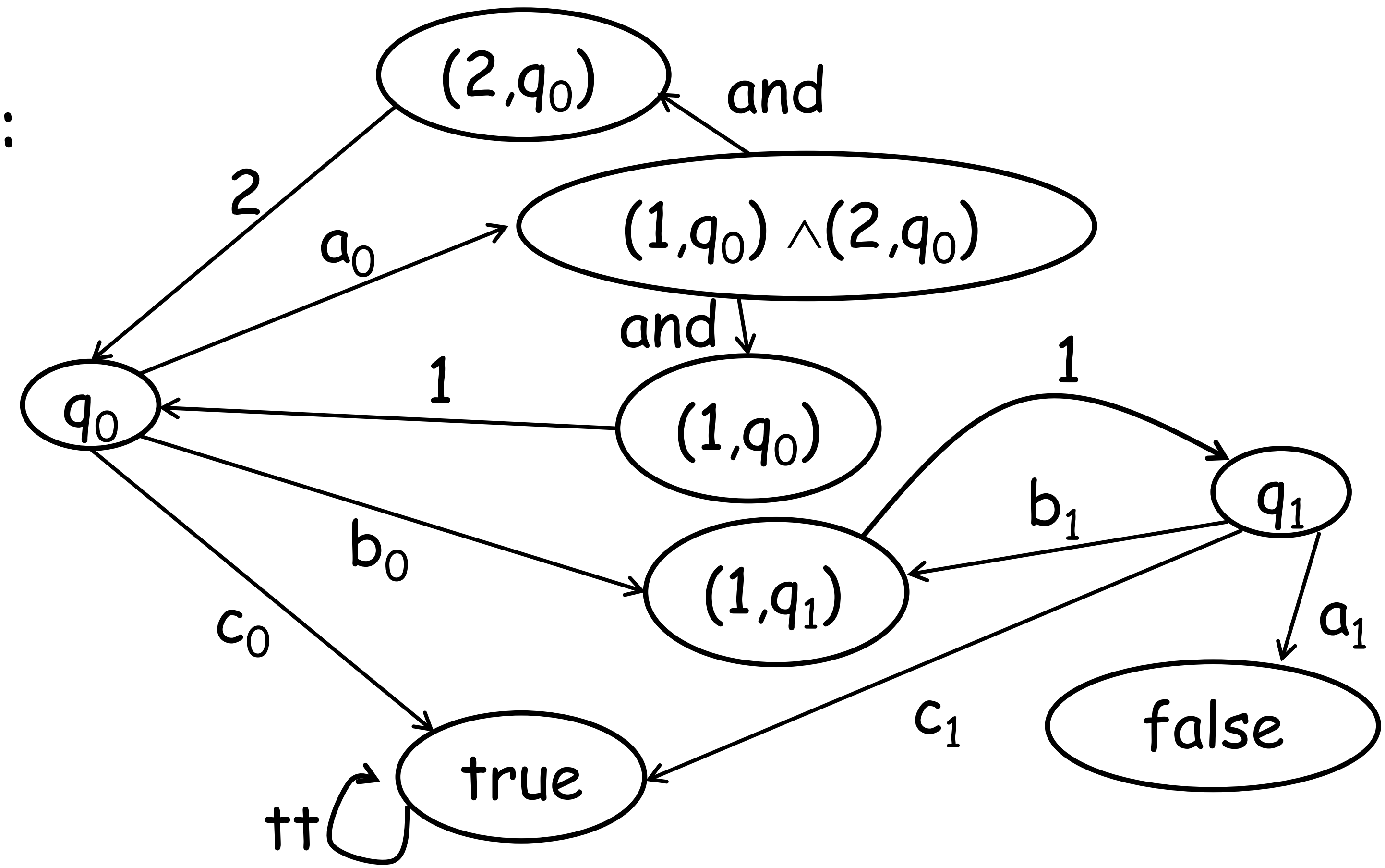
A:

$\delta(q_0,a)= (1,q_0) \wedge (2,q_0)$ $\delta(q_1,a)=$ false

$\delta(q_0,b)= \delta(q_1,b)= (1,q_1)$ $\delta(q_0,c)= \delta(q_1,c)=$ true

$\Omega(q_0)=0$ $\Omega(q_1)=1$

L_A:



Construction of L_A

A:

$\delta(q_0,a)= (1,q_0) \wedge (2,q_0)$ $\delta(q_1,a)=$ false

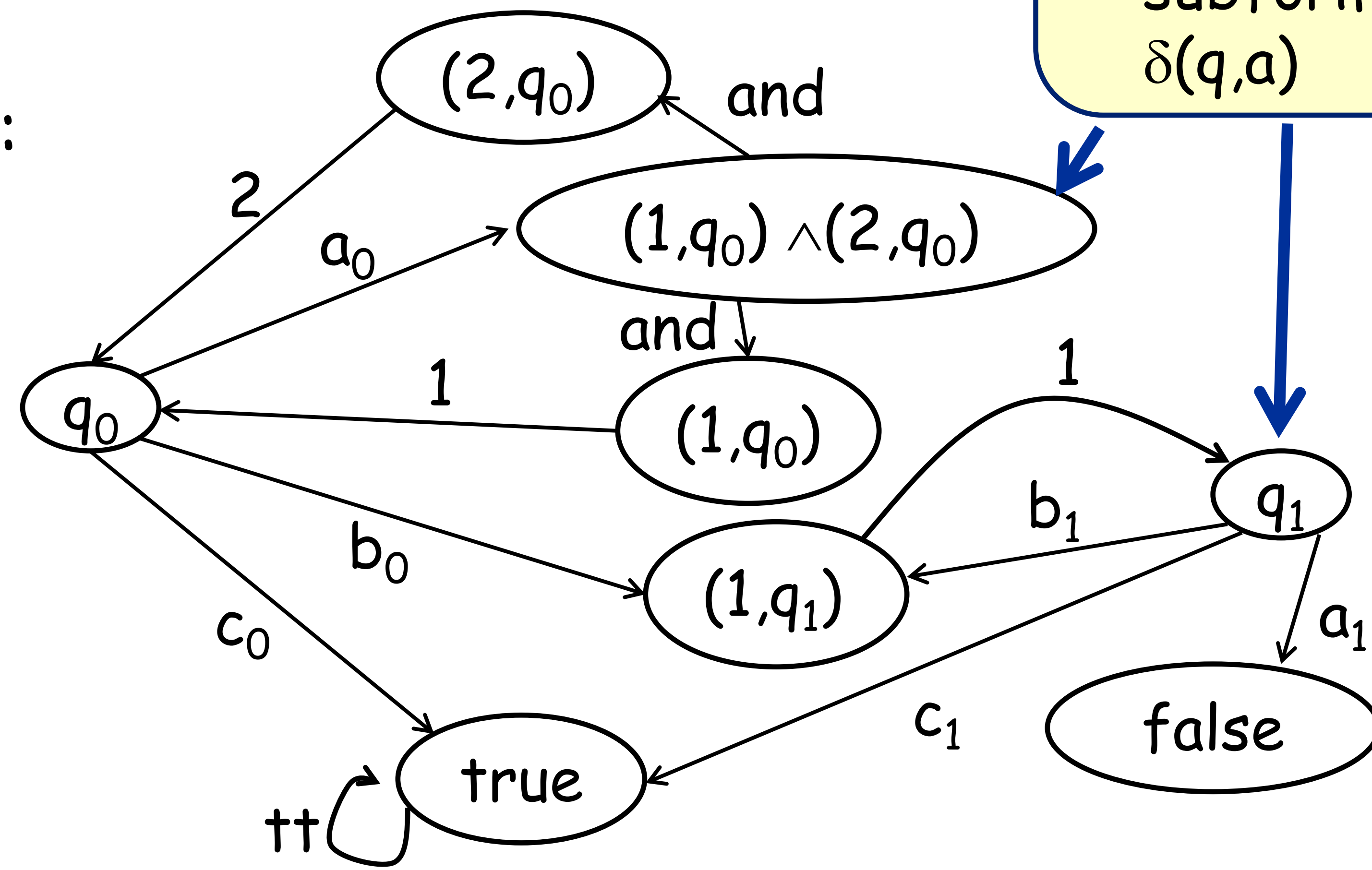
$\delta(q_0,b)= \delta(q_1,b)= (1,q_1)$ $\delta(q_0,c)= \delta(q_1,$

$\Omega(q_0)=0$ $\Omega(q_1)=1$

The states of L_A consist of:

- states of A and
- subformulas of $\delta(q,a)$

L_A:



Construction of L_A

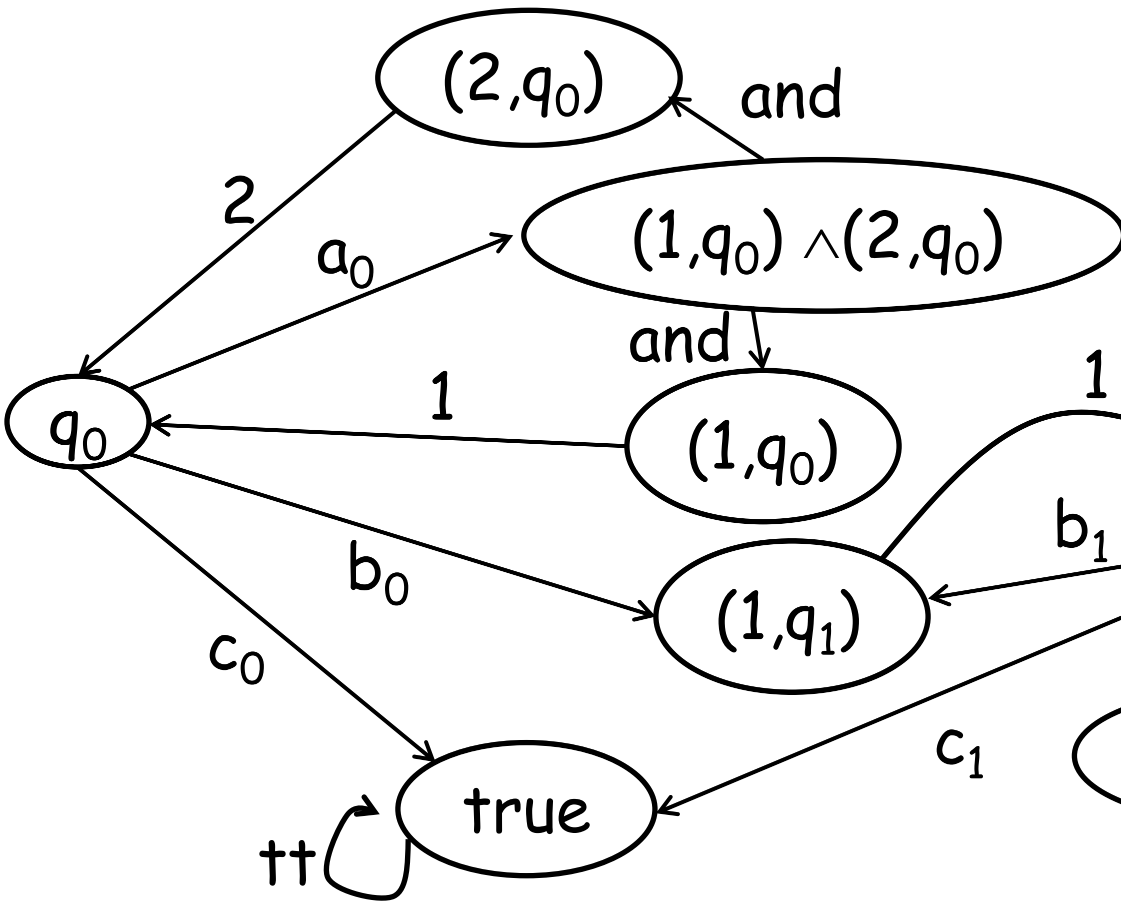
A:

$\delta(q_0,a)= (1,q_0) \wedge (2,q_0)$ $\delta(q_1,a)=$ false

$\delta(q_0,b)= \delta(q_1,b)= (1,q_1)$ $\delta(q_0,c)= \delta(q_1,c)=$ true

$\Omega(q_0)=0$ $\Omega(q_1)=1$

L_A:



A transition label is an input symbol annotated with a priority; or ...

Construction of L_A

A:

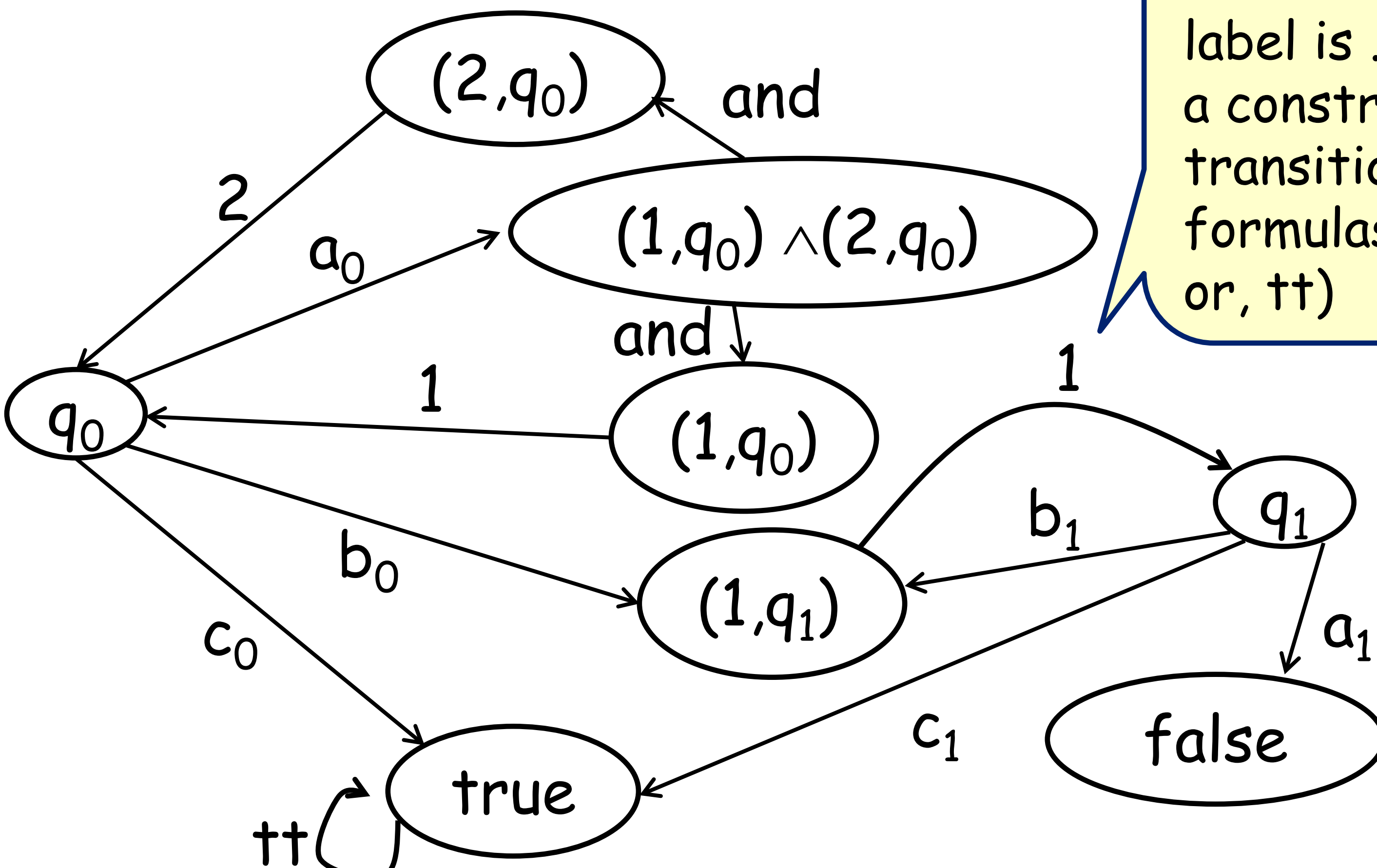
$\delta(q_0,a)= (1,q_0) \wedge (2,q_0)$ $\delta(q_1,a)=$ false

$\delta(q_0,b)= \delta(q_1,b)= (1,q_1)$ $\delta(q_0,c)= \delta(q_1,c)=$ true

$\Omega(q_0)=0$ $\Omega(q_1)=1$

A transition label is ...; or a constructor of transition formulas (i, and, or, tt)

L_A:



Outline

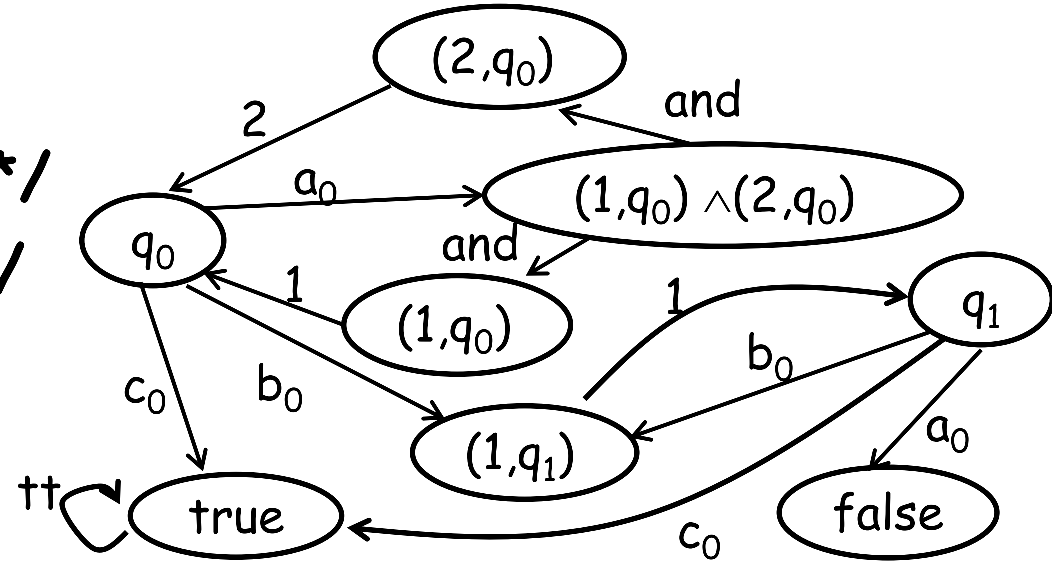
- Reviews of HORS model checking and HFL model checking
- From HORS to HFL model checking
  - construction of L_A
  - construction of $\varphi_{G,p}$
    - case p=0
    - general case
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$\varphi_{a\ c\ (b\ c)}$ =
 $\langle a_0 \rangle$ "can visit 1st and 2nd children with states satisfying φ_c and $\varphi_{b\ c}$ respectively"

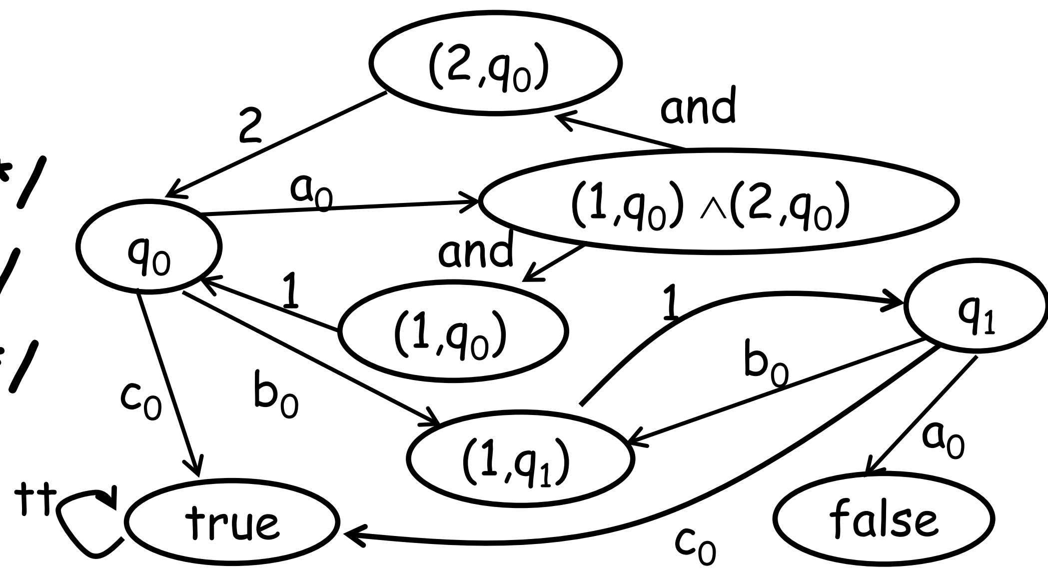
= $\langle a_0 \rangle$(
 $\langle 1 \rangle \varphi_c$ /* case (1,q) */
 $\vee \langle 2 \rangle \varphi_{b\ c}$ /* case (2,q) */
 $\vee \langle tt \rangle$true /* case true */



From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$\varphi_{a\ c\ (b\ c)}$ =
$\langle a_0 \rangle$ "can visit 1st and 2nd children with states satisfying φ_c and $\varphi_{b\ c}$ respectively"
= $\langle a_0 \rangle$(
$\langle 1 \rangle \varphi_c$ /* case (1,q) */
$\vee \langle 2 \rangle \varphi_{b\ c}$ /* case (2,q) */
$\vee \langle tt \rangle$true /* case true */
$\vee$($\langle$and$\rangle$true /*case f$\wedge$g */
$\wedge$ [and]

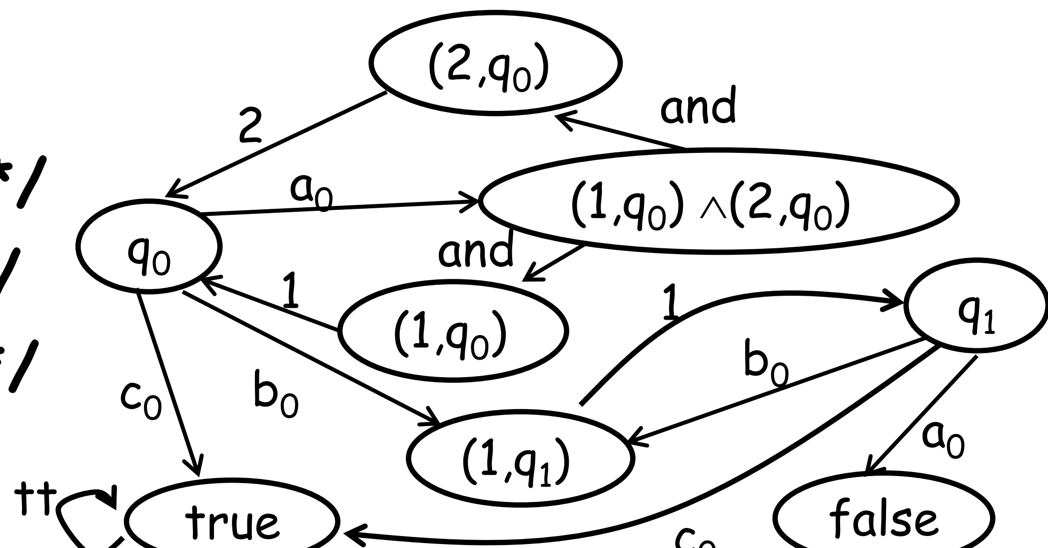


From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$\varphi_{a\ c\ (b\ c)}$ =
$\langle a_0 \rangle$ "can visit 1st and 2nd children with states satisfying φ_c and $\varphi_{b\ c}$ respectively"

$= \langle a_0 \rangle(\nu X.$
$\quad \langle 1 \rangle \varphi_c$ /* case (1,q) */
$\quad \vee \langle 2 \rangle \varphi_{b\ c}$ /* case (2,q) */
$\quad \vee \langle tt \rangle$true /* case true */
$\quad \vee (\langle and \rangle$true /*case $f \wedge g$ */
$\quad\quad \wedge$ [and]X)

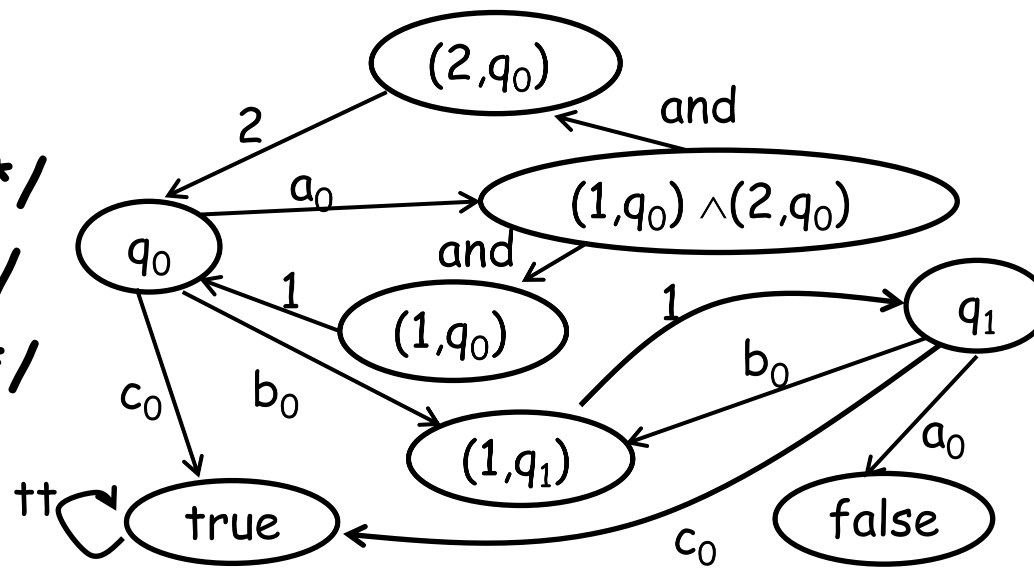


From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$\varphi_{a\ c\ (b\ c)}$ =
$\langle a_0 \rangle$ "can visit 1st and 2nd children with states satisfying φ_c and $\varphi_{b\ c}$ respectively"

$= \langle a_0\rangle(\nu X.$
$\quad \langle 1\rangle\varphi_c$ /* case (1,q) */
$\quad \vee\langle 2\rangle\varphi_{b\ c}$ /* case (2,q) */
$\quad \vee\langle tt\rangle$true /* case true */
$\quad \vee(\langle and\rangle$true /*case $f\wedge g$ */
$\qquad \wedge\ [and]X)$
$\quad \vee\ \langle or\rangle X)$ /*case $f\vee g$ */



From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$\varphi_{a\ c\ (b\ c)} = \langle a_0 \rangle (H_2\ \varphi_c\ \varphi_{b\ c})$

where $H_2 = \lambda Y_1 . \lambda Y_2 . \nu X.$

$\langle 1 \rangle Y_1$ /* case (1,q) */

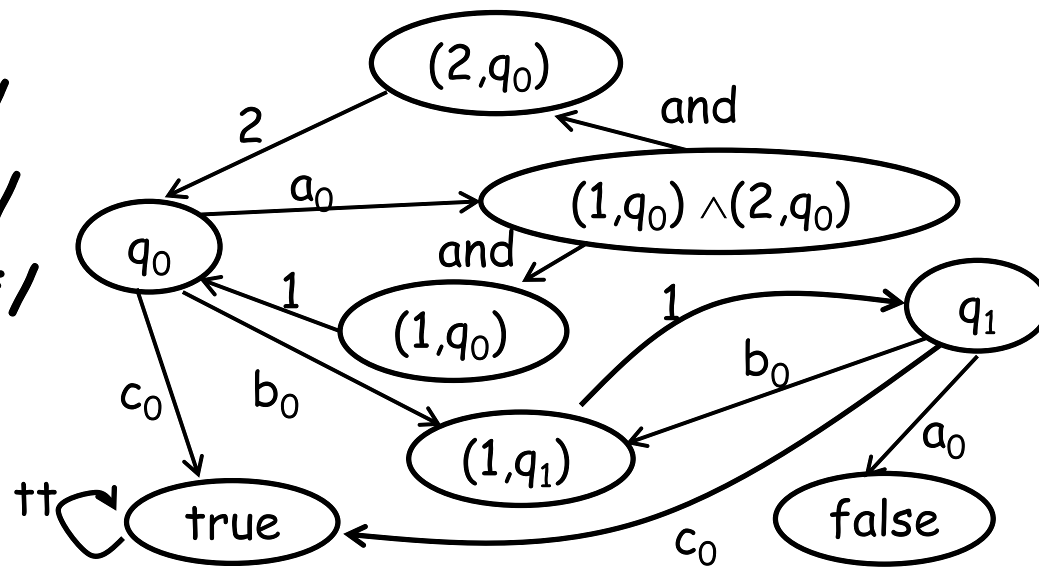
$\vee \langle 2 \rangle Y_2$ /* case (2,q) */

$\vee \langle tt \rangle \mathrm{true}$ /* case true */

$\vee (\langle \mathrm{and} \rangle \mathrm{true}$ /*case $f \wedge g$ */

$\wedge\ [\mathrm{and}]X)$

$\vee\ \langle \mathrm{or} \rangle X)$ /*case $f \vee g$ */



From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$$\varphi_{a\ c\ (b\ c)} = \langle a_0\rangle(H_2\ \varphi_c\ \varphi_{b\ c})$$
$$= \langle a_0\rangle(H_2\,(\langle c_0\rangle H_0)\ (\langle b_0\rangle H_1\ (\langle c_0\rangle H_0)))$$

where $H_2 = \lambda Y_1.\,\lambda Y_2.\nu X.$

$\langle 1\rangle Y_1$ /* case (1,q) */

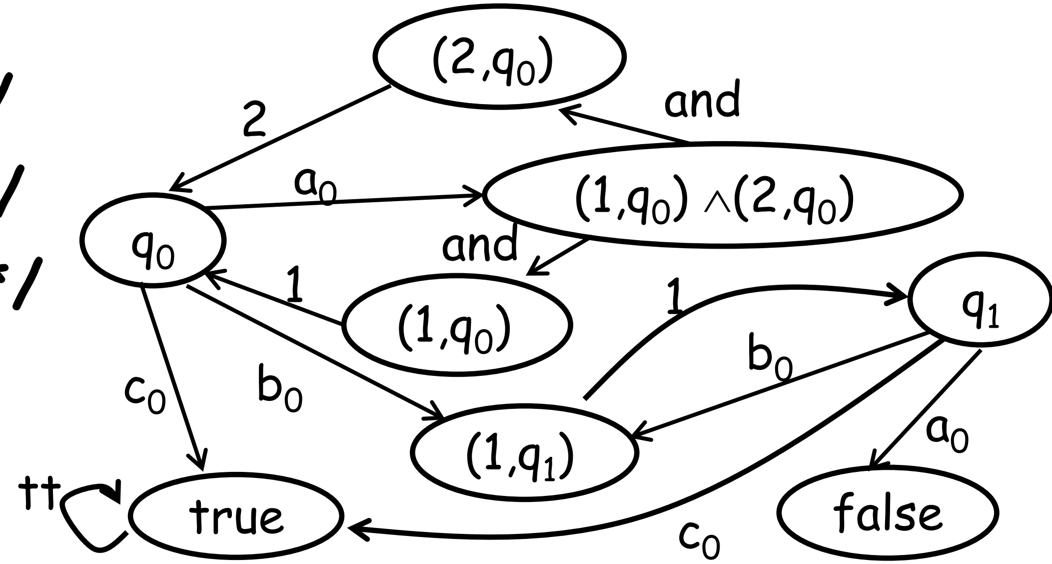
$\vee\langle 2\rangle Y_2$ /* case (2,q) */

$\vee\langle tt\rangle$true /* case true */

$\vee(\langle and\rangle$true /*case $f\wedge g$ */

$\wedge$ [and]X)

$\vee$ $\langle or\rangle$X) /*case $f\vee g$ */



From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$\varphi_{a\ c\ (b\ c)} = \langle a_0 \rangle (H_2\ \varphi_c\ \varphi_{b\ c})$

$= \langle a_0 \rangle (H_2\ (\langle c_0 \rangle H_0)\ (\langle b_0 \rangle H_1\ (\langle c_0 \rangle H_0)))$

where $H_2 = \lambda Y_1 . \lambda Y_2 . \nu X.$

$\langle 1 \rangle Y_1$ /* case (1,q) */

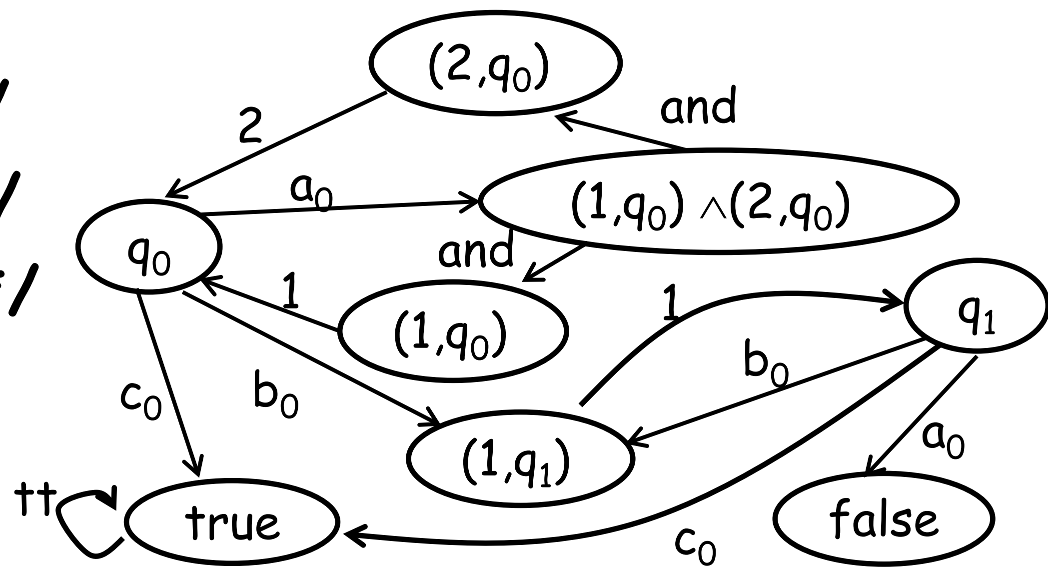
$\vee \langle 2 \rangle Y_2$ /* case (2,q) */

$\vee \langle tt \rangle \mathrm{true}$ /* case true */

$\vee (\langle and \rangle \mathrm{true}$ /*case $f \wedge g$ */

$\wedge\ [and] X)$

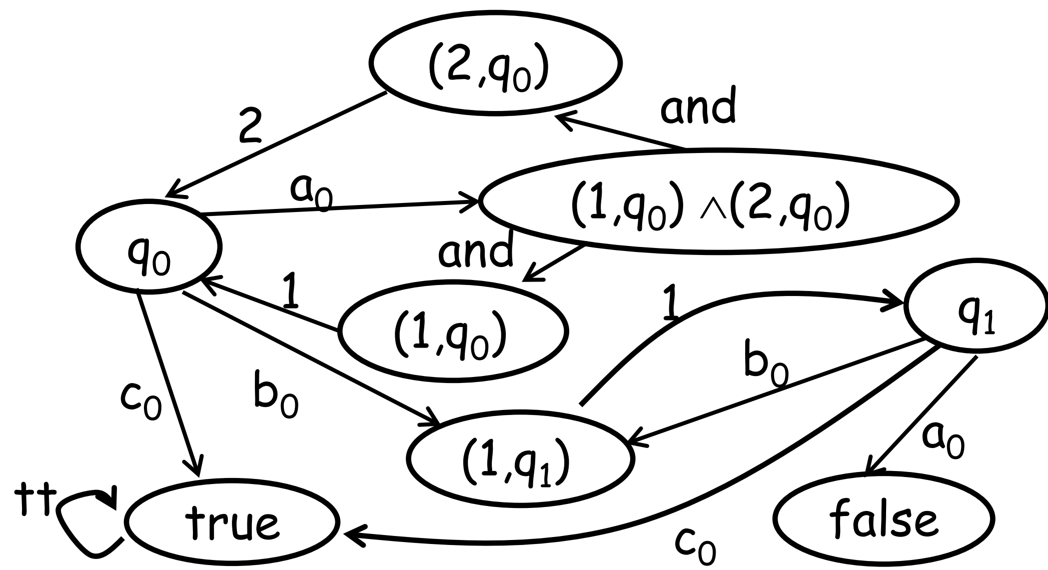
$\vee\ \langle or \rangle X)$ /*case $f \vee g$ */



From trees to HFL formulas

φ_T: "the current state has transitions corresponding to an accepting run for T"

$$\varphi_{a\ T1\ \ldots\ Tk} = \langle a_0 \rangle (H_k\ \varphi_{T1}\ \ldots\ \varphi_{Tk})$$
(co-inductively defined)



From HORS to HFL

$F \to t$

$\Rightarrow\ F =_\nu t^\#$

where:

$F^\# = F \qquad x^\# = x$

$(t_1 t_2)^\# = (t_1)^\#(t_2)^\#$

$(\lambda x.t)^\# = \lambda x.(t)^\#$

$a^\# = \lambda x_1 \ldots \lambda x_k.\langle a_0 \rangle(H_k\ x_1\ \ldots\ x_k)$

Example

HORS G

$S \rightarrow F\ c$

$F\ x \rightarrow a\ \ x\ \ (F\ (b\ x))$

A:

$\delta(q_0,a) = (1,q_0) \wedge (2,q_0)$

$\delta(q_1,a) = \text{false}$

$\delta(q_0,b) = \delta(q_1,b) = (1,q_1)$

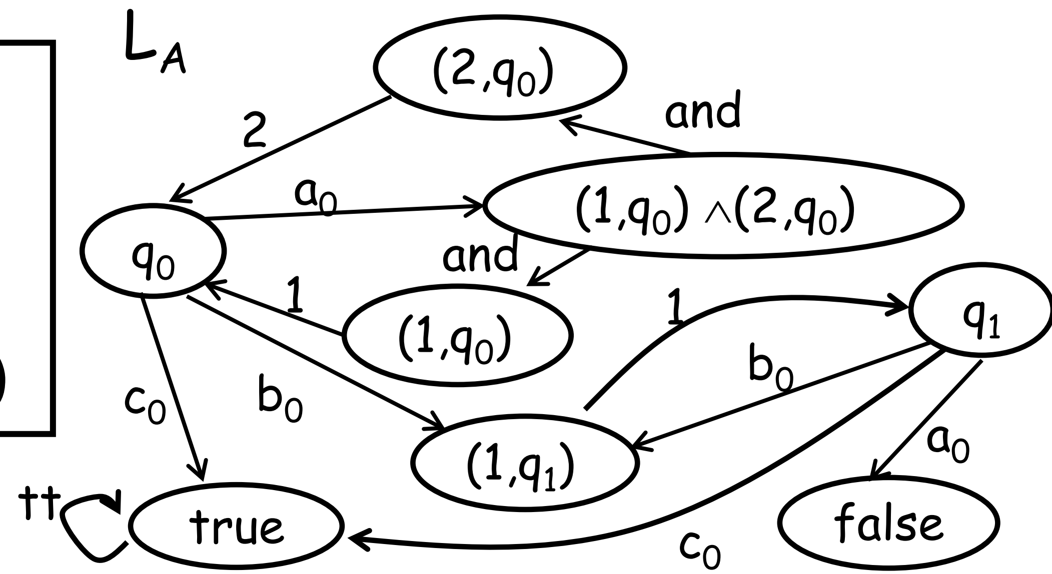
$\delta(q_0,c) = \delta(q_1,c) = \text{true}$

$\varphi_{G,0}$

$S =_\nu F\ (\langle c_0\rangle H_0)$

$F\ x =_\nu$

$\langle a_0\rangle(H_2\ x\ (F(\langle b_0\rangle(H_1\ x))))$



Outline

- Reviews of HORS model checking and HFL model checking
- From HORS to HFL model checking
  - construction of L_A
  - construction of $\varphi_{G,p}$
    - case p=0
    - *general case*
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

Challenge

- How to translate the parity condition of APT:

  "for every path of a run-tree, the largest priority visited infinitely often is even"

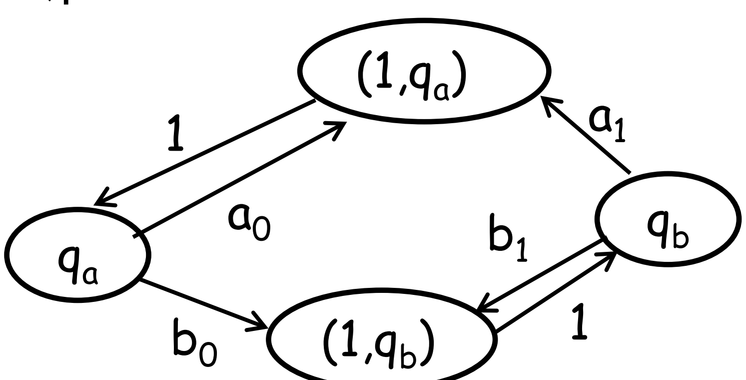
  to a proper nesting of least/greatest fixpoint formulas?

e.g. A: $\delta(q_a,a)=\delta(q_b,a)=(1,q_a)$ $\delta(q_a,b)=\delta(q_b,b)=(1,q_b)$

$\Omega(q_a)=0$, $\Omega(q_b)=1$

G: $S \rightarrow a\ (b\ F)$ $F \rightarrow a\ S$

$G \not\models A$ but



$\models$

$S =_\nu \langle a_0 \rangle (H_1\ (\langle b_0 \rangle (H_1\ F)));$

$F =_\nu \langle a_1 \rangle (H_1\ S)$

Ideas

- Duplicate each non-terminal for each priority

An appropriate copy is chosen, depending on the largest priority seen since the last unfolding

$$S \to a\ (b\ F) \qquad F \to a\ S$$

$$
\begin{aligned}
S^1 &=_\mu \langle a_0\rangle(H_1\ (\langle b_0\rangle(H_1\ F^0)));\\
F^1 &=_\mu \langle a_1\rangle(H_1\ S^1);\\
S^0 &=_\nu \langle a_0\rangle(H_1\ (\langle b_0\rangle(H_1\ F^0)));\\
F^0 &=_\nu \langle a_1\rangle(H_1\ S^1);
\end{aligned}
$$

The largest priority seen since the previous unfolding of a non-terminal.

Ideas

- Duplicate each non-terminal for each priority
- Duplicate also each argument, so that a function can choose an appropriate copy

$S \to F\ G \qquad F\ x \to b\ (x\ S) \qquad G\ y \to a\ y$

We cannot locally decide the priority annotation for G; only F knows when G is unfolded.

⬇

$$S^1 =_\mu F^0\ G^0\ G^1$$
$$F^1\ x^0\ x^1 =_\mu \langle b_0\rangle(H_1\ (x^0\ S^0\ S^1)) \vee \langle b_1\rangle(H_1\ (x^1\ S^1\ S^1))$$

...

General construction of $\varphi_{G,p}$

$G: F_1\ x_1\ \ldots\ x_{k1} \to t_1,\ \ldots,\ F_n\ x_1\ \ldots\ x_{kn} \to t_n$

$\Downarrow$

$F_1^p\ x_1^0\ \ldots\ x_1^p\ \ldots\ x_{k1}^0\ \ldots\ x_{k1}^p =_{\alpha(p)} t_1^{\#0};\ \ldots;$

$F_n^p\ x_1^0\ \ldots\ x_1^p\ \ldots\ x_{k1}^0\ \ldots\ x_{k1}^p =_{\alpha(p)} t_n^{\#0};$

$\ldots;$

$F_1^0\ x_1^0\ \ldots\ x_1^p\ \ldots\ x_{kn}^0\ \ldots\ x_{kn}^p =_{\alpha(0)} t_1^{\#0};\ \ldots;$

$F_n^0\ x_1^0\ \ldots\ x_1^p\ \ldots\ x_{kn}^0\ \ldots\ x_{kn}^p =_{\alpha(0)} t_n^{\#0}$

where $\alpha(i) = \nu$ if i is even and μ otherwise

General construction of $\varphi_{G,p}$

$G: F_1\ x_1\ \ldots\ x_{k1} \to t_1,\ \ldots,\ F_n\ x_1\ \ldots\ x_{kn} \to t_n$

$\Downarrow$

$F_1^p\ x_1^0\ \ldots\ x_1^p\ \ldots\ x_{k1}^0\ \ldots\ x_{k1}^p =_{\alpha(p)} t_1^{\#0};$

$\ldots;$

$F_n^0\ x_1^0\ \ldots\ x_1^p\ \ldots\ x_{kn}^0\ \ldots\ x_{kn}^p =_{\alpha(0)} t_n^{\#0}$

$(a)^{\#i} = \lambda x_{1,0} \ldots \lambda x_{1,p} \ldots \lambda x_{k,0} \ldots \lambda x_{k,p}\ .$
$\quad \langle a_0 \rangle (H_k\ \ x_{1,0} \ldots x_{k,0}) \vee \ldots \vee \langle a_p \rangle (H_k\ \ x_{1,p} \ldots x_{k,p})$

$(x)^{\#i} = x^i$

$(F)^{\#i} = F^i$

$(s\ t)^{\#i} = (s)^{\#i}\ \ (t)^{\#\max(0,i)} \ldots (t)^{\#\max(p,i)}$

Correctness of Translation

- **Theorem:**

  $G \models A$

  if and only if

  $L_A \models \varphi_{G,p}$

Follows from the type-based characterizations of HORS and HFL model checking:

$$G \models A \underset{\text{[K\&Ong 09]}}{\Leftrightarrow} \vdash_A G \underset{\text{(new)}}{\Leftrightarrow} \vdash_{L_A} \varphi_{G,p} \underset{\text{(new)}}{\Leftrightarrow} L_A \models \varphi_{G,p}$$

Correctness of Translation

- **Theorem:**

  $G \models A$

  if and only if

  $L_A \models \varphi_{G,p}$

$|L_A|$ is polynomial in $|A|$

$|\varphi_{G,p}|$ is polynomial in $|G|$, p

Outline

- Reviews of HORS model checking and HFL model checking
- From HORS to HFL model checking
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

From HFL to HORS model checking

- Input:
  - LTS L
  - HFL formula φ
- Output:
  - HORS $G_{\varphi,c}$
  - APT A_L

  such that $L \models \varphi$ iff $G_{\varphi,c} \models A_L$ for sufficiently large c

Intuition:

- $G_{\varphi,c}$ generates tree representation of the formula obtained from φ by unfolding fixedpoint operators sufficiently many times
- A_L accepts trees representing valid formulas

HFL-to-HORS Translation: Overview

$\boxed{F\ X =_\nu \varphi}$

Remove fixpoint operators by finite unfoldings

$\boxed{F^{(c)}\ X=[F^{(c-1)}/F]\varphi\ ;\ldots;F^{(1)}\ X=[F^{(0)}/F]\ \varphi;F^{(0)}\ X=\mathrm{true}}$

Convert it to HORS, which generates the tree representation of the formula

$\boxed{F^{(c)}\ X\to[F^{(c-1)}/F]\varphi';\ldots;\ F^{(1)}\ X\to[F^{(0)}/F]\ \varphi';\ F^{(0)}\ X\to\mathrm{true}}$

Parameterize F by a number, and implement numbers (up to $k\left\{2^{2^{\cdot^{\cdot^{2^n}}}}\right.$) as functions (cf. [Jones01])

$\boxed{F\ m\ X\to\ \mathrm{if}\ (\mathrm{Zero?}\ m)\ \mathrm{true}\ ([F\ (m-1)\ /F]\varphi')}$

Correctness of Translation

- **Theorem:**

  $L \models \varphi$

  if and only if

  $G_{\varphi,|L|} \models A_L$

  $|G_{\varphi,|L|}|$ is polynomial in $|\varphi|$ and $|L|$

  $|A_L|$ is polynomial in $|L|$

Outline

- Reviews of HORS model checking and HFL model checking
- From HORS to HFL model checking
- From HFL to HORS model checking
- Type system for HFL model checking
- Conclusion

Goal

- Design a type system $\vdash_L$ such that:

  $L \models \varphi$

  if and only if

  $\vdash_L \varphi$

  (cf. K-Ong type system for HORS model checking [K&Ong, LICS09])

  Applications:

  - correctness proof of HORS-to-HFL translation
  - practical model checkers for HFL
    (cf. practical HORS model checkers based on intersection types)

Types

τ ::= s — type of propositions that hold at state s

(i.e. $s \models \varphi \Leftrightarrow \vdash \varphi : s$)

$\sigma \rightarrow \tau$ — type of functions from σ to τ

$\sigma ::= \tau_1 \wedge \ldots \wedge \tau_n$ — intersection types

$\wedge : s \rightarrow s \rightarrow s$ for every s

$\vee : (s \rightarrow T \rightarrow s) \wedge (T \rightarrow s \rightarrow s)$ for every s

Typing Rules

$$\Gamma \vdash \mathbf{true}: s$$

$$\frac{\Gamma \vdash \varphi: s' \quad s \to_a s'}{\Gamma \vdash \langle a \rangle \varphi: s}$$

$$\frac{\Gamma \vdash \varphi: s \quad \Gamma \vdash \psi: s}{\Gamma \vdash \varphi \wedge \psi: s}$$

$$\frac{\Gamma \vdash \varphi: s' \text{ for every } s' \text{ such that } s \to_a s'}{\Gamma \vdash [a]\varphi: s}$$

$$\Gamma, X{:}\tau \vdash X{:}\tau$$

$$\frac{\Gamma \vdash \varphi: \tau_1 \wedge \ldots \wedge \tau_k \to \tau \quad \Gamma \vdash \psi: \tau_i \text{ for each } i}{\Gamma \vdash \varphi\ \psi: \tau}$$

$$\frac{\Gamma, x{:}\tau_1, \ldots, x{:}\tau_k \vdash \varphi{:}\tau}{\Gamma \vdash \lambda x.\varphi: \tau_1 \wedge \ldots \wedge \tau_k \to \tau}$$

Typing Fixpoint Formulas

$$\frac{\emptyset \vdash \varphi : \tau \qquad X =_{\alpha} \varphi}{\Gamma \vdash X : \tau}$$

Definition:

$|\!-_L X_1 =_{\alpha 1} \varphi_1 \; ; \; \ldots ; \; X_n =_{\alpha n} \varphi_n$

if there is a possibly infinite derivation for

$\emptyset |\!- X_1 : s_{init}$

such that, for each infinite derivation path,

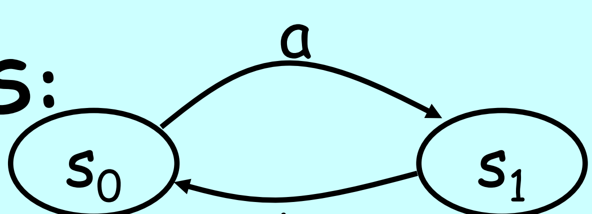
$\alpha_j = \nu$ for the least j such that

X_j is unfolded infinitely often.

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X: s_0 \mid\!\!- X: s_0}{X: s_0 \mid\!\!- [b]X: s_1}}{\emptyset \vdash \lambda X.[b]X: s_0 \to s_1}}{\emptyset \vdash F: s_0 \to s_1} \qquad \dfrac{\dfrac{\dots}{\emptyset \vdash \langle a\rangle(F\ A): s_0}}{\emptyset \vdash A: s_0}}{\emptyset \vdash F\ A: s_1}}{\dfrac{\emptyset \vdash \langle a\rangle(F\ A): s_0}{\emptyset \vdash A: s_0}}$$

LTS:



HES

$A =_\nu \langle a\rangle(F\ A);$

$F =_\mu \lambda X.[b]X$

Correctness of Type System

- **Theorem:**

  $L \models \varphi$

  if and only if

  $\vdash_L \varphi$

- **Corollary:**

  $L \models \varphi$ can be decided in time polynomial in the size of φ, if the following parameters are fixed:
  - L
  - the largest size of types in φ
  - alternation depth of φ

Outline

- Reviews of HORS model checking and HFL model checking
- From HORS to HFL model checking
- From HFL to HORS model checking
- Type system for HFL model checking
- Related work and Conclusion

Related Work

- **HORS model checking**
  - decidability [Knapik+02][Ong06]...
  - type-based characterization [K09][K&Ong09]
  - algorithms [K09][K11][Ramsay+14]...
  - applications [K09][K+11][Ong+11]...
- **HFL model checking**
  - decidability [Viswanathan² 04]
  - complexity [Axelsson+ 07]
  - applications [Viswanathan² 04][Lange+ 12]

Related Work

- Type-based characterization of HORS model checking [K 09][K&Ong 09]
  inspired:
  - translation from HORS to HFL model checking
  - type-based characterization

- Encoding of big numbers as functions [Jones 01][Tsukada&K 14]

- Reduction from HORS model checking to nested least/greatest fixedpoint computation [Salvati&Walukiewicz, CSL15]

Conclusion

- Revealed close relationships between HORS/HFL model checking through:
  - order-preserving mutual reductions
  - type-based characterization of HFL model checking similar to that of HORS model checking
- Future work: mutual transfer of results (e.g. practical model checking algorithms)