Nested Timed Automata

Guoqiang Li

Shanghai Jiao Tong University

Feb. 9, 2014

Joint work with Xiaojuan Cai, Mizuhito Ogawa and Shoji Yuen.

Motivation

Hybrid automata extend timed automata with various rates of clocks;
We would like to extend timed automata with (time-sensitive) context switches.

- (Recursive) Procedure calls
- Multi-level interrupt handlings

Need to deal with 'local' clocks.

A Usual Automata-Based Program Analysis

```
int lek = 10;
Foo()
int x, y;
⋮
if x > y then Foo();
⋮
```

A Usual Automata-Based Program Analysis

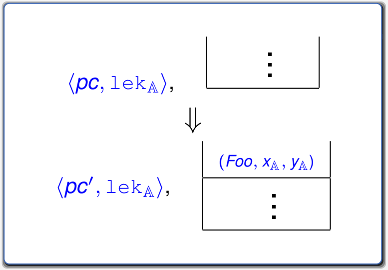
int $lek = 10$;
`Foo()`

int x, y;

⋮

if $x > y$ **then** `Foo();`

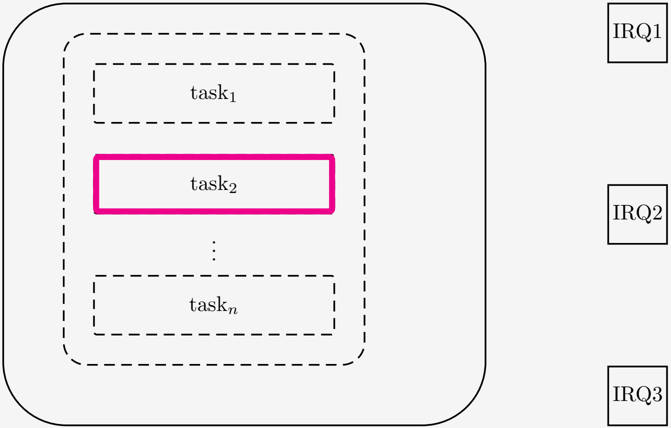
⋮



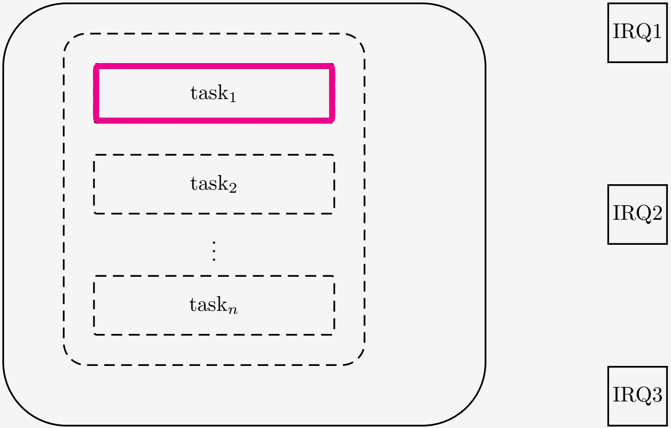
Procedure with Local Clocks

```
Tfoo() {
 clock x, y;
    ⋮
 reset(y);
    ⋮
 if x < 10 && y <= 5 then Tfoo();
 else return;
  ...
}
```

Multilevel (Nested) Interrupts

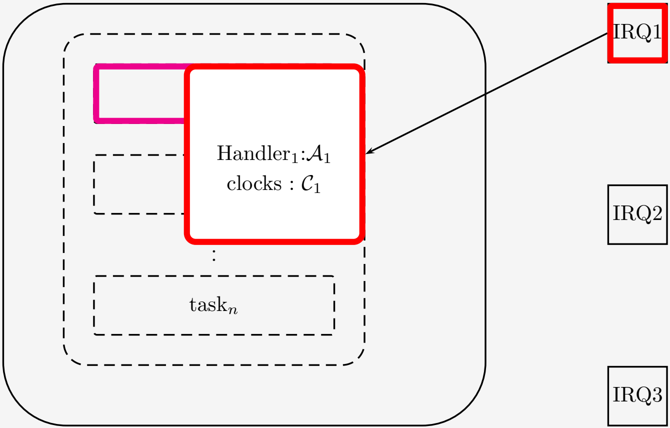


Multilevel (Nested) Interrupts



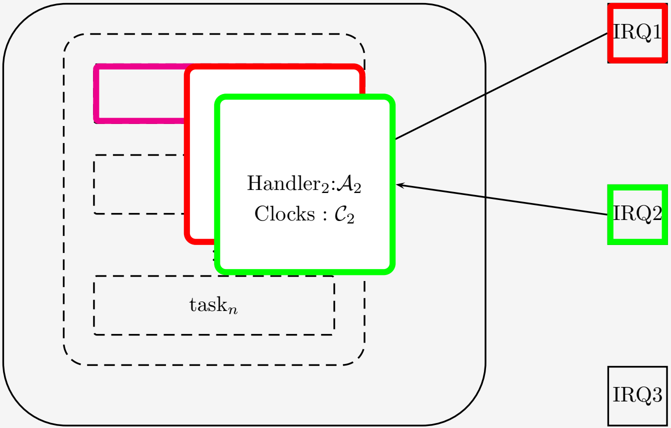
Multilevel (Nested) Interrupts

- Interrupt handlers override the behavior by $\mathcal{A}_i$.



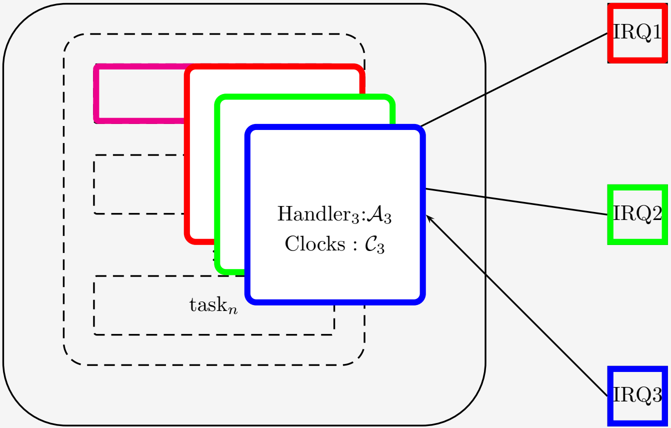
Multilevel (Nested) Interrupts

- Interrupt handlers override the behavior by $\mathcal{A}_i$.



Multilevel (Nested) Interrupts

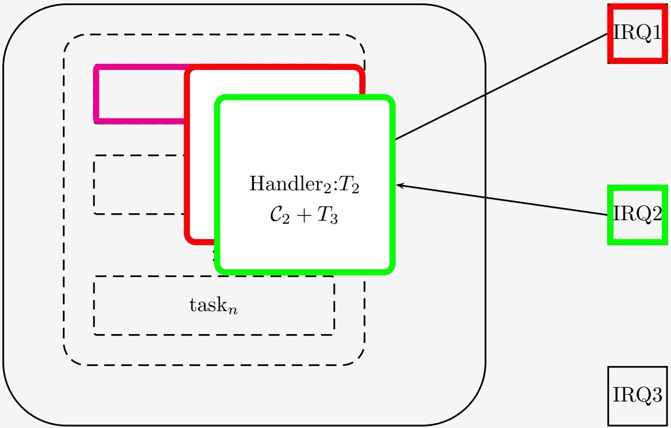
- Interrupt handlers override the behavior by $\mathcal{A}_i$.



Multilevel (Nested) Interrupts

- The behavior is resumed after the handlers terminate.

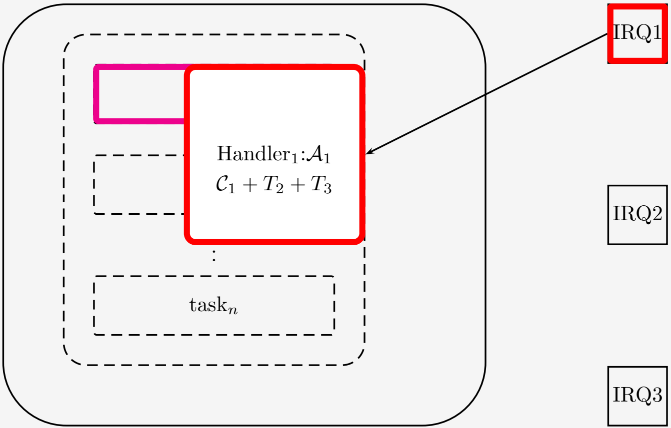
Clock values of $\mathcal{C}_2$ are changed.



Multilevel (Nested) Interrupts

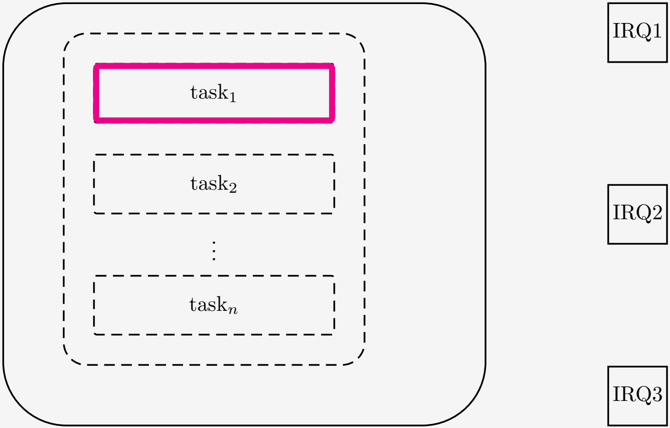
- The behavior is resumed after the handlers terminate.

Clock values of $\mathcal{C}_1$ are changed.



Multilevel (Nested) Interrupts

- The behavior is resumed after the handlers terminate.



Behavioral Model

- A nested timed automaton is a pushdown system whose stack symbols are *timed automata*.
- It either behaves as the top TA in the stack, or switches from one TA to another by *pushing*, *popping*, and *altering* the top TA.
- When time passage happens, all clocks of these TAs in the stack *elapse uniformly*.

Contents

- Timed Automata
- Nested timed automata (NeTA)
- State reachability is decidable via translation into DTPDA (dense timed pushdown automata [Abdulla et.al. LICS2012])
- Correctness of the translation.
- Conclusion

Timed Automata (TA)

$\mathcal{A} = (Q, q_0, F, X, \Delta)$, where

- Q is a finite set of control locations, with the initial location $q_0 \in Q$,
- $F \subseteq Q$ is the set of final locations,
- X is a finite set of clocks,
- $\Delta \subseteq Q \times \mathcal{O} \times Q$, where $\mathcal{O}$ is a set of operations. A transition $q_1 \xrightarrow{\phi} q_2$, where ϕ is either of

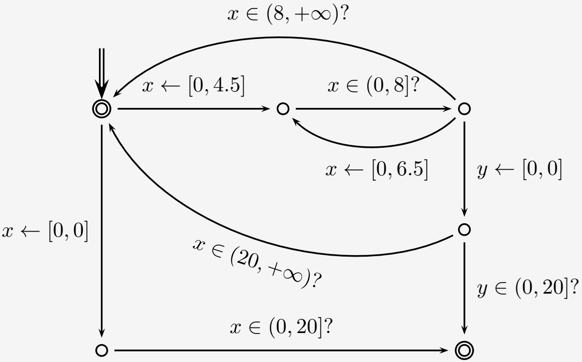
  Local ϵ,

  Test $x \in I?$,

  Assignment $x \leftarrow I$.

Clock updates, Diagnal-free and convex constraints, No invariants

Timed Automata (TA) [An Example]



Nested Timed Automata

Nested Timed Automata

$\mathcal{N} = (T, \mathcal{A}_0, \Delta)$, where

- T is a finite set of TA, with the initial timed automaton $\mathcal{A}_0 \in T$,
- $\Delta \subseteq T \times \mathcal{P} \times (T \cup \{\varepsilon\})$, where $\mathcal{P} = \{push, pop, internal\}$.
- A rule $(\mathcal{A}_i, \Phi, \mathcal{A}_j) \in \Delta$ is written as $\mathcal{A}_i \xrightarrow{\Phi} \mathcal{A}_j$, where

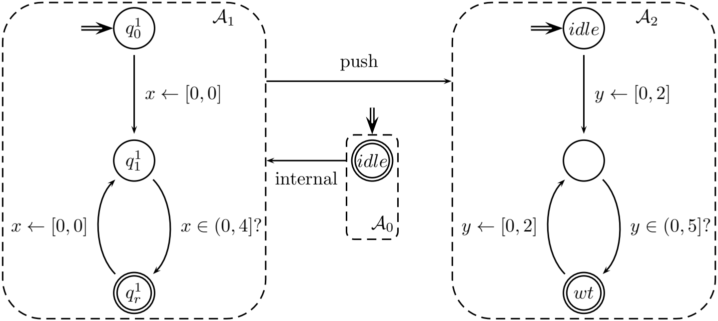
$$\begin{aligned} \text{Push} \quad & \mathcal{A}_i \xrightarrow{push} \mathcal{A}_j, \\ \text{Pop} \quad & \mathcal{A}_i \xrightarrow{pop} \varepsilon, \text{ and} \\ \text{Internal} \quad & \mathcal{A}_i \xrightarrow{internal} \mathcal{A}_j. \end{aligned}$$

Semantics of NeTA

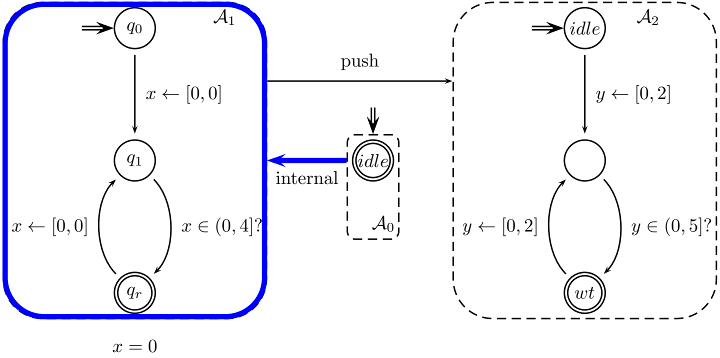
Given an NeTA $(T, \mathcal{A}_0, \Delta)$, a configuration is a stack, and the stack alphabet is a tuple $\langle \mathcal{A}, q, \nu \rangle$, The transition of NeTA is represented as follows:

- Progress transitions: $c \xrightarrow{t}_{\mathcal{N}} c + t$.
- Discrete transitions: $c \xrightarrow{\phi}_{\mathcal{N}} c'$
  - **Intra-action** $\langle \mathcal{A}, q, \nu \rangle c \xrightarrow{\phi}_{\mathcal{N}} \langle \mathcal{A}, q', \nu' \rangle c$
  - **Push** $\langle \mathcal{A}, q, \nu \rangle c \xrightarrow{push}_{\mathcal{N}} \langle \mathcal{A}', q_0(\mathcal{A}'), \nu'_{\mathbf{0}} \rangle \langle \mathcal{A}, q, \nu \rangle c$
  - **Pop** $\langle \mathcal{A}, q, \nu \rangle c \xrightarrow{pop}_{\mathcal{N}} c$ if $q \in F(\mathcal{A})$.
  - **Inter-action** $\langle \mathcal{A}, q, \nu \rangle c \xrightarrow{internal}_{\mathcal{N}} \langle \mathcal{A}', q_0(\mathcal{A}'), \nu'_{\mathbf{0}} \rangle c$ if $q \in F(\mathcal{A})$.

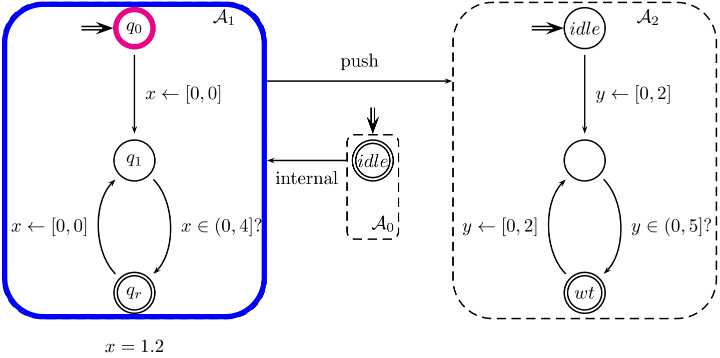
Nested Timed Automata [An Example]



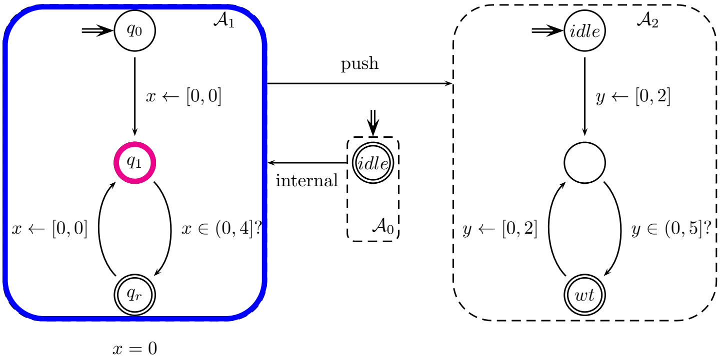
Nested Timed Automata [An Example]



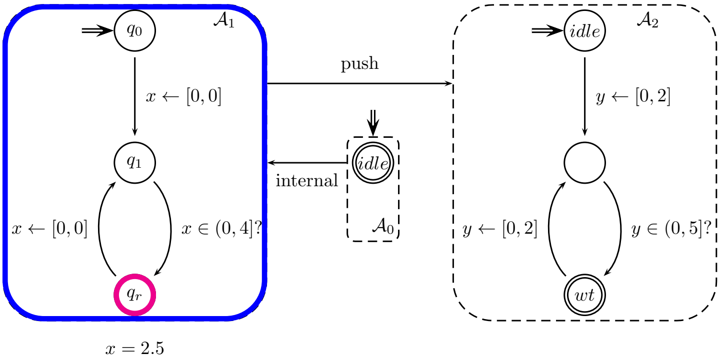
Nested Timed Automata [An Example]



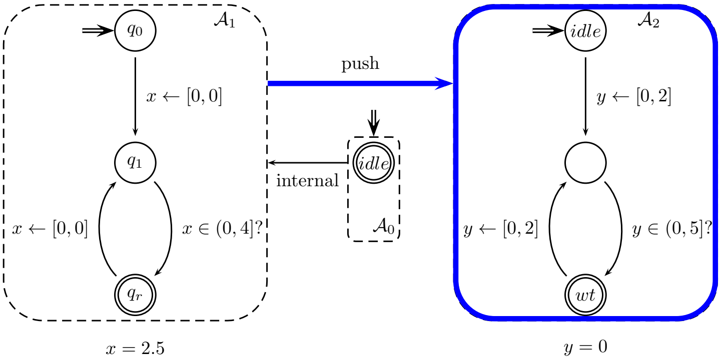
Nested Timed Automata [An Example]



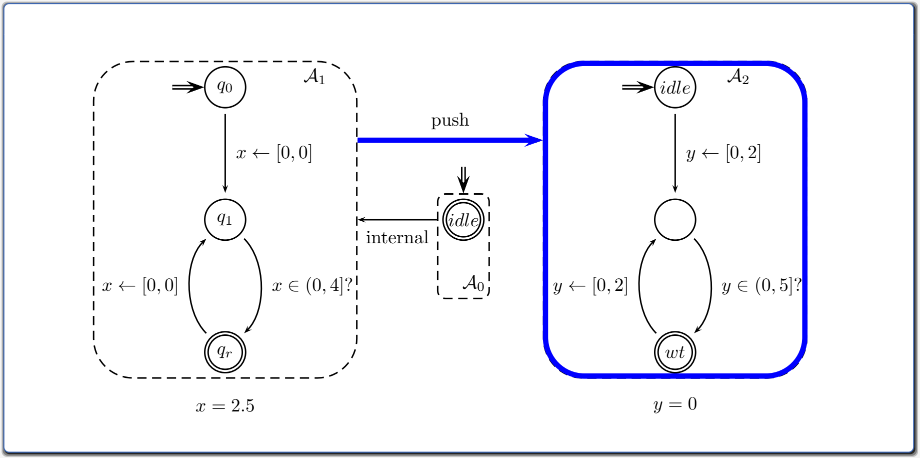
Nested Timed Automata [An Example]



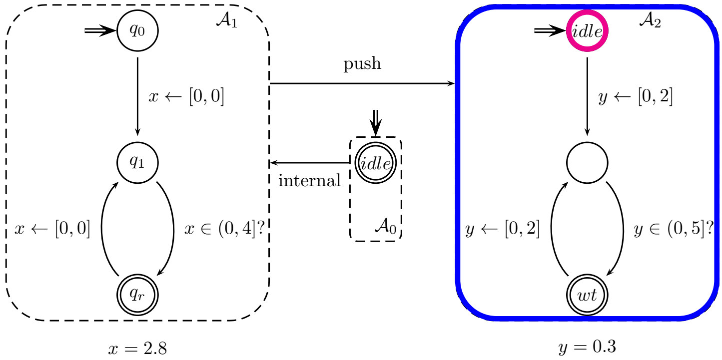
Nested Timed Automata [An Example]



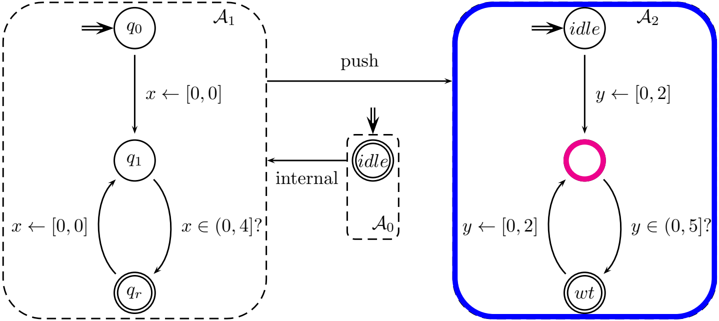
Nested Timed Automata [An Example]



Nested Timed Automata [An Example]



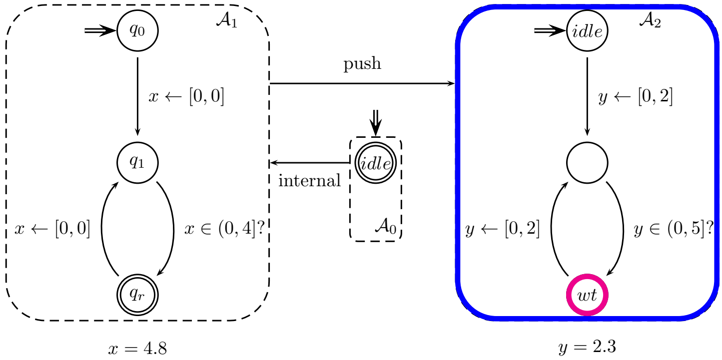
Nested Timed Automata [An Example]



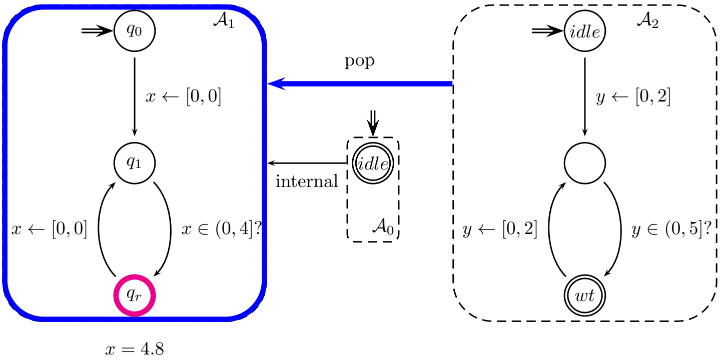
$x = 3.5$

$y = 1$

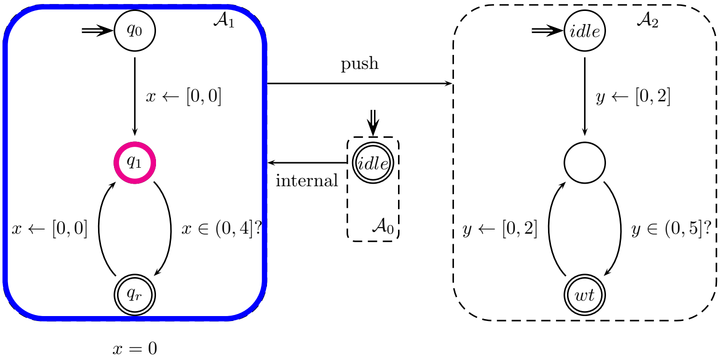
Nested Timed Automata [An Example]



Nested Timed Automata [An Example]

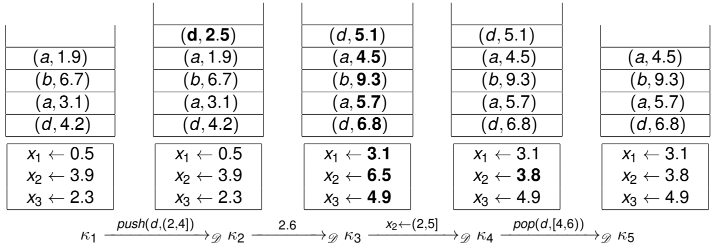


Nested Timed Automata [An Example]



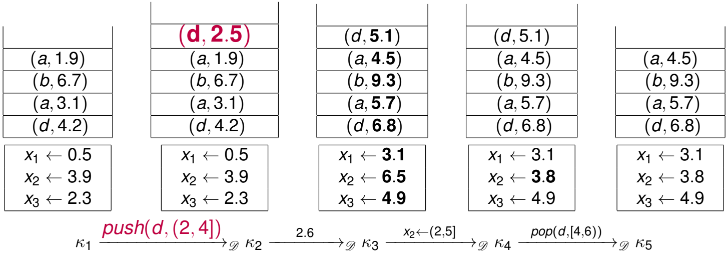
TRANSLATION TO DTPDA

- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$



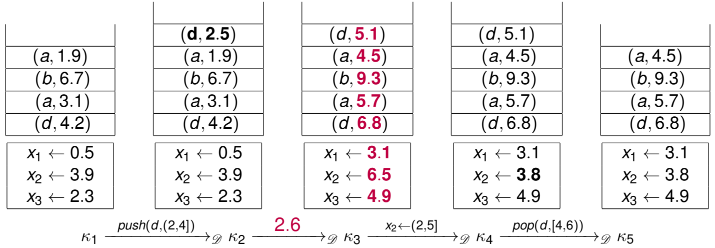
Dense Timed PDA [Abdulla et.al. 2012]

- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$



Dense Timed PDA [Abdulla et.al. 2012]

- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$



Dense Timed PDA [Abdulla et.al. 2012]

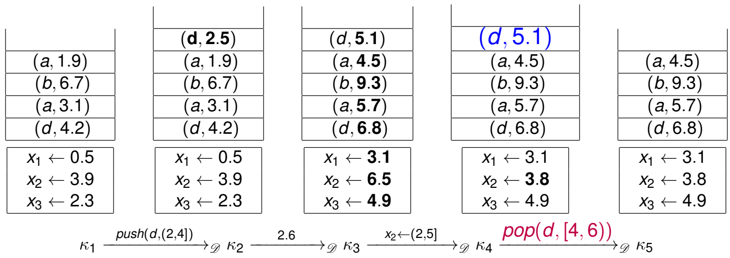
- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$

κ_1	κ_2	κ_3	κ_4	κ_5
	$(\mathbf{d}, \mathbf{2.5})$	$(d, \mathbf{5.1})$	$(d, 5.1)$	
$(a, 1.9)$	$(a, 1.9)$	$(a, \mathbf{4.5})$	$(a, 4.5)$	$(a, 4.5)$
$(b, 6.7)$	$(b, 6.7)$	$(b, \mathbf{9.3})$	$(b, 9.3)$	$(b, 9.3)$
$(a, 3.1)$	$(a, 3.1)$	$(a, \mathbf{5.7})$	$(a, 5.7)$	$(a, 5.7)$
$(d, 4.2)$	$(d, 4.2)$	$(d, \mathbf{6.8})$	$(d, 6.8)$	$(d, 6.8)$
$x_1 \leftarrow 0.5$	$x_1 \leftarrow 0.5$	$x_1 \leftarrow \mathbf{3.1}$	$x_1 \leftarrow 3.1$	$x_1 \leftarrow 3.1$
$x_2 \leftarrow 3.9$	$x_2 \leftarrow 3.9$	$x_2 \leftarrow \mathbf{6.5}$	$x_2 \leftarrow \mathbf{3.8}$	$x_2 \leftarrow 3.8$
$x_3 \leftarrow 2.3$	$x_3 \leftarrow 2.3$	$x_3 \leftarrow \mathbf{4.9}$	$x_3 \leftarrow 4.9$	$x_3 \leftarrow 4.9$

$$\kappa_1 \xrightarrow{push(d,(2,4])}_{\mathscr{D}} \kappa_2 \xrightarrow{2.6}_{\mathscr{D}} \kappa_3 \xrightarrow{x_2 \leftarrow (2,5]}_{\mathscr{D}} \kappa_4 \xrightarrow{pop(d,[4,6))}_{\mathscr{D}} \kappa_5$$

Dense Timed PDA [Abdulla et.al. 2012]

- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$



DTPDA

$\mathcal{D} = \langle S, s_0, \Gamma, C, \Delta \rangle$, where

- S is a finite set of states with the initial state $s_0 \in S$,
- Γ is a finite stack alphabet,
- C is a finite set of clocks, and
- $\Delta \subseteq S \times \mathcal{O} \times S$ is a finite set of transitions.

A transition $s_1 \xrightarrow{\phi} s_2$, where ϕ is either of

Local:	ϵ		
Test:	$x \in I?$	Assignment:	$x \leftarrow I$
Push:	$push(\gamma, I)$	Pop:	$pop(\gamma, I)$

A Variation of DTPDA for Encoding NeTA

Push *push*(γ, I) pushes γ to the top of the stack, with the age in the interval I.

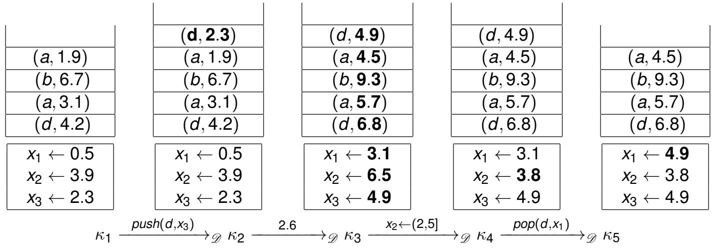
Pop *pop*(γ, I) pops the top-most stack symbol provided that this symbol is γ and its age belongs to I.

Push$_A$ *push*(γ, x) pushes γ to the stack associated with a local age with the value of the x's value.

Pop$_A$ *pop*(γ, x) pops γ from a stack and assigns value of its local age to the global clock x.

An Example of the DTPDA Variant

- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$



An Example of the DTPDA Variant

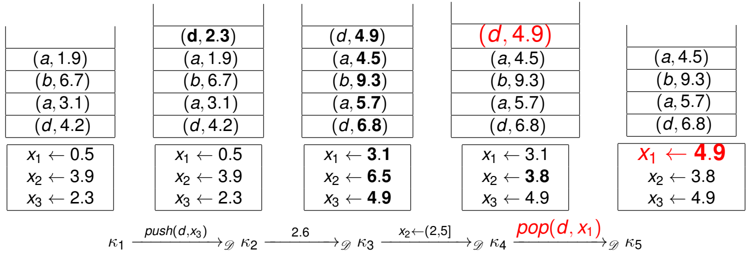
- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$

κ_1	κ_2	κ_3	κ_4	κ_5
	$(d, 2.3)$	$(d, 4.9)$	$(d, 4.9)$	
$(a, 1.9)$	$(a, 1.9)$	$(a, 4.5)$	$(a, 4.5)$	$(a, 4.5)$
$(b, 6.7)$	$(b, 6.7)$	$(b, 9.3)$	$(b, 9.3)$	$(b, 9.3)$
$(a, 3.1)$	$(a, 3.1)$	$(a, 5.7)$	$(a, 5.7)$	$(a, 5.7)$
$(d, 4.2)$	$(d, 4.2)$	$(d, 6.8)$	$(d, 6.8)$	$(d, 6.8)$
$x_1 \leftarrow 0.5$	$x_1 \leftarrow 0.5$	$x_1 \leftarrow 3.1$	$x_1 \leftarrow 3.1$	$x_1 \leftarrow 4.9$
$x_2 \leftarrow 3.9$	$x_2 \leftarrow 3.9$	$x_2 \leftarrow 6.5$	$x_2 \leftarrow 3.8$	$x_2 \leftarrow 3.8$
$x_3 \leftarrow 2.3$	$x_3 \leftarrow 2.3$	$x_3 \leftarrow 4.9$	$x_3 \leftarrow 4.9$	$x_3 \leftarrow 4.9$

$$\kappa_1 \xrightarrow{push(d, x_3)}_{\mathscr{D}} \kappa_2 \xrightarrow{2.6}_{\mathscr{D}} \kappa_3 \xrightarrow{x_2 \leftarrow (2,5]}_{\mathscr{D}} \kappa_4 \xrightarrow{pop(d, x_1)}_{\mathscr{D}} \kappa_5$$

An Example of the DTPDA Variant

- State: $S = \{\bullet\}$
- clocks: $\mathcal{C} = \{x_1, x_2, x_3\}$,
- Stack symbols: $\Gamma = \{a, b, d\}$



State Reachability of DTPDA

State reachability

s is reachabile if for some w' and ν',
$\langle s_0, w, \nu\rangle \rightarrow^* \langle s, w', \nu'\rangle$
where $\rightarrow = \xrightarrow{\text{Local}} \cup \xrightarrow{\text{Test}} \cup \xrightarrow{\text{Assignment}} \cup \xrightarrow{\text{Push}} \cup \xrightarrow{\text{Pop}}$

Theorem

The state reachability of DTPDA is decidable.

[Abdulla et.al. LICS2012]
Region construction by fractional parts of ages
Rotation at poping with shadow variables consistency

State Reachability of DTPDA Variant

State reachability

s is reachabile if for some w' and ν',
$\langle s_0, w, \nu \rangle \rightarrow^* \langle s, w', \nu' \rangle$
where $\rightarrow = \xrightarrow{\text{Local}} \cup \xrightarrow{\text{Test}} \cup \xrightarrow{\text{Assignment}} \cup \xrightarrow{\text{Push}} \cup \xrightarrow{\text{Pop}} \cup \xrightarrow{\text{Push}_A} \cup \xrightarrow{\text{Pop}_A}$

Theorem

The state reachability of DTPDA variant is decidable.

Small modification to LICS2012 proof works for the variant.

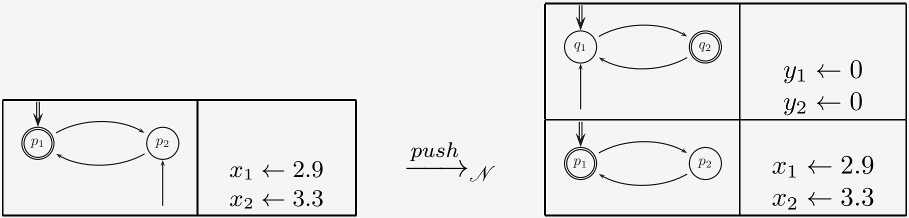
Another proof idea via WSPDS: [Cai et.al. 2013, 2014]

Encoding to DTPDA Variant

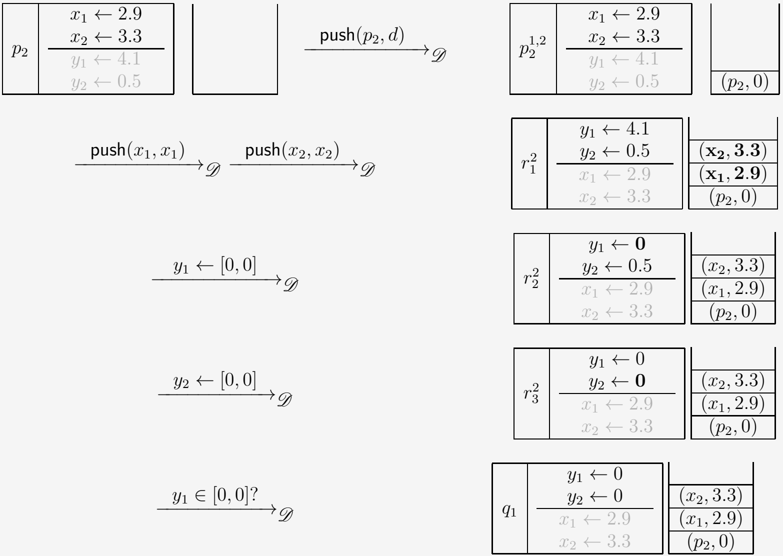
$\mathcal{A}_i \xrightarrow{push} \mathcal{A}_j$	$q_i \xrightarrow{push(q_i,d)} p_1^{i,j} \xrightarrow{push(x_1^i,x_1^i)} \cdots p_{m_i}^{i,j} \xrightarrow{push(x_{m_i}^i,x_{m_i}^i)} p_{m_i+1}^{i,j} \xrightarrow{\epsilon}$ $r_1^j \xrightarrow{x_1^j \leftarrow [0,0]} r_2^j \cdots r_{m_j+1}^j \xrightarrow{\epsilon} t^j \xrightarrow{x_1^j \in [0,0]?} q_0(\mathcal{A}_j)$
$\mathcal{A}_i \xrightarrow{pop} \epsilon$	$q_i \xrightarrow{\epsilon} o \xrightarrow{pop(x_{m_i+1}^i,x_{m_i+1}^i)} \cdots \xrightarrow{pop(x_1^i,x_1^i)} o \xrightarrow{pop(q,d)} q$
$\mathcal{A}_i \xrightarrow{internal} \mathcal{A}_j$	$q_i \xrightarrow{\epsilon} r_1^j \xrightarrow{x_1^j \leftarrow [0,0]} r_2^j \cdots r_{m_j+1}^j \xrightarrow{\epsilon} t^j \xrightarrow{x_1^j \in [0,0]?} q_0(\mathcal{A}_j)$

- The key of the encoding is to synchronize the initial value of clocks, and
- Storing and restoring clocks values simultaneously when timed context switches.

Encoding Push of NeTA to DTPDA



Encoding Push of NeTA to DTPDA



Correctness

Lemma

Given an NeTA $\mathcal{N}$, its encoding $\mathcal{E}(\mathcal{N})$, and configurations c, c' of $\mathcal{N}$.

- **(Preservation)** *if* $c \longrightarrow c'$*, then* $[\![c]\!] \hookrightarrow^* [\![c']\!]$*;*
- **(Reflection)** *if* $[\![c]\!] \hookrightarrow^* \kappa$*,*
  1. *there exists* c' *such that* $\kappa = [\![c']\!]$ *and* $c \longrightarrow^* c'$*, or*
  2. κ *is not an encoded configuration, and there exists* c' *such that* $\kappa \hookrightarrow^* [\![c']\!]$ *by discrete transitions and* $c \longrightarrow^* c'$*.*

Correctness

Lemma

Given an NeTA $\mathcal{N}$, its encoding $\mathcal{E}(\mathcal{N})$, and configurations c, c' of $\mathcal{N}$.

- **(Preservation)** *if* $c \longrightarrow c'$, *then* $[\![c]\!] \hookrightarrow^* [\![c']\!]$;
- **(Reflection)** *if* $[\![c]\!] \hookrightarrow^* \kappa$,
  1. *there exists* c' *such that* $\kappa = [\![c']\!]$ *and* $c \longrightarrow^* c'$, *or*
  2. κ *is not an encoded configuration, and there exists* c' *such that* $\kappa \hookrightarrow^* [\![c']\!]$ *by discrete transitions and* $c \longrightarrow^* c'$.

Theorem

The state reachability problem of NeTA is decidable.

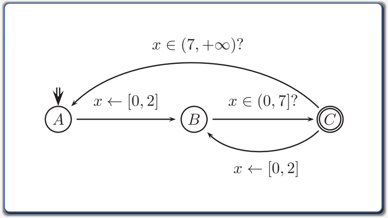
Deadline Analysis for multilevel interrupt handling

Deadline Analysis

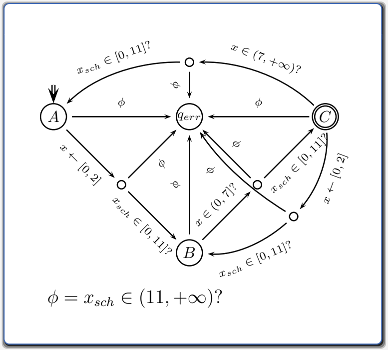
- Interrupt handlers as guarded timed automata.
- Interrupt request as $push_A$ operation.
- Return from interrupt as pop_A operation.
- Deadline violation as the reachability to *err* state.

Timed Automata with Deadline

Add a stopwatch to check deadline and an error state



$\Rightarrow$
Deadline
=11



Fall into q_{err} when deadline passed.

Deadline Analysis

$$\textit{Handler}_i \quad : \quad \text{guard}(\mathcal{A}_i, d_i)$$

guard($\mathcal{A}_i, d_i$) adds the deadline to $\mathcal{A}$

$$\textit{Interrupt} \quad : \quad \mathcal{A}_i \xrightarrow{\text{push}} \mathcal{A}_|$$

A_i may interrupt A_j

$$\textit{Initial} \quad : \quad \mathcal{A}_0 = \textit{Task}_1 \mid\mid \textit{Task}_2 \mid\mid \cdots \mid\mid \textit{Task}_n$$

Interrupt fails to be handled if q_{err} in some $\mathcal{A}_i (i > 0)$ is reachabile.

Ongoing Work

- Three kinds of clocks: global clocks, local clocks and stopwatch clocks.
- Reachability problems of pushdown systems under respective kind of clocks are decidable, however:

Ongoing Work

- Three kinds of clocks: global clocks, local clocks and stopwatch clocks.
- Reachability problems of pushdown systems under respective kind of clocks are decidable, however:
  - Under global and stopwatch clocks: Undecidable [Benerecetti et.al, 2010]

Ongoing Work

- Three kinds of clocks: global clocks, local clocks and stopwatch clocks.
- Reachability problems of pushdown systems under respective kind of clocks are decidable, however:
  - Under global and stopwatch clocks: Undecidable [Benerecetti et.al, 2010]
  - Under global and local clocks: Decidable [Li et. al., 2014]
  - Under local and stopwatch clocks: ???

Ongoing Work

- Three kinds of clocks: global clocks, local clocks and stopwatch clocks.
- Reachability problems of pushdown systems under respective kind of clocks are decidable, however:
  - Under global and stopwatch clocks: Undecidable [Benerecetti et.al, 2010]
  - Under global and local clocks: Decidable [Li et. al., 2014]
  - Under local and stopwatch clocks: ???
- Reachability problem of NeTA with invariant is positive.

Future Implementation

- Develop a tool based on a restrictive class such that a pop action occurs only with an integer-valued age.
- This subclass can be encoded into UTPDA (without local age).
- Encode UTPDA to weighted pushdown system to gain the efficiency.

Related Work

- Timed PDA [Bouajjani et. al.,1994]
  PDA with Global clocks
- Timed recursive state machines [Benerecetti et.al, 2010]
  Extended PDA with two stacks for states and clocks
- Recursive timed automata [Trivedi et.al., 2010]
  Local clocks stop
- Hierarchical timed automata [David et.al., 2001]
  Static hierarchy

Conclusion

- An NeTA is a pushdown system with a finite set of TA as stack symbols.
- All clocks in the stack elapse uniformly.
- The state reachability is decidable by encoding to DTPDA with an extension of local clock assignment.

Thank you!

li.g@sjtu.edu.cn