Real-time Synthesis is Hard!

@ Highlights 2016

Benjamin Monmege (LIF, Aix-Marseille Université)
Thomas Brihaye, Morgane Estiévenart, Hsi-Ming Ho (UMONS)
Gilles Geeraerts (ULB)
Nathalie Sznajder (UPMC, LIP6)

7 September 2016

Reactive synthesis

$\Sigma = \Sigma_C \uplus \Sigma_E$

- controllable actions owned by controller C:
  $\{MoveUp, MoveDown, OpenDoor, Opened, \ldots\}$
- uncontrollable actions owned by environment E:
  $\{0F\text{-}Up, 0F\text{-}Down, \ldots, -1F, 0F, \ldots Open, Close, \ldots\}$

Reactive synthesis

$\Sigma = \Sigma_C \uplus \Sigma_E$

- controllable actions owned by controller C:
  $\{\mathit{MoveUp}, \mathit{MoveDown}, \mathit{OpenDoor}, \mathit{Opened}, \ldots\}$
- uncontrollable actions owned by environment E:
  $\{\mathit{0F\text{-}Up}, \mathit{0F\text{-}Down}, \ldots, \mathit{-1F}, \mathit{0F}, \ldots \mathit{Open}, \mathit{Close}, \ldots\}$

$+$ state (at which floor, opening, $\ldots$)

Reactive synthesis

$\Sigma = \Sigma_C \uplus \Sigma_E$

- controllable actions owned by controller C:
  $\{\mathit{MoveUp}, \mathit{MoveDown}, \mathit{OpenDoor}, \mathit{Opened}, \ldots\}$
- uncontrollable actions owned by environment E:
  $\{\mathit{0F\text{-}Up}, \mathit{0F\text{-}Down}, \ldots, \mathit{-1F}, \mathit{0F}, \ldots \mathit{Open}, \mathit{Close}, \ldots\}$

$+$ state (at which floor, opening, ...)

$+$ timing restrictions (latency, ...)

Reactive synthesis

$\Sigma = \Sigma_C \uplus \Sigma_E$

- controllable actions owned by controller C:
  $\{\mathit{MoveUp}, \mathit{MoveDown}, \mathit{OpenDoor}, \mathit{Opened}, \ldots\}$
- uncontrollable actions owned by environment E:
  $\{\mathit{0F\text{-}Up}, \mathit{0F\text{-}Down}, \ldots, \mathit{-1F}, \mathit{0F}, \ldots \mathit{Open}, \mathit{Close}, \ldots\}$

$+$ state (at which floor, opening, ...)

$+$ timing restrictions (latency, ...)

$=$

Plant $\mathcal{P}$: a DTA over Σ

Reactive synthesis

$\Sigma = \Sigma_C \uplus \Sigma_E$

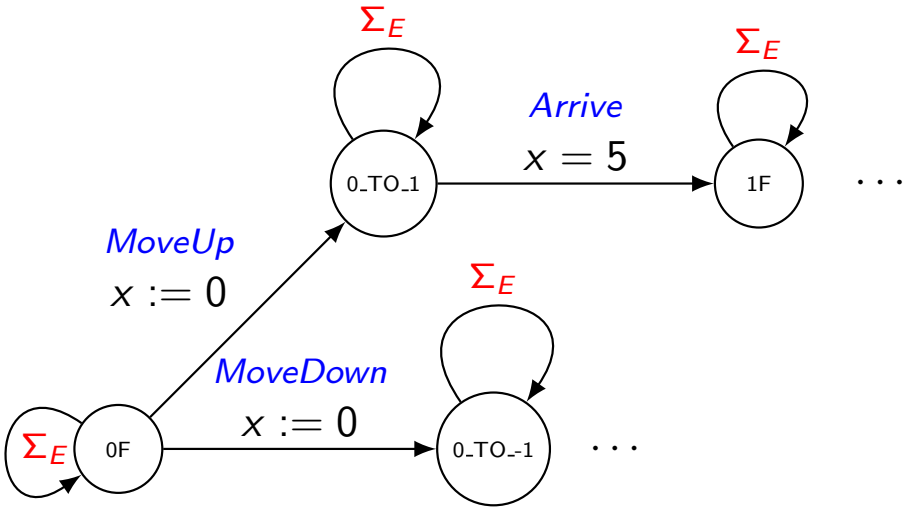
- controllable actions owned by controller C: $\{\mathit{MoveUp}, \mathit{MoveDown}, \mathit{OpenDoor}, \mathit{Opened}, \ldots\}$
- uncontrollable actions owned by environment E: $\{\mathit{0F\text{-}Up}, \mathit{0F\text{-}Down}, \ldots, \mathit{-1F}, \mathit{0F}, \ldots \mathit{Open}, \mathit{Close}, \ldots\}$

$+$ state (at which floor, opening, ...)

$+$ timing restrictions (latency, ...)

$=$

Plant $\mathcal{P}$: a DTA over Σ



Reactive synthesis

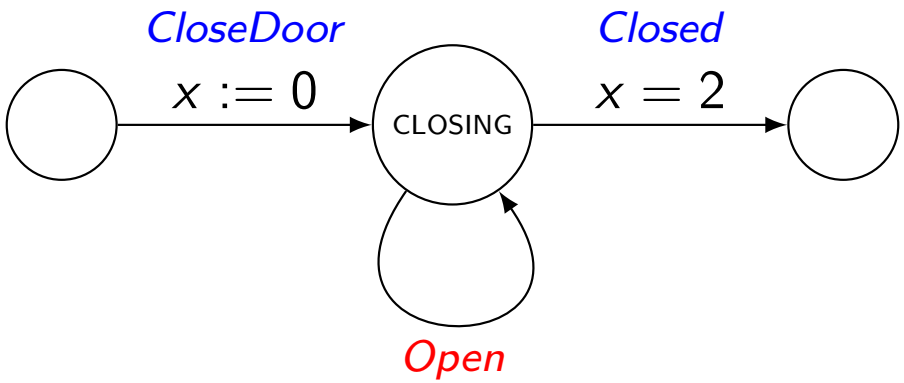
A run of $\mathcal{P}$ can be seen as a play of the *timed game* between C and E.

Reactive synthesis

A run of $\mathcal{P}$ can be seen as a play of the *timed game* between C and E.
In each round, each player proposes a pair (delay, action) **enabled in** $\mathcal{P}$:

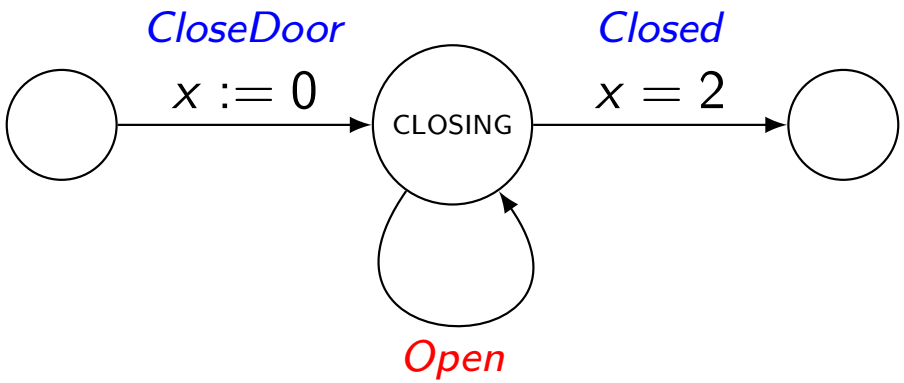
Reactive synthesis

A run of $\mathcal{P}$ can be seen as a play of the *timed game* between C and E.
In each round, each player proposes a pair (delay, action) **enabled in** $\mathcal{P}$:



Reactive synthesis

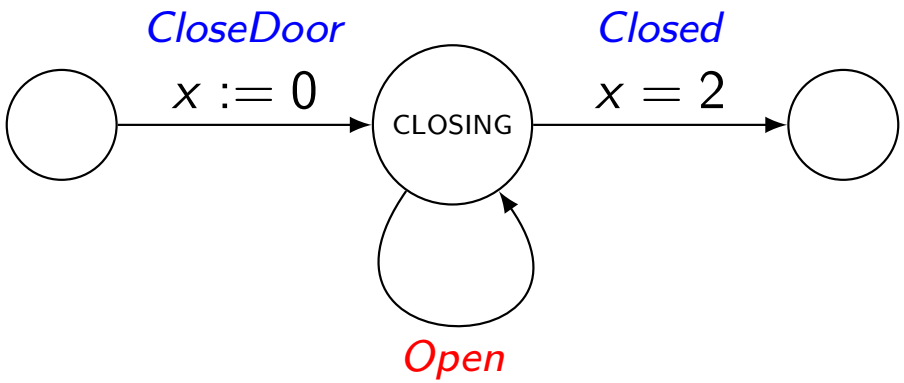
A run of $\mathcal{P}$ can be seen as a play of the *timed game* between C and E.
In each round, each player proposes a pair (delay, action) **enabled in** $\mathcal{P}$:



$(\Delta_C, \mathit{Closed})$ $(\Delta_E, \mathit{Open})$

Reactive synthesis

A run of $\mathcal{P}$ can be seen as a play of the *timed game* between C and E.
In each round, each player proposes a pair (delay, action) **enabled in** $\mathcal{P}$:



$(\Delta_C, \mathit{Closed})$ $(\Delta_E, \mathit{Open})$

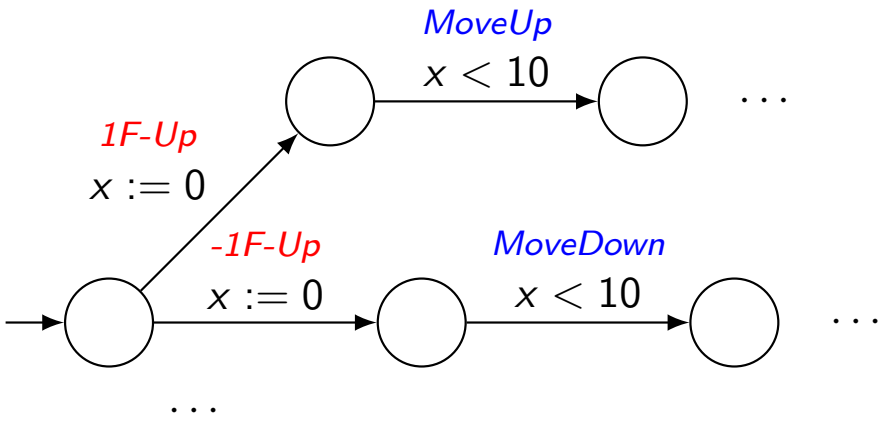
Only action(s) with the shortest delay $\min(\Delta_C, \Delta_E)$ may be played.

Reactive synthesis

‘The lift responds to any call in less than 10 t.u.’

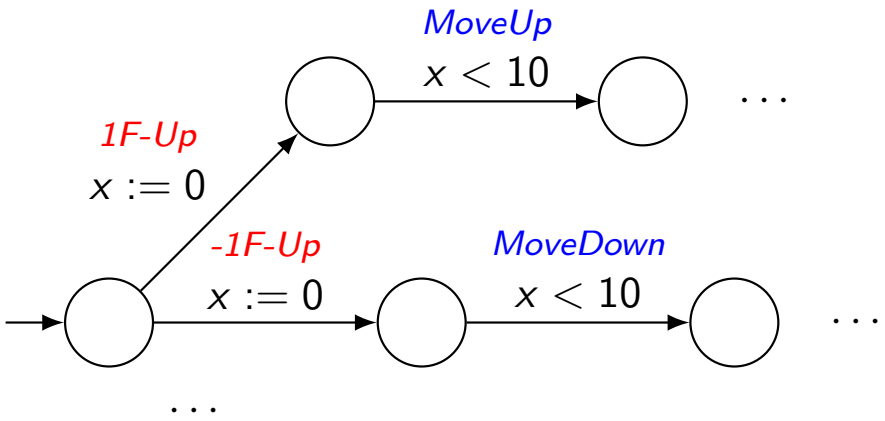
Reactive synthesis

‘The lift responds to any call in less than 10 t.u.’



Reactive synthesis

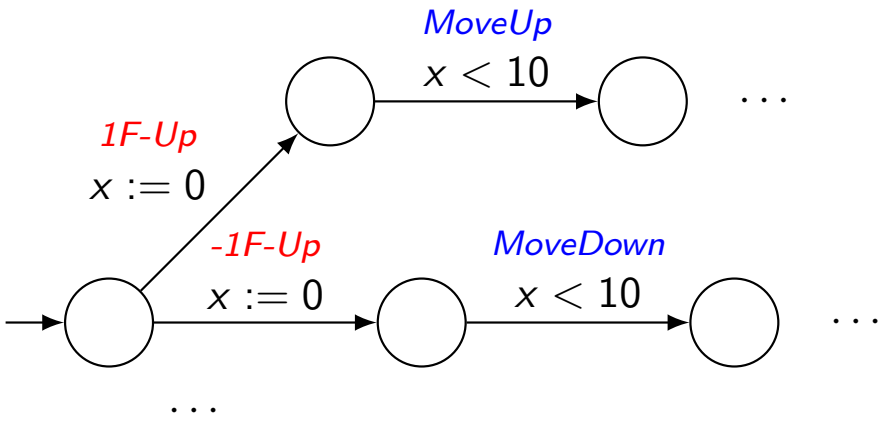
'The lift responds to any call in less than 10 t.u.'



Specification $\mathcal{L}$: a set of timed words over Σ

Reactive synthesis

'The lift responds to any call in less than 10 t.u.'



Specification $\mathcal{L}$: a set of timed words over Σ

Reactive synthesis problem (RS)

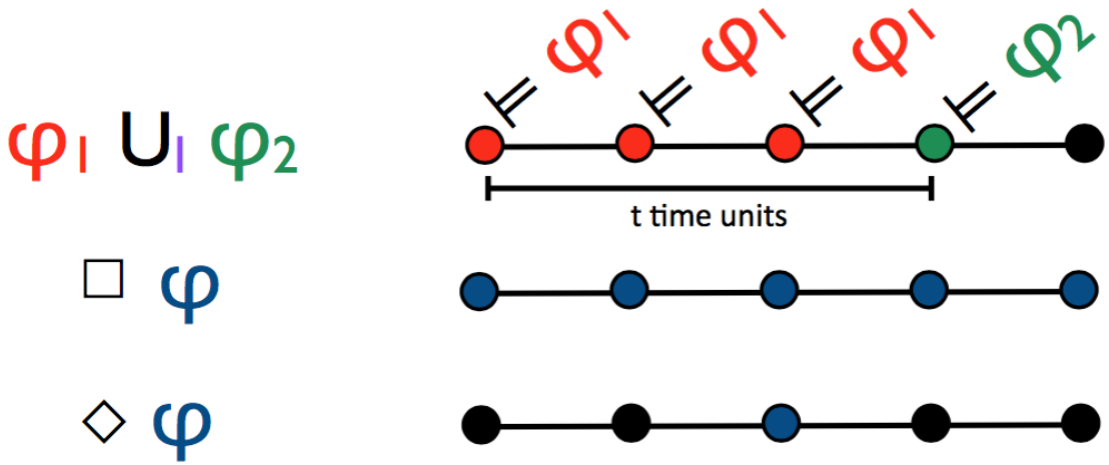
Given plant $\mathcal{P}$ and specification $\mathcal{L}$, find a strategy of Controller such that no matter what Environment does, every play satisfies the specification.

Metric Temporal Logic (MTL)

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathsf{U}_I \, \varphi$$

with $a \in \Sigma$, $I \subseteq [0, \infty)$ with bounds in $\mathbb{Q} \cup \{+\infty\}$

Models: finite (or infinite) timed words $\sigma = (a_1, t_1)(a_2, t_2) \cdots$



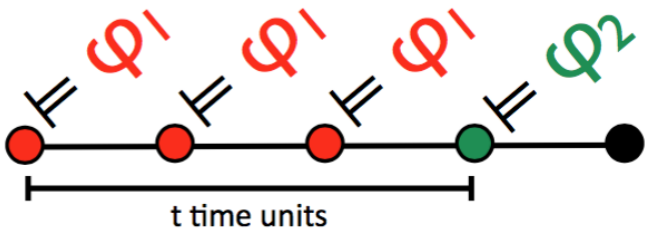
Metric Temporal Logic (MTL)

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathsf{U}_I \, \varphi$$

with $a \in \Sigma$, $I \subseteq [0, \infty)$ with bounds in $\mathbb{Q} \cup \{+\infty\}$

Models: finite (or infinite) timed words $\sigma = (a_1, t_1)(a_2, t_2) \cdots$

$\varphi_1 \, \mathsf{U}_I \, \varphi_2$



$\Box \, \varphi$

$\Diamond \, \varphi$

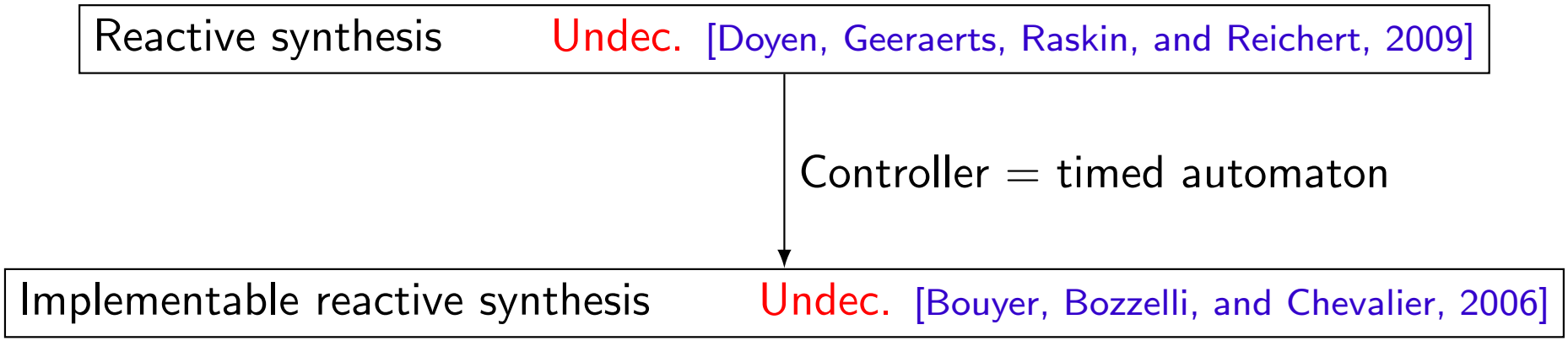
Theorem: [Doyen, Geeraerts, Raskin, and Reichert, 2009]

Reactive synthesis problem is undecidable for ECL (hence, MTL) specifications, even without plant.

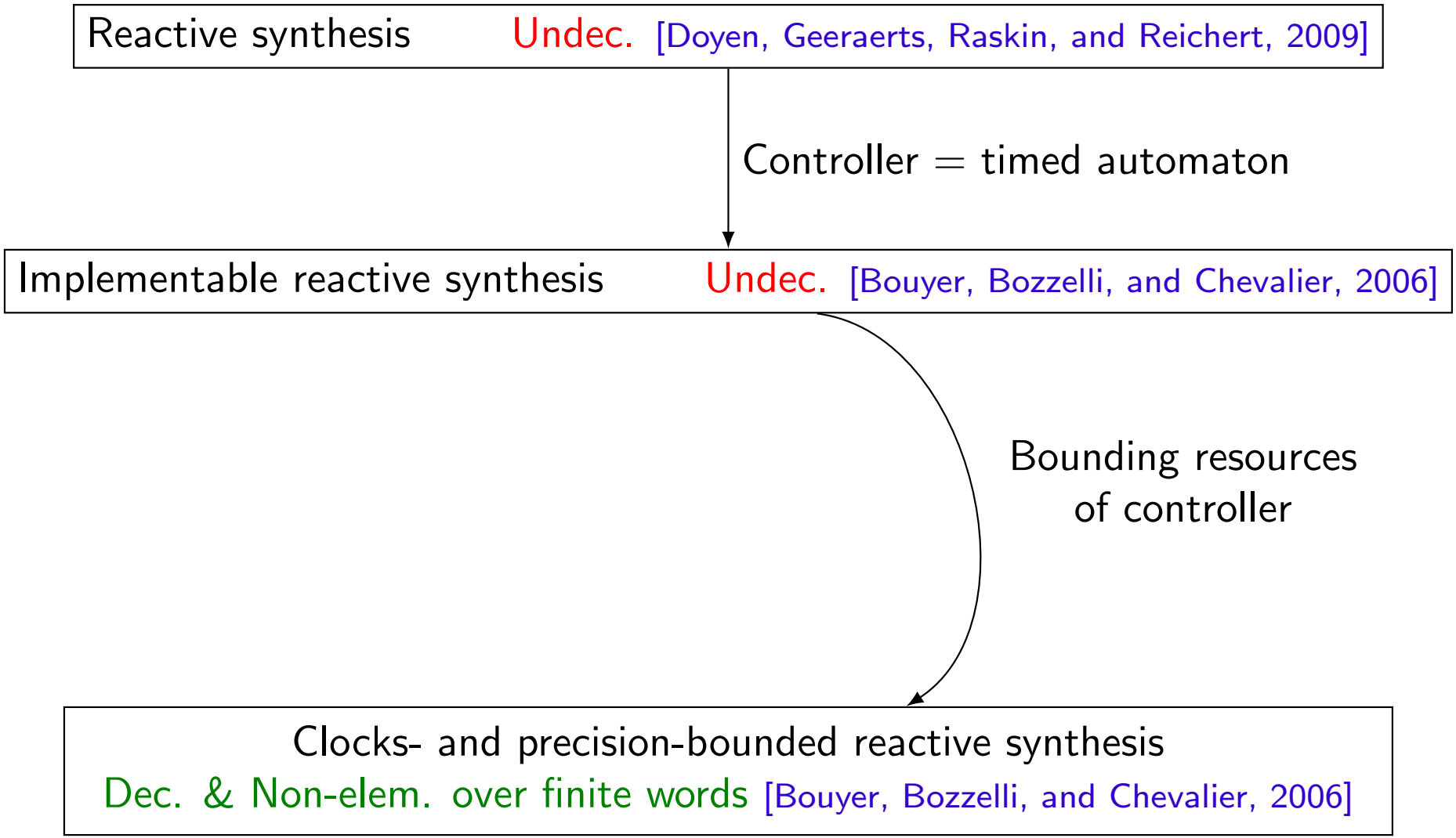
Reactive synthesis for MTL

Reactive synthesis	Undec. [Doyen, Geeraerts, Raskin, and Reichert, 2009]

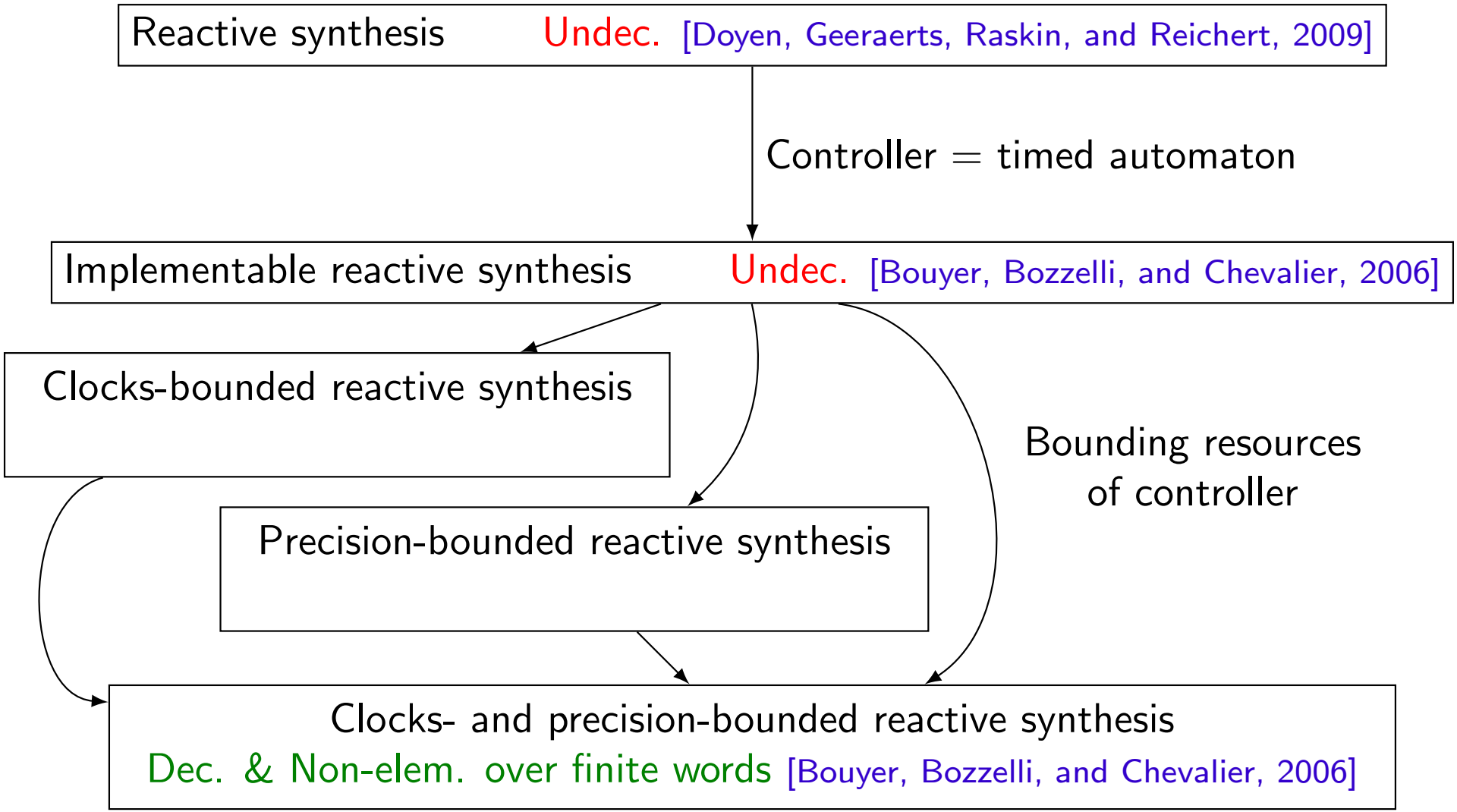
Reactive synthesis for MTL



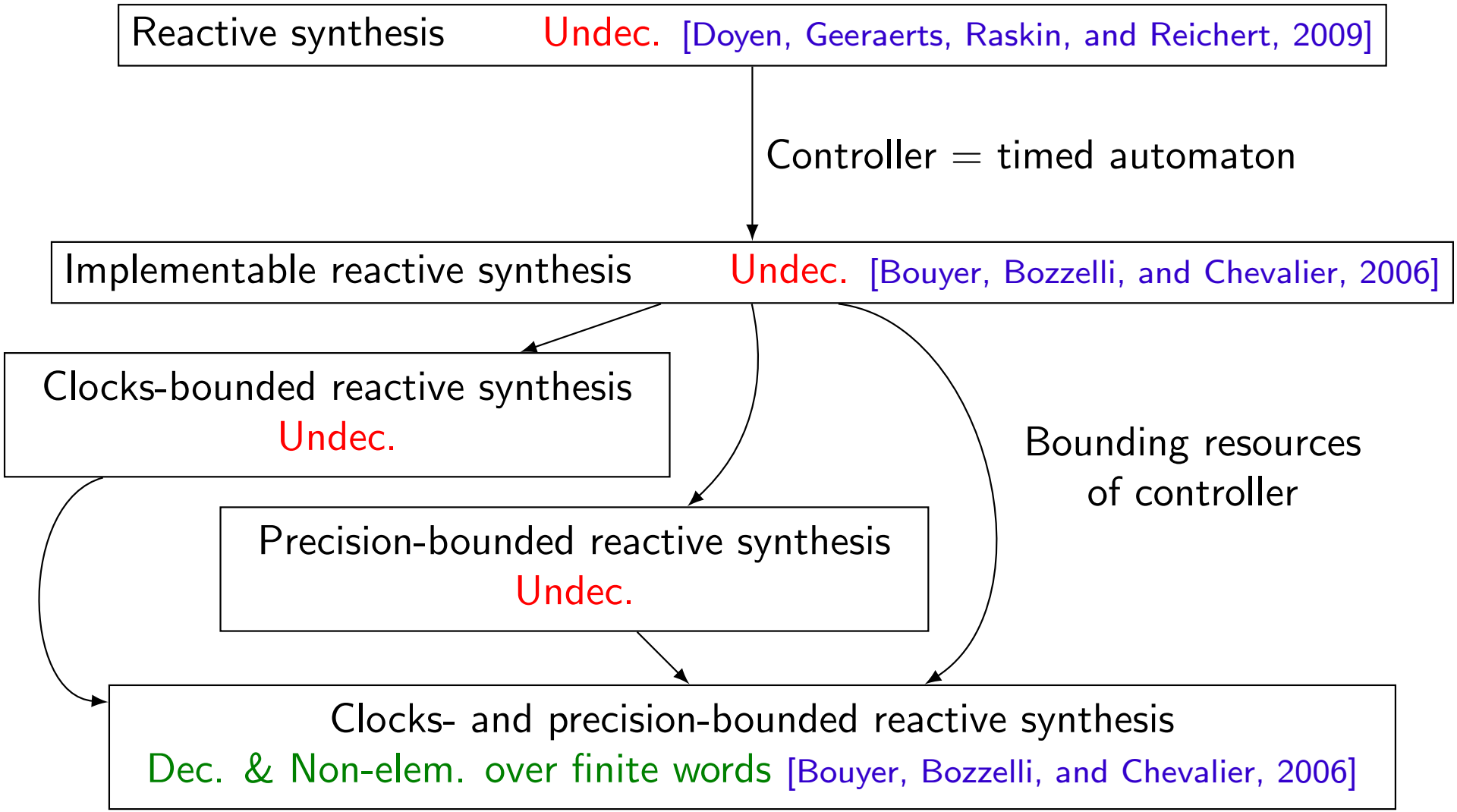
Reactive synthesis for MTL



Reactive synthesis for MTL



Reactive synthesis for MTL



Regaining hope? Less expressive specifications

Undecidability proofs heavily use ‘punctuality’ of MTL: $\mathit{request} \rightarrow \Diamond_{=1} \mathit{grant}$

Regaining hope? Less expressive specifications

Undecidability proofs heavily use 'punctuality' of MTL: $request \rightarrow \Diamond_{=1} grant$

$$request \rightarrow \Diamond_{[1,2]} grant$$

$$request \rightarrow \Diamond_{\leqslant 3} grant$$

Regaining hope? Less expressive specifications

Undecidability proofs heavily use 'punctuality' of MTL: $\mathit{request} \rightarrow \Diamond_{=1} \mathit{grant}$

$$\mathit{request} \rightarrow \Diamond_{[1,2]} \mathit{grant}$$

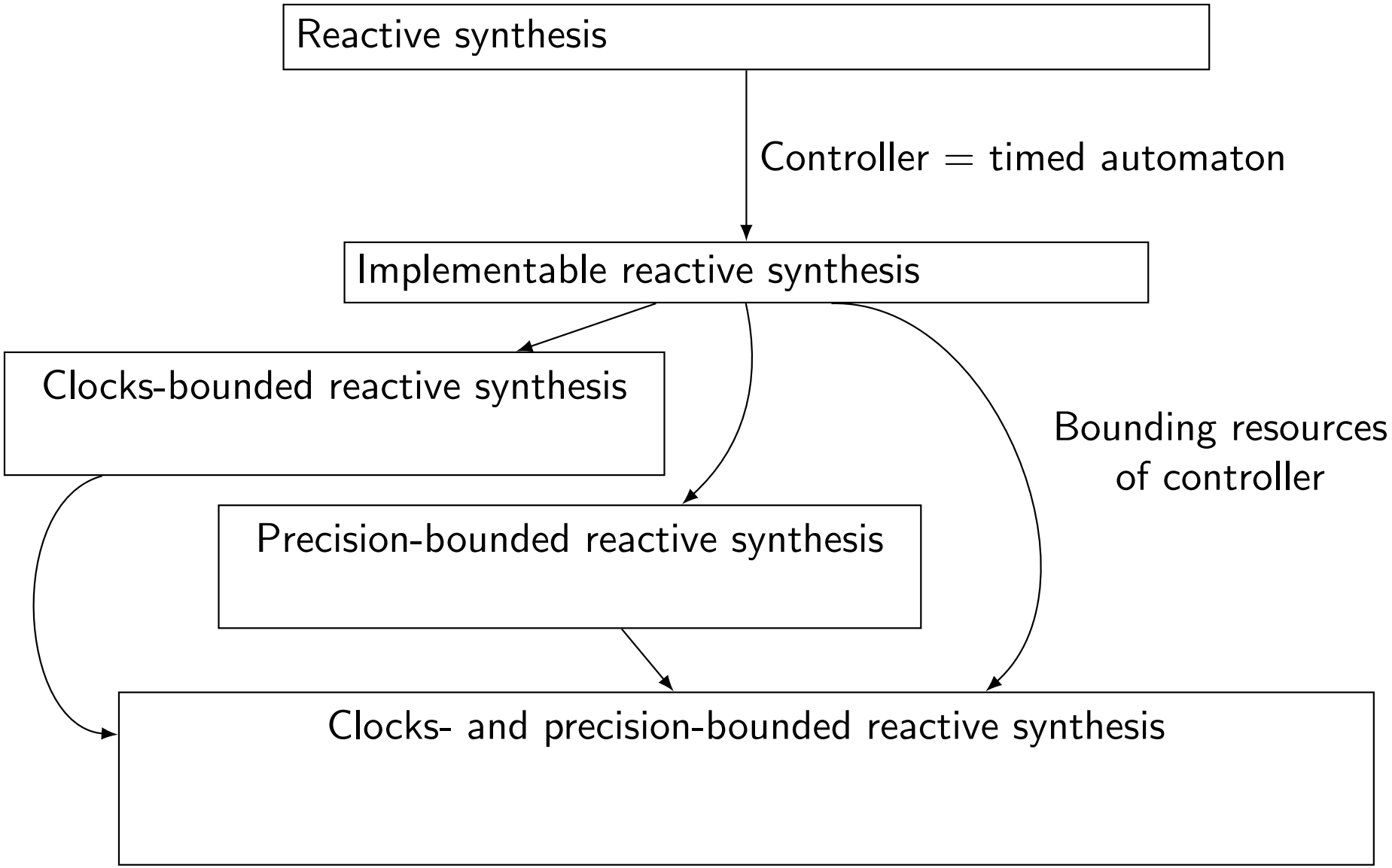
$$\mathit{request} \rightarrow \Diamond_{\leqslant 3} \mathit{grant}$$

MITL = non-punctual fragment of MTL:

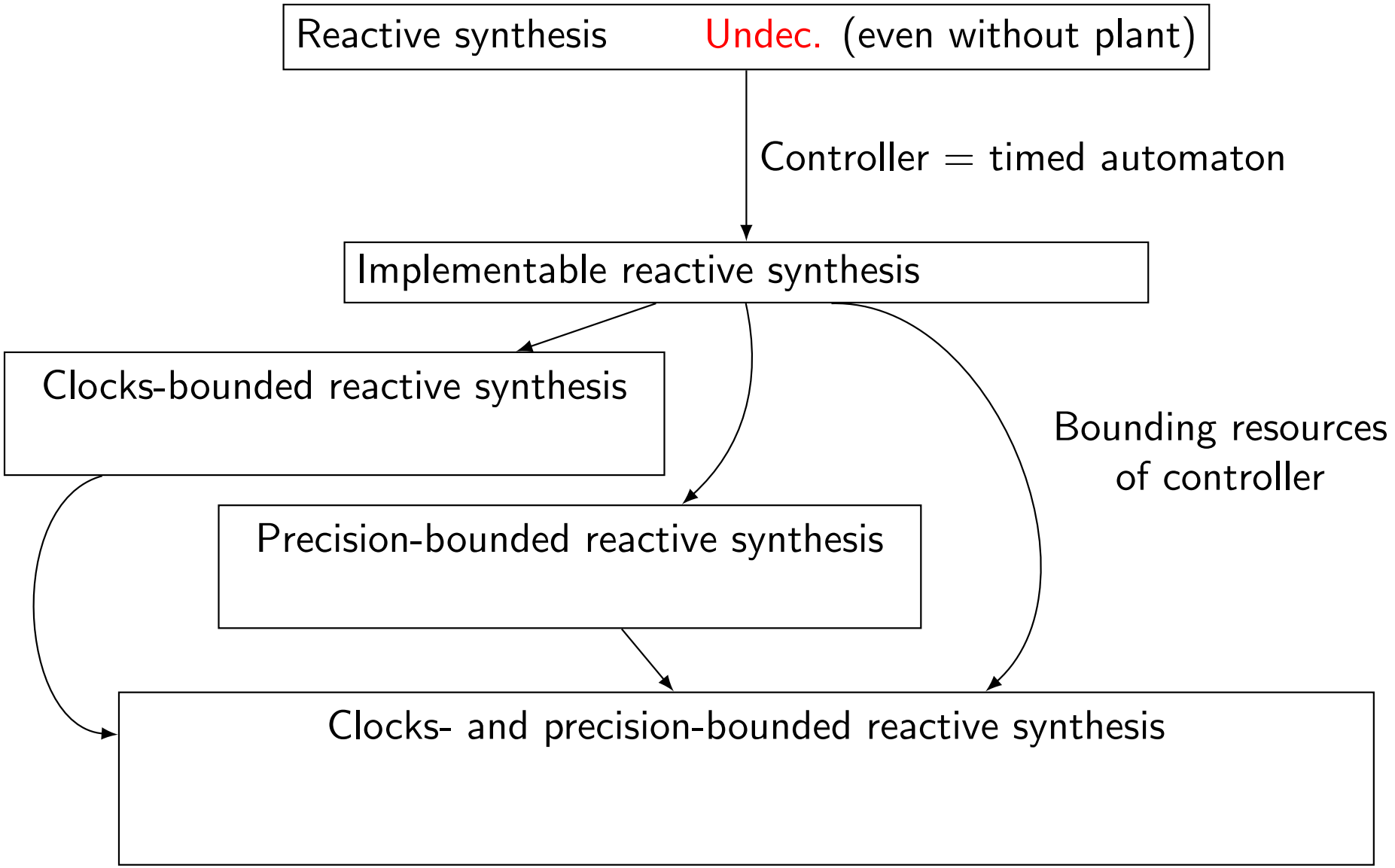
$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathsf{U}_I \, \varphi$$

with $a \in \Sigma$, $I \subseteq [0, \infty)$ is a **non-singular** with bounds in $\mathbb{Q} \cup \{+\infty\}$

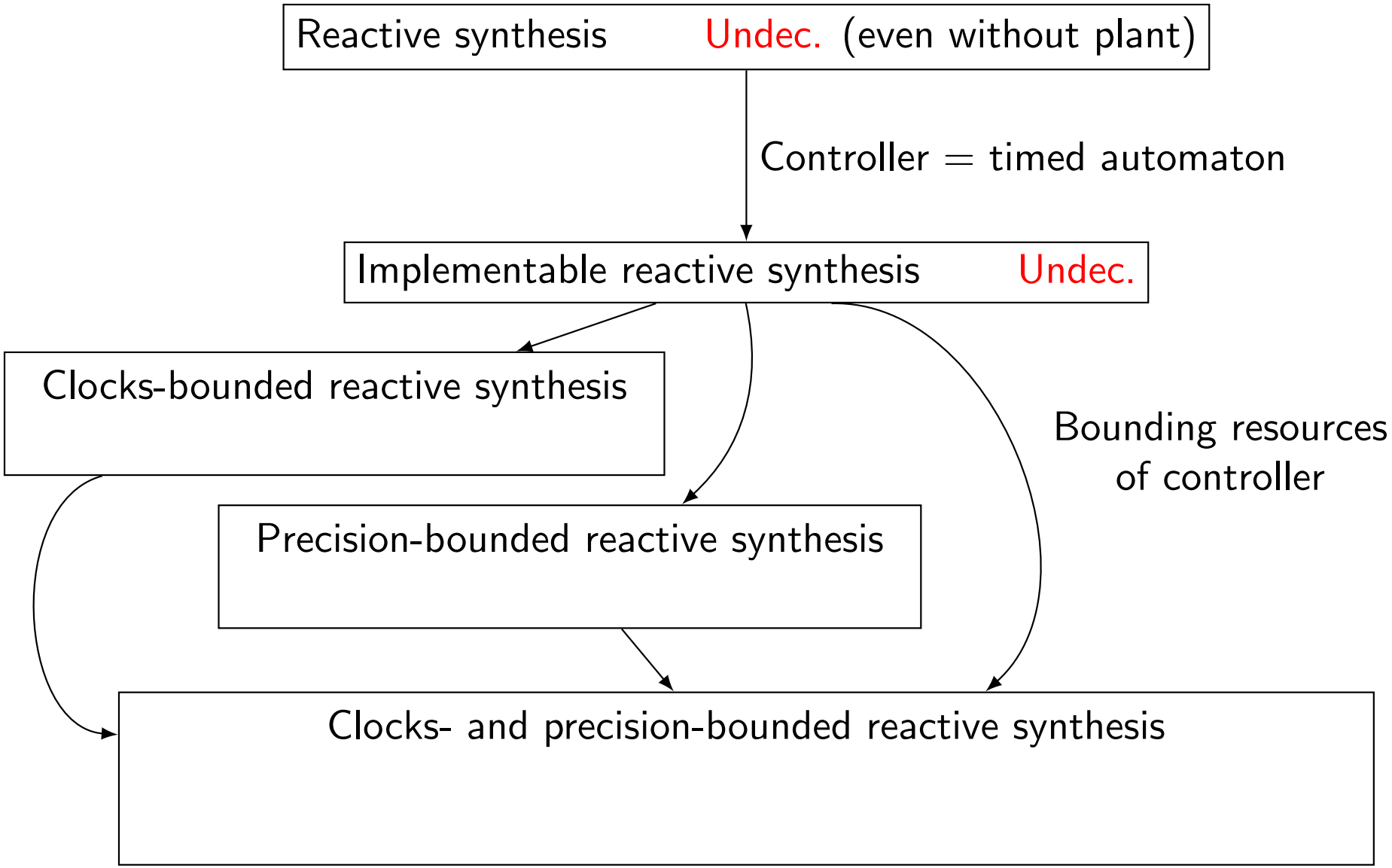
Contribution: reactive synthesis for MITL



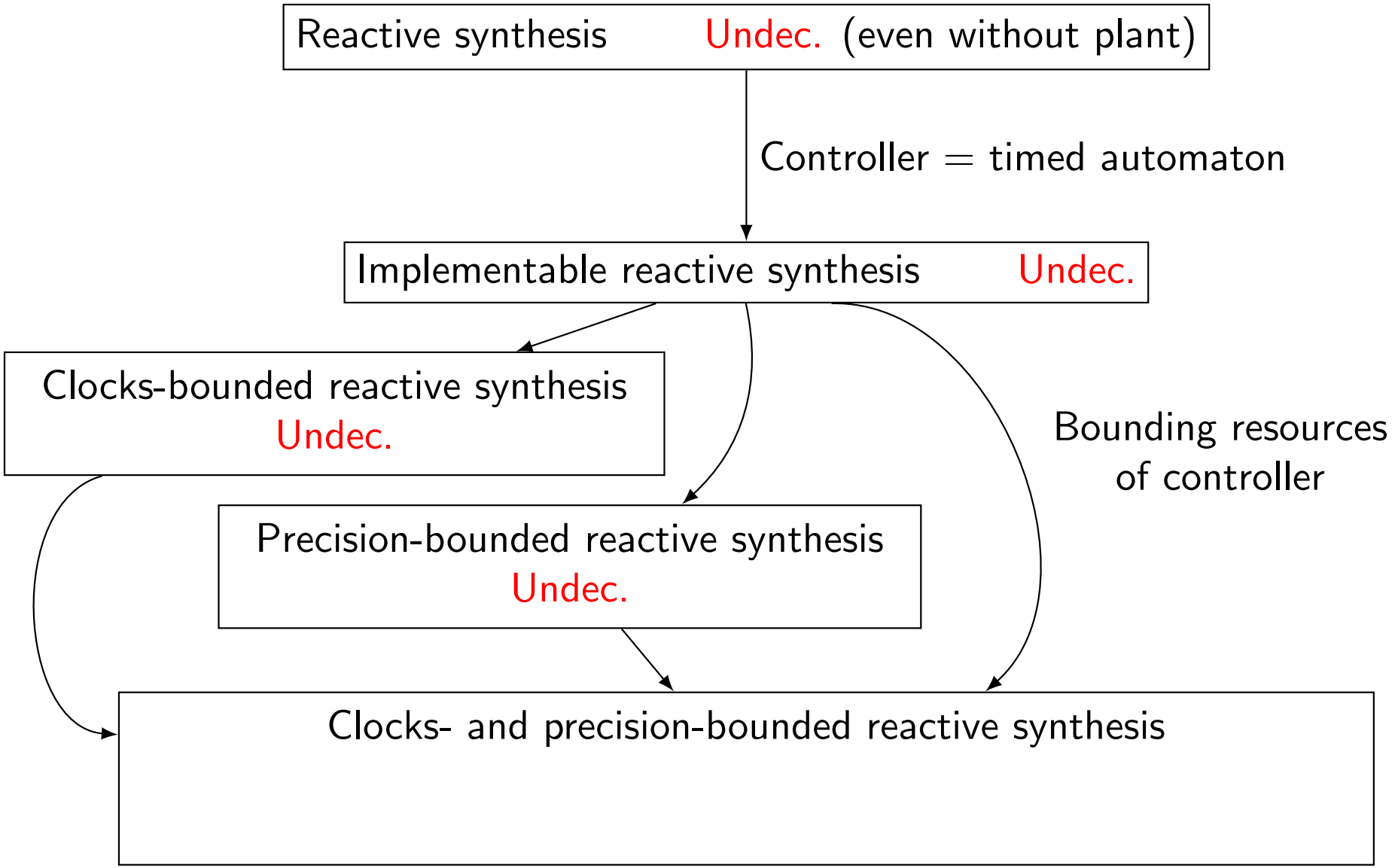
Contribution: reactive synthesis for MITL



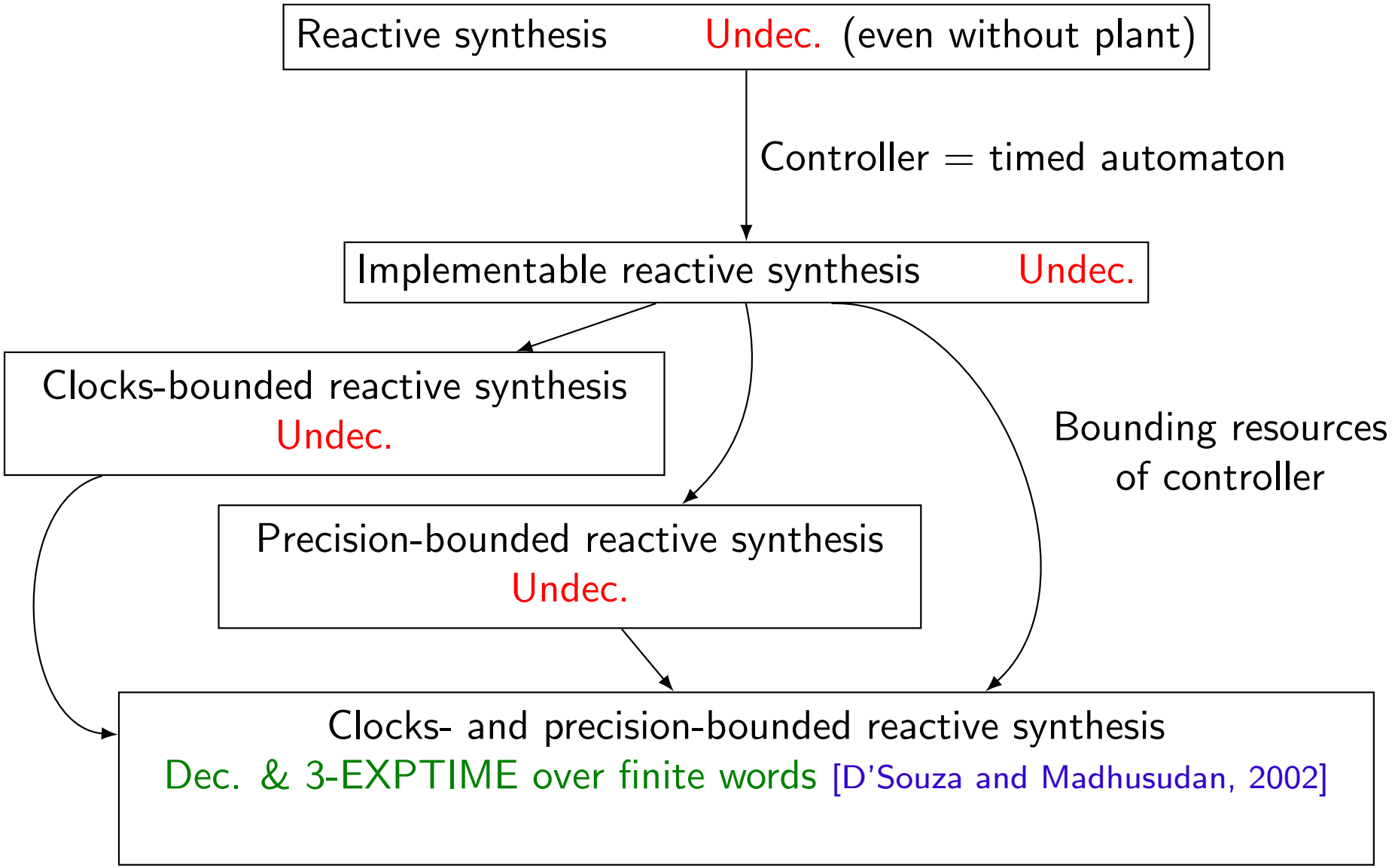
Contribution: reactive synthesis for MITL



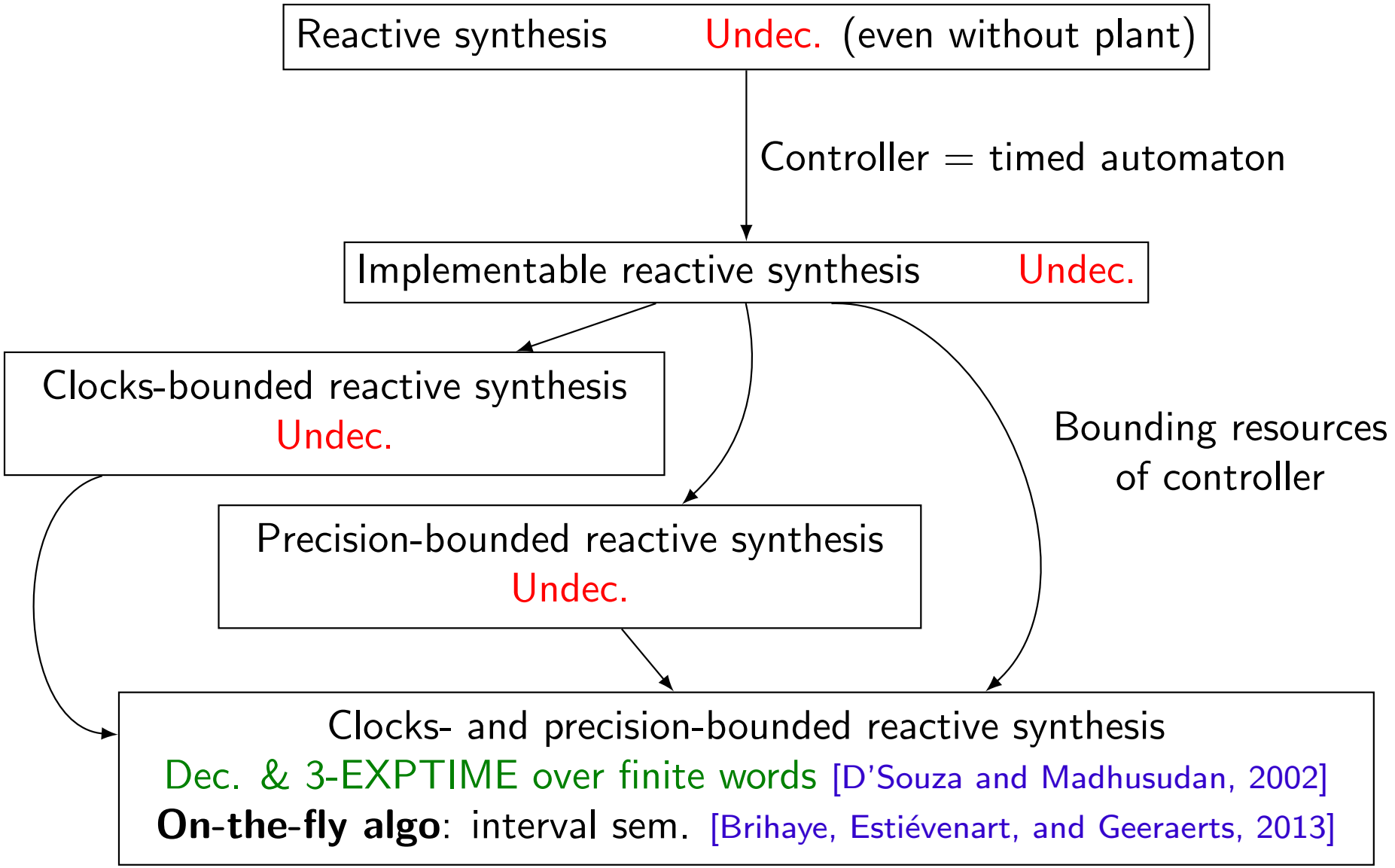
Contribution: reactive synthesis for MITL



Contribution: reactive synthesis for MITL



Contribution: reactive synthesis for MITL



Conclusion

For almost all reactive synthesis problems, MITL is as hard as MTL...

Conclusion

For almost all reactive synthesis problems, MITL is as hard as MTL...

... except for **resources-bounded problem** over finite words:

- Non-elementary for MTL;
- 3-EXPTIME for MITL;
- on-the-fly algorithm

Conclusion

For almost all reactive synthesis problems, MITL is as hard as MTL...

... except for **resources-bounded problem** over finite words:

- Non-elementary for MTL;
- 3-EXPTIME for MITL;
- on-the-fly algorithm

Other fragments?? Hopeless!

	Safety-MTL	coFlat-MTL	Open-MITL	Closed-MITL
implementable RS				
clock-bounded RS				
precision-bounded RS				

Conclusion

For almost all reactive synthesis problems, MITL is as hard as MTL...

... except for **resources-bounded problem** over finite words:

- Non-elementary for MTL;
- 3-EXPTIME for MITL;
- on-the-fly algorithm

Other fragments?? Hopeless!

	Safety-MTL	coFlat-MTL	Open-MITL	Closed-MITL
implementable RS	undec.	undec.	undec.	undec.
clock-bounded RS	undec.	undec.	undec.	undec.
precision-bounded RS	undec.	undec.	undec.	undec.

Conclusion

For almost all reactive synthesis problems, MITL is as hard as MTL...

... except for **resources-bounded problem** over finite words:

- Non-elementary for MTL;
- 3-EXPTIME for MITL;
- on-the-fly algorithm

Other fragments?? Hopeless!

	Safety-MTL	coFlat-MTL	Open-MITL	Closed-MITL
implementable RS	undec.	undec.	undec.	undec.
clock-bounded RS	undec.	undec.	undec.	undec.
precision-bounded RS	undec.	undec.	undec.	undec.

Possible future directions:

- Decidable fragments for BPrecRS/BClockRS
- Heuristics for speed-up for the on-the-fly algorithm
- Experiments of the on-the-fly algorithm over the fragments
- Robustness of controllers

Conclusion

For almost all reactive synthesis problems, MITL is as hard as MTL...

... except for **resources-bounded problem** over finite words:

- Non-elementary for MTL;
- 3-EXPTIME for MITL;
- on-the-fly algorithm

Other fragments?? Hopeless!

	Safety-MTL	coFlat-MTL	Open-MITL	Closed-MITL
implementable RS	undec.	undec.	undec.	undec.
clock-bounded RS	undec.	undec.	undec.	undec.
precision-bounded RS	undec.	undec.	undec.	undec.

Possible future directions:

- Decidable fragments for BPrecRS/BClockRS
- Heuristics for speed-up for the on-the-fly algorithm
- Experiments of the on-the-fly algorithm over the fragments
- Robustness of controllers

Thank you for your attention! Questions?

References

Patricia Bouyer, Laura Bozzelli, and Fabrice Chevalier. Controller synthesis for MTL specifications. In *Proceedings of the 17th International Conference on Concurrency Theory (CONCUR'06)*, volume 4137 of *Lecture Notes in Computer Science*, pages 450–464. Springer, 2006.

Patricia Bouyer, Nicolas Markey, Joël Ouaknine, and James Worrell. The cost of punctuality. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07)*, pages 109–120. IEEE Computer Society, 2007.

Thomas Brihaye, Morgane Estiévenart, and Gilles Geeraerts. On MITL and alternating timed automata. In *Proceedings of the 11th international conference on Formal Modeling and Analysis of Timed Systems (FORMATS'13)*, volume 8053 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2013.

Thomas Brihaye, Morgane Estiévenart, Gilles Geeraerts, Hsi-Ming Ho, Benjamin Monmege, and Nathalie Sznajder. Real-time synthesis is hard! In Martin Fränzle and Nicolas Markey, editors, *Proceedings of the 14th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'16)*, volume 9884 of *Lecture Notes in Computer Science*, pages 105–120. Springer, August 2016. doi: 10.1007/978-3-319-44878-7_7.

Laurent Doyen, Gilles Geeraerts, Jean-François Raskin, and Julien Reichert. Realizability of real-time logics. In *Proceedings of the 7th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'09)*, volume 5813 of *Lecture Notes in Computer Science*, pages 133–148. Springer, 2009.

Deepak D'Souza and P. Madhusudan. Timed control synthesis for external specifications. In *Proceedings of the 19th Annual conference on Theoretical Aspects of Computer Science (STACS'02)*, volume 2285 of *Lecture Notes in Computer Science*, pages 571–582. Springer, 2002.

Joël Ouaknine and James Worrell. Safety metric temporal logic is fully decidable. In *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2006.

Implementable reactive synthesis (IRS)

C = deterministic symbolic transition system $\mathcal{T}$

- set of locations; if finite $\rightarrow \mathcal{T}$ is a DTA
- finite set of clocks X
- finite set of possible clock constraints *precision* (m, K):

$$g ::= \top \mid g \wedge g \mid x < \alpha/m \mid x \leqslant \alpha/m \mid x = \alpha/m \mid x \geqslant \alpha/m \mid x > \alpha/m$$

with $x \in X$, $m \in \mathbb{N}$ and $0 \leqslant \alpha \leqslant K$.

Implementable reactive synthesis (IRS)

C = deterministic symbolic transition system $\mathcal{T}$

- set of locations; if finite $\rightarrow \mathcal{T}$ is a DTA
- finite set of clocks X
- finite set of possible clock constraints *precision* (m, K):

$$g ::= \top \mid g \wedge g \mid x < \alpha/m \mid x \leqslant \alpha/m \mid x = \alpha/m \mid x \geqslant \alpha/m \mid x > \alpha/m$$

with $x \in X$, $m \in \mathbb{N}$ and $0 \leqslant \alpha \leqslant K$.

Implementable reactive synthesis problem (IRS)

Given $\mathcal{P}$ and $\mathcal{L}$, find such a $\mathcal{T}$ that no matter what E does, every play satisfies the specification.

Recovering decidability...

Clock constraints in $\mathcal{T}$:

$$g ::= \top \mid g \wedge g \mid x < \alpha/m \mid x \leqslant \alpha/m \mid x = \alpha/m \mid x \geqslant \alpha/m \mid x > \alpha/m$$

with $x \in X$, $m \in \mathbb{N}$ and $0 \leqslant \alpha \leqslant K$.

Recovering decidability...

Clock constraints in $\mathcal{T}$:

$$g ::= \top \mid g \wedge g \mid x < \alpha/m \mid x \leqslant \alpha/m \mid x = \alpha/m \mid x \geqslant \alpha/m \mid x > \alpha/m$$

with $x \in X$, $m \in \mathbb{N}$ and $0 \leqslant \alpha \leqslant K$.

Fix X and $(m, K) \Longrightarrow$ the alphabet of $\mathcal{T}$ is given!

Recovering decidability...

Clock constraints in $\mathcal{T}$:

$$g ::= \top \mid g \wedge g \mid x < \alpha/m \mid x \leqslant \alpha/m \mid x = \alpha/m \mid x \geqslant \alpha/m \mid x > \alpha/m$$

with $x \in X$, $m \in \mathbb{N}$ and $0 \leqslant \alpha \leqslant K$.

Fix X and $(m, K) \Longrightarrow$ the alphabet of $\mathcal{T}$ is given!

Bounded-resources reactive synthesis problem (BResRS)

Given $\mathcal{P}$, $\mathcal{L}$, and **a set of clocks X and precision (m, K)**, *find such a resource-bounded $\mathcal{T}$ that no matter what E does, every play satisfies the specification.*

Other fragments of MTL

IRS/BPrecRS/BClockRS for MITL are undecidable.
What about other fragments of MTL?

Theorem: [Ouaknine and Worrell, 2006]

Satisfiability and model-checking for Safety-MTL are decidable.

Theorem: [Bouyer, Markey, Ouaknine, and Worrell, 2007]

Model-checking for coFlat-MTL is decidable.

However with some more efforts, we can rewrite the formula in these fragments.

IRS/BPrecRS/BClockRS over infinite timed words:

	Safety-MTL	coFlat-MTL
desired	undec.	**undec.**
undesired	**undec.**	undec.

On the decidable side

Theorem: [D'Souza and Madhusudan, 2002]

BResRS is decidable for **undesired** specifications given in TA.

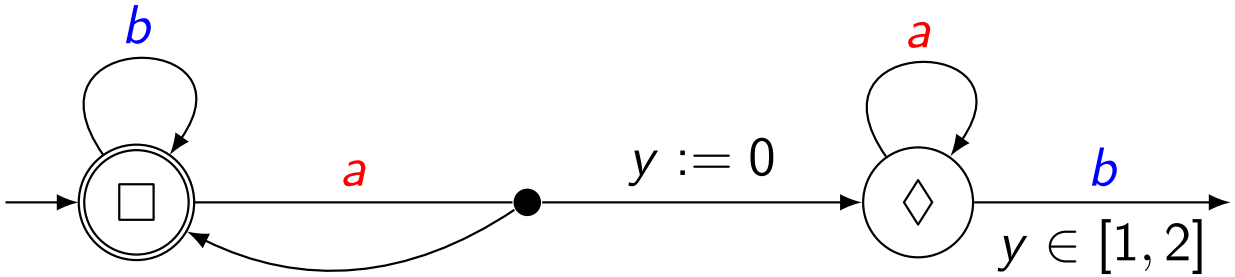
Theorem: [Bouyer, Bozzelli, and Chevalier, 2006]

BResRS is decidable for MTL specifications with non-primitive recursive complexity (over finite timed words).

We propose an **on-the-fly** algorithm that solves BResRS for MITL specifications with 3-EXPTIME complexity (over finite words).

From MTL to OCATA

$$\varphi = \Box(a \Rightarrow \Diamond_{[1,2]}\, b)$$



Execution on the timed word $(a, 0.5)(a, 0.6)(a, 1.2)(b, 2.3)$: