SYNTHESIS OF TIMELINE-BASED PLANNING STRATEGIES AVOIDING DETERMINIZATION

Dario Della Monica
University of Udine, Italy

GandALF 2024
June 20, 2024
Reykjavik, Iceland

joint work with R. Acampora, L. Geatti, N. Gigante, A. Montanari, and P. Sala

TIMELINE-BASED PLANNING

Timeline-based planning is an approach to planning mostly focused on temporal reasoning:

- no clear separation between actions, states, and goals;
- planning problems are modeled as systems made of a number of independent, but interacting, components;
- components are described by **state variables**;
- the **timelines** describe their evolution over time;
- the evolution of the system is governed by a set of temporal constraints called **synchronization rules**.

WHY TIMELINES?

Timeline-based planning systems shine when:

- modeling and reasoning about systems made of many **components**, rather than the behavior of a single agent
- approaching problems where **temporal reasoning** is key

TIMELINES AND SPACE EXPLORATION

Timeline-based planning was born in the space operations field, and has been used in real-world mission planning and scheduling systems, both on-board and on-ground.

N. Muscettola. *HSTS: Integrating Planning and Scheduling*. Intelligent Scheduling 1994

J. Barreiro et al. *EUROPA: A Platform for AI Planning, Scheduling, Constraint Programming, and Optimization*. ICKEPS 2012

S. Chien et al. ASPEN – Automated Planning and Scheduling for Space Mission Operations. 2000

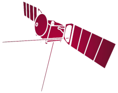
European Space Agency

A. Cesta, G. Cortellessa, S. Fratini, and A. Oddi. *Developing an End-to-End Planning Application from a Timeline Representation Framework*. IAAI 2009

S. Fratini, A. Cesta, R. De Benedictis, A. Orlandini, and R. Rasconi. *APSI-Based Deliberation in Goal Oriented Autonomous Controllers*. ASTRA 2011

DOMAIN EXAMPLE

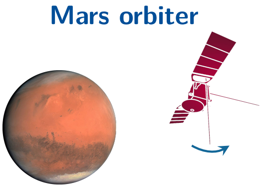
Mars orbiter



Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": **Mars**, Slewing, Earth
2. Four "activities": Science, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

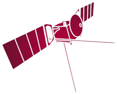


Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, **Slewing**, Earth
2. Four "activities": Science, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

Mars orbiter

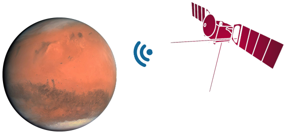


Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, Slewing, **Earth**
2. Four "activities": Science, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

Mars orbiter

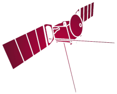


Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, Slewing, Earth
2. Four "activities": **Science**, Communication, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

Mars orbiter

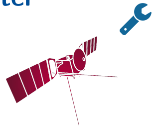




Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, Slewing, Earth
2. Four "activities": Science, **Communication**, Maintenance, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

Mars orbiter

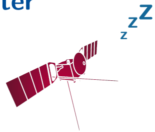


Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, Slewing, Earth
2. Four "activities": Science, Communication, **Maintenance**, Idle
3. Temporal constraints:
   - Scientific measurements can be done only when pointing to Mars
   - Communication can happen:
     - only when pointing to Earth
     - only when a receiving ground station is visible
4. Goals:
   - Perform at least a given number of scientific measurements

DOMAIN EXAMPLE

Mars orbiter

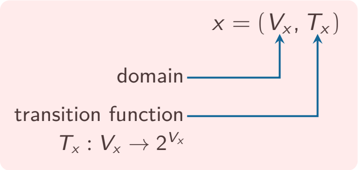


Toy example of a Mars orbiter doing scientific measurements:

1. Three "pointing modes": Mars, Slewing, Earth
2. Four "activities": Science, Communication, Maintenance, **Idle**
3. Temporal constraints:
    - Scientific measurements can be done only when pointing to Mars
    - Communication can happen:
        - only when pointing to Earth
        - only when a receiving ground station is visible
4. Goals:
    - Perform at least a given number of scientific measurements

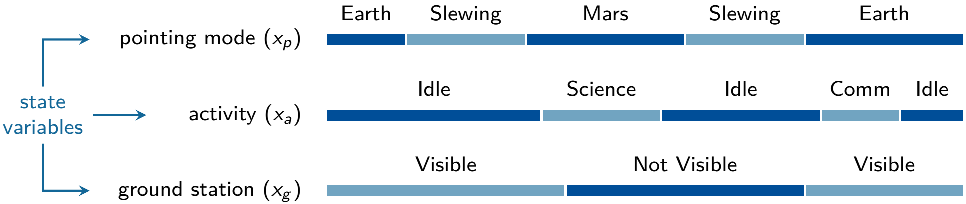
STATE VARIABLES AND TIMELINES

State variables represent the components of the system:

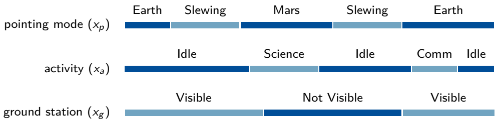


Timelines encode possible evolutions of state variables

- sequences of **tokens**, i.e., time intervals where the variable holds a single value
- token transitions respect T_x
- **tokens have a duration** (positive integer) – **time is discrete**



SYNCHRONISATION RULES



The interaction of the components is governed by the synchronization rules.

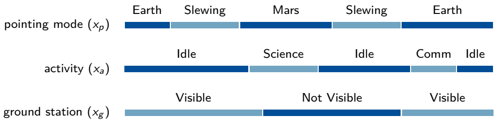
Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \rightarrow \exists b[x_p = Mars] \,.\, start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

for all tokens a **where** $x_a = Science$,
there is a token b **where** $x_p = Mars$,
such that b **contains** a.

SYNCHRONISATION RULES



The interaction of the components is governed by the synchronization rules.

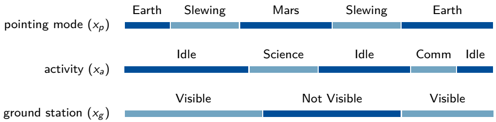
Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \rightarrow \exists b[x_p = Mars] \,.\, start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

for all tokens a **where** $x_a = Science$,
there is a token b **where** $x_p = Mars$,
such that b **contains** a.

SYNCHRONISATION RULES



The interaction of the components is governed by the synchronization rules.

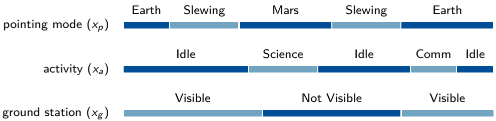
Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \rightarrow \exists b[x_p = Mars] \,.\, start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

for all tokens a **where** $x_a = Science$,
there is a token b **where** $x_p = Mars$,
such that b **contains** a.

SYNCHRONISATION RULES



The interaction of the components is governed by the synchronization rules.

Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \rightarrow \exists b[x_p = Mars] \, . \, start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

for all tokens a **where** $x_a = Science$,
there is a token b **where** $x_p = Mars$,
such that b **contains** a.

SYNTAX – SYNCHRONIZATION RULES

Each rule has a fixed structure:

$$a[x = u] \quad \rightarrow \quad \exists b[y = v] \;.\; \langle body \rangle$$

$$\bigvee$$

$$\exists c[z = w] d[k = r] \;.\; \langle body \rangle$$

$$\bigvee$$

$$\ldots$$

optional trigger

existential statements

Example

Scientific measurements can be done only when pointing to Mars:

$$a[x_a = Science] \rightarrow \exists b[x_p = Mars] \;.\; start(b) \leqslant start(a) \wedge end(a) \leqslant end(b)$$

SYNTAX – ATOMS

The body is made of a **conjunction** of **atoms** (temporal relations):

$$start(a) \leqslant end(b)$$

endpoint → *start*

token name → *a*

SYNTAX – ATOMS

The body is made of a **conjunction** of **atoms** (temporal relations):

$$start(a) \leqslant end(b)$$

endpoint

token name

Quantitative atoms allow to bind min and max distance between start/end of tokens

$$start(a) \leqslant_{[l,u]} end(b)$$

lower bound
$l \in \mathbb{N}$

upper bound
$u \in \mathbb{N} \cup \{+\infty\}$

The body is made of a **conjunction** of **atoms** (temporal relations):

$$start(a) \leqslant end(b)$$

endpoint → $start$

token name → a

we focus on qualitative atoms

Quantitative atoms allow to bind min and max distance between start/end of tokens

$$start(a) \leqslant_{[l,u]} end(b)$$

lower bound $l \in \mathbb{N}$

upper bound $u \in \mathbb{N} \cup \{+\infty\}$

QUANTITATIVE TIMELINE-BASED PLANNING

Quantitative timeline-based planning problem ***(quantitative atoms allowed)***

Given a set of state variables SV and a set of synchronization rules S,
is there a set of timelines for SV satisfying S?

Theorem [Gigante, Montanari, Cialdea Mayer, Orlandini, ICAPS 2017]

Quantitative timeline-based planning is EXPSPACE-**complete** *(quantitative atoms **allowed**)*

QUANTITATIVE TIMELINE-BASED PLANNING

Quantitative timeline-based planning problem ***(quantitative atoms allowed)***

Given a set of state variables SV and a set of synchronization rules S,
is there a set of timelines for SV satisfying S?

Theorem [Gigante, Montanari, Cialdea Mayer, Orlandini, ICAPS 2017]

Quantitative timeline-based planning is EXPSPACE-**complete** *(quantitative atoms* ***allowed****)*

Automata-based procedure in EXPSPACE [Della Monica, Gigante, Montanari, Sala, KR 2018]

QUALITATIVE TIMELINE-BASED PLANNING

we focus on
qualitative atoms

Theorem [Della Monica, Gigante, La Torre, Montanari, TIME 2020]

Qualitative timeline-based planning is PSPACE-**complete** *(quantitative atoms **not allowed**)*

- automata-based procedure

AUTOMATA-BASED PROCEDURE

The problem is encoded into an **automaton**:

- the encoding produces a **nondeterministic** finite automaton
- the NFA accepts (a word representing) a plan iff the plan is a solution for the problem
- the size of the automaton is **exponential** in the size of the problem
- solving the reachability problem **on-the-fly** gives us the **PSPACE** upper bound

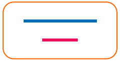
NFA ENCODING PLAN EXISTENCE

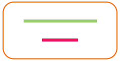
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

NFA ENCODING PLAN EXISTENCE

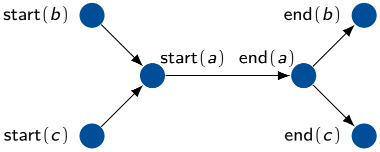
for all tokens a there are tokens b and c such that b contains a and c contains a

NFA ENCODING PLAN EXISTENCE



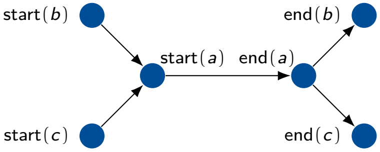


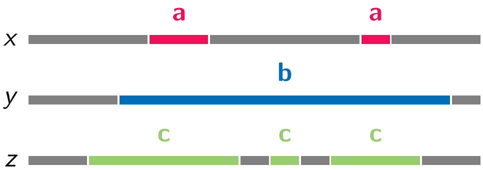
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**



NFA ENCODING PLAN EXISTENCE

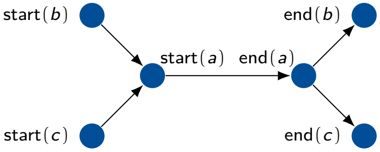
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

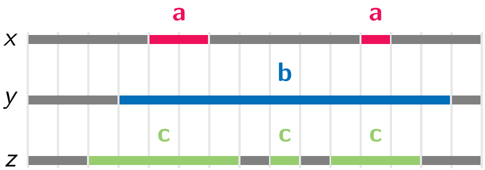




NFA ENCODING PLAN EXISTENCE

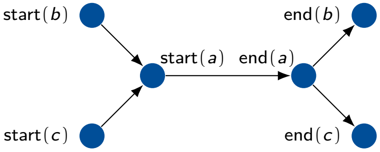
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

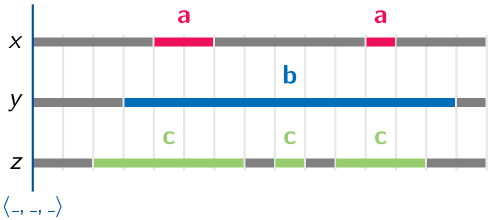




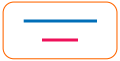
NFA ENCODING PLAN EXISTENCE

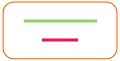
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**



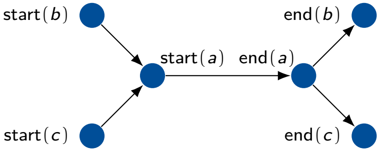


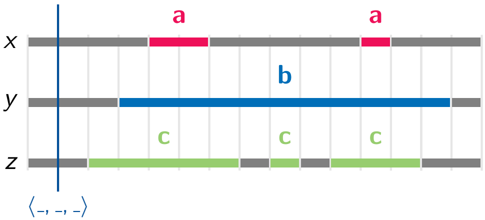
NFA ENCODING PLAN EXISTENCE





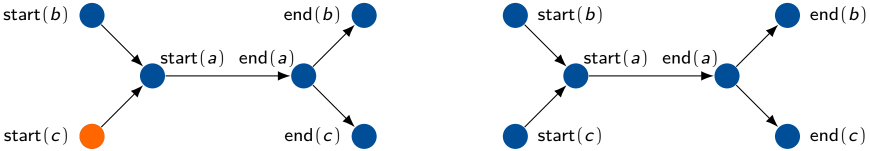
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

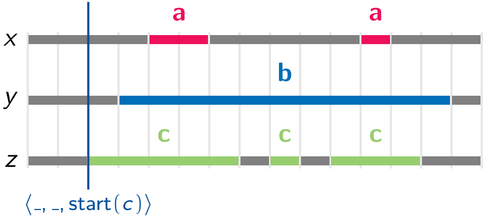




NFA ENCODING PLAN EXISTENCE

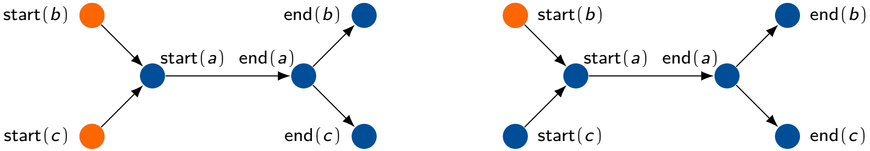
for all tokens a there are tokens b and c such that b contains a and c contains a

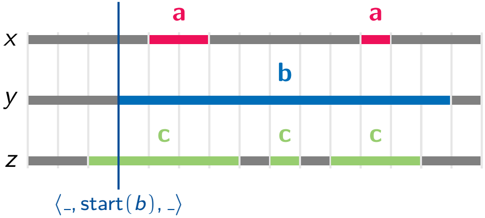




NFA ENCODING PLAN EXISTENCE

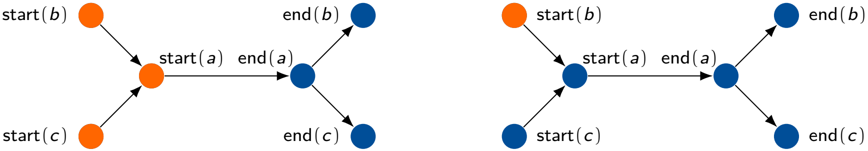
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

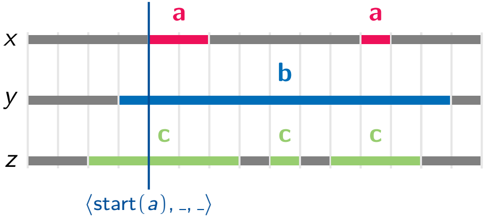




NFA ENCODING PLAN EXISTENCE

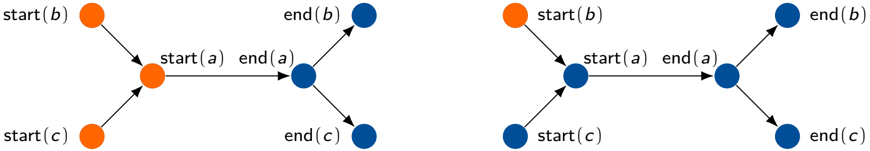
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

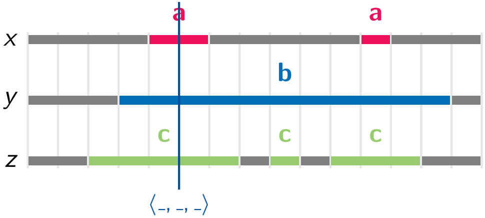




NFA ENCODING PLAN EXISTENCE

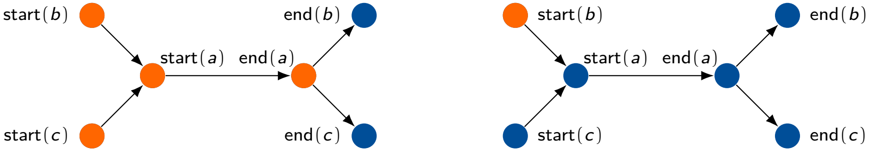
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

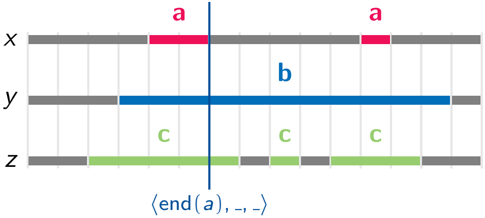




NFA ENCODING PLAN EXISTENCE

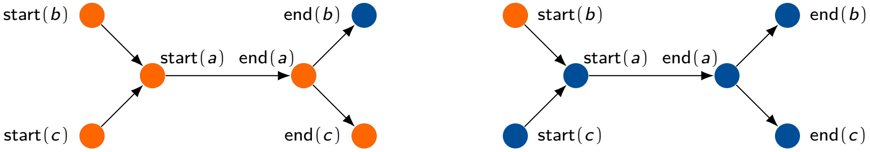
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

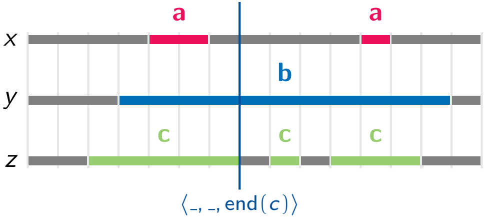




NFA ENCODING PLAN EXISTENCE

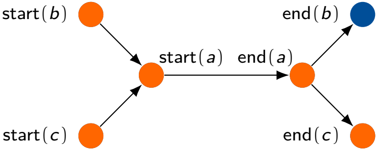
for all tokens a there are tokens b and c such that b contains a and c contains a

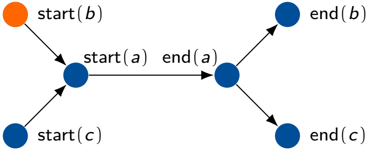


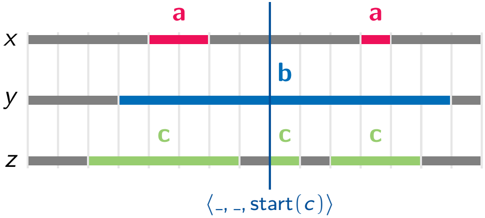


NFA ENCODING PLAN EXISTENCE

for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**

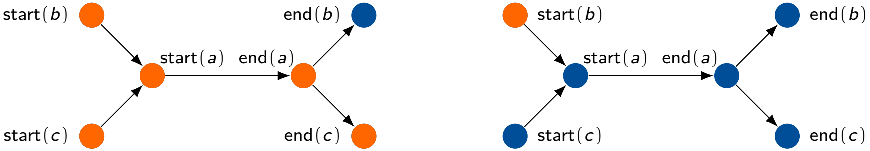


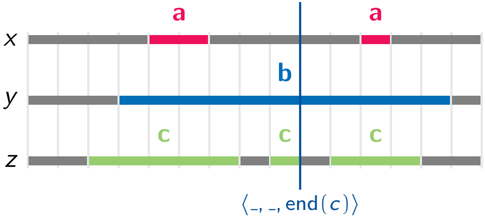




NFA ENCODING PLAN EXISTENCE

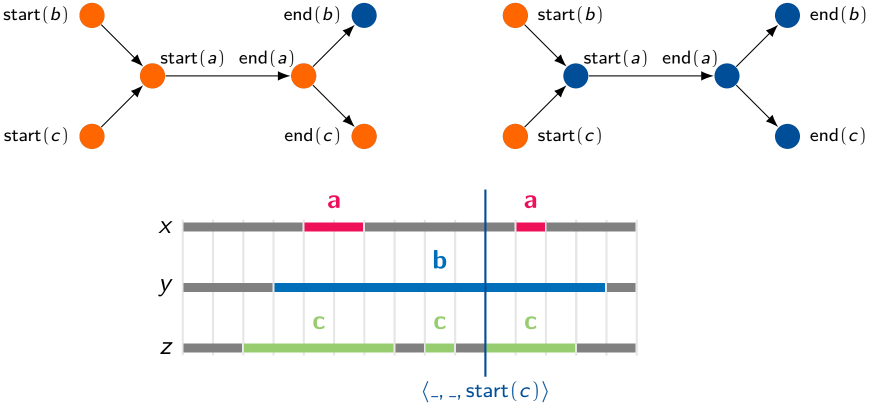
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**





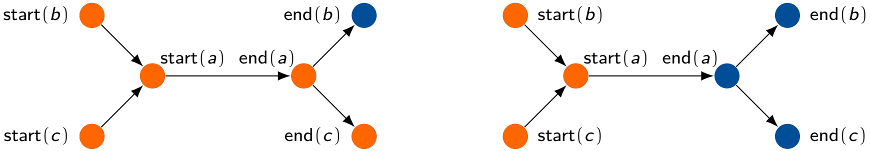
NFA ENCODING PLAN EXISTENCE

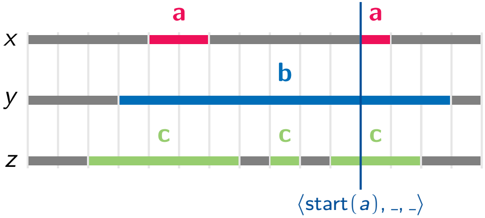
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**



NFA ENCODING PLAN EXISTENCE

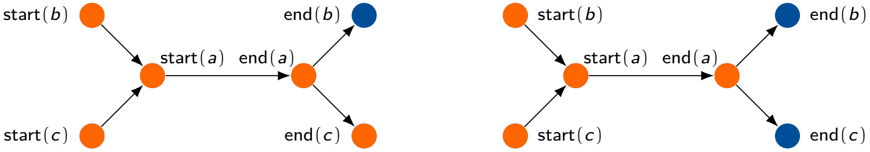
for all tokens a there are tokens b and c such that b contains a and c contains a

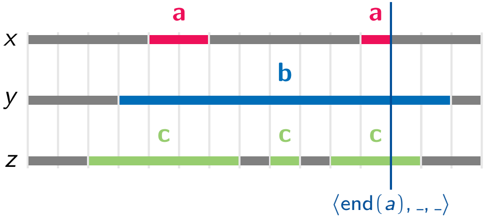




NFA ENCODING PLAN EXISTENCE

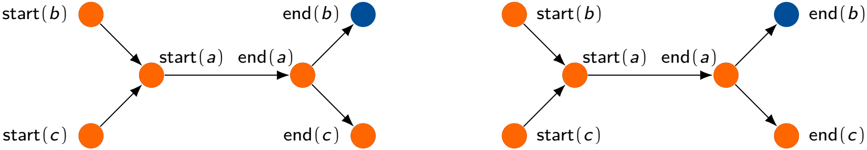
for all tokens a there are tokens b and c such that b contains a and c contains a

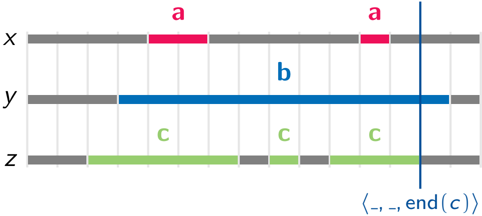




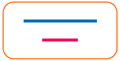
NFA ENCODING PLAN EXISTENCE

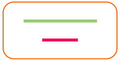
for all tokens **a** there are tokens **b** and **c** such that **b** contains **a** and **c** contains **a**



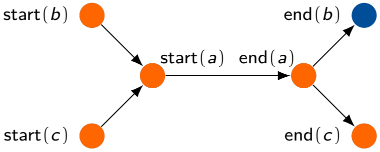


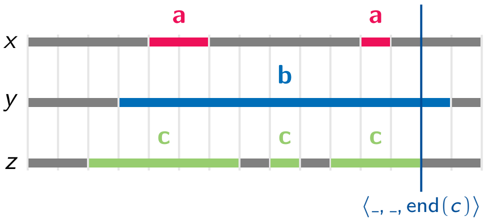
NFA ENCODING PLAN EXISTENCE



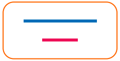


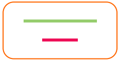
for all tokens a there are tokens b and c such that b contains a and c contains a



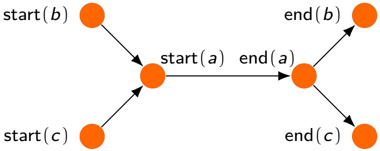


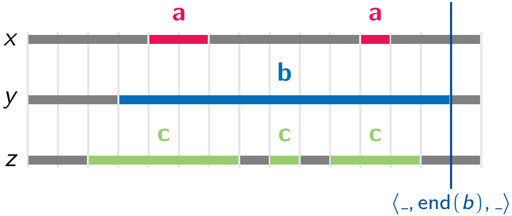
NFA ENCODING PLAN EXISTENCE





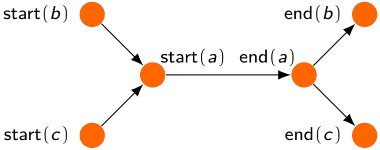
for all tokens a there are tokens b and c such that b contains a and c contains a

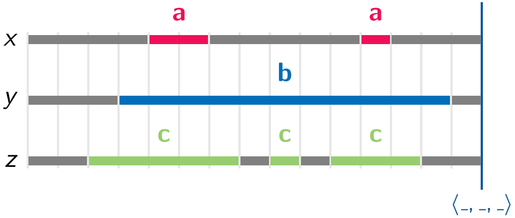




NFA ENCODING PLAN EXISTENCE

for all tokens a there are tokens b and c such that b contains a and c contains a





PLAN SYNTHESIS

To **synthesize** a plan (possibly against an adversarial environment) one needs **determinism**

- DFA provides **strategies** (functions)
- determinization costs an exponential in space
    - a 2-EXPTIME trivial solution to the synthesis problem

PLAN SYNTHESIS

To **synthesize** a plan (possibly against an adversarial environment) one needs **determinism**

- DFA provides **strategies** (functions)
- determinization costs an exponential in space
  - a 2-EXPTIME trivial solution to the synthesis problem
- **alternative**: direct encoding of plan existence into DFA
  - **eager rules for determinism**

A SYNTACTIC RESTRICTION: EAGER RULES

Definition (eager rules for trigger tokens)

A. Token a is a trigger token, token b is not

1. atoms $\mathsf{start}(b) \leqslant \mathsf{start}(a)$ and $\mathsf{start}(a) \leqslant \mathsf{end}(b)$ imply atom $\mathsf{start}(b) = \mathsf{start}(a)$
   - if a is required to start during b, then *eager rules* also require that a and b start together
2. atoms $\mathsf{start}(b) \leqslant \mathsf{end}(a)$ and $\mathsf{end}(a) \leqslant \mathsf{end}(b)$ imply atom $\mathsf{start}(b) = \mathsf{start}(a)$
   or atom $\mathsf{start}(b) = \mathsf{end}(a)$
   - if a is required to end during b, then *eager rules* also require that the start of b coincides with the start or the end of a
3. atoms $\mathsf{start}(a) \leqslant \mathsf{start}(b)$ and $\mathsf{end}(a) \leqslant \mathsf{end}(b)$ imlpy atom $\mathsf{start}(b) = \mathsf{start}(a)$
   or atom $\mathsf{start}(b) = \mathsf{end}(a)$
   - if a is required to start/end not later than b, then *eager rules* also require that b coincides with the start or the end of a

A SYNTACTIC RESTRICTION: EAGER RULES – CONT'D

Definition (eager rules for non-trigger tokens)

B Neither a nor b is a trigger token

1. atoms $\mathsf{start}(b) \leqslant \mathsf{start}(a)$ and $\mathsf{start}(a) \leqslant \mathsf{end}(b)$ imply atom $\mathsf{start}(b) = \mathsf{start}(a)$
   - if a is required to start during b, then *eager rules* also require that a and b start together
2. atoms $\mathsf{start}(b) \leqslant \mathsf{end}(a)$ and $\mathsf{end}(a) \leqslant \mathsf{end}(b)$ imply atom $\mathsf{start}(b) = \mathsf{end}(a)$
   - if a is required to end during b, then *eager rules* also require that the start of b coincides with the end of a

ALLEN RELATIONS IN THE EAGER FRAGMENT

graphical representation	Allen's relation	*a* is trigger	neither *a* nor *b* is trigger
	*a before b*	✓	✓
	a after b	✓	✓
	*a meets b*	✓	✓
	a met-by b	✓	✓
	*a ends b*	✗	✗
	a ended-by b	✗	✗
	*a begins b*	✓	✗
	a begun-by b	✓	✗
	*a overlaps b*	✗	✗
	a overlapped-by b	✗	✗
	*a during b*	✗	✗
	a contains b	✓	✗
	*a equals b*	✓	✗

CONCLUSIONS

- A direct DFA encoding of the plan existence problem for the **eager** fragment of qualitative timeline-based planning
- An EXPTIME procedure for the **synthesis** problem for the eager fragment of qualitative timeline-based planning

Future challenges

- A direct DFA construction to deal with the **whole qualitative** timeline-based planning (if any)
- Study of the **quantitative** version of timeline-based planning

Thank you

Questions?