The complexity of soundness in workflow nets

Philip Offtermatt

Joint work with
Michael Blondin and Filip Mazowiecki

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

UNIVERSITAS VARSOVIENSIS

UNIVERSITÉ DE SHERBROOKE

Powered by BeamerikZ

The complexity of soundness in workflow nets

Philip Offtermatt

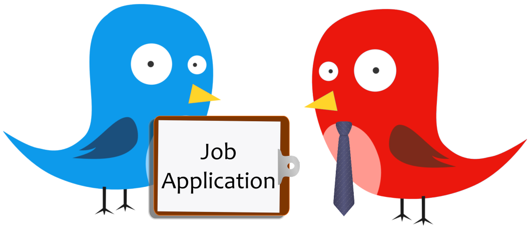
Joint work with
Michael Blondin and Filip Mazowiecki

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

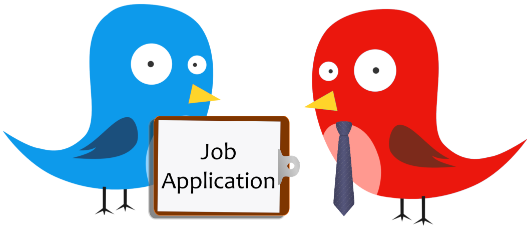
UNIVERSITAS VARSOVIENSIS

UNIVERSITÉ DE SHERBROOKE

Processes are everywhere!

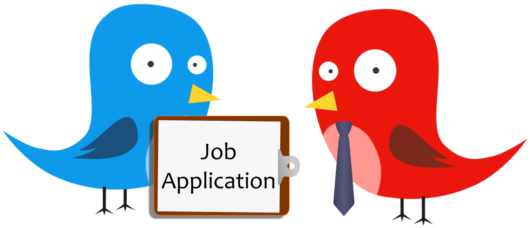


Processes are everywhere!



- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/ Recheck application

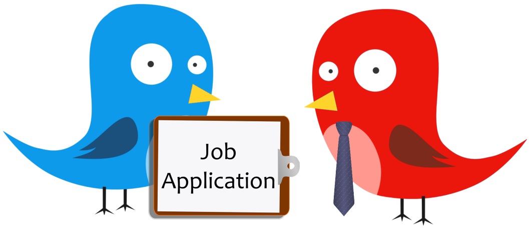
Processes are everywhere!



- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application

How many applicants will we need until we find a new hire?

Processes are everywhere!

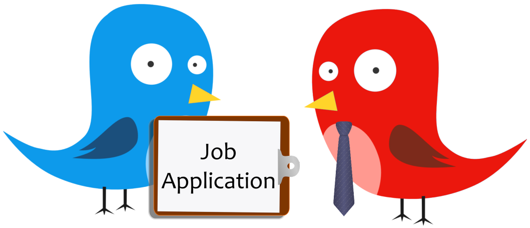


- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application

How many applicants will we need until we find a new hire?

Can we handle applications faster?

Processes are everywhere!



- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/ Recheck application

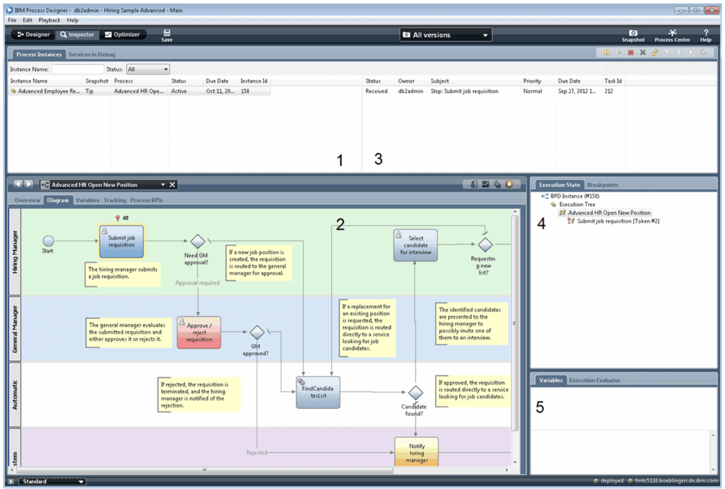
How many applicants will we need until we find a new hire?

Can we handle applications faster?

Will every applicant hear back from us?

Processes are everywhere!

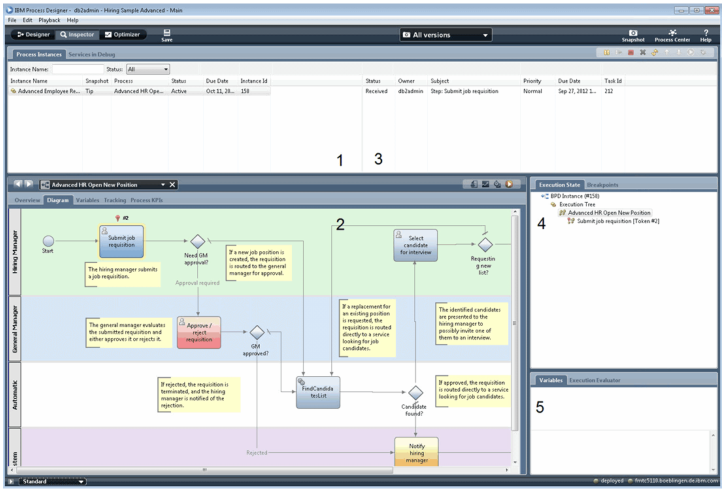
Modelled by humans...



Processes are everywhere!

Modelled by humans...

...or mined from logs

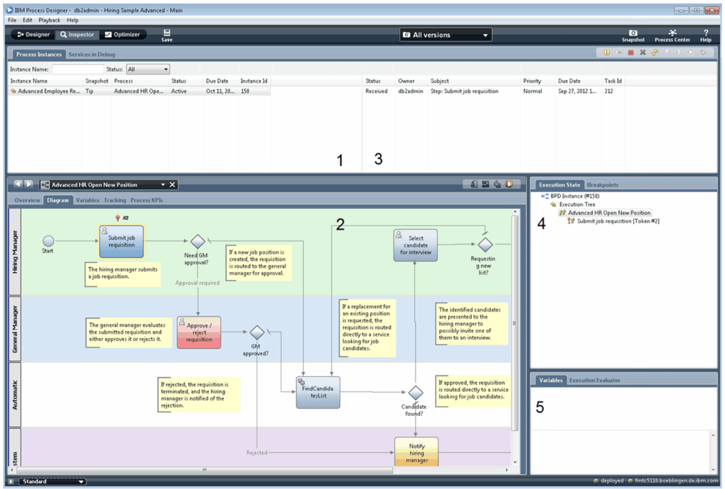


Case ID	Task Name	Resource	Date	Time
1	Receive Application	Peter	04/12/2020	06:37:11
1	Check legal requirements	Anne	05/12/2020	19:21:54
2	Receive Application	Peter	05/12/2020	02:04:19
3	Receive Application	Peter	06/12/2020	11:27:20
1	Check applicant suitability	Eva	06/12/2020	11:25:53
4	Receive Application	Peter	06/12/2020	14:18:20
5	Receive Application	Peter	07/12/2020	12:54:57
2	Check applicant suitability	Eva	08/12/2020	17:20:30
1	Accept	Eva	08/12/2020	06:45:23
3	Check legal requirements	Anne	08/12/2020	06:36:26
4	Check applicant suitability	Eva	16/12/2020	00:21:57
2	Check legal requirements	Anne	16/12/2020	09:03:05
2	Recheck Application	Chris	18/12/2020	19:44:24
2	Check legal requirements	Anne	19/12/2020	20:26:55
4	Check legal requirements	Anne	19/12/2020	17:38:49
4	Reject	Chris	20/12/2020	09:37:59
3	Check applicant suitability	Anne	20/12/2020	01:32:44
2	Check applicant suitability	Peter	27/12/2020	03:35:57
3	Accept	Eva	29/12/2020	17:18:55
2	Reject	Peter	29/12/2020	03:48:06
5	Check legal requirements	Anne	29/12/2020	03:37:39

Processes are everywhere!

Modelled by humans...

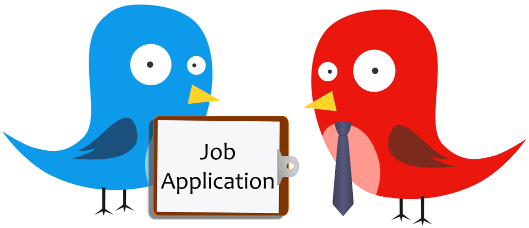
...or mined from logs



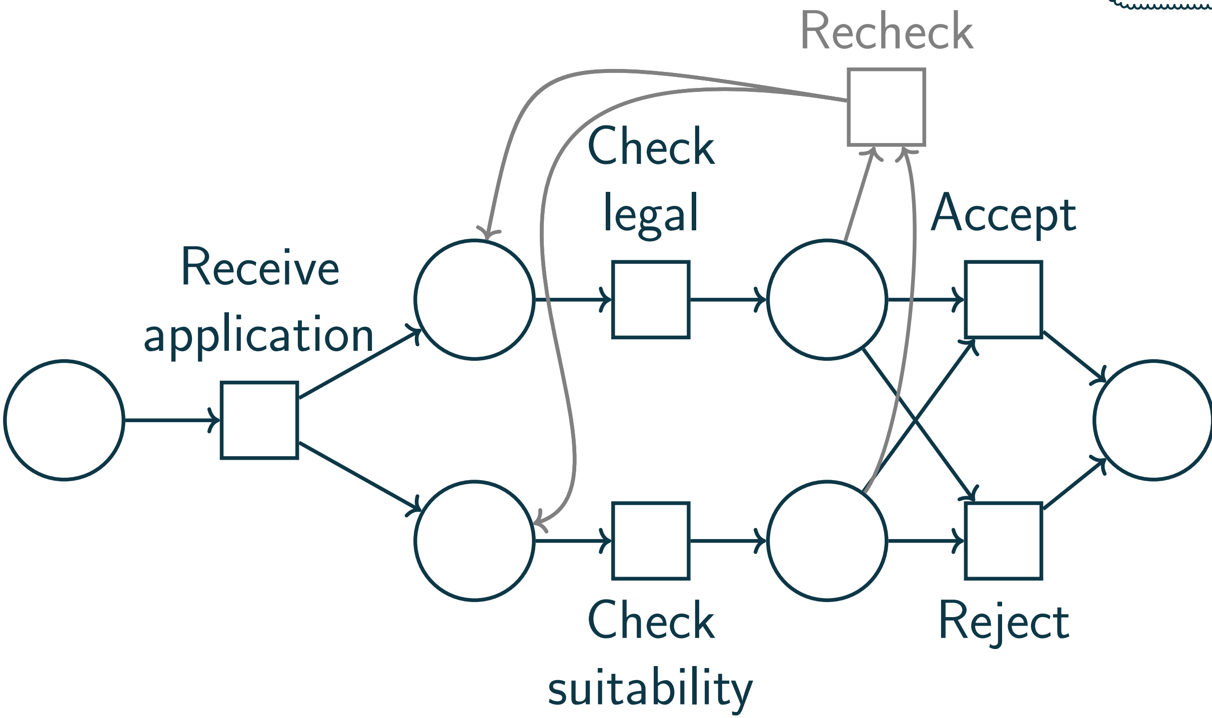
Case ID	Task Name	Resource	Date	Time
1	Receive Application	Peter	04/12/2020	06:37:11
1	Check legal requirements	Anne	05/12/2020	19:21:54
2	Receive Application	Peter	05/12/2020	02:04:19
3	Receive Application	Peter	06/12/2020	11:27:20
1	Check applicant suitability	Eva	06/12/2020	11:25:53
4	Receive Application	Peter	06/12/2020	14:18:20
5	Receive Application	Peter	07/12/2020	12:54:57
2	Check applicant suitability	Eva	08/12/2020	17:20:30
1	Accept	Eva	08/12/2020	06:45:23
3	Check legal requirements	Anne	08/12/2020	06:36:26
4	Check applicant suitability	Eva	16/12/2020	00:21:57
2	Check legal requirements	Anne	16/12/2020	09:03:05
2	Recheck Application	Chris	18/12/2020	19:44:24
2	Check legal requirements	Anne	19/12/2020	20:26:55
4	Check legal requirements	Anne	19/12/2020	17:38:49
4	Reject	Chris	20/12/2020	09:37:59
3	Check applicant suitability	Anne	20/12/2020	01:32:44
2	Check applicant suitability	Peter	27/12/2020	03:35:57
3	Accept	Eva	29/12/2020	17:18:55
2	Reject	Peter	29/12/2020	03:48:06
5	Check legal requirements	Anne	29/12/2020	03:37:39

How can we formally reason about processes?

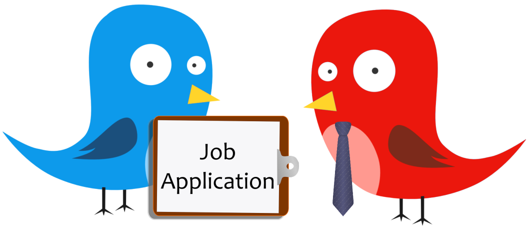
Formally modelling processes: Workflow nets



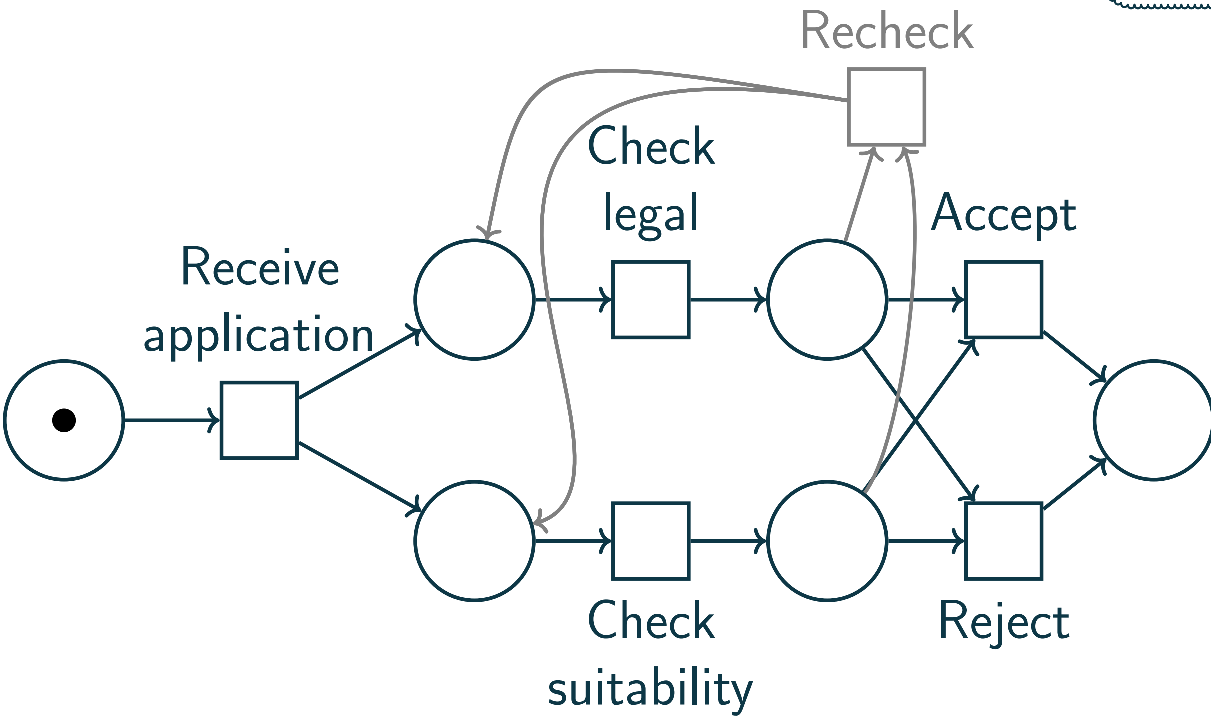
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



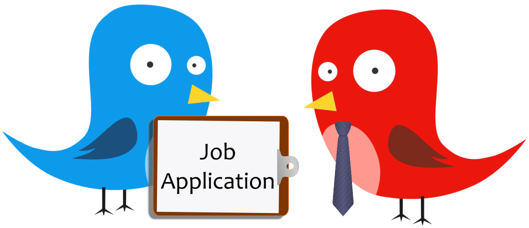
Formally modelling processes: Workflow nets



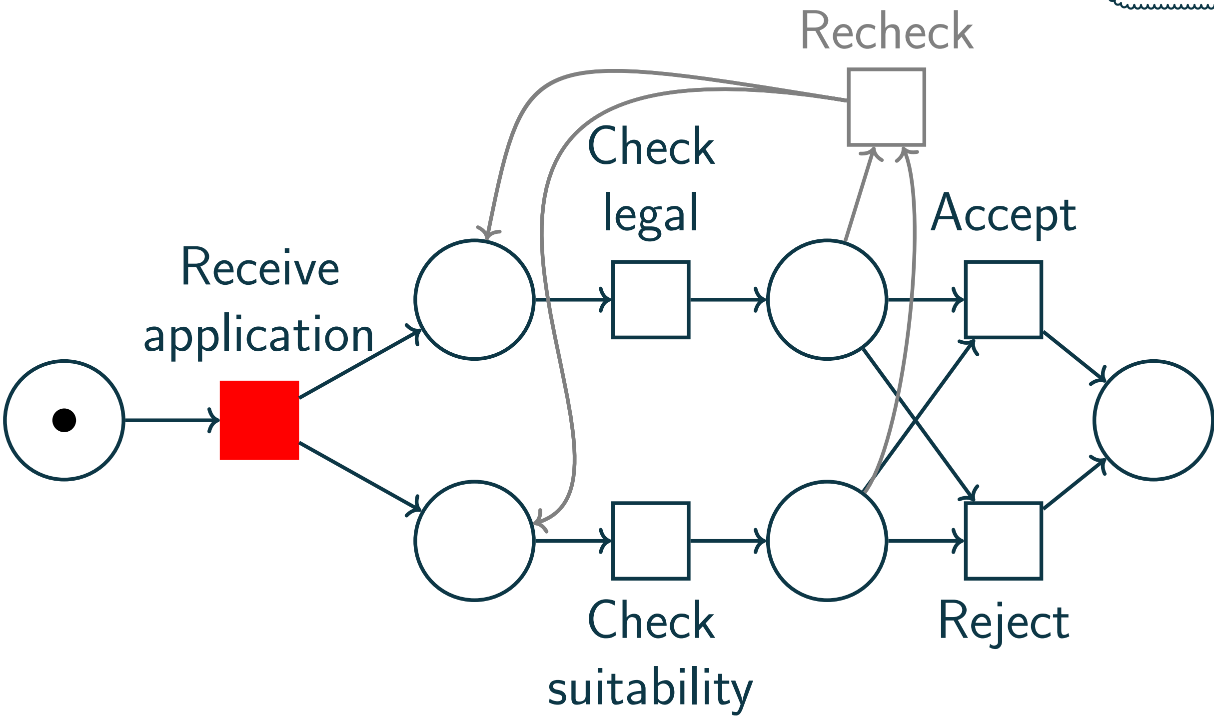
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



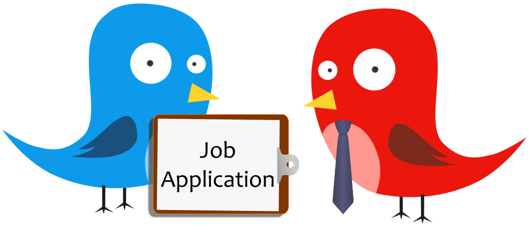
Formally modelling processes: Workflow nets



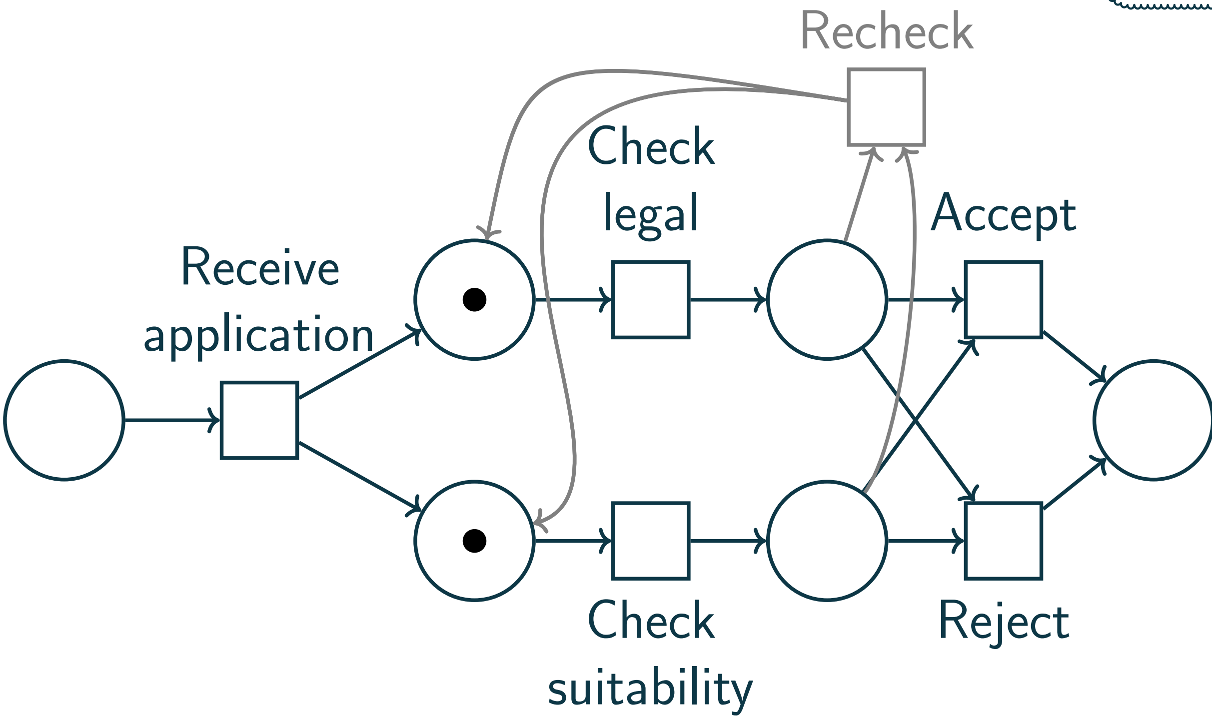
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



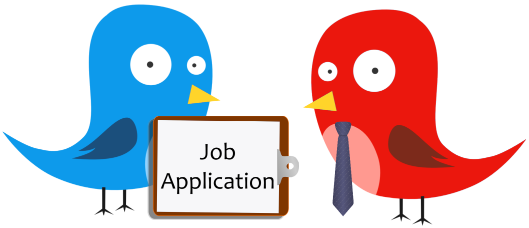
Formally modelling processes: Workflow nets



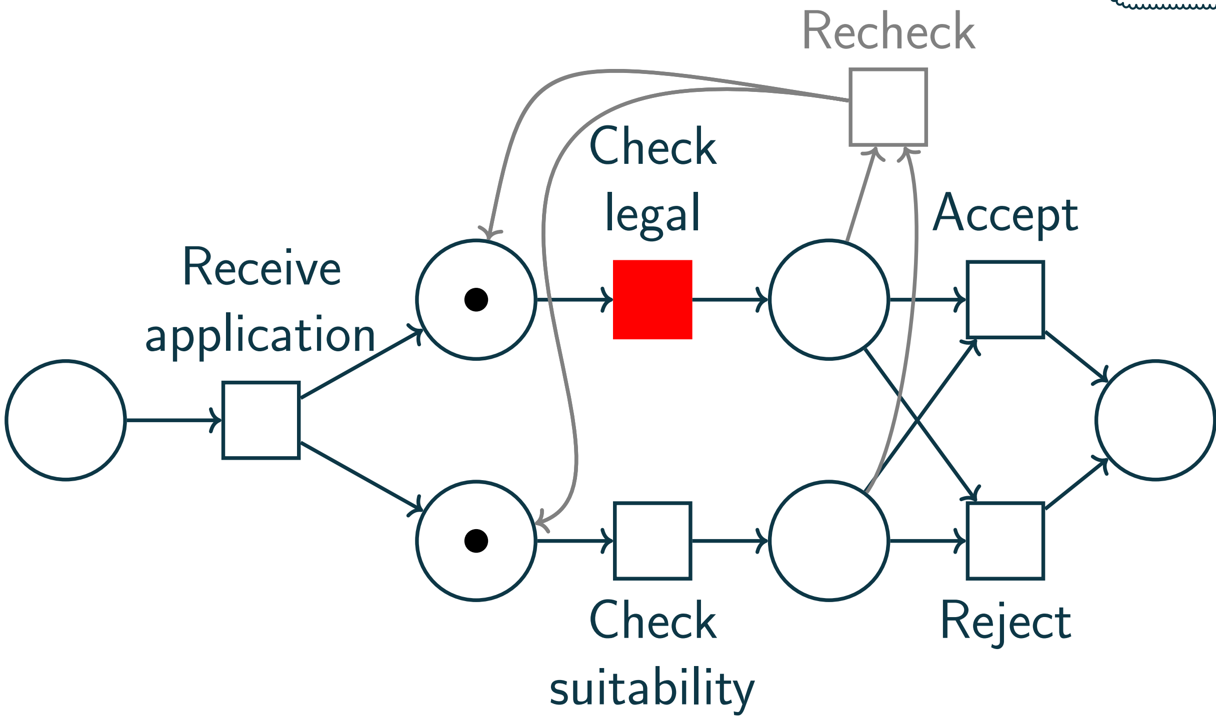
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



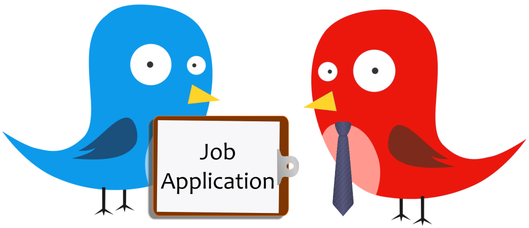
Formally modelling processes: Workflow nets



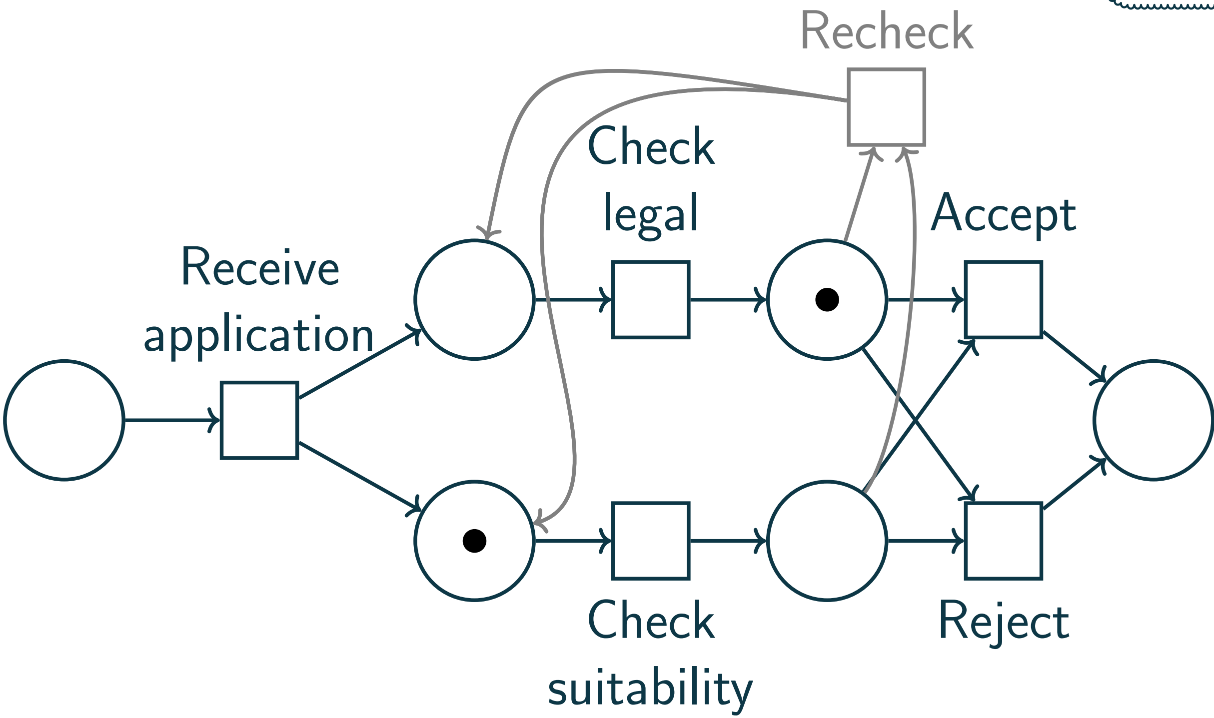
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



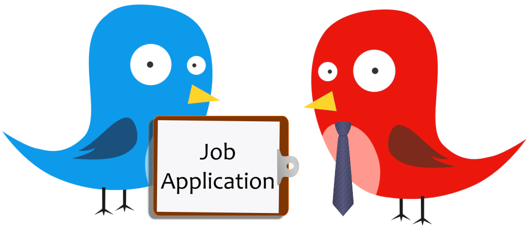
Formally modelling processes: Workflow nets



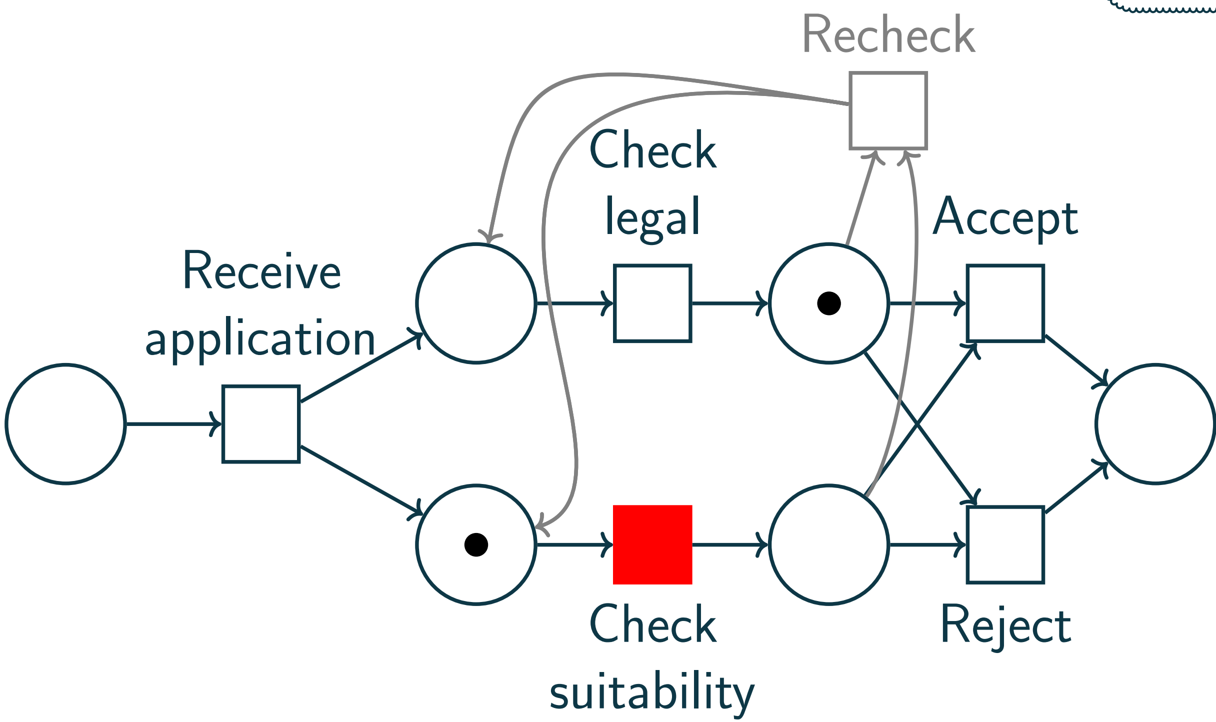
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



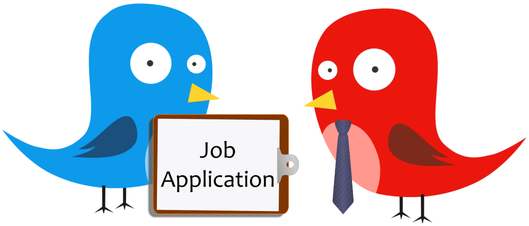
Formally modelling processes: Workflow nets



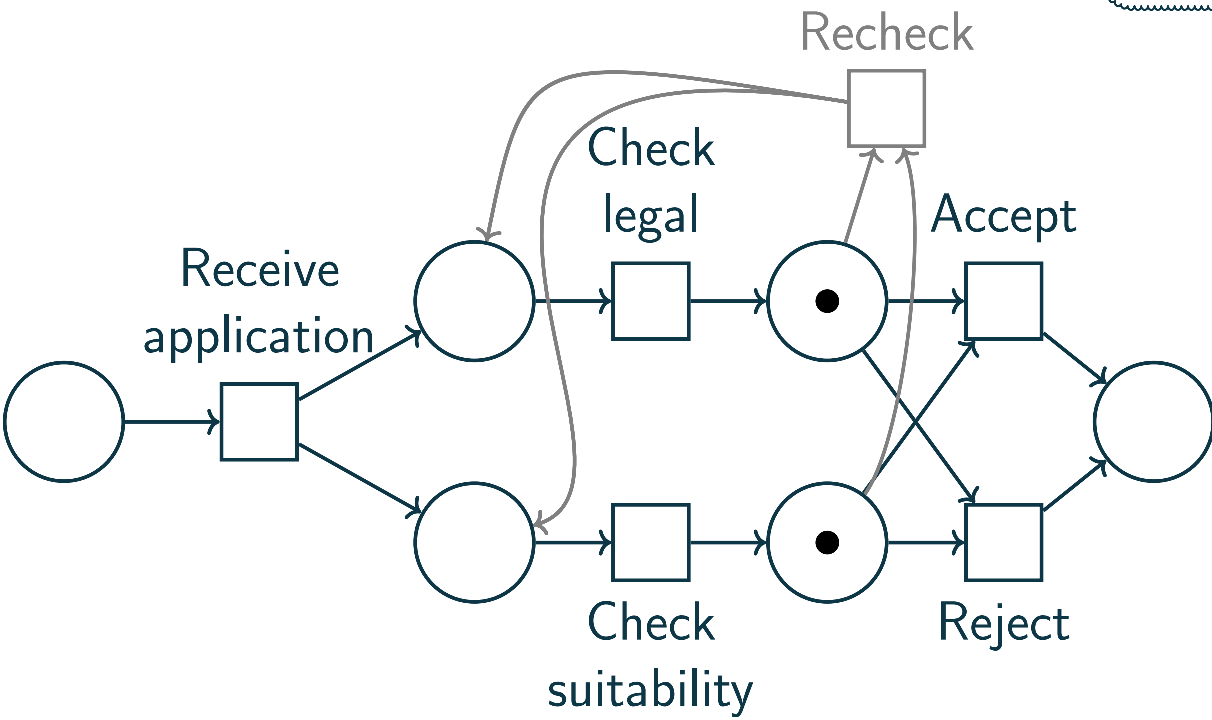
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



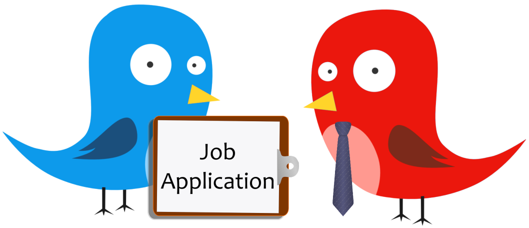
Formally modelling processes: Workflow nets



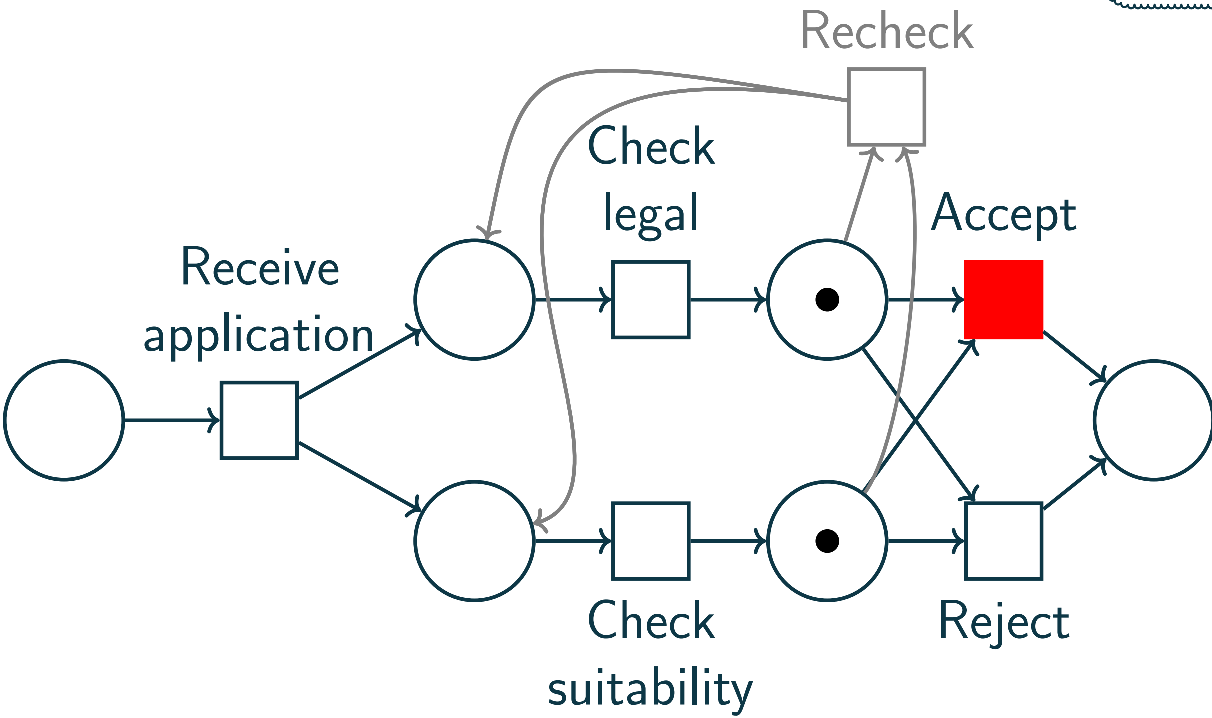
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



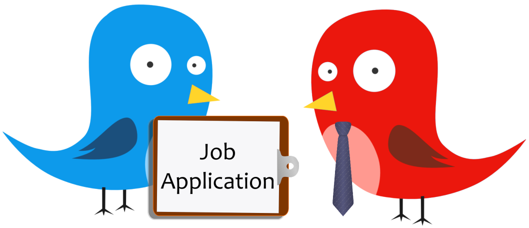
Formally modelling processes: Workflow nets



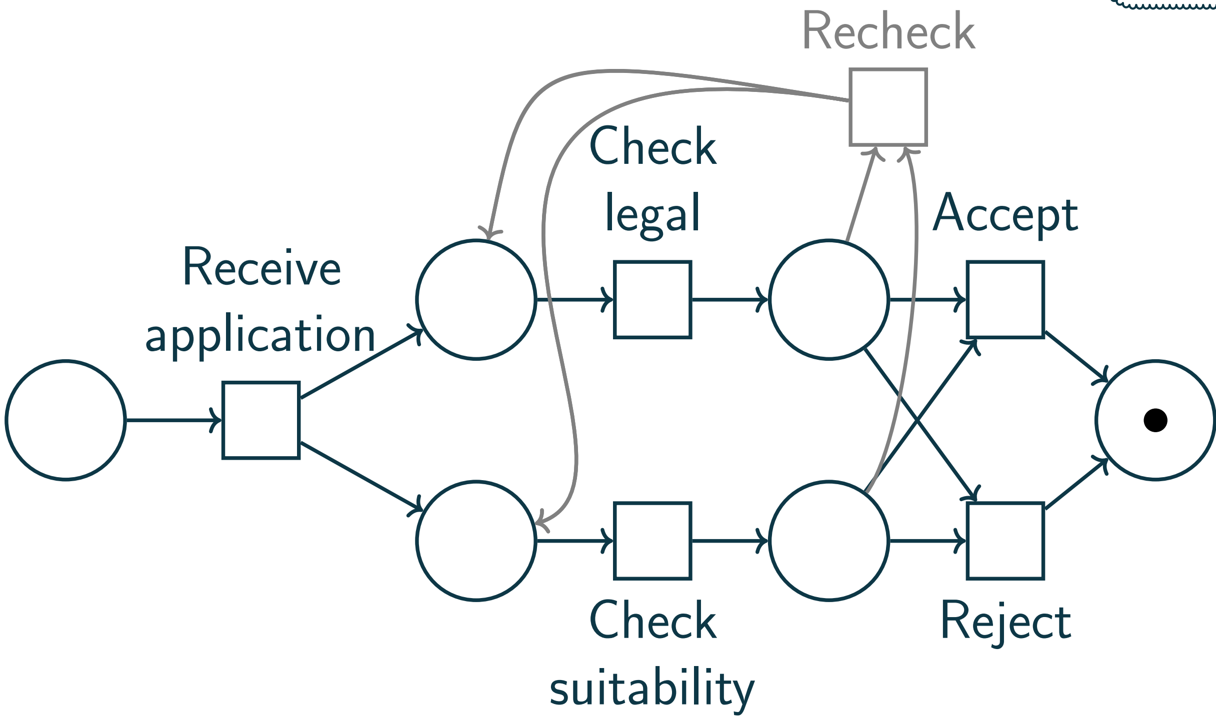
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



Formally modelling processes: Workflow nets

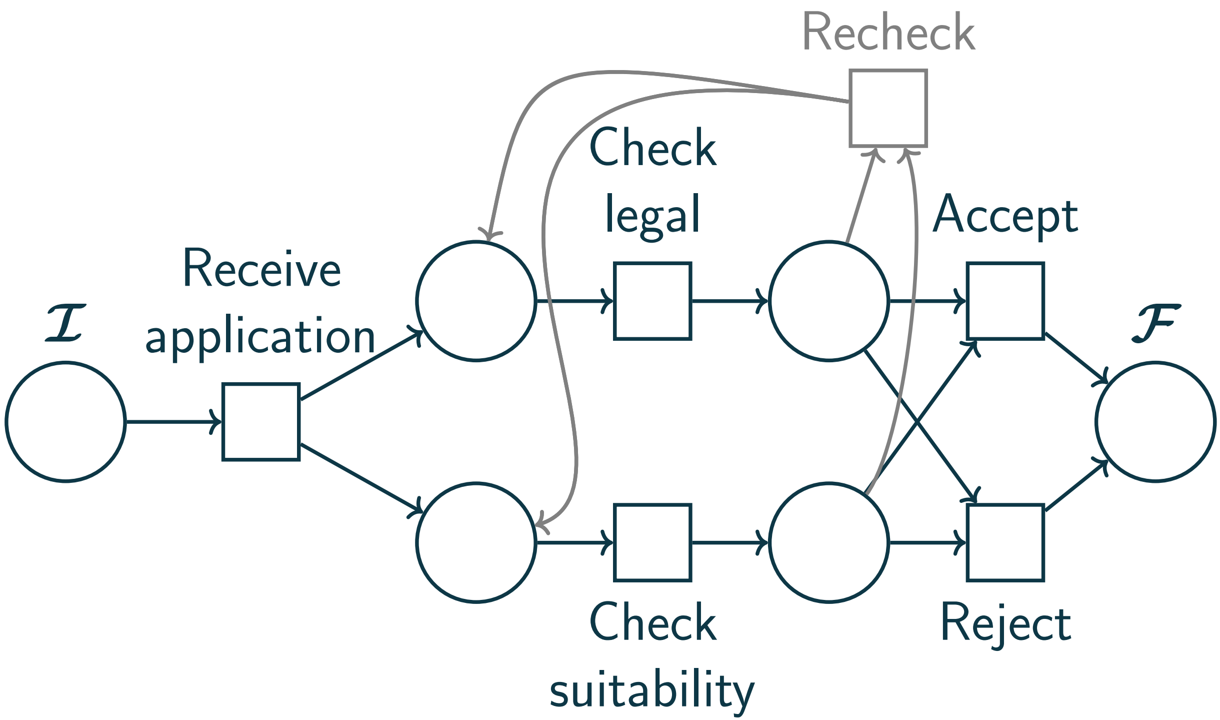


- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



Workflow nets

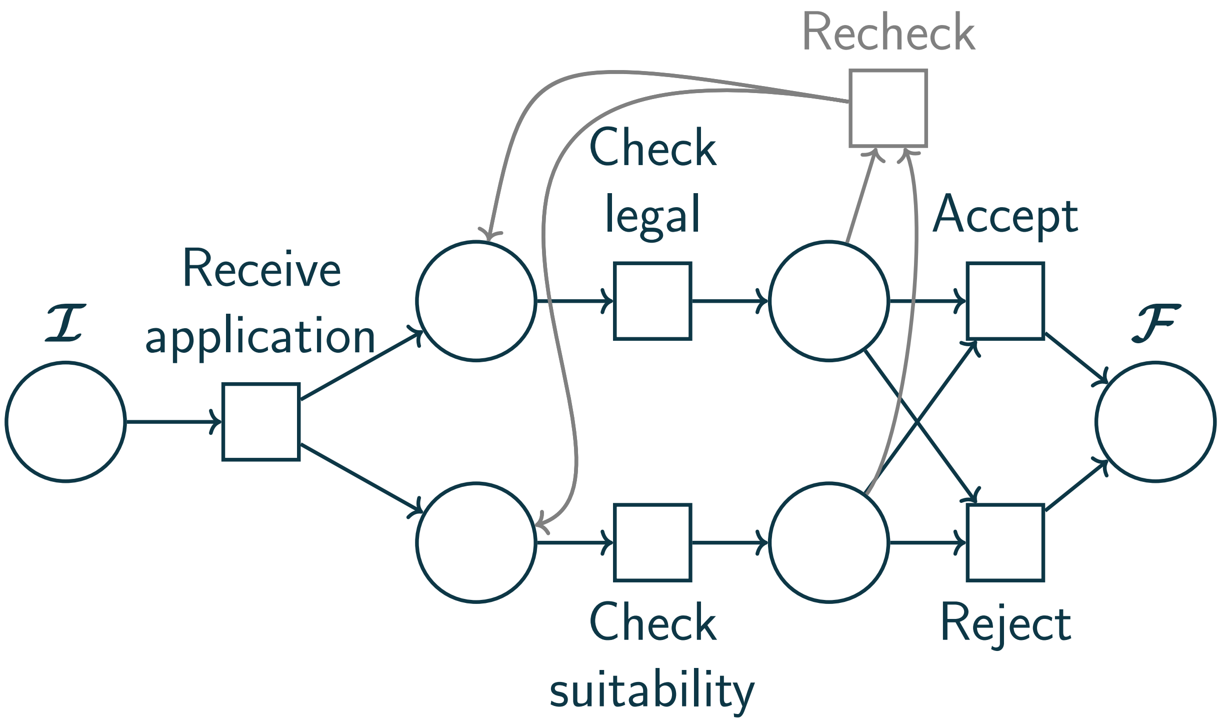
Formally: Petri nets of a specific shape



Workflow nets

Formally: Petri nets of a specific shape

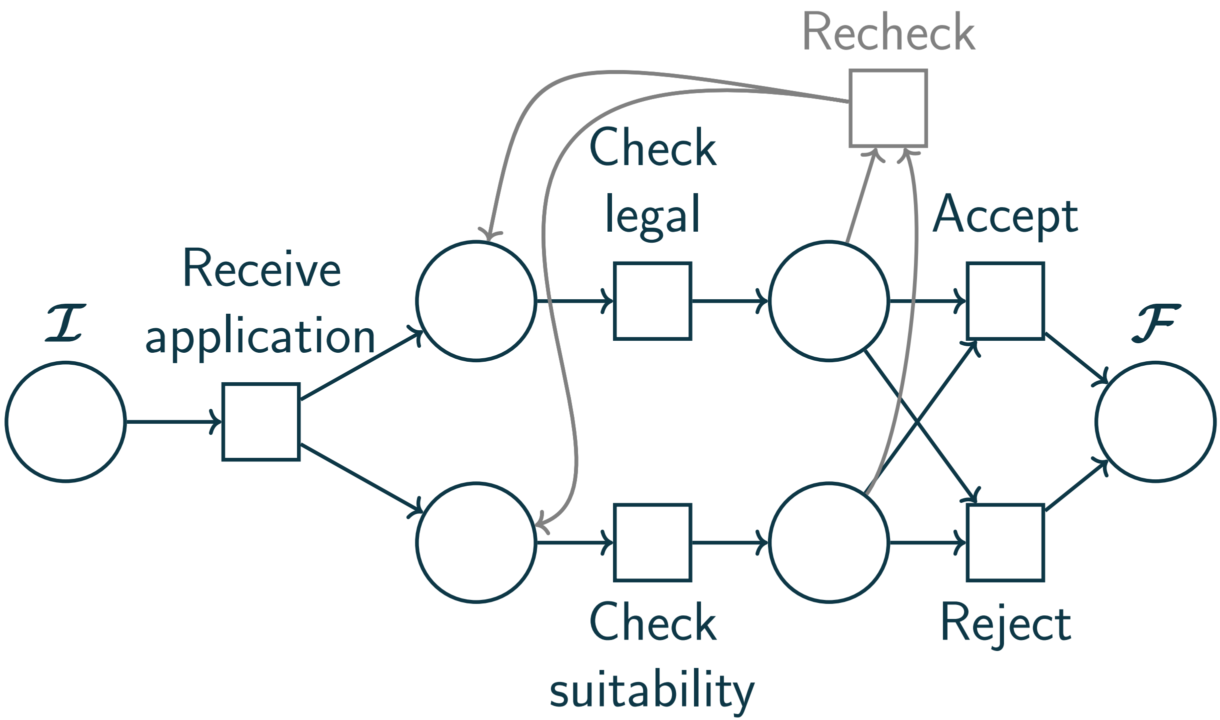
1. $\mathcal{I}$ has no incoming arcs



Workflow nets

Formally: Petri nets of a specific shape

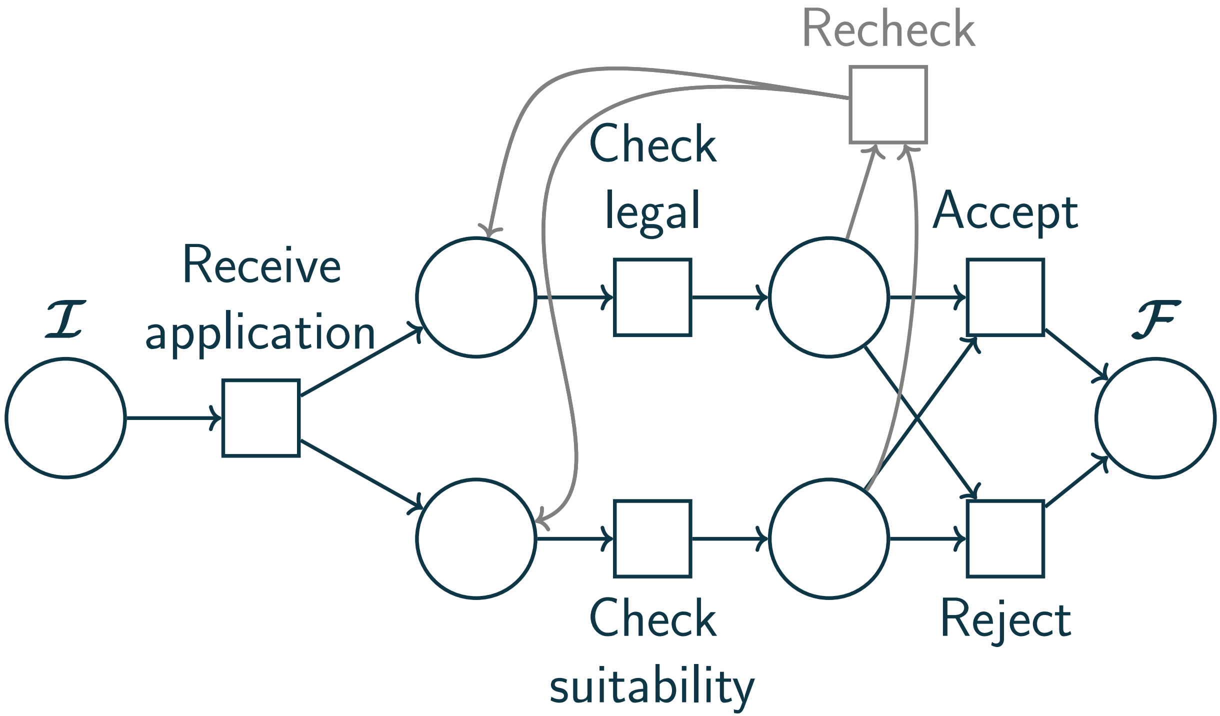
1. $\mathcal{I}$ has no incoming arcs
2. $\mathcal{F}$ has no outgoing arcs



Workflow nets

Formally: Petri nets of a specific shape

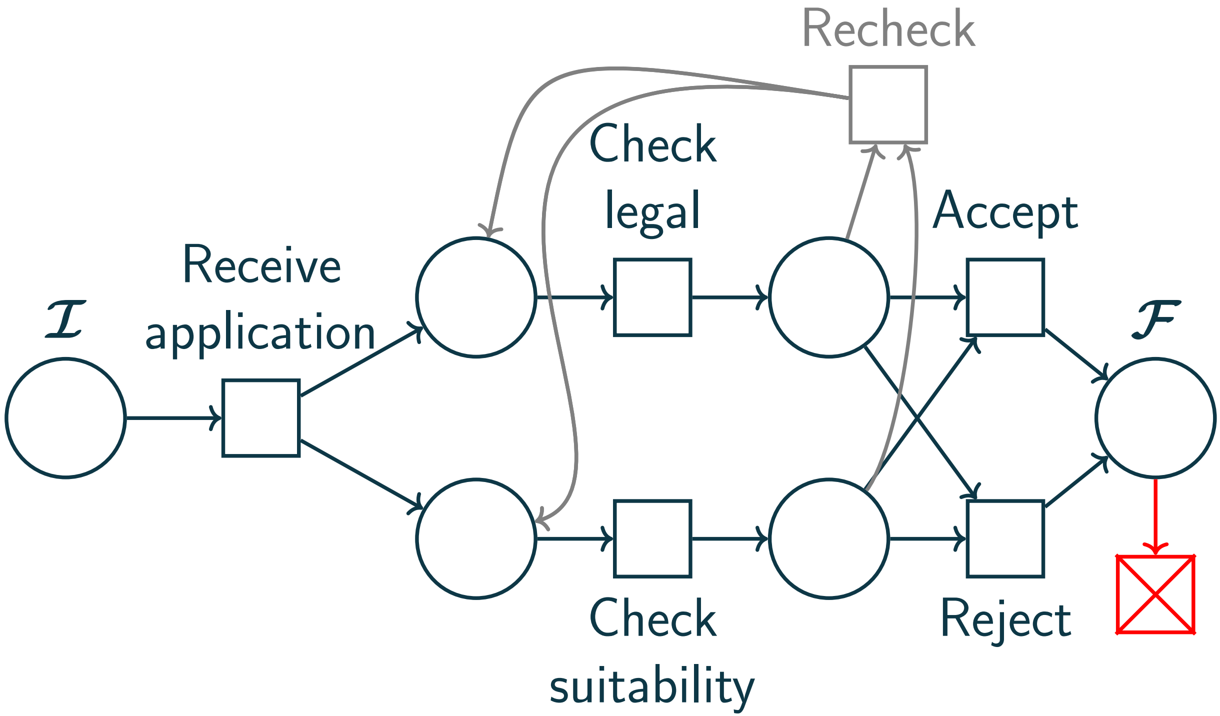
1. $\mathcal{I}$ has no incoming arcs
2. $\mathcal{F}$ has no outgoing arcs
3. All transitions are on a path from $\mathcal{I}$ to $\mathcal{F}$



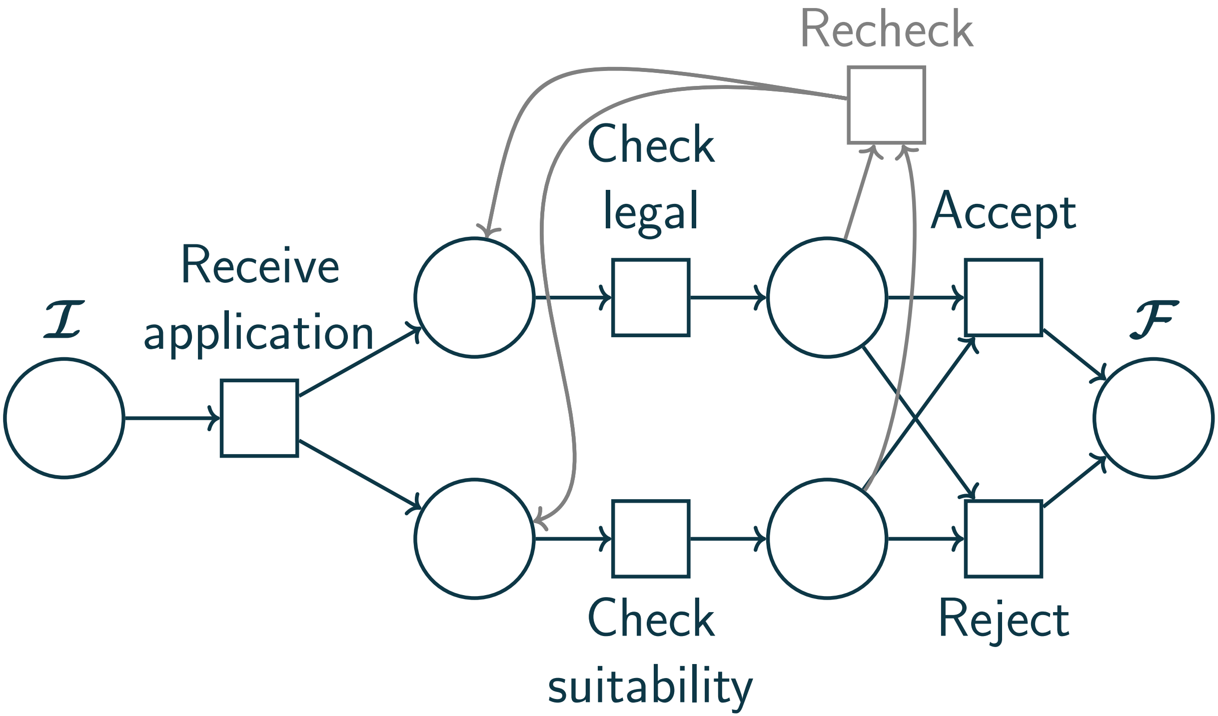
Workflow nets

Formally: Petri nets of a specific shape

1. $\mathcal{I}$ has no incoming arcs
2. $\mathcal{F}$ has no outgoing arcs
3. All transitions are on a path from $\mathcal{I}$ to $\mathcal{F}$

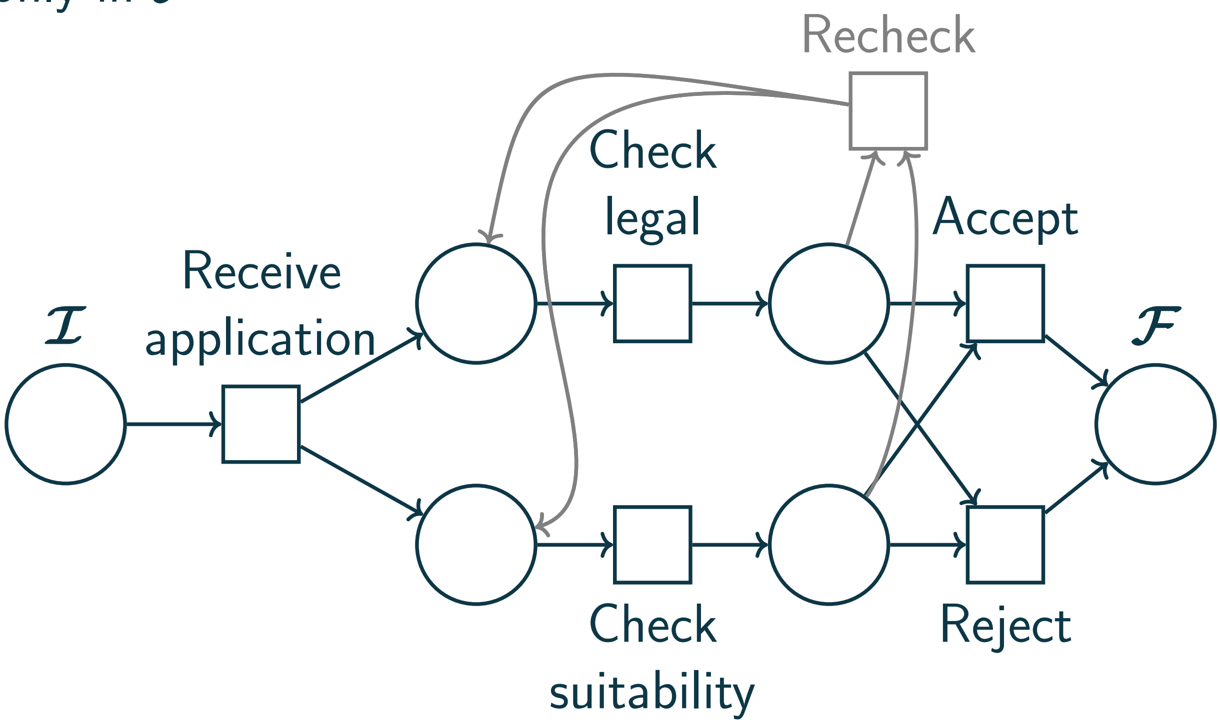


Correctness conditions for processes



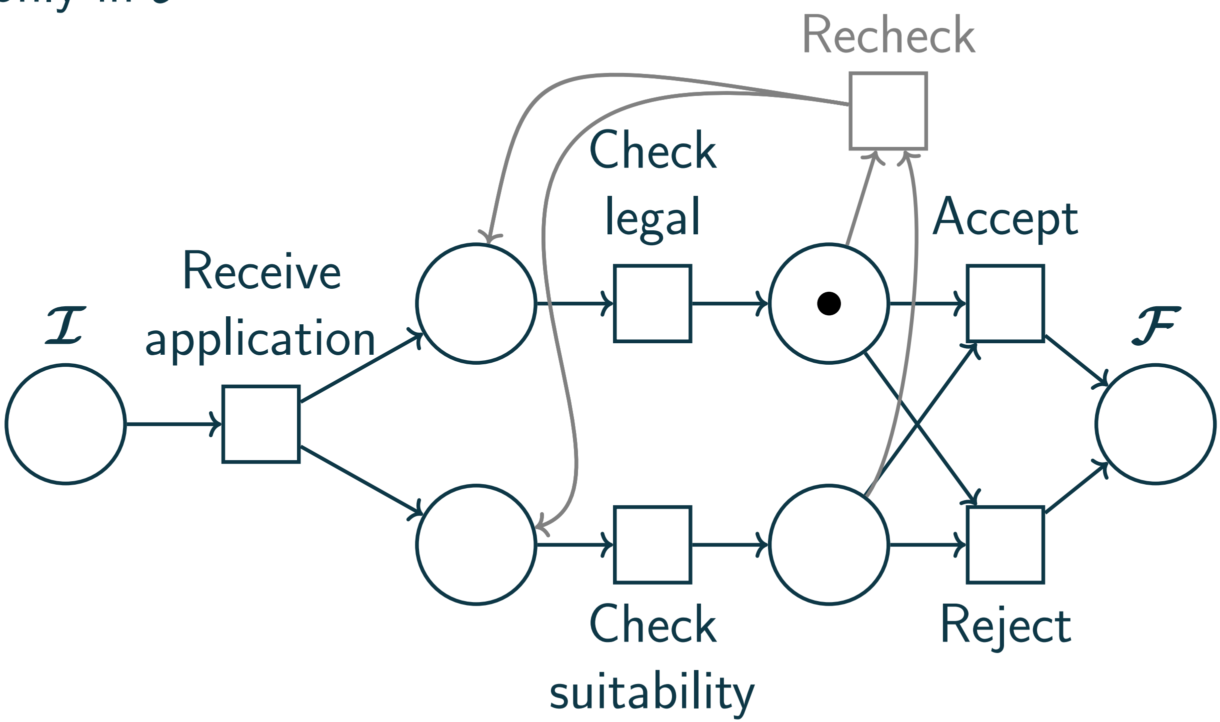
Correctness conditions for processes

Option to complete:
We should be able to reach a
a marking that has tokens
only in $\mathcal{F}$



Correctness conditions for processes

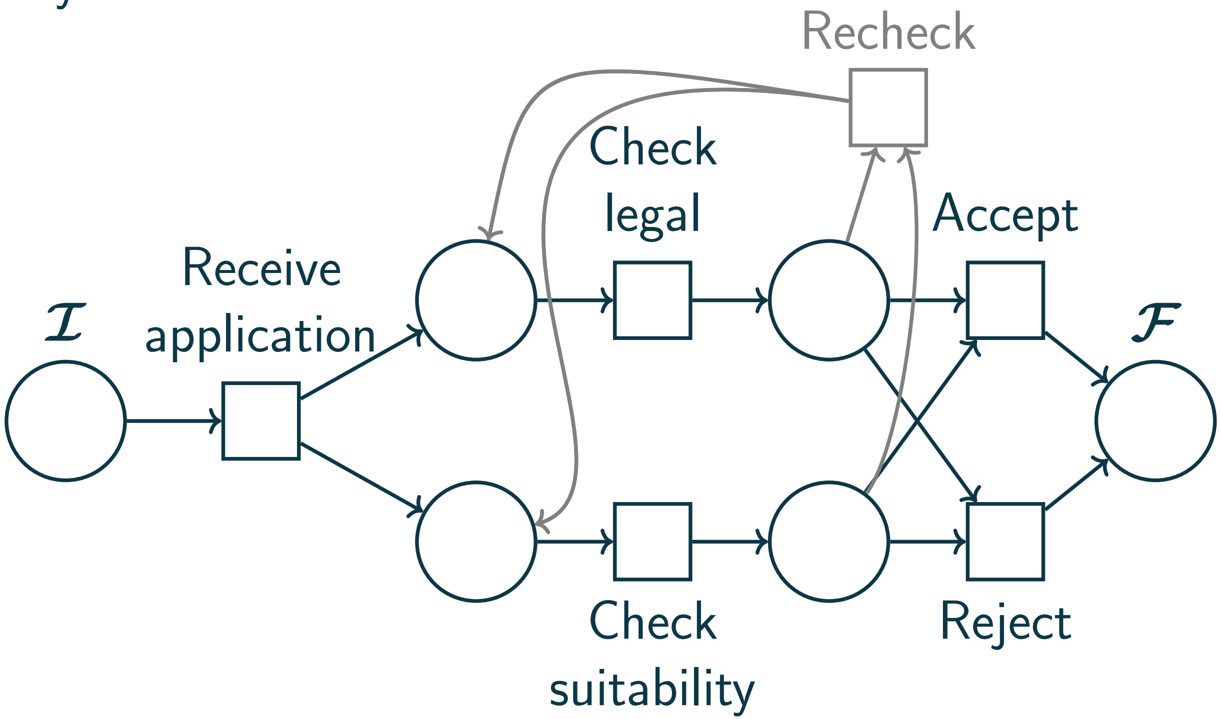
Option to complete:
We should be able to reach a
a marking that has tokens
only in $\mathcal{F}$



Correctness conditions for processes

Option to complete:
We should be able to reach a a marking that has tokens only in $\mathcal{F}$

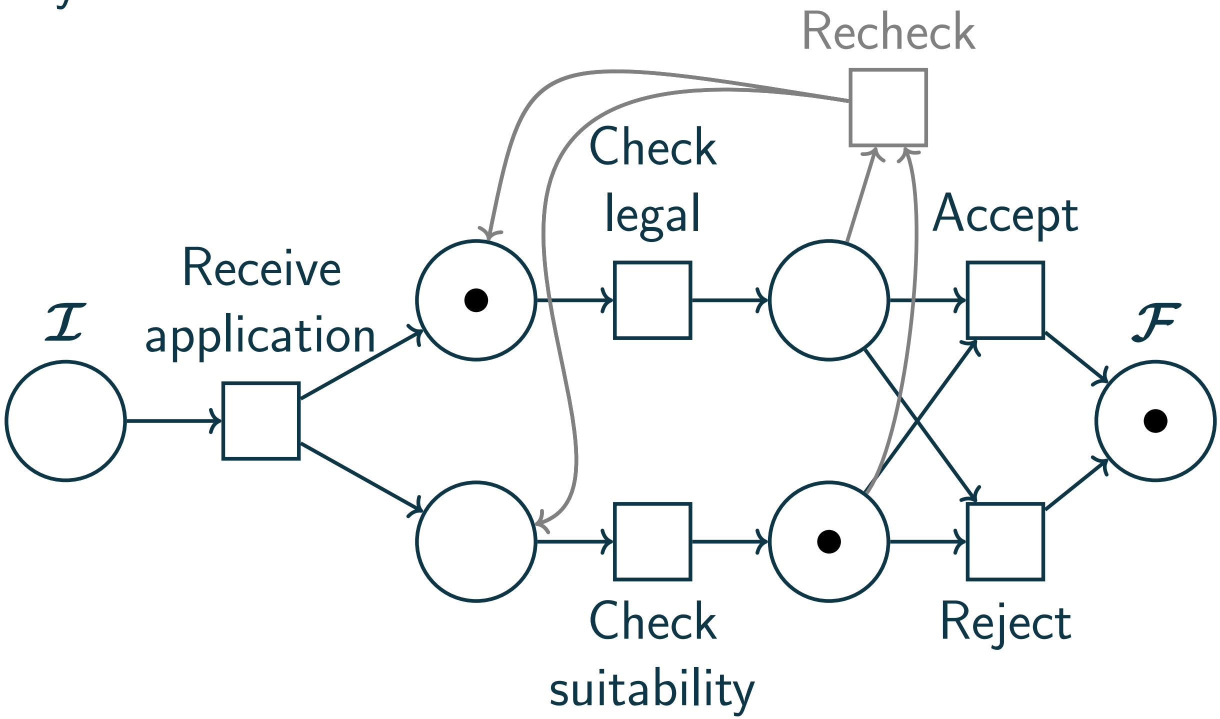
Proper completion:
When $\mathcal{F}$ is marked the rest of the net is empty



Correctness conditions for processes

Option to complete:
We should be able to reach a a marking that has tokens only in $\mathcal{F}$

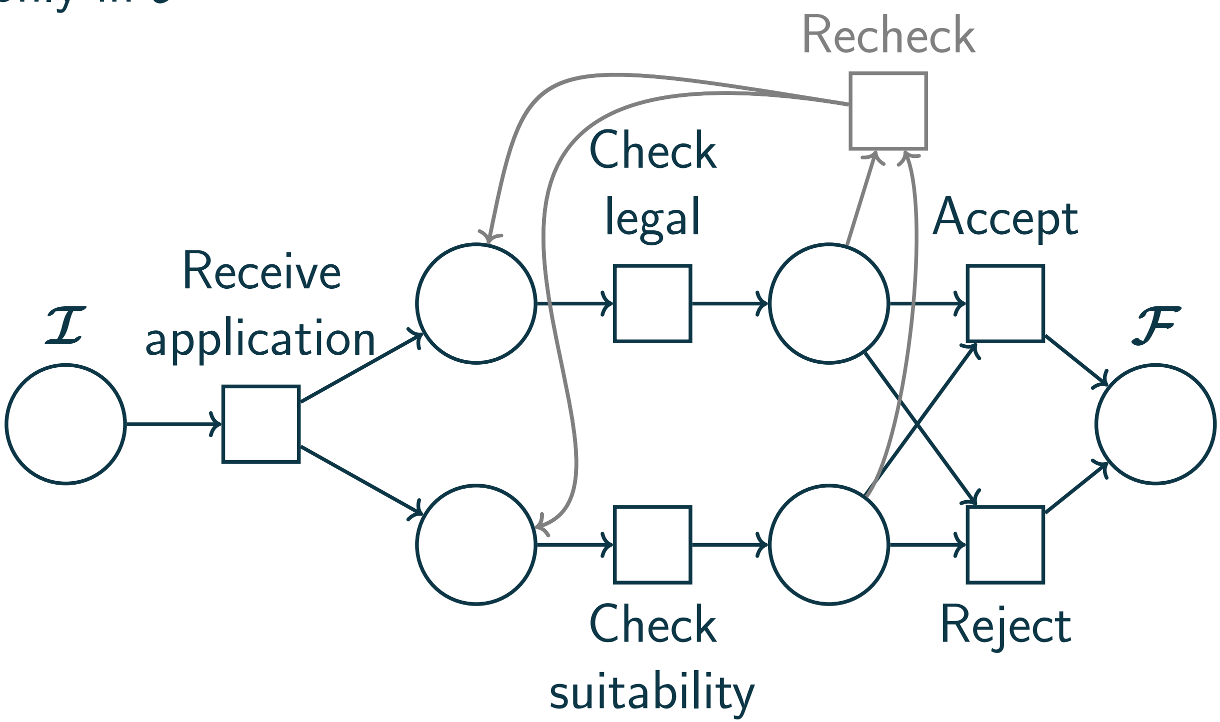
Proper completion:
When $\mathcal{F}$ is marked the rest of the net is empty



Correctness conditions for processes

Option to complete:
We should be able to reach a a marking that has tokens only in $\mathcal{F}$

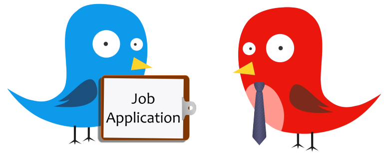
Proper completion:
When $\mathcal{F}$ is marked the rest of the net is empty



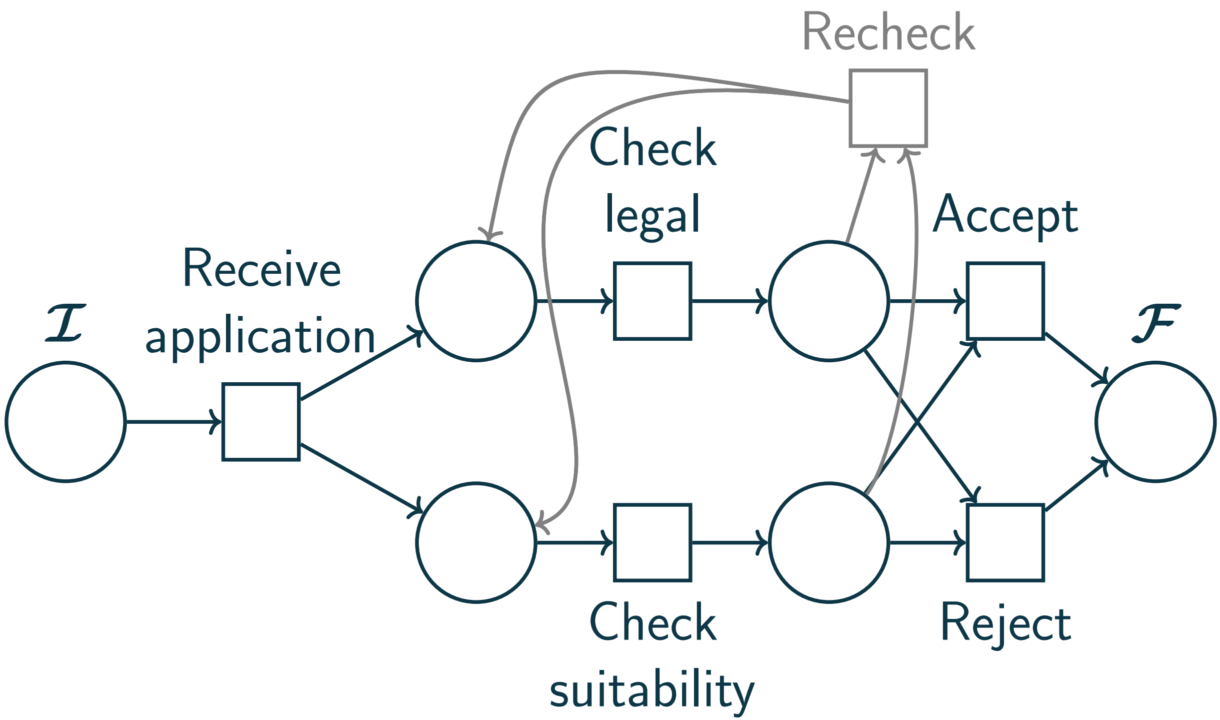
Can we condense these into a single condition?

A concise correctness condition

Soundness:
From any marking reachable from $\{\mathcal{I}: 1\}$, the final marking $\{\mathcal{F}: 1\}$ can be reached



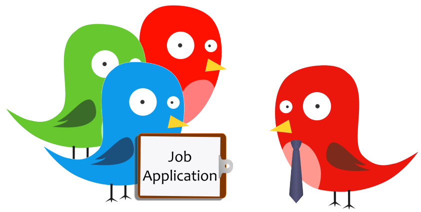
$\forall$ runs π $\exists$ run π' : $\{\mathcal{I}: 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}: 1\}$

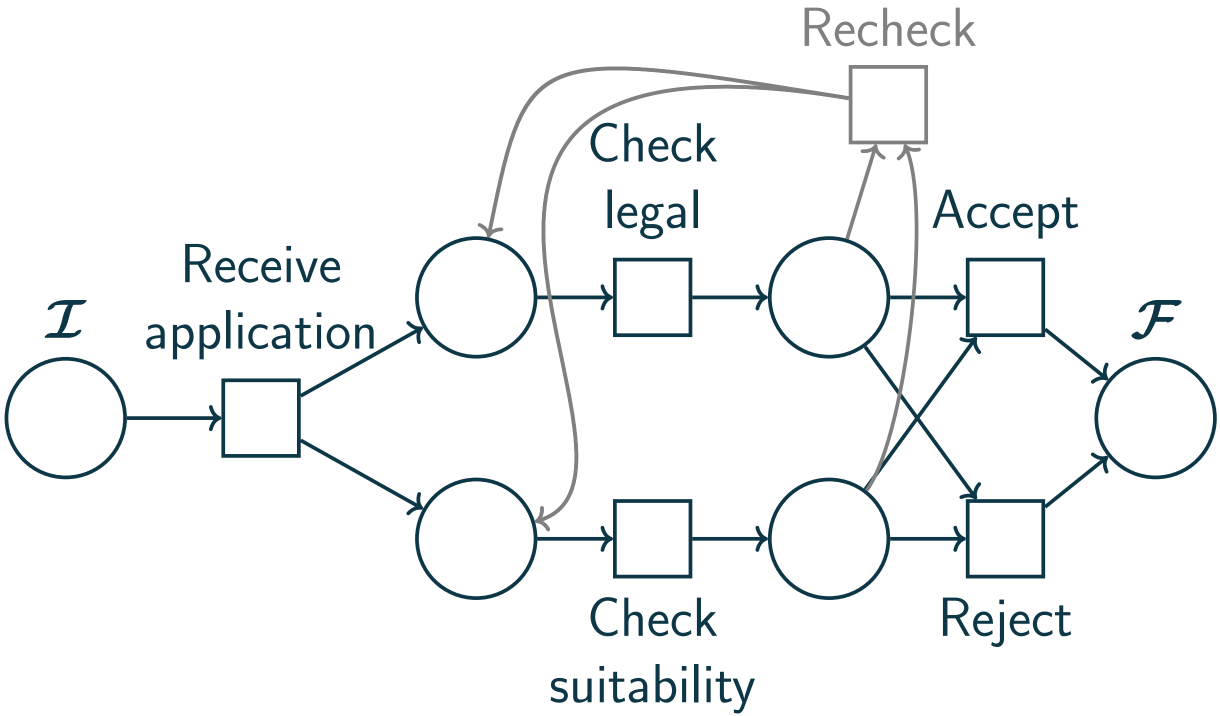


A concise correctness condition

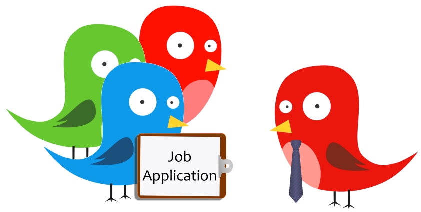
Soundness:
From any marking reachable from $\{\mathcal{I}: 1\}$, the final marking $\{\mathcal{F}: 1\}$ can be reached

$\forall$ runs π $\exists$ run π' : $\{\mathcal{I}: 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}: 1\}$

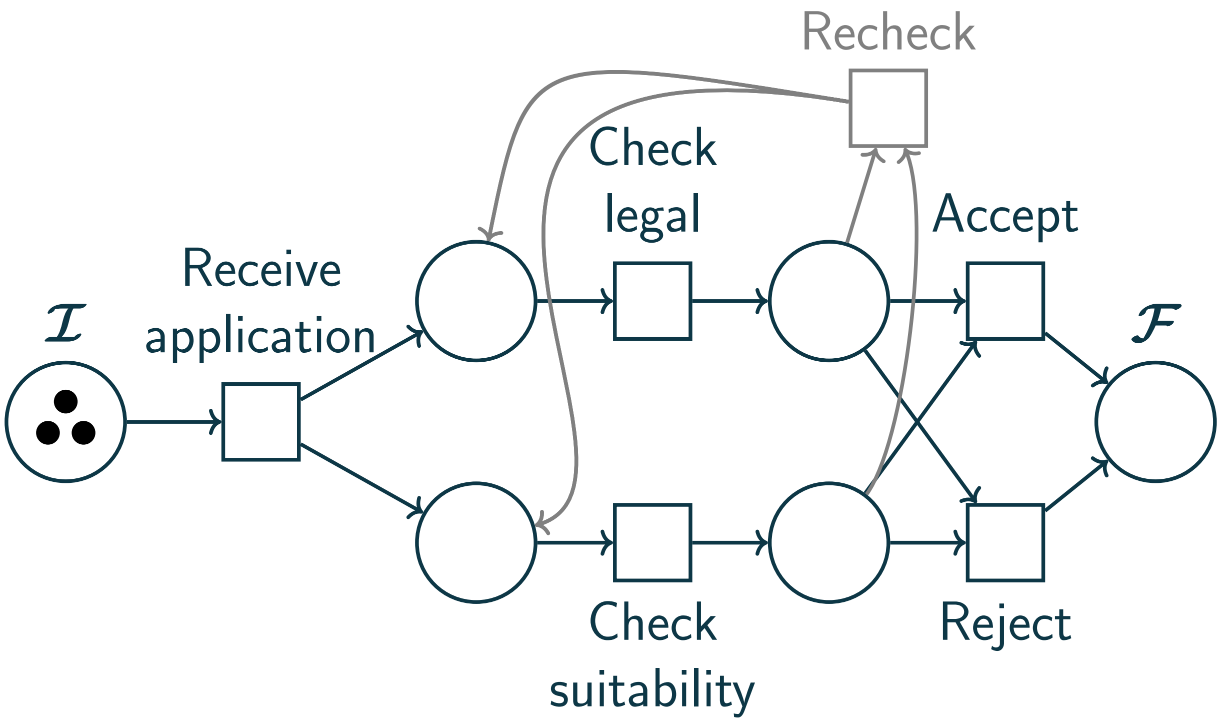




Extending soundness



k-**soundness:**
From any marking reachable from $\{\mathcal{I}: k\}$, the final marking $\{\mathcal{F}: k\}$ can be reached



Variants of soundness

k-soundness:
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

Variants of soundness

k-soundness:
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

Generalised soundness:
$\forall k$: k-sound

Variants of soundness

k-soundness:
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

Generalised soundness:
$\forall k$: k-sound

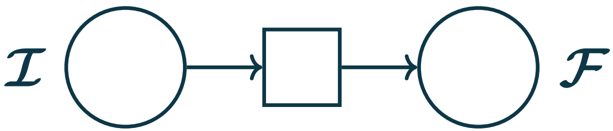
Structural soundness:
$\exists k$: k-sound

Variants of soundness

k-soundness:
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

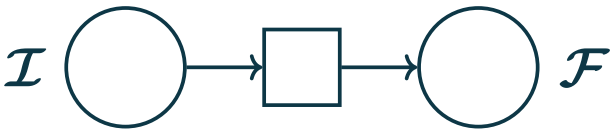
Generalised soundness:
$\forall k$: k-sound

Structural soundness:
$\exists k$: k-sound



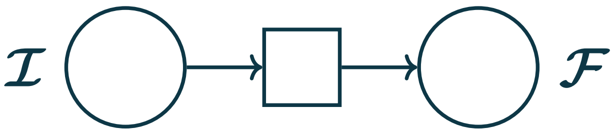
Variants of soundness

***k*-soundness:**
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k\colon$ k-sound	**Structural soundness:** $\exists k\colon$ k-sound
	✓	

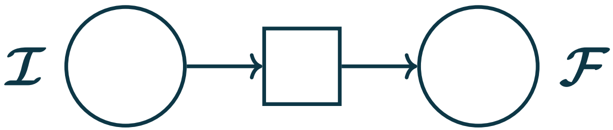
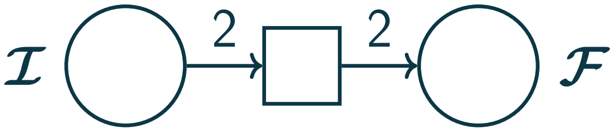
Variants of soundness

k-soundness:
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k$: k-sound	**Structural soundness:** $\exists k$: k-sound
	✓	✓

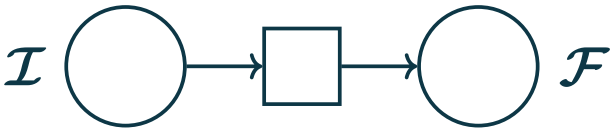
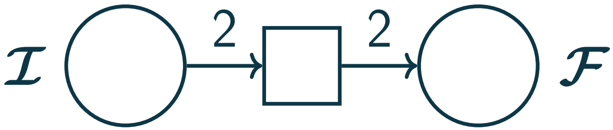
Variants of soundness

k-soundness:
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k\colon$ k-sound	**Structural soundness:** $\exists k\colon$ k-sound
	✓	✓
		

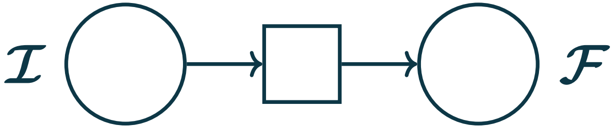
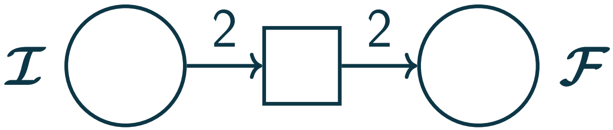
Variants of soundness

k-**soundness:**
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k$: k-sound	**Structural soundness:** $\exists k$: k-sound
	✓	✓
	X Not 1-sound	

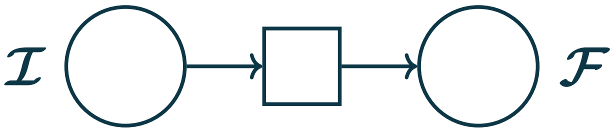
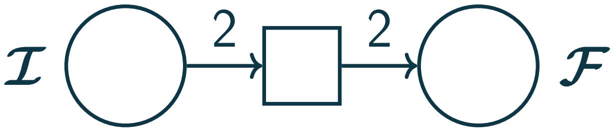
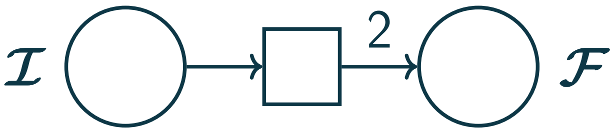
Variants of soundness

***k*-soundness:**
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k$: k-sound	**Structural soundness:** $\exists k$: k-sound
	✓	✓
	X Not 1-sound	✓ 2-sound

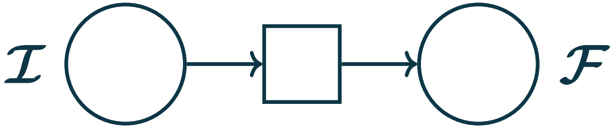
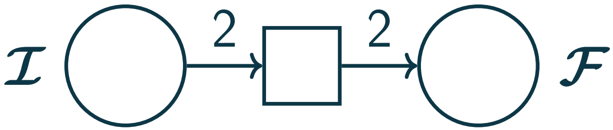
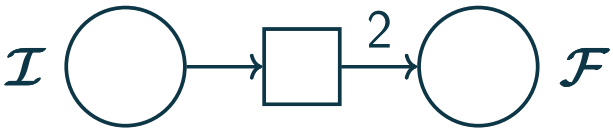
Variants of soundness

***k*-soundness:**
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k$: k-sound	**Structural soundness:** $\exists k$: k-sound
	✓	✓
	X Not 1-sound	✓ 2-sound
		

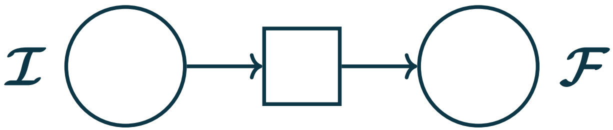
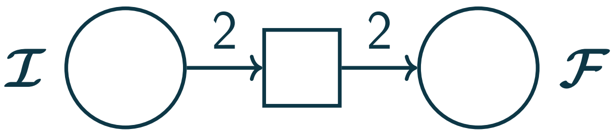
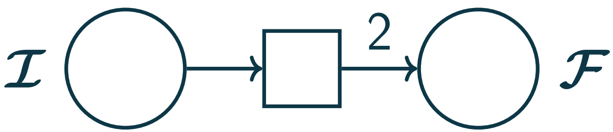
Variants of soundness

***k*-soundness:**
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

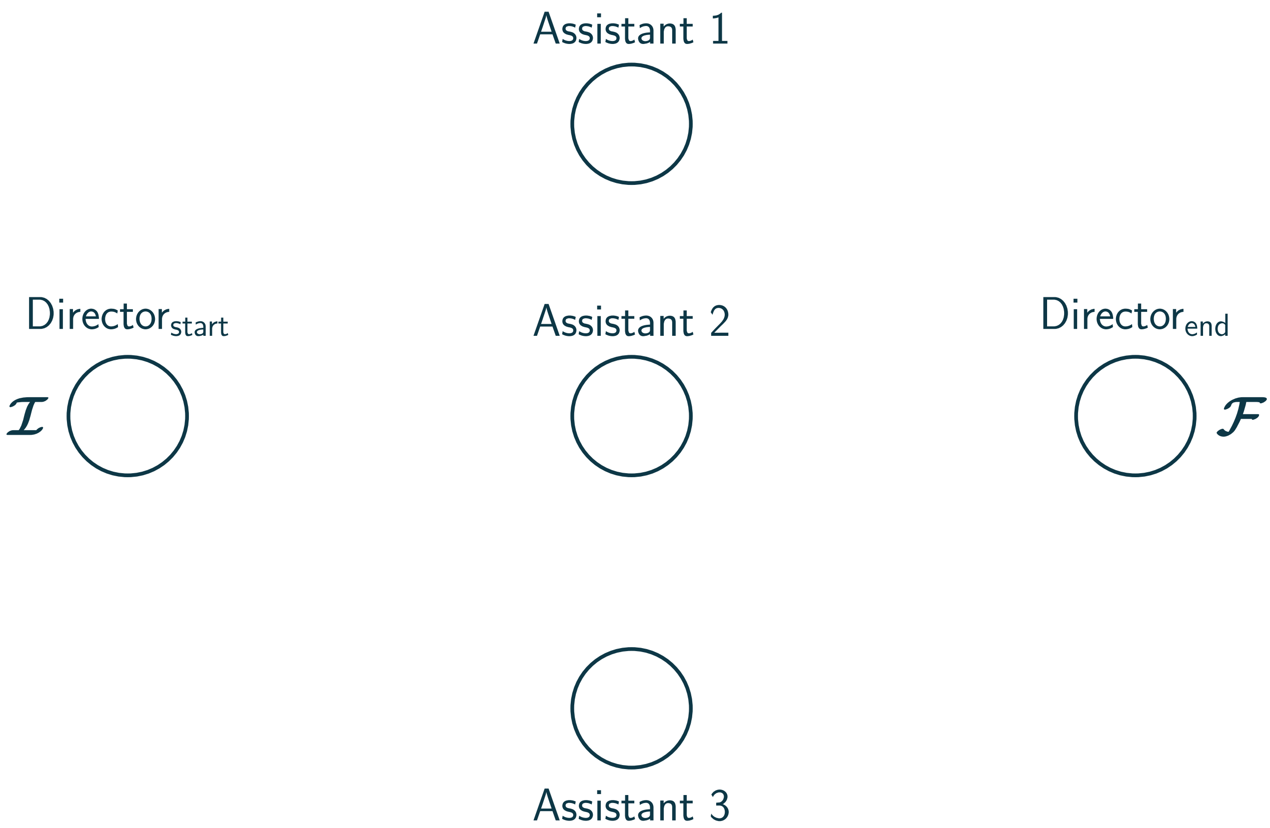
	Generalised soundness: $\forall k\colon k$-sound	**Structural soundness:** $\exists k\colon k$-sound
	✓	✓
	X Not 1-sound	✓ 2-sound
	X	

Variants of soundness

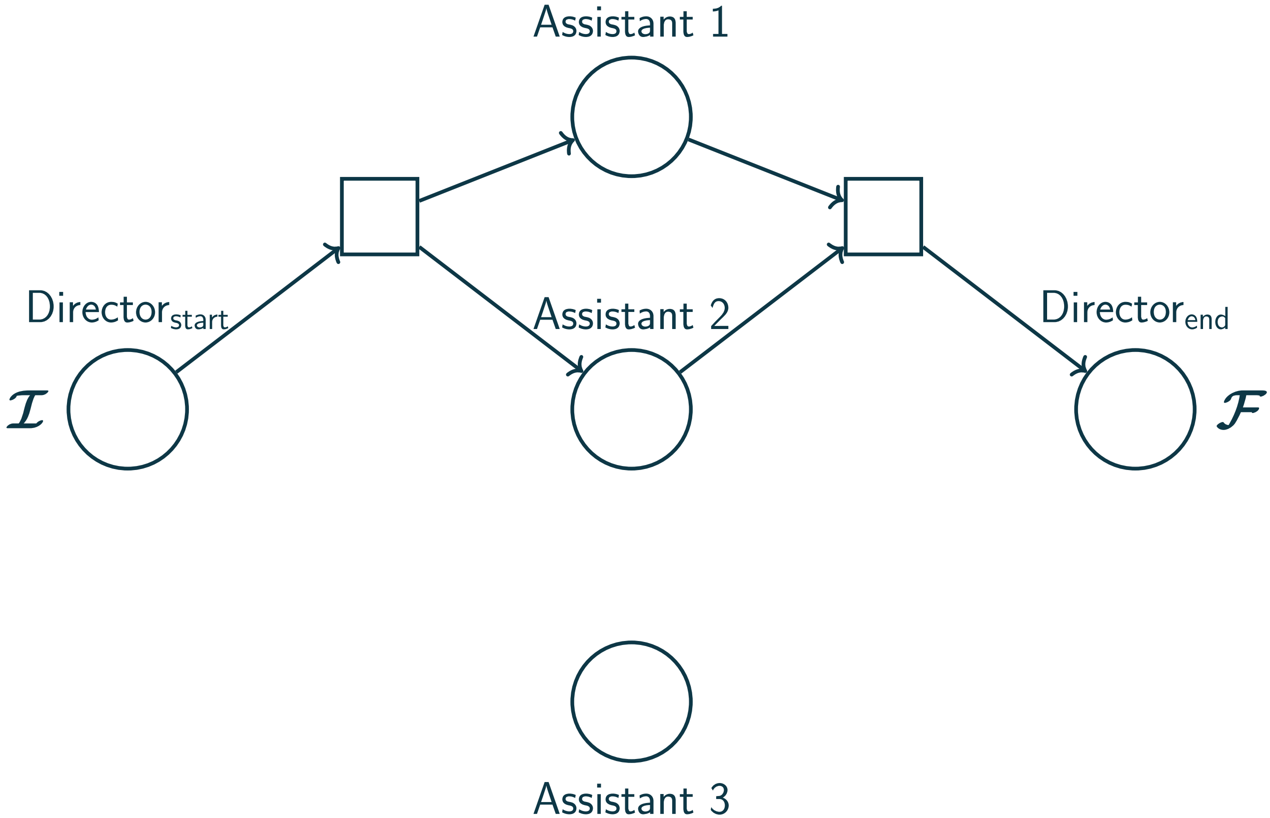
***k*-soundness:**
From any marking reachable from $\{\mathcal{I}\colon k\}$,
the final marking $\{\mathcal{F}\colon k\}$ can be reached

	Generalised soundness: $\forall k\colon$ k-sound	**Structural soundness:** $\exists k\colon$ k-sound
	✓	✓
	X Not 1-sound	✓ 2-sound
	X	X

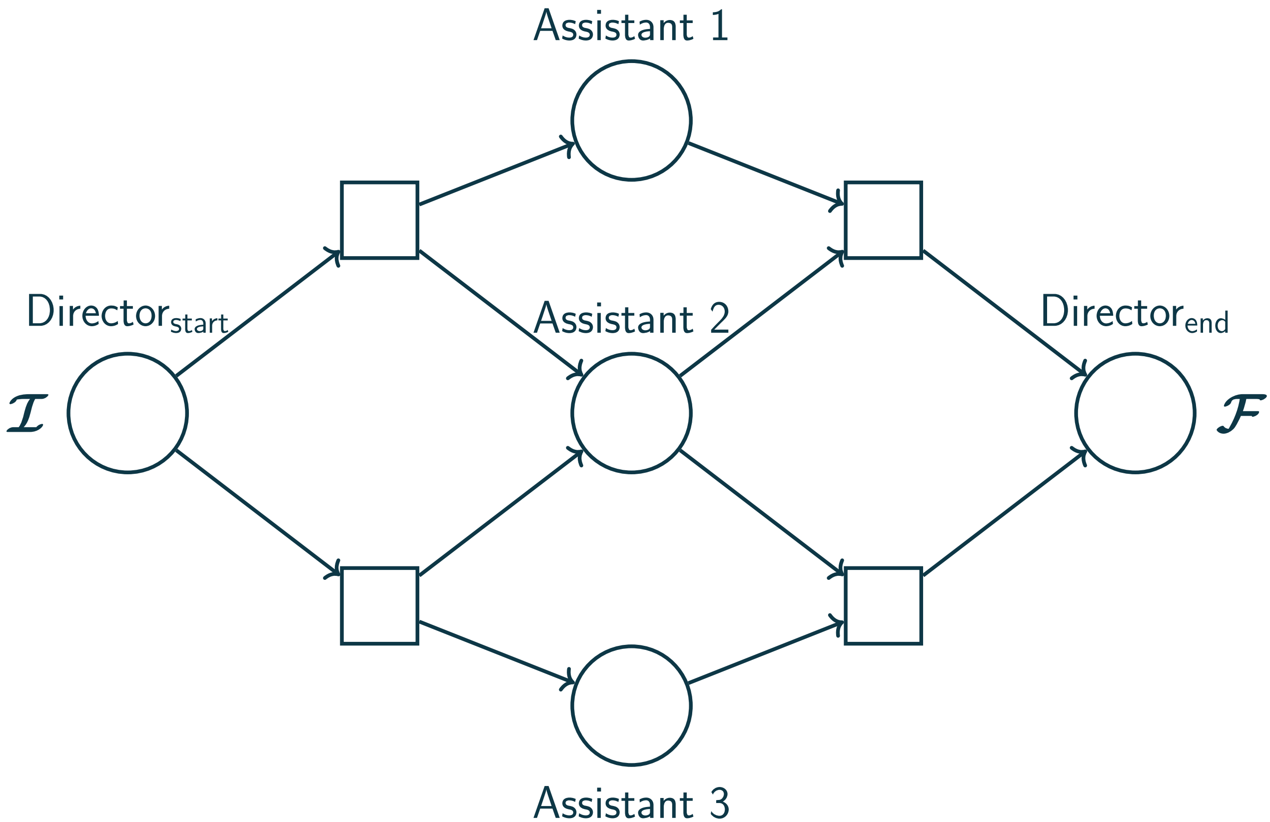
Variants of soundness



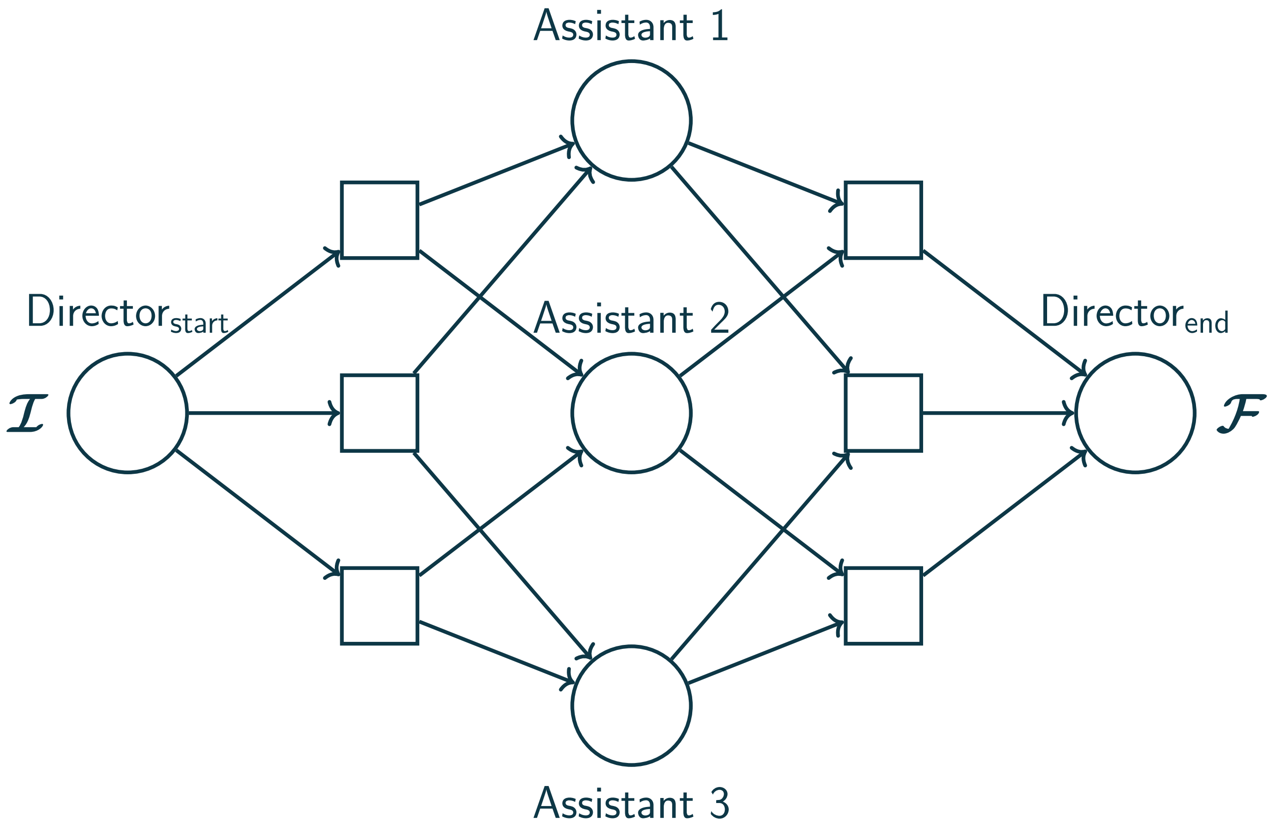
Variants of soundness



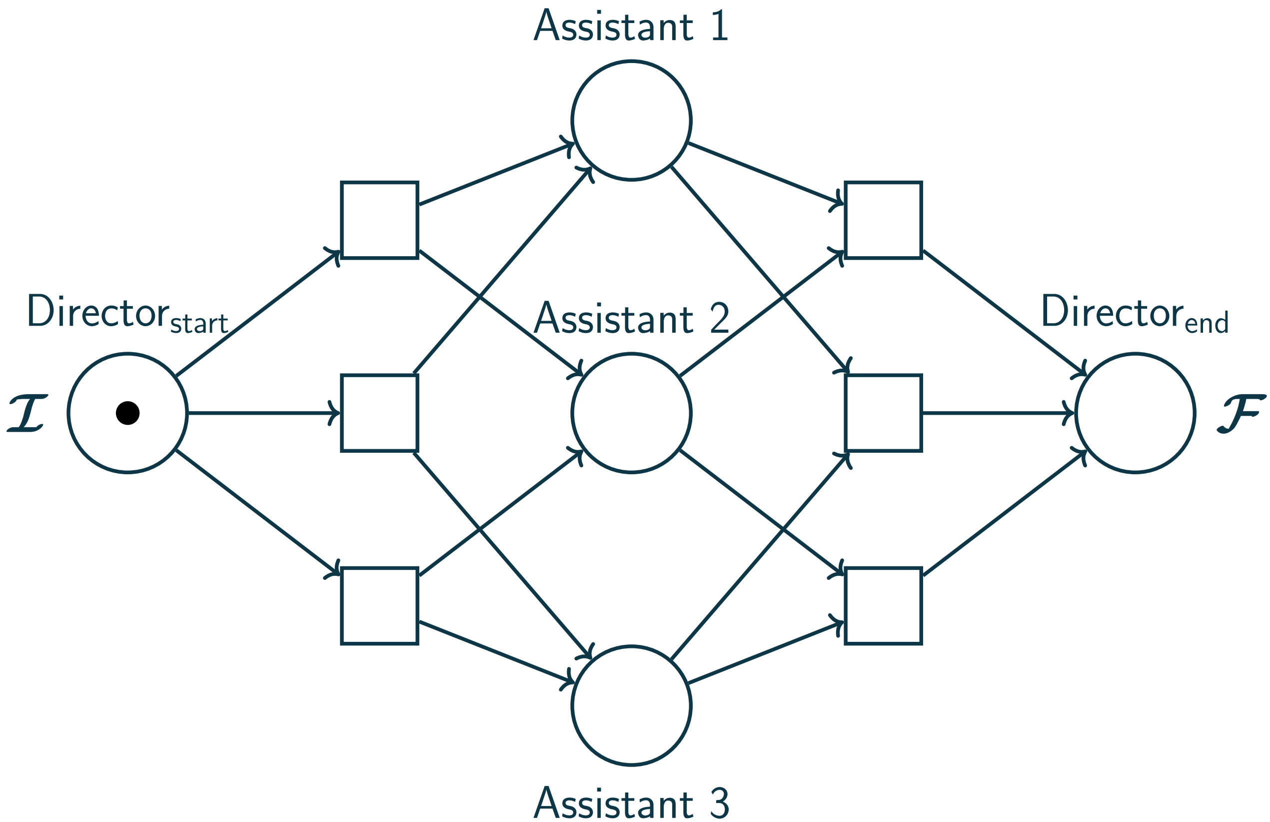
Variants of soundness



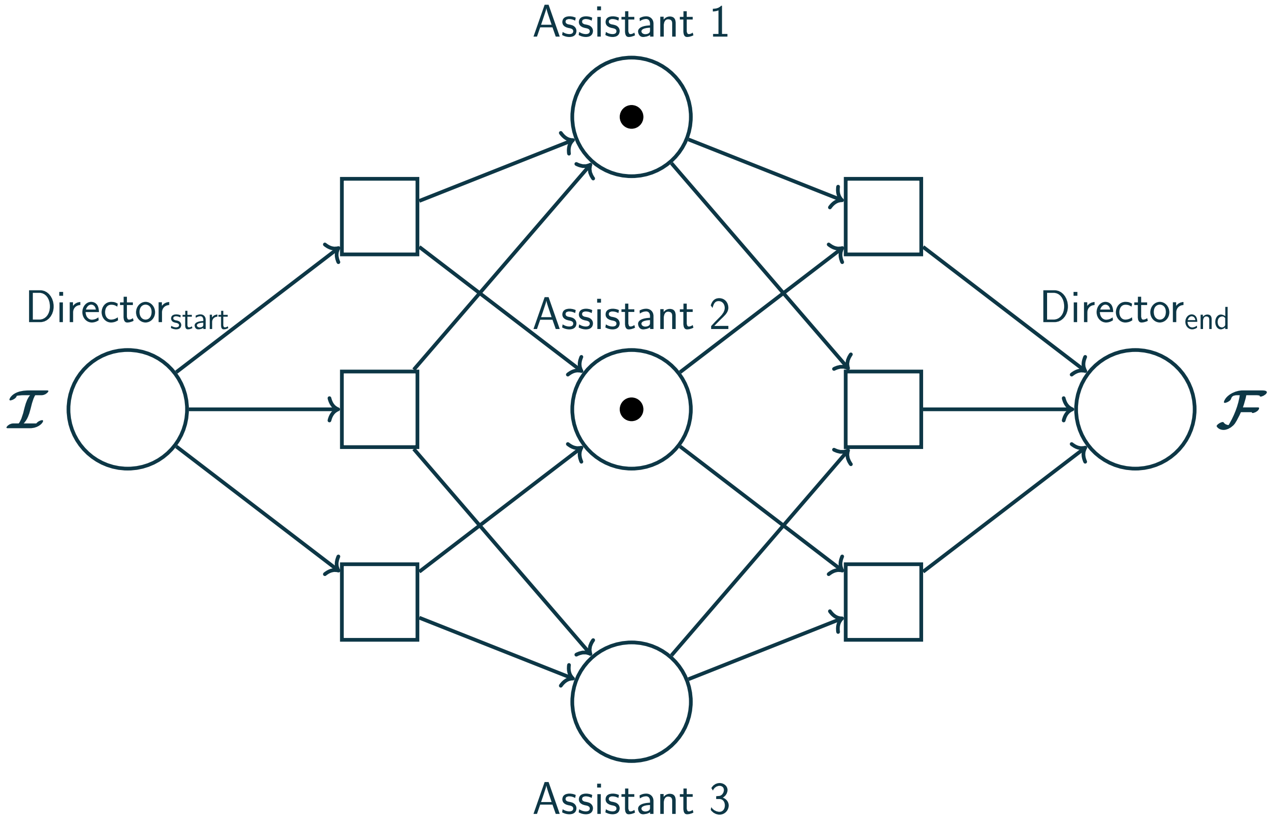
Variants of soundness



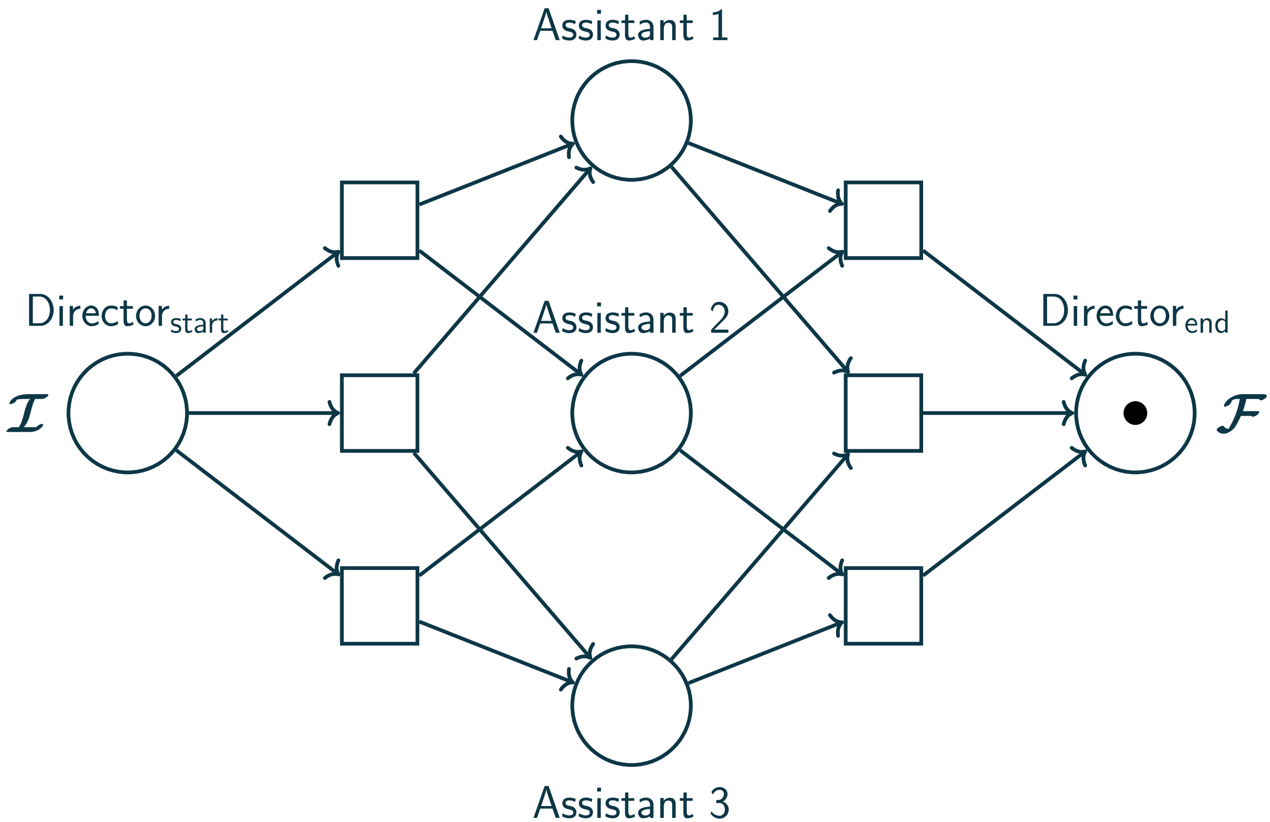
Variants of soundness



Variants of soundness

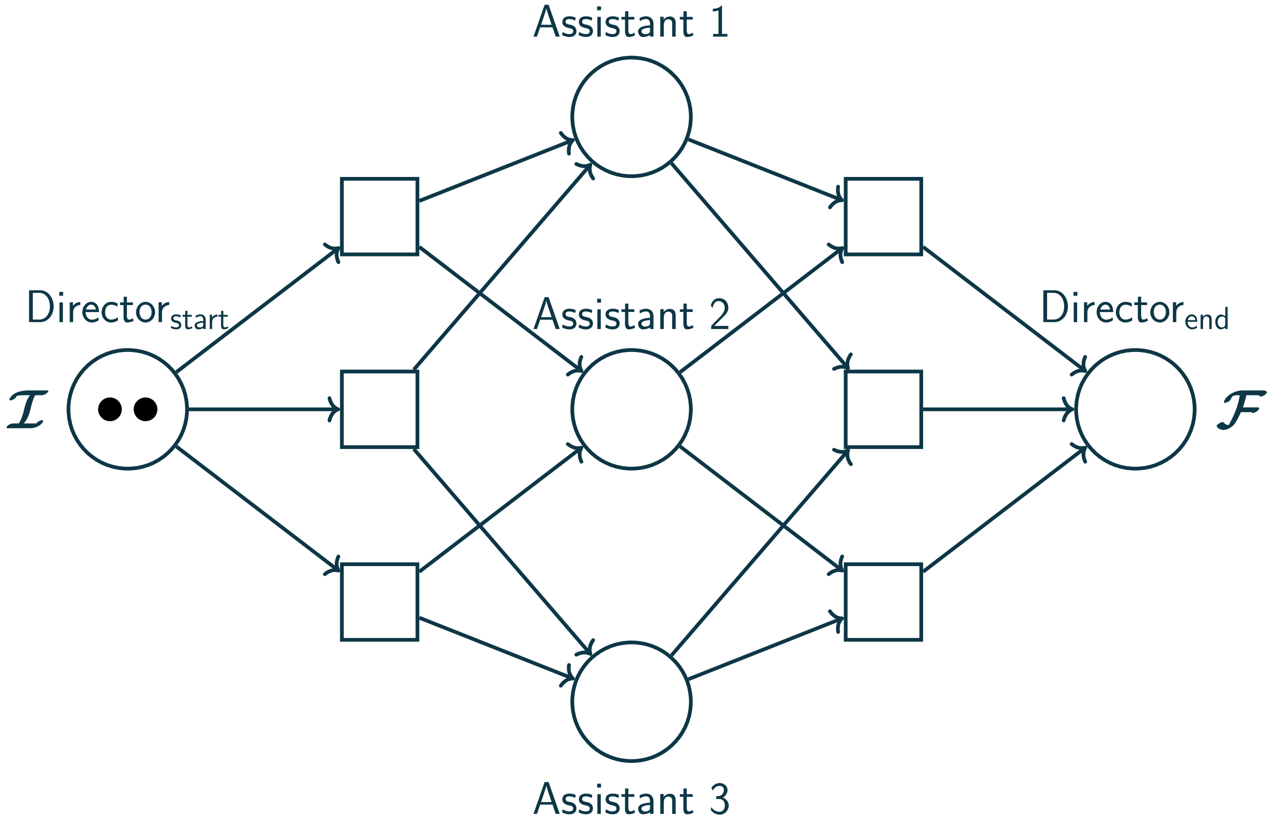


Variants of soundness



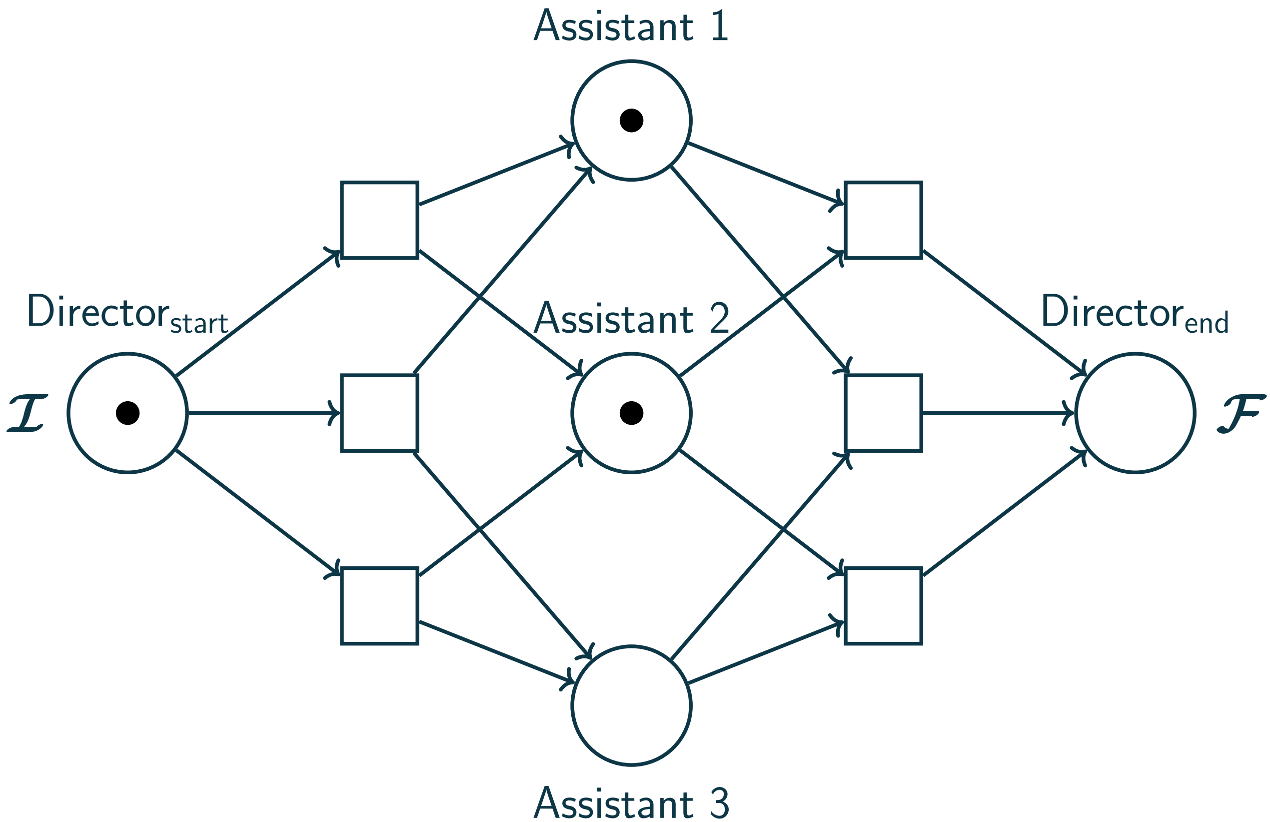
1-sound ✓

Variants of soundness



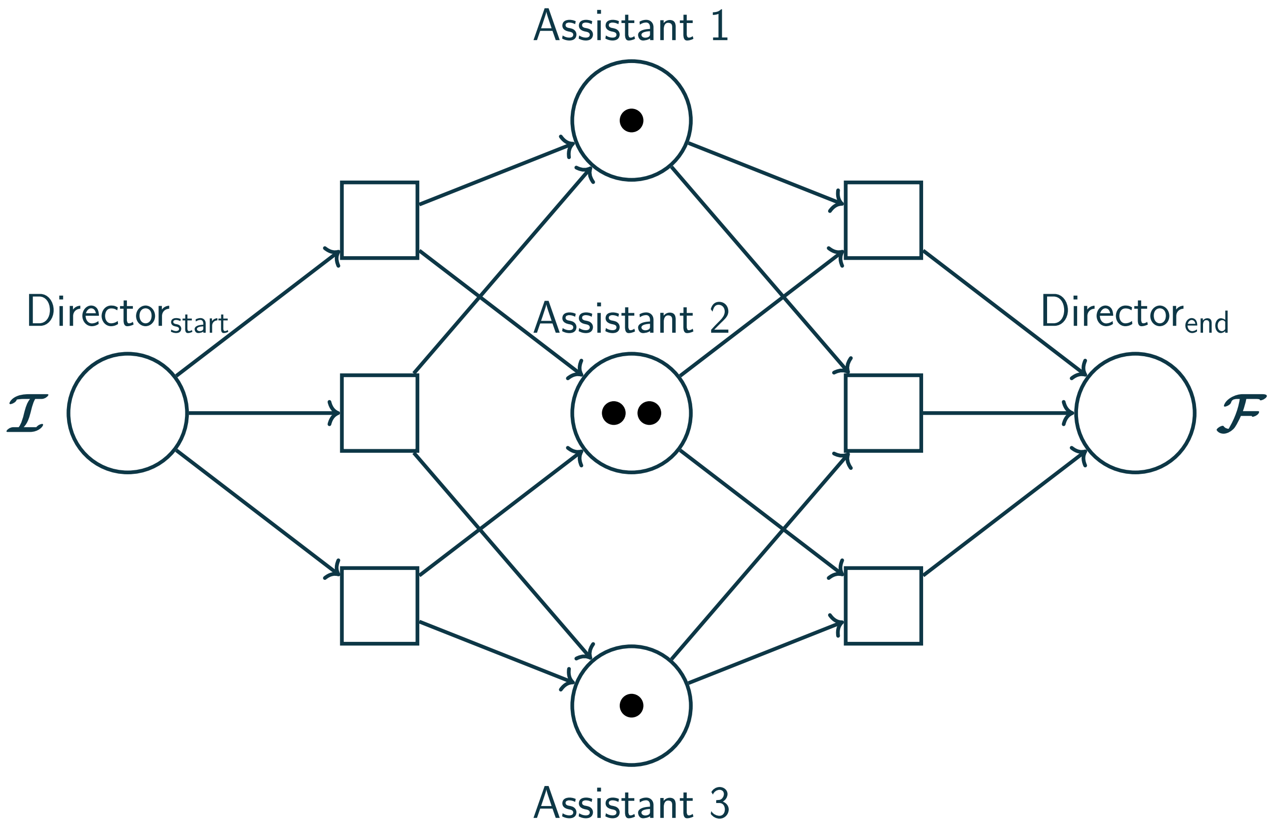
1-sound ✓

Variants of soundness



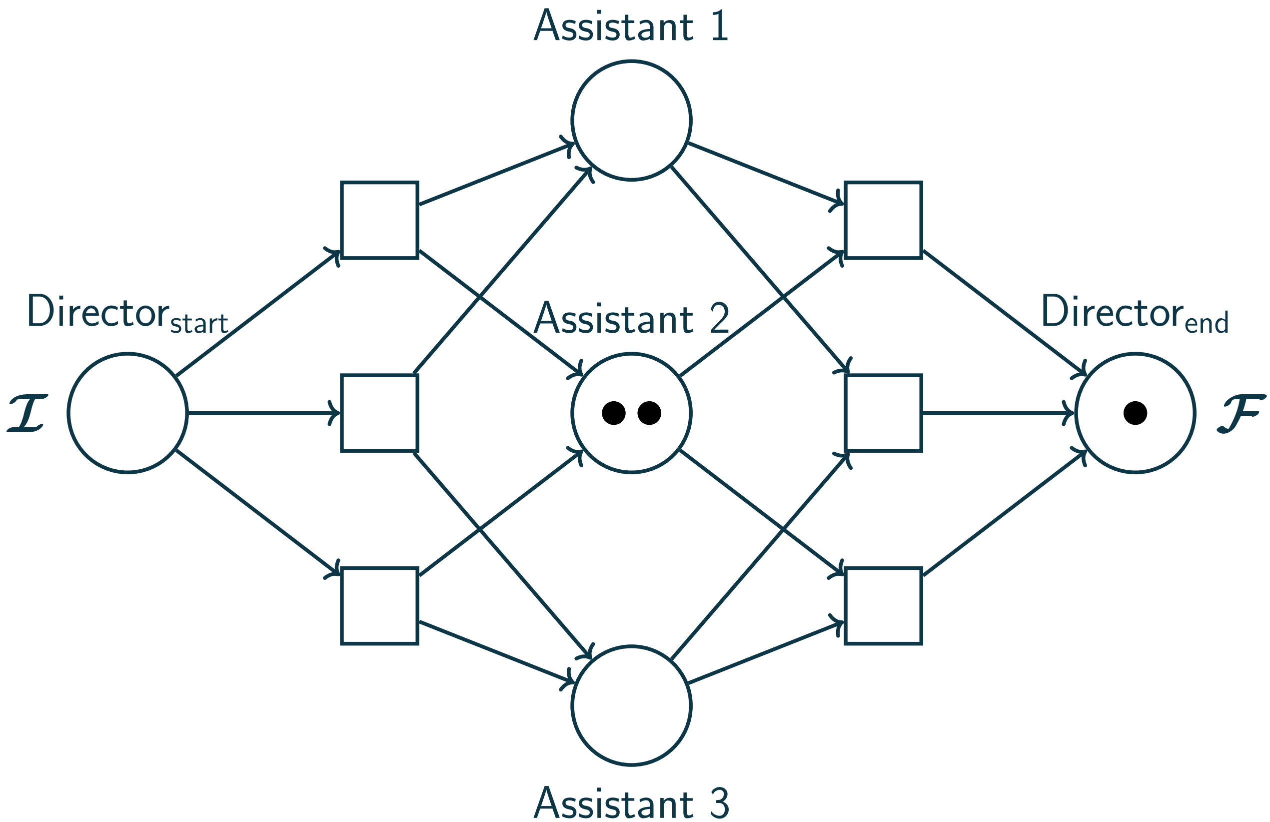
1-sound ✓

Variants of soundness



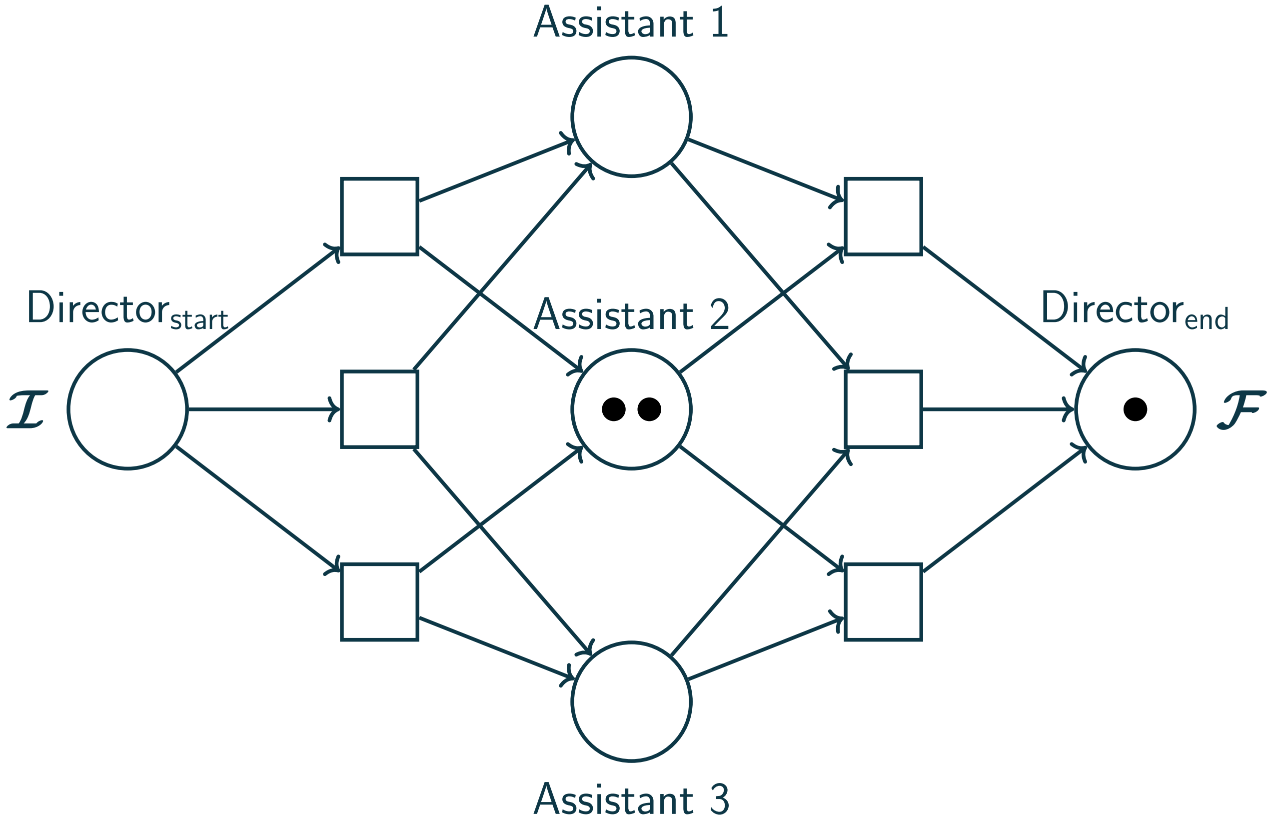
1-sound ✓

Variants of soundness



1-sound ✓

Variants of soundness



1-sound ✓ 2-sound X

Checking soundness - complexity?

	known results	our work
k-Soundness		
Generalised Soundness		
Structural Soundness		

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	
Generalised Soundness		
Structural Soundness		

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	
Generalised Soundness	Decidable [van Hee et al.;'04]	
Structural Soundness		

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	
Generalised Soundness	Decidable [van Hee et al.;'04]	
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	

Checking soundness - complexity?

	known results	our work
k-Soundness	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	

Checking soundness - complexity?

	known results	our work
k-Soundness	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

Exact algorithms are impractical in general; instead:

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*
  co-NP complete necessary condition for generalised soundness

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets*

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets*
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

[LICS '22]

1.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets*
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete

[LICS '22]

1.

2.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets*
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	[LICS '22]
k-**Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness*
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets*
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	[LICS '22] 1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead: [CAV '22]

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets*
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	[LICS '22] **1.**
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	**2.**
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	**3.**

Exact algorithms are impractical in general; instead: [CAV '22]

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
k-**Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	[LICS '22] 1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead: [CAV '22]

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* 5.
  Soundness in Ptime, and all soundness variants are equivalent

N **is** 1**-sound** $\Leftrightarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I} : 1\})$ **is cyclic** $+$ **bounded**

$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1**-sound** $\Leftrightarrow$ **cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

$\boxed{(N_{\mathrm{SC}}, \{\mathcal{I} : 1\})}$ **is**

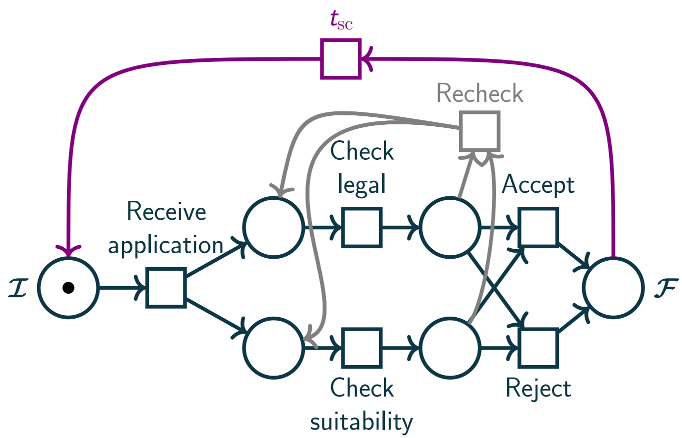
N **is** 1**-sound** $\Leftrightarrow$ **cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F} : 1\}$

$\boxed{(N_{\text{SC}}, \{\mathcal{I}\colon 1\})}$ **is**

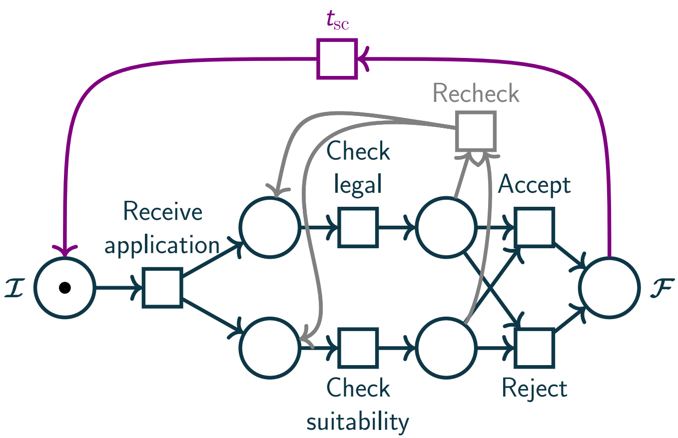
N **is** 1**-sound** $\Leftrightarrow$ **cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$



N is 1-sound $\Leftrightarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I}: 1\})$ **is** $\boxed{\textbf{cyclic}}$ **+ bounded**

Any reachable marking can reach $\{\mathcal{F}: 1\}$

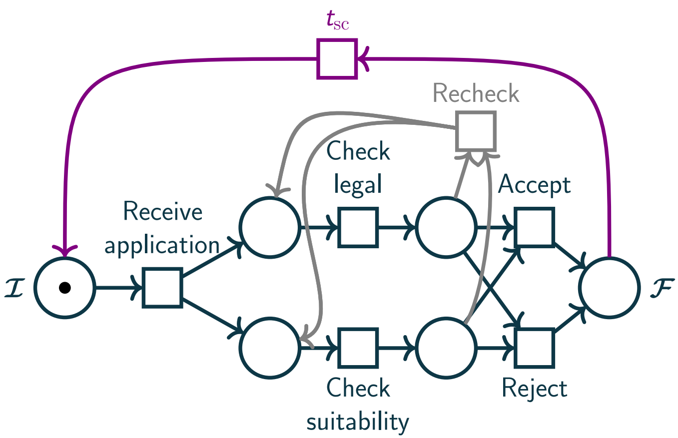


$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1-**sound** $\Leftrightarrow$ $\boxed{\textbf{cyclic}}$ $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

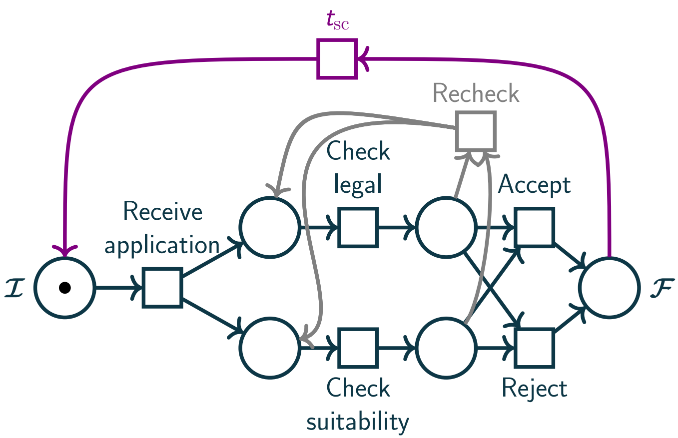


$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1-**sound** $\iff$ **cyclic** + $\boxed{\textbf{bounded}}$

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$



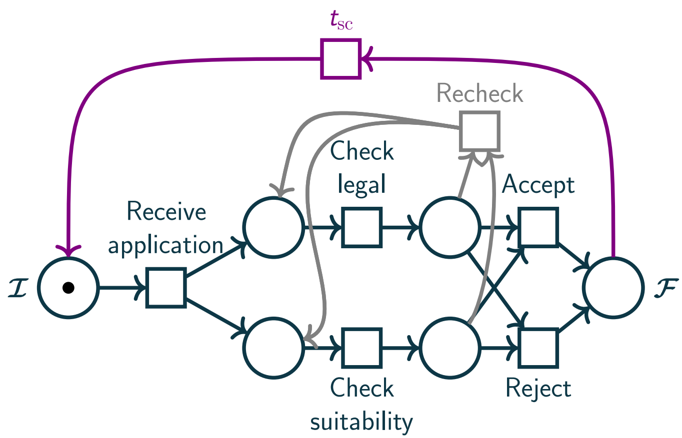
$(N_{\mathrm{SC}}, \{\mathcal{I}: 1\})$ **is**

N **is** 1-**sound** $\Leftrightarrow$ **cyclic** + $\boxed{\textbf{bounded}}$

Any reachable marking can reach $\{\mathcal{F}: 1\}$

Any reachable marking can reach $\{\mathcal{I}: 1\}$

Unbounded: $\{\mathcal{I}: 1\}$ can reach m which can reach m′ with m < m′



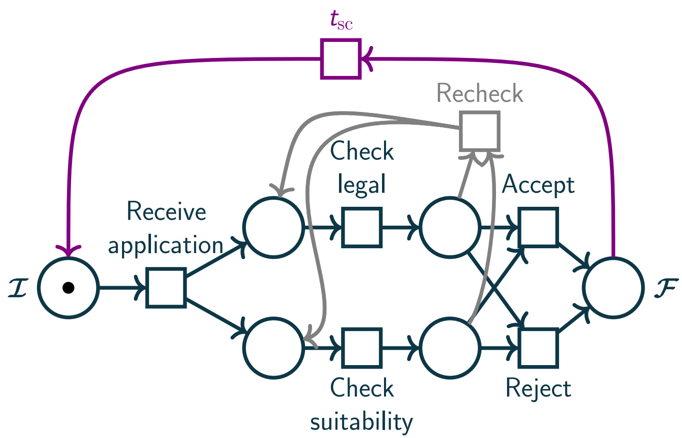
$$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$$ **is**

N **is 1-sound** $\Leftarrow$ **cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with m < m′



$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

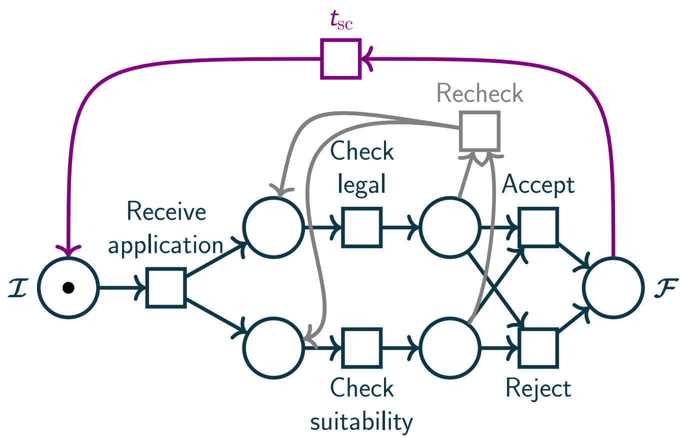
N **is 1-sound** $\Leftarrow$ **cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with m < m′

$\{\mathcal{I}\colon 1\}$ reaches m implies m reaches $\{\mathcal{I}\colon 1\}$



$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1-**sound** $\Leftarrow$ **cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

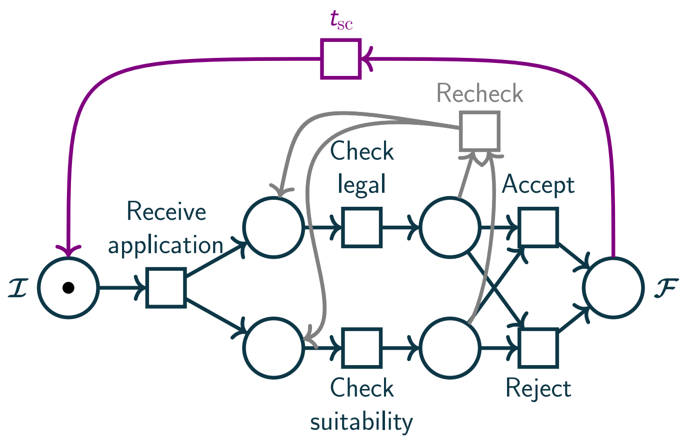
Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with $\mathrm{m} < \mathrm{m}'$

$\{\mathcal{I}\colon 1\}$ reaches m implies m reaches $\{\mathcal{I}\colon 1\}$

$\downarrow$

m reaches m′ which marks $\mathcal{F}$

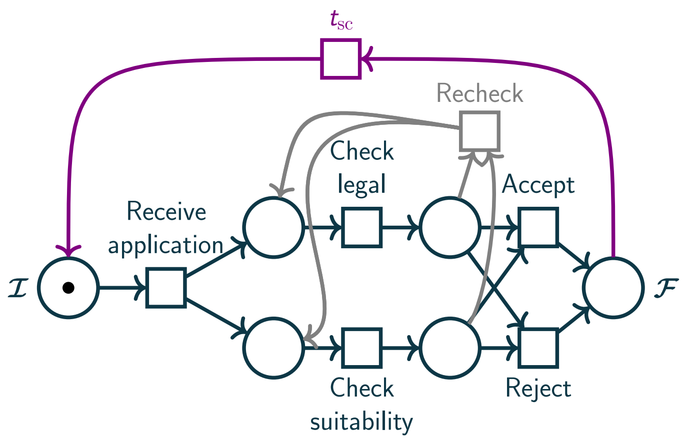


N **is** 1**-sound** $\Leftarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m$'$ with m $<$ m$'$



$\{\mathcal{I}\colon 1\}$ reaches m implies m reaches $\{\mathcal{I}\colon 1\}$

$\downarrow$

m reaches m$'$ which marks $\mathcal{F}$

$\downarrow$

m$'$ $= \{\mathcal{F}\colon 1\}$
otherwise N_{SC} is unbounded

$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1-**sound** $\Leftarrow$ **cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with m < m′

$\{\mathcal{I}\colon 1\}$ reaches m implies m reaches $\{\mathcal{I}\colon 1\}$

↓

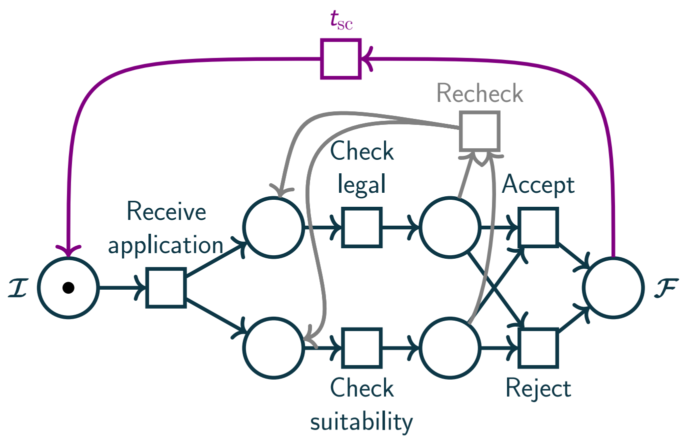
m reaches m′ which marks $\mathcal{F}$

↓

m′ = $\{\mathcal{F}\colon 1\}$
otherwise N_{SC} is unbounded

↓

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

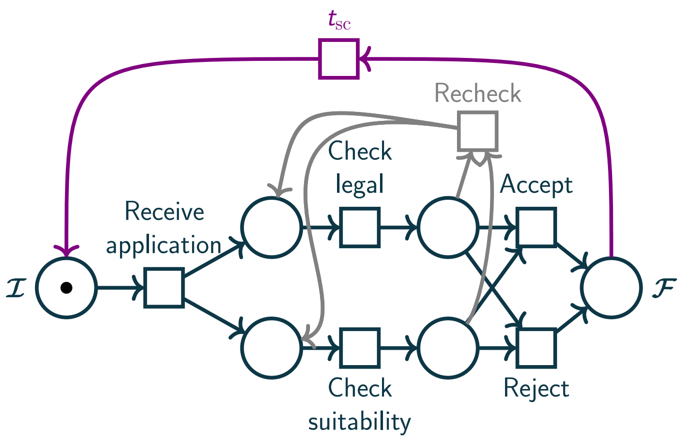


$$N \textbf{ is 1-sound} \Rightarrow (N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\}) \textbf{ is cyclic} + \textbf{bounded}$$

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with $\mathrm{m} < \mathrm{m}'$



$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

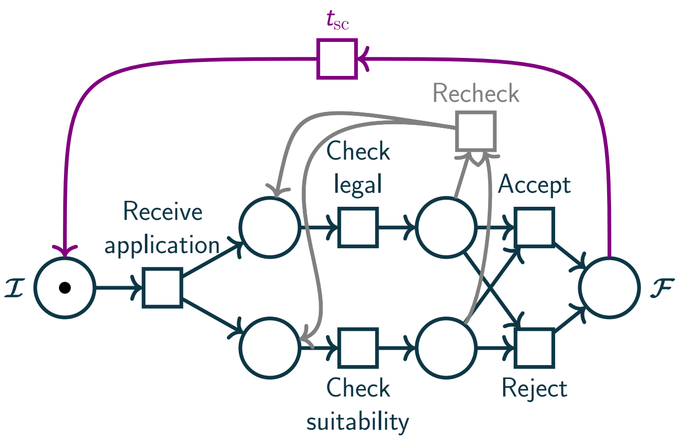
N **is** 1**-sound** $\Rightarrow$ **cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m$'$ with m $<$ m$'$

$\{\mathcal{I}\colon 1\}$ reaches m implies m reaches $\{\mathcal{F}\colon 1\}$



N **is 1-sound** $\Rightarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I}: 1\})$ **is cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}: 1\}$

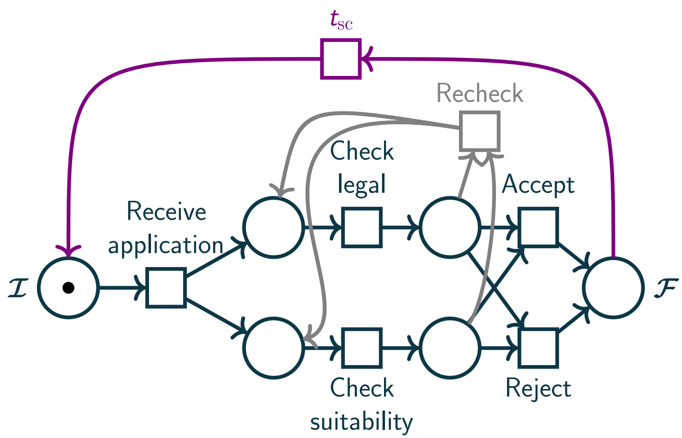
Any reachable marking can reach $\{\mathcal{I}: 1\}$

Unbounded: $\{\mathcal{I}: 1\}$ can reach m which can reach m′ with m < m′

$\{\mathcal{I}: 1\}$ reaches m implies m reaches $\{\mathcal{F}: 1\}$

$\downarrow$

$\{\mathcal{F}: 1\}$ can reach $\{\mathcal{I}: 1\}$

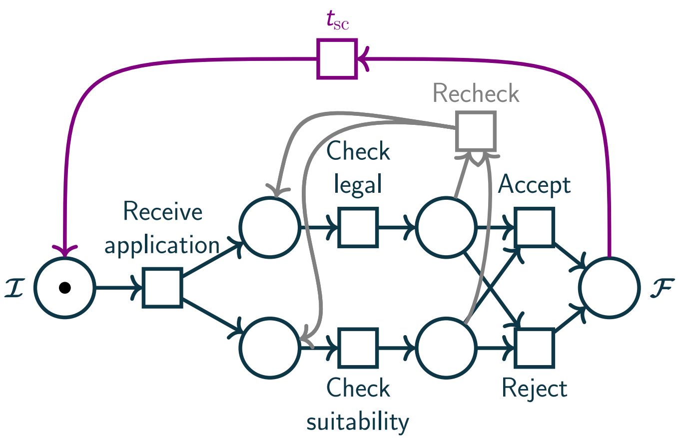


N **is** 1**-sound** $\Rightarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m$'$ with m $<$ m$'$



$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

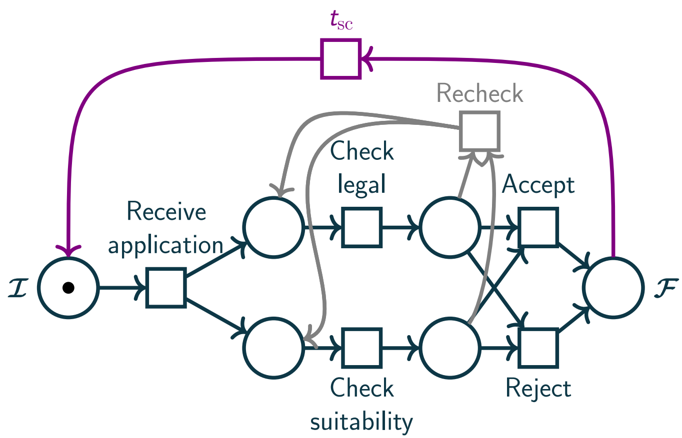
N **is** 1**-sound** $\Rightarrow$ **cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with m $<$ m′

Assume N_{SC} is unbounded but N is 1-sound



$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is 1-sound** $\Rightarrow$ **cyclic** + **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

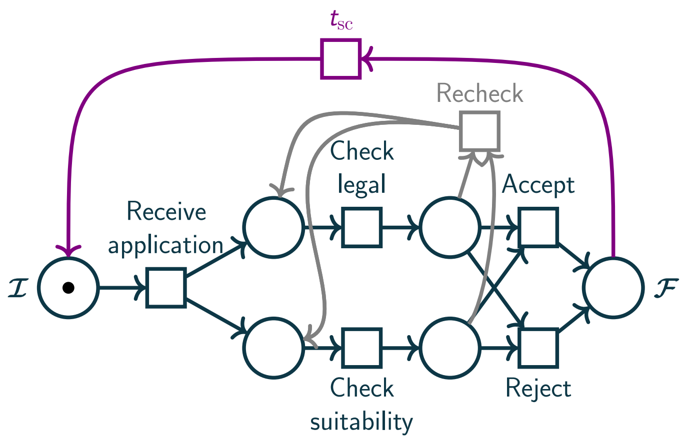
Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m$'$ with m $<$ m$'$

Assume N_{SC} is unbounded but N is 1-sound

$\downarrow$

$\{\mathcal{I}\colon 1\}$ can reach m which can reach m$' >$ m: m$'$ = m + n



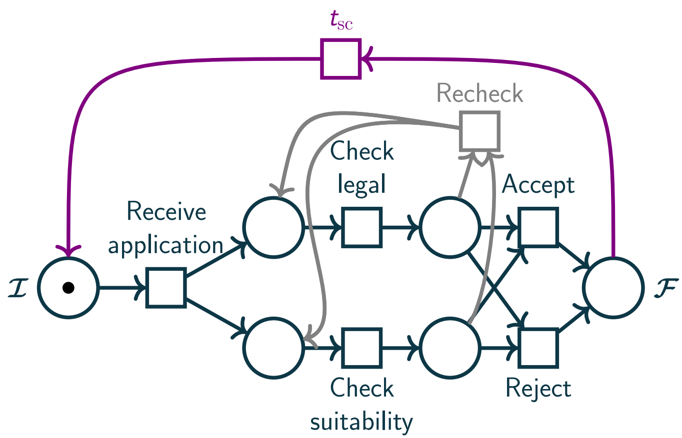
$(N_{\text{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1**-sound** $\Rightarrow$ **cyclic** $+$ **bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m$'$ with m $<$ m$'$



Assume N_{SC} is unbounded but N is 1-sound

$\downarrow$

$\{\mathcal{I}\colon 1\}$ can reach m which can reach m$' >$ m: m$' =$ m $+$ n

$\downarrow$

m can reach $\{\mathcal{F}\colon 1\}$

m $+$ n can reach $\{\mathcal{F}\colon 1\} +$ n

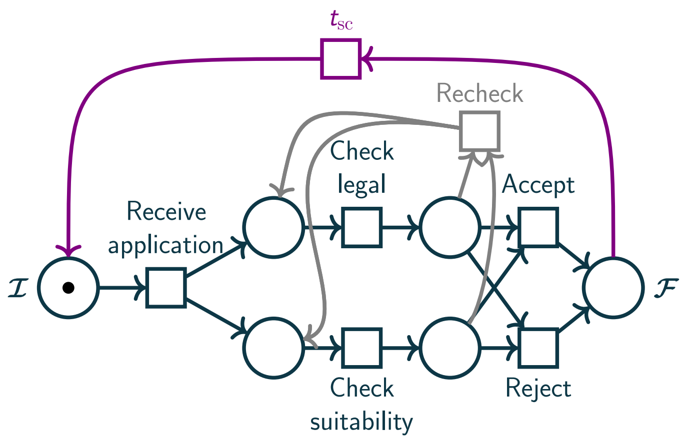
$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1**-sound** $\Rightarrow$ **cyclic + bounded**

Any reachable marking can reach $\{\mathcal{F}\colon 1\}$

Any reachable marking can reach $\{\mathcal{I}\colon 1\}$

Unbounded: $\{\mathcal{I}\colon 1\}$ can reach m which can reach m′ with $\mathrm{m} < \mathrm{m}'$



Assume N_{SC} is unbounded but N is 1-sound

↓

$\{\mathcal{I}\colon 1\}$ can reach m which can reach $\mathrm{m}' > \mathrm{m}$: $\mathrm{m}' = \mathrm{m} + \mathrm{n}$

↓

m can reach $\{\mathcal{F}\colon 1\}$

$\mathrm{m} + \mathrm{n}$ can reach $\{\mathcal{F}\colon 1\} + \mathrm{n}$

↓

$\{\mathcal{F}\colon 1\} + \mathrm{n}$ cannot reach $\{\mathcal{F}\colon 1\}$

$\Rightarrow N$ is not 1-sound!

N is 1-sound $\Leftrightarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ is cyclic + bounded

N **is** 1**-sound** $\Leftrightarrow$ $(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is cyclic** + **bounded**

In EXPSPACE
[Bouziane & Finkel, '97]

$(N_{\text{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1**-sound** $\Leftrightarrow$ **cyclic** + **bounded**

In EXPSPACE [Bouziane & Finkel, '97]

In EXPSPACE [Rackoff, '78]

N **is** 1**-sound** $\Leftrightarrow$ $(N_{\text{SC}}, \{\mathcal{I}\colon 1\})$ **is cyclic** + **bounded**

In EXPSPACE!

In EXPSPACE [Bouziane & Finkel, '97]

In EXPSPACE [Rackoff, '78]

$(N_{\mathrm{SC}}, \{\mathcal{I}\colon 1\})$ **is**

N **is** 1**-sound** $\Leftrightarrow$ **cyclic** + **bounded**

In EXPSPACE!

In EXPSPACE [Bouziane & Finkel, '97]

In EXPSPACE [Rackoff, '78]

EXPSPACE-hardness is by reduction from reachability in **reversible Petri nets**

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* 5.
  Soundness in Ptime, and all soundness variants are equivalent

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} \colon k\} \to m$ implies $m \to \{\mathcal{F} \colon k\}$

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \to m$ implies $m \to \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} \colon k\} \rightarrow m$ implies $m \rightarrow \{\mathcal{F} \colon k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \to m$ implies $m \to \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$
unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$
not generalised sound

Only enumerate small markings: Big marking reachable $\Rightarrow$
not $\mathbb{Z}$-bounded

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \rightarrow m$ implies $m \rightarrow \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded

Algorithm:

- Guess small k
- Check k-soundness: enumerate reachable markings
- If large markings are encountered: not generalised sound

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \to m$ implies $m \to \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound **1.**

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded **2.**

Algorithm:

- Guess small k
- Check k-soundness: enumerate reachable markings
- If large markings are encountered: not generalised sound

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I}: k\} \to m$ implies $m \to \{\mathcal{F}: k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound **1.**

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded **2.**

Algorithm:

- Guess small k
- Check k-soundness: enumerate reachable markings
- If large markings are encountered: not generalised sound

Generalised Soundness requires $\mathbb{Z}$-boundedness

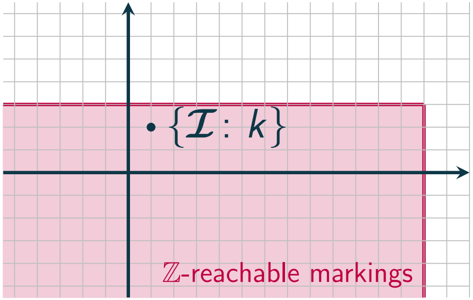
$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}: k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0

Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}\colon k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0

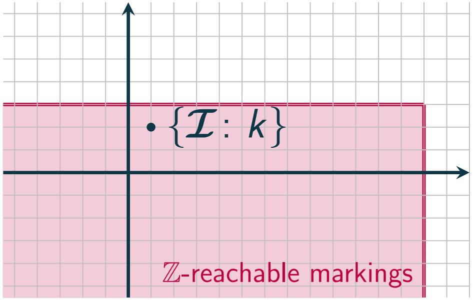


✓ $\mathbb{Z}$-bounded

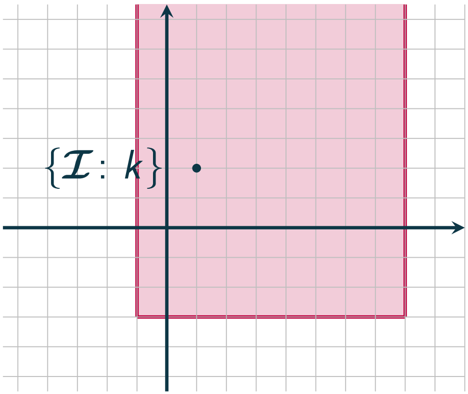
Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}\colon k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0



✓ $\mathbb{Z}$-bounded

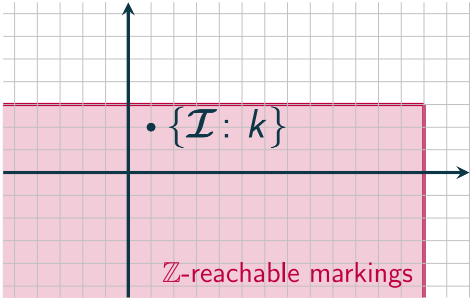


X Not $\mathbb{Z}$-bounded

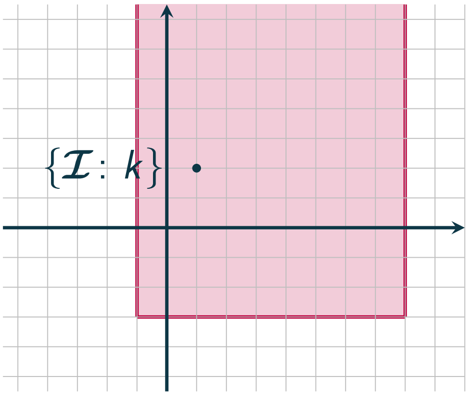
Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}: k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0



✓ $\mathbb{Z}$-bounded



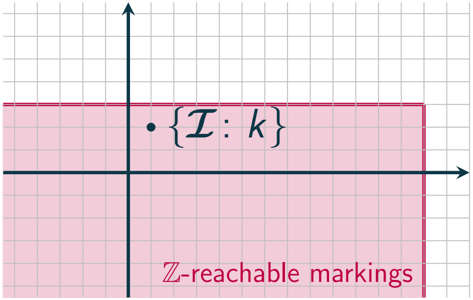
X Not $\mathbb{Z}$-bounded

Why does gen. soundness require $\mathbb{Z}$-boundedness?

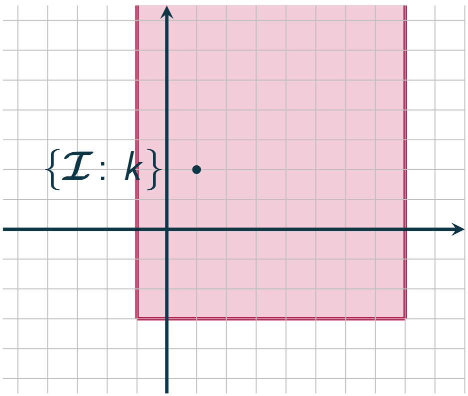
Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}\colon k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0



✓ $\mathbb{Z}$-bounded



X Not $\mathbb{Z}$-bounded

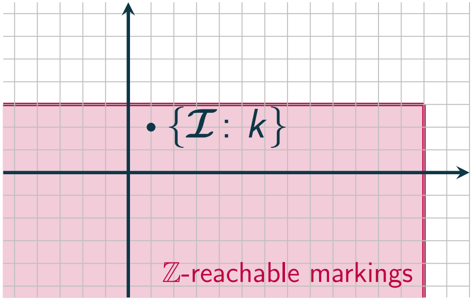
Why does gen. soundness require $\mathbb{Z}$-boundedness?

Recall: k-soundness requires boundedness from $\{\mathcal{I}\colon k\}$

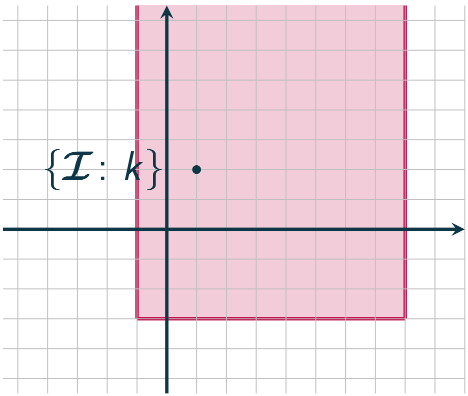
Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-boundedness: $\forall k \exists \vec{b}$: $\{\mathcal{I} : k\} \to_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\to_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0



✓ $\mathbb{Z}$-bounded



X Not $\mathbb{Z}$-bounded

Why does gen. soundness require $\mathbb{Z}$-boundedness?

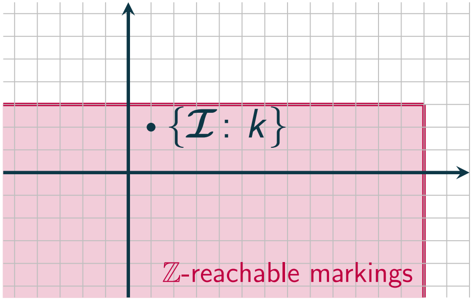
Recall: k-soundness requires boundedness from $\{\mathcal{I} : k\}$

$\Rightarrow$ Generalised soundness requires boundedness for all k

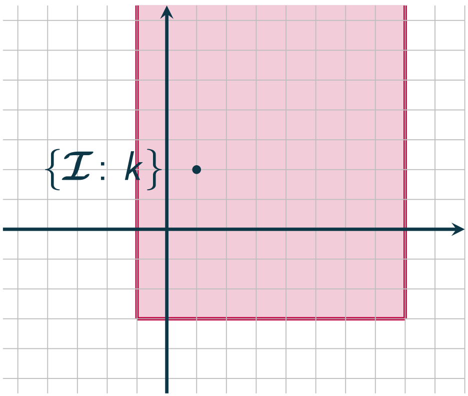
Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}: k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0



✓ $\mathbb{Z}$-bounded



X Not $\mathbb{Z}$-bounded

Why does gen. soundness require $\mathbb{Z}$-boundedness?

Recall: k-soundness requires boundedness from $\{\mathcal{I}: k\}$

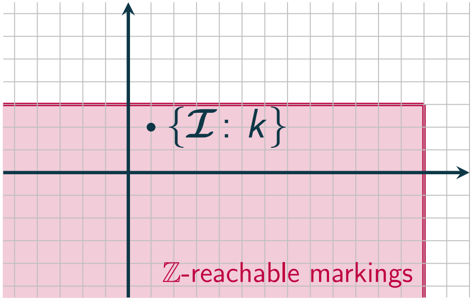
$\Rightarrow$ Generalised soundness requires boundedness for all k

If a net is $\mathbb{Z}$-unbounded, then for some k it is unbounded over $\mathbb{N}$

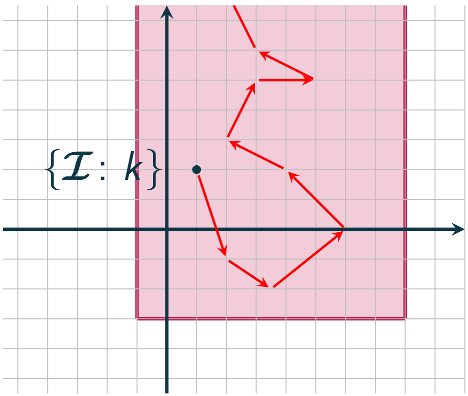
Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-boundedness: $\forall k \exists \vec{b}$: $\{\mathcal{I}: k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0



✓ $\mathbb{Z}$-bounded



X Not $\mathbb{Z}$-bounded

Why does gen. soundness require $\mathbb{Z}$-boundedness?

Recall: k-soundness requires boundedness from $\{\mathcal{I}: k\}$

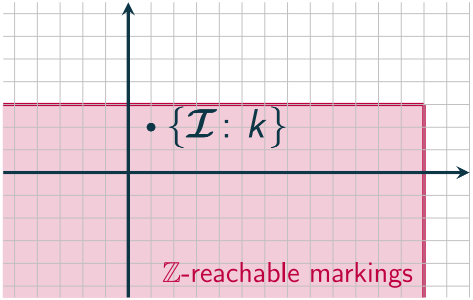
$\Rightarrow$ Generalised soundness requires boundedness for all k

If a net is $\mathbb{Z}$-unbounded, then for some k it is unbounded over $\mathbb{N}$

Generalised Soundness requires $\mathbb{Z}$-boundedness

$\mathbb{Z}$-**boundedness**: $\forall k \exists \vec{b}$: $\{\mathcal{I}\colon k\} \rightarrow_{\mathbb{Z}} m > 0$ implies $m \leq \vec{b}$

$\rightarrow_{\mathbb{Z}}$: $\mathbb{Z}$-reachability – may drop below 0

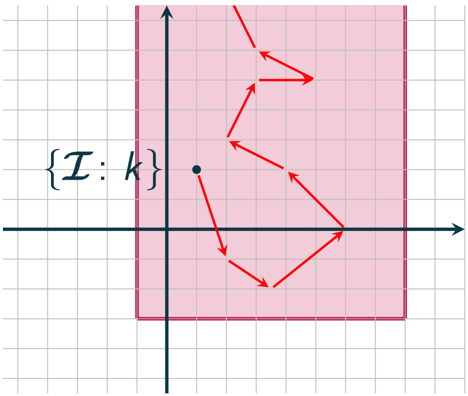


✓ $\mathbb{Z}$-bounded

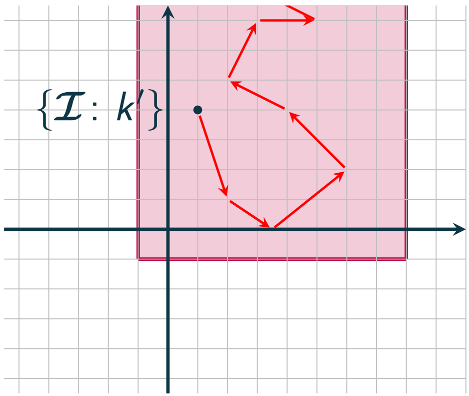
Why does gen. soundness require $\mathbb{Z}$-boundedness?

Recall: k-soundness requires boundedness from $\{\mathcal{I}\colon k\}$

$\Rightarrow$ Generalised soundness requires boundedness for all k



X Not $\mathbb{Z}$-bounded



X Not bounded from k'

If a net is $\mathbb{Z}$-unbounded, then for some k it is unbounded over $\mathbb{N}$

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} \colon k\} \to m$ implies $m \to \{\mathcal{F} \colon k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound **1.**

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded **2.**

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \rightarrow m$ implies $m \rightarrow \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound **1.**

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded **2.**

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \rightarrow m$ implies $m \rightarrow \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound **1.**

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded **2.**

Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$

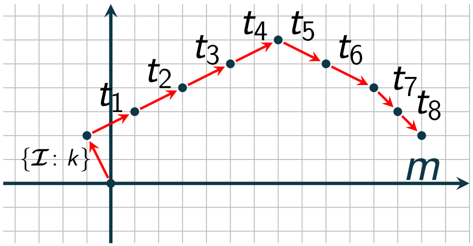
Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$ Big markings must be reached by long runs

Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$

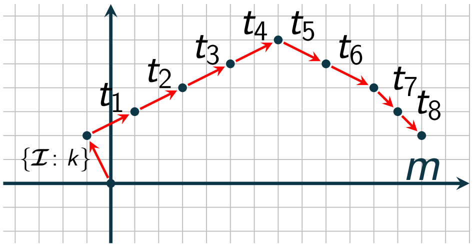
Big markings must be reached by long runs



Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$

Big markings must be reached by long runs

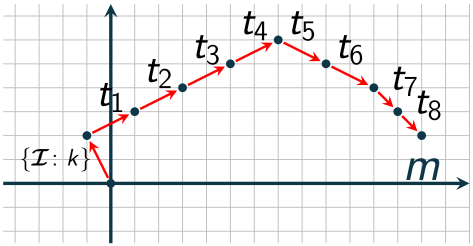


Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m

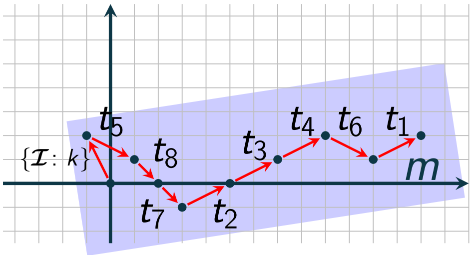
Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$

Big markings must be reached by long runs



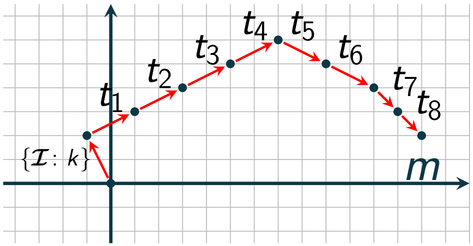
Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m



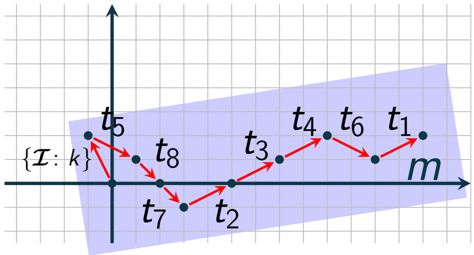
Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I} \colon k\} \xrightarrow{\text{very large}} m$

Big markings must be reached by long runs



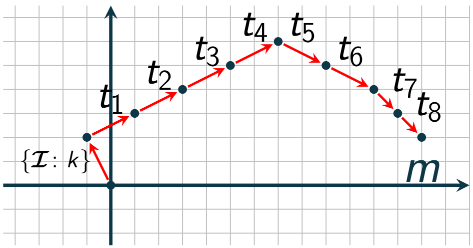
Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m



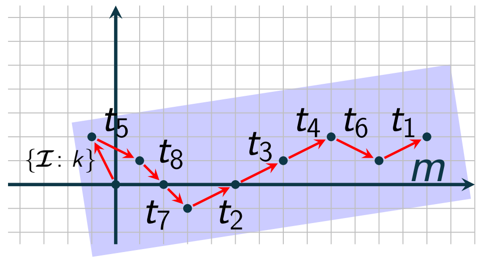
Long runs $\Rightarrow$ Many vectors $\Rightarrow$ Many points

Big reachable markings imply $\mathbb{Z}$-unboundedness

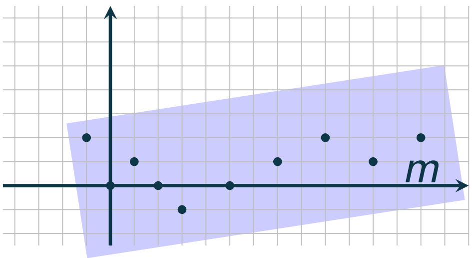
$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$ Big markings must be reached by long runs



Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m



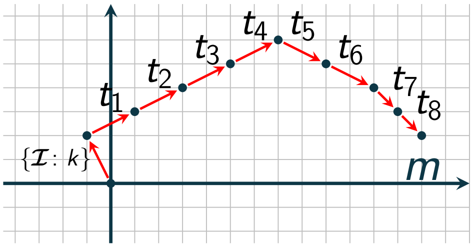
Long runs $\Rightarrow$ Many vectors $\Rightarrow$ Many points



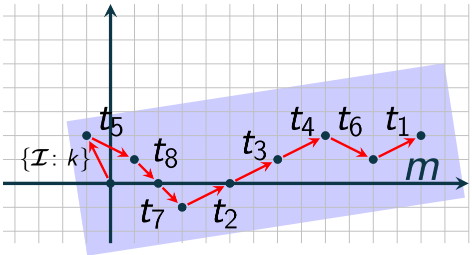
Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$

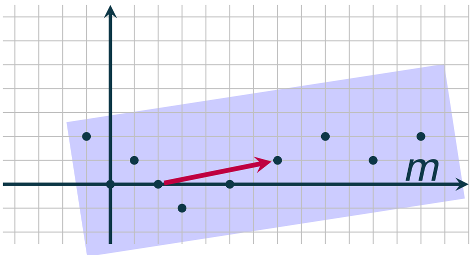
Big markings must be reached by long runs



Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m



Long runs $\Rightarrow$ Many vectors $\Rightarrow$ Many points

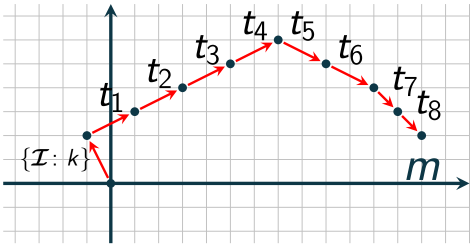


Enough points $\xRightarrow{\text{Pigeonhole}}$ Strict increases $\Rightarrow$ $\mathbb{Z}$-unboundedness

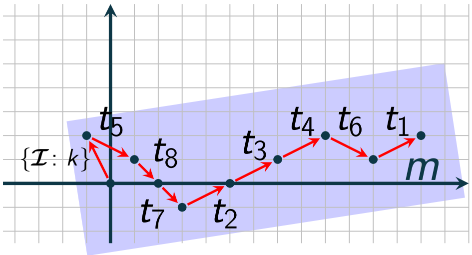
Big reachable markings imply $\mathbb{Z}$-unboundedness

$\{\mathcal{I} : k\} \xrightarrow{\text{very large}} m$

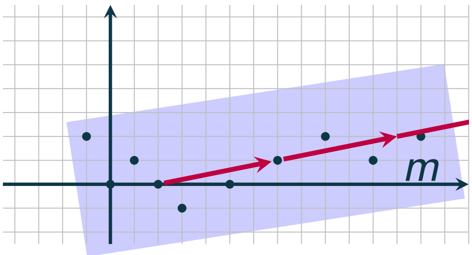
Big markings must be reached by long runs



Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m



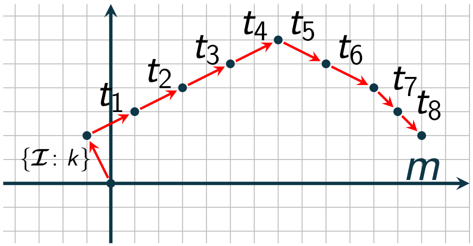
Long runs $\Rightarrow$ Many vectors $\Rightarrow$ Many points



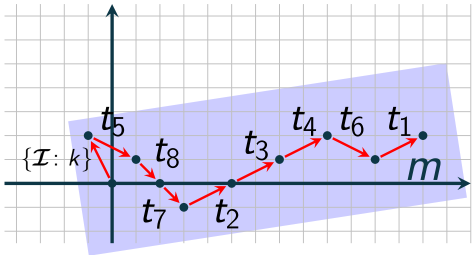
Enough points $\xRightarrow{\textit{Pigeonhole}}$ Strict increases $\Rightarrow$ $\mathbb{Z}$-unboundedness

Big reachable markings imply $\mathbb{Z}$-unboundedness

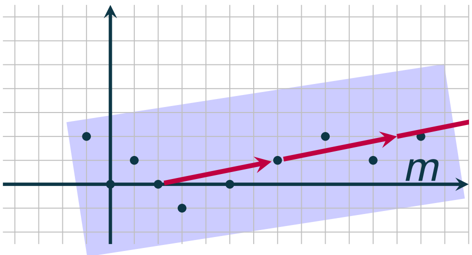
$\{\mathcal{I}\colon k\} \xrightarrow{\text{very large}} m$ Big markings must be reached by long runs



Steinitz Lemma:
Reorder vectors to stay close to straight line from $\vec{0}$ to m



Long runs $\Rightarrow$ Many vectors $\Rightarrow$ Many points



Enough points $\xRightarrow{\textit{Pigeonhole}}$ Strict increases $\Rightarrow$ $\mathbb{Z}$-unboundedness

Big reachable markings imply $\mathbb{Z}$-unboundedness!

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I}: k\} \to m$ implies $m \to \{\mathcal{F}: k\}$

Witness k's are small: Not generalised sound $\Rightarrow$ unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$ not generalised sound **1.**

Only enumerate small markings: Big marking reachable $\Rightarrow$ not $\mathbb{Z}$-bounded **2.**

Algorithm:

- Guess small k
- Check k-soundness: enumerate reachable markings
- If large markings are encountered: not generalised sound

Generalised soundness is in PSPACE

N is **generalised sound:**
$\forall k : \{\mathcal{I} : k\} \to m$ implies $m \to \{\mathcal{F} : k\}$

Witness k's are small: Not generalised sound $\Rightarrow$
unsound for a small k

A helpful necessary condition: Not $\mathbb{Z}$-bounded $\Rightarrow$
not generalised sound

1.

Only enumerate small markings: Big marking reachable $\Rightarrow$
not $\mathbb{Z}$-bounded

2.

Algorithm:

- Guess small k
- Check k-soundness: enumerate reachable markings
- If large markings are encountered: not generalised sound

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Deciding structural soundness in EXPSPACE

Characterize the **set of sound numbers**

Deciding structural soundness in EXPSPACE

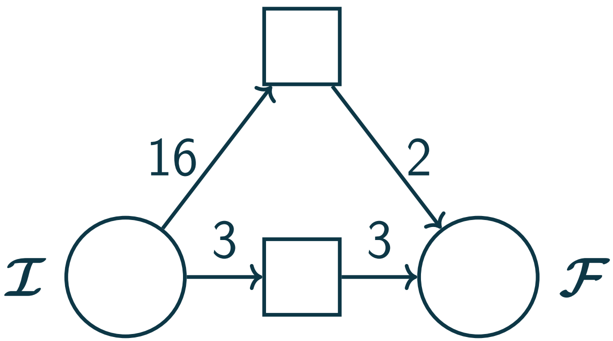
Characterize the **set of sound numbers**

Theorem:
$\text{Sound}_N = \{p, 2p, 3p, \ldots, kp\}$
with $p \in \mathbb{N}, k \in \mathbb{N} \cup \{\infty\}$

Deciding structural soundness in EXPSPACE

Characterize the **set of sound numbers**

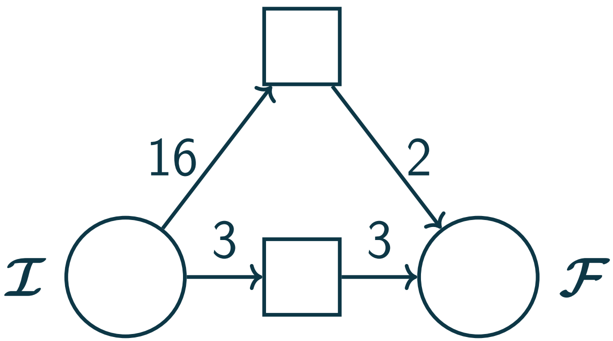
Theorem:
$\text{Sound}_N = \{p, 2p, 3p, \ldots, kp\}$
with $p \in \mathbb{N}, k \in \mathbb{N} \cup \{\infty\}$



Deciding structural soundness in EXPSPACE

Characterize the **set of sound numbers**

Theorem:
$\text{Sound}_N = \{p, 2p, 3p, \ldots, kp\}$
with $p \in \mathbb{N}$, $k \in \mathbb{N} \cup \{\infty\}$

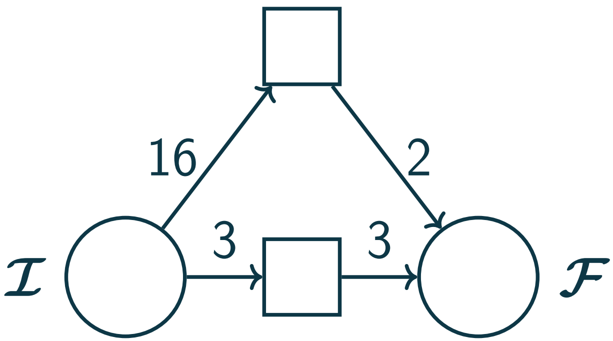


$\{3, 6, 9, 12, 15\}$-sound

Deciding structural soundness in EXPSPACE

Characterize the **set of sound numbers**

Theorem:
$\mathrm{Sound}_N = \{p, 2p, 3p, \ldots, kp\}$
with $p \in \mathbb{N}, k \in \mathbb{N} \cup \{\infty\}$
...and p, k (if $\neq \infty$) are at most exponential in size(N)

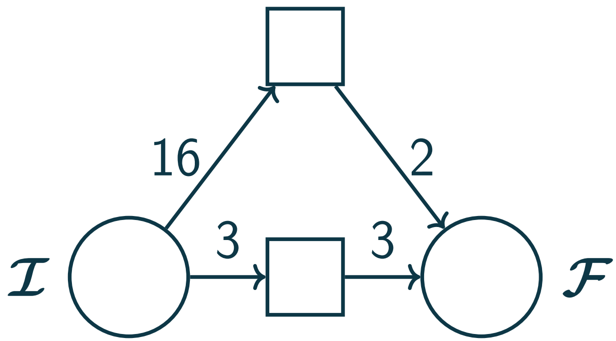


$\{3, 6, 9, 12, 15\}$-sound

Deciding structural soundness in EXPSPACE

Characterize the **set of sound numbers**

Theorem:
$\text{Sound}_N = \{p, 2p, 3p, \ldots, kp\}$
with $p \in \mathbb{N}, k \in \mathbb{N} \cup \{\infty\}$
...and p, k (if $\neq \infty$) are at most exponential in size(N)



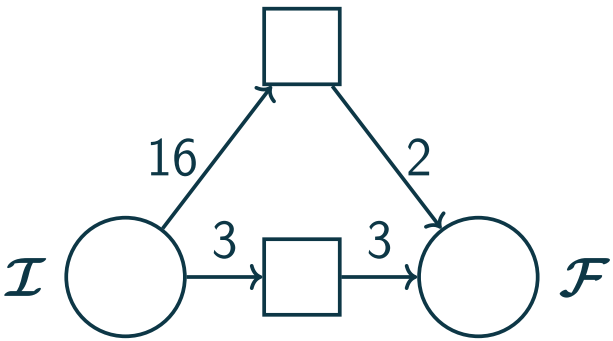
$\{3, 6, 9, 12, 15\}$-sound

EXPSPACE-algorithm for structural soundness:

Deciding structural soundness in EXPSPACE

Characterize the **set of sound numbers**

Theorem:
$\mathrm{Sound}_N = \{p, 2p, 3p, \ldots, kp\}$
with $p \in \mathbb{N}, k \in \mathbb{N} \cup \{\infty\}$
...and p, k (if $\neq \infty$) are at most exponential in size(N)



$\{3, 6, 9, 12, 15\}$-sound

EXPSPACE-algorithm for structural soundness:

Check k-soundness for all "small" k

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* 5.
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

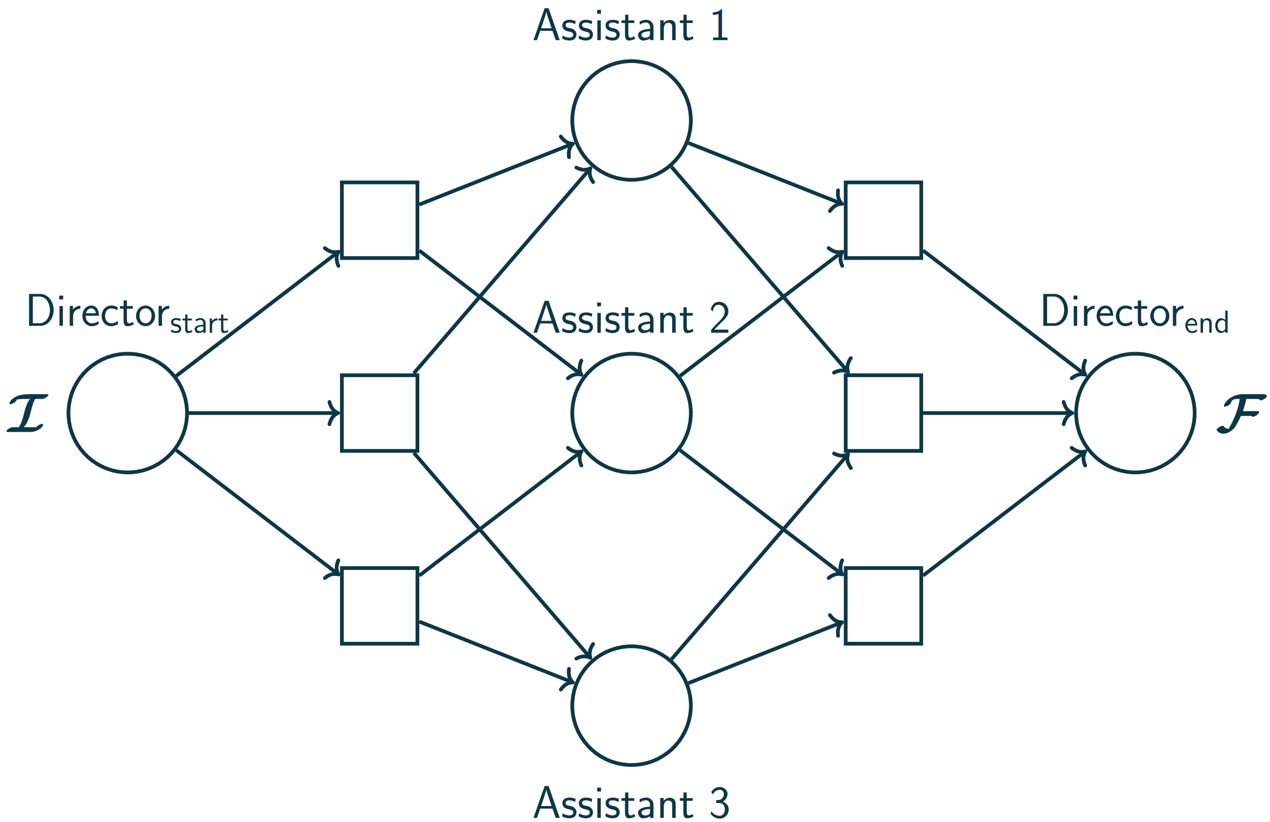
- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Relaxing generalised soundness

Continuous Reachability $\rightarrow_{\mathbb{Q}}$: Allow firing transitions partially

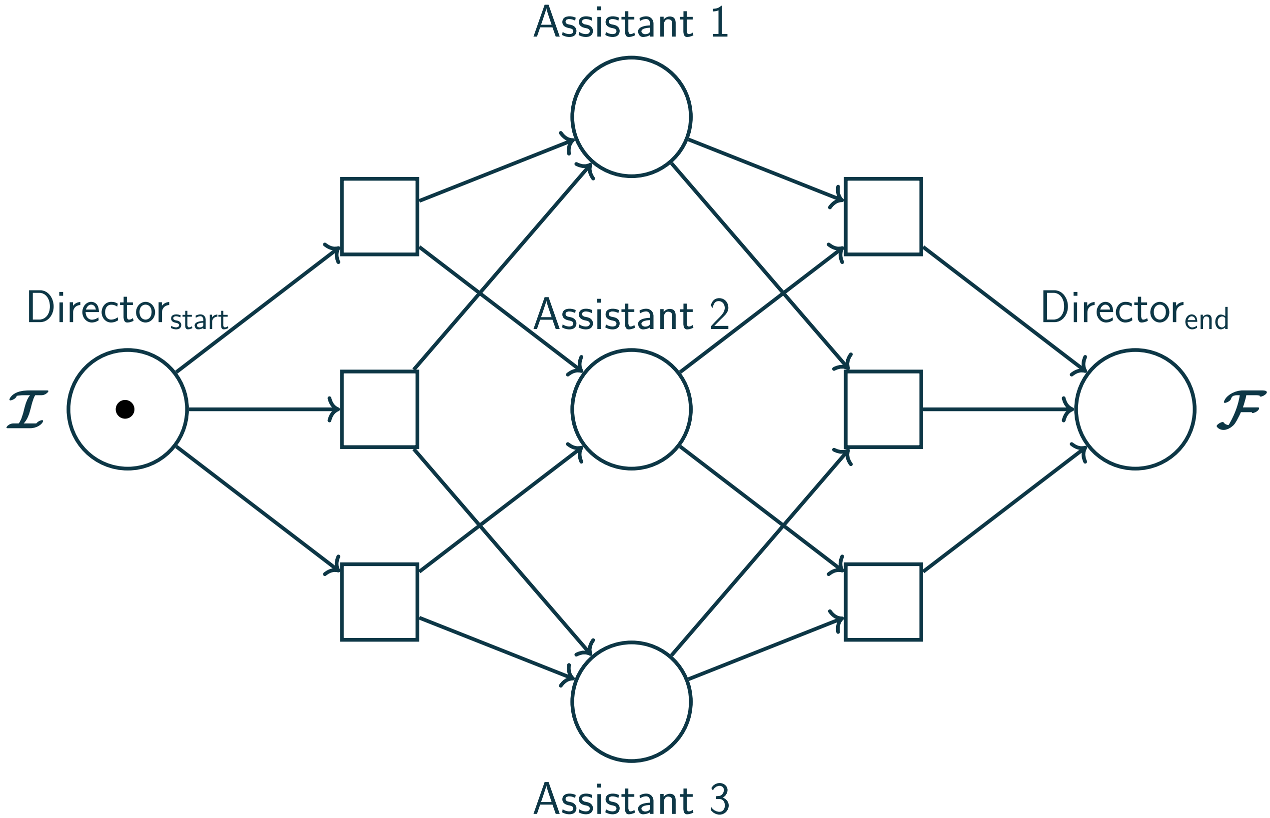
Relaxing generalised soundness

Continuous Reachability $\to_{\mathbb{Q}}$: Allow firing transitions partially



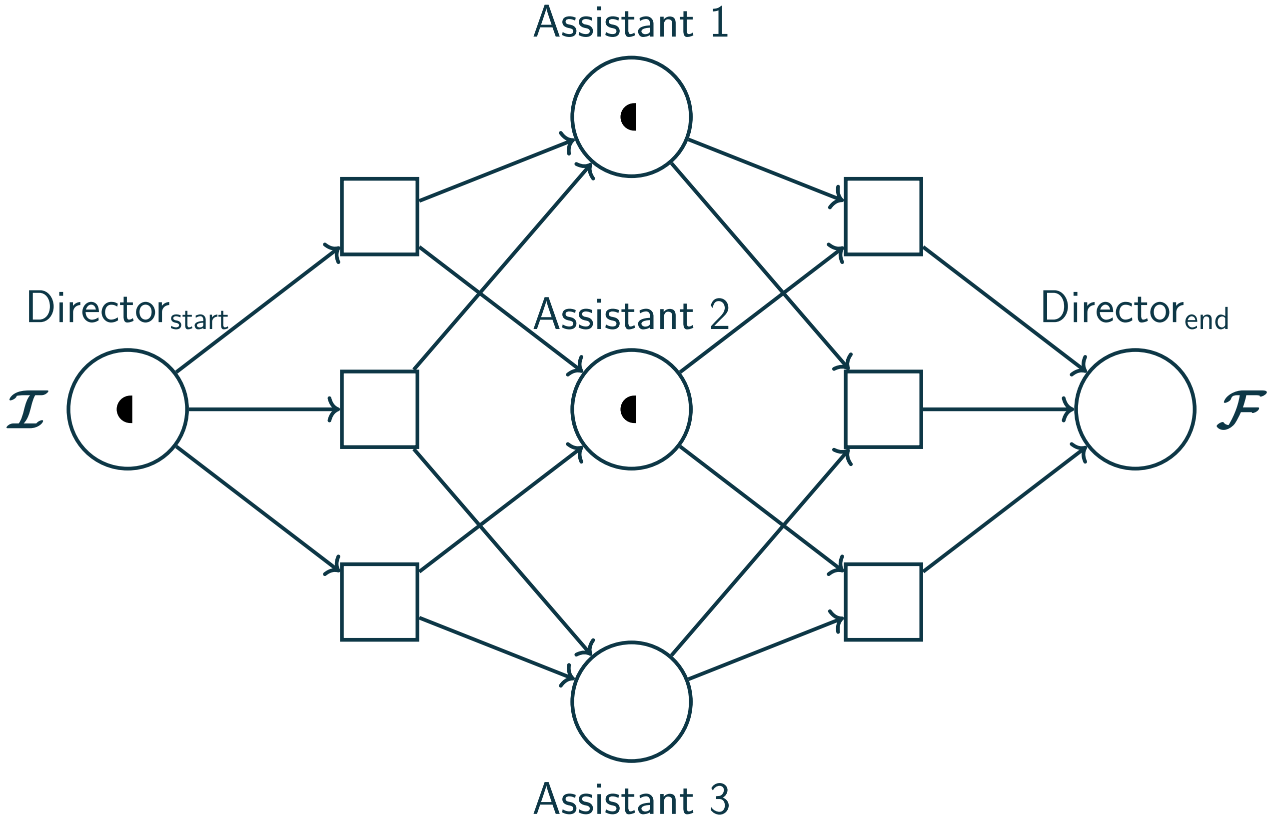
Relaxing generalised soundness

Continuous Reachability $\to_{\mathbb{Q}}$: Allow firing transitions partially



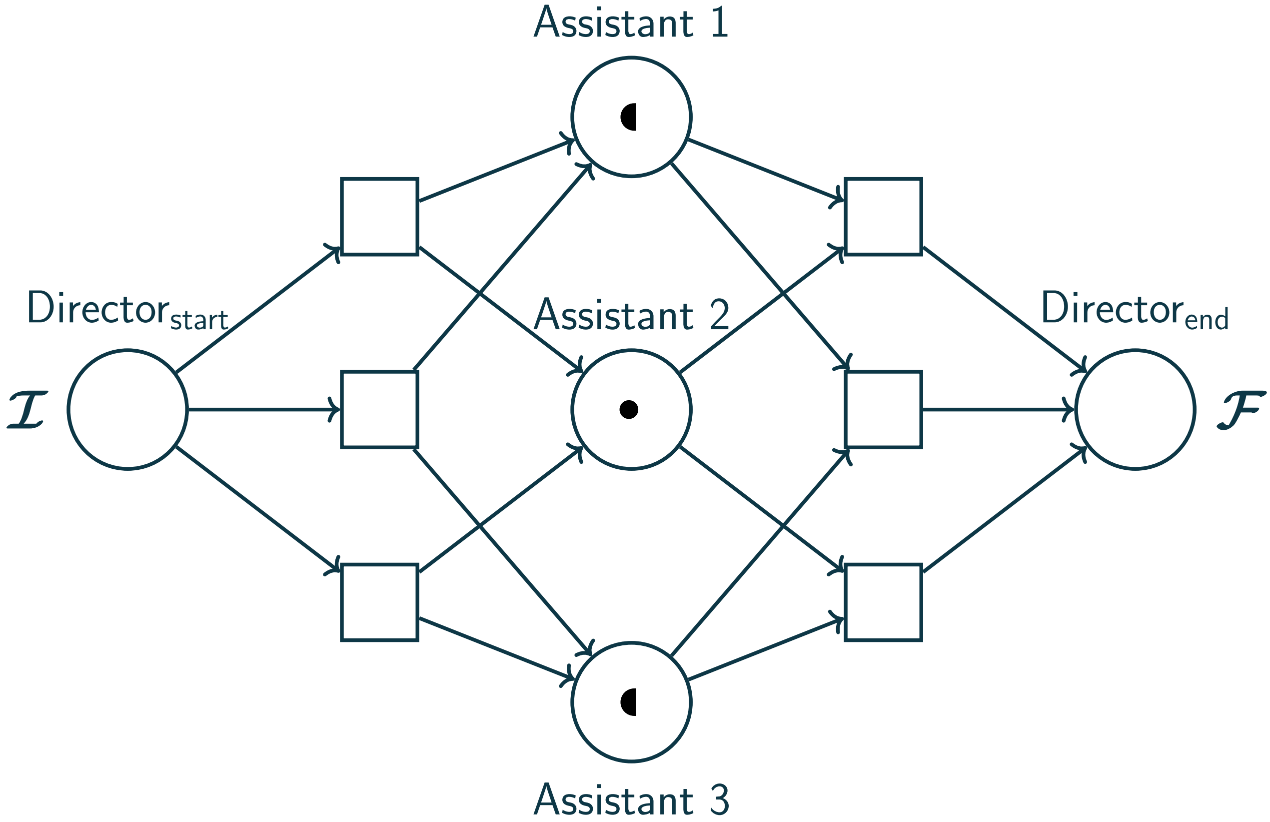
Relaxing generalised soundness

Continuous Reachability $\to_{\mathbb{Q}}$: Allow firing transitions partially



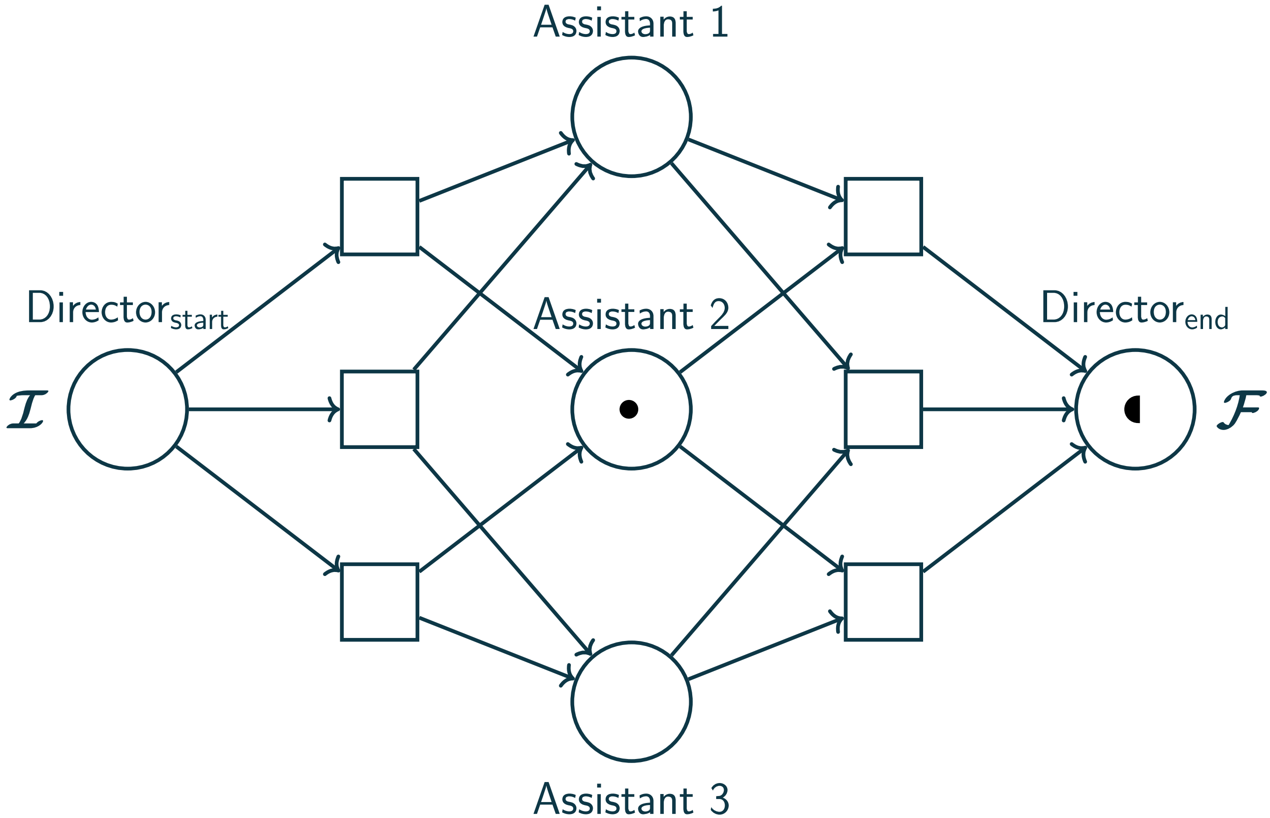
Relaxing generalised soundness

Continuous Reachability $\rightarrow_{\mathbb{Q}}$: Allow firing transitions partially



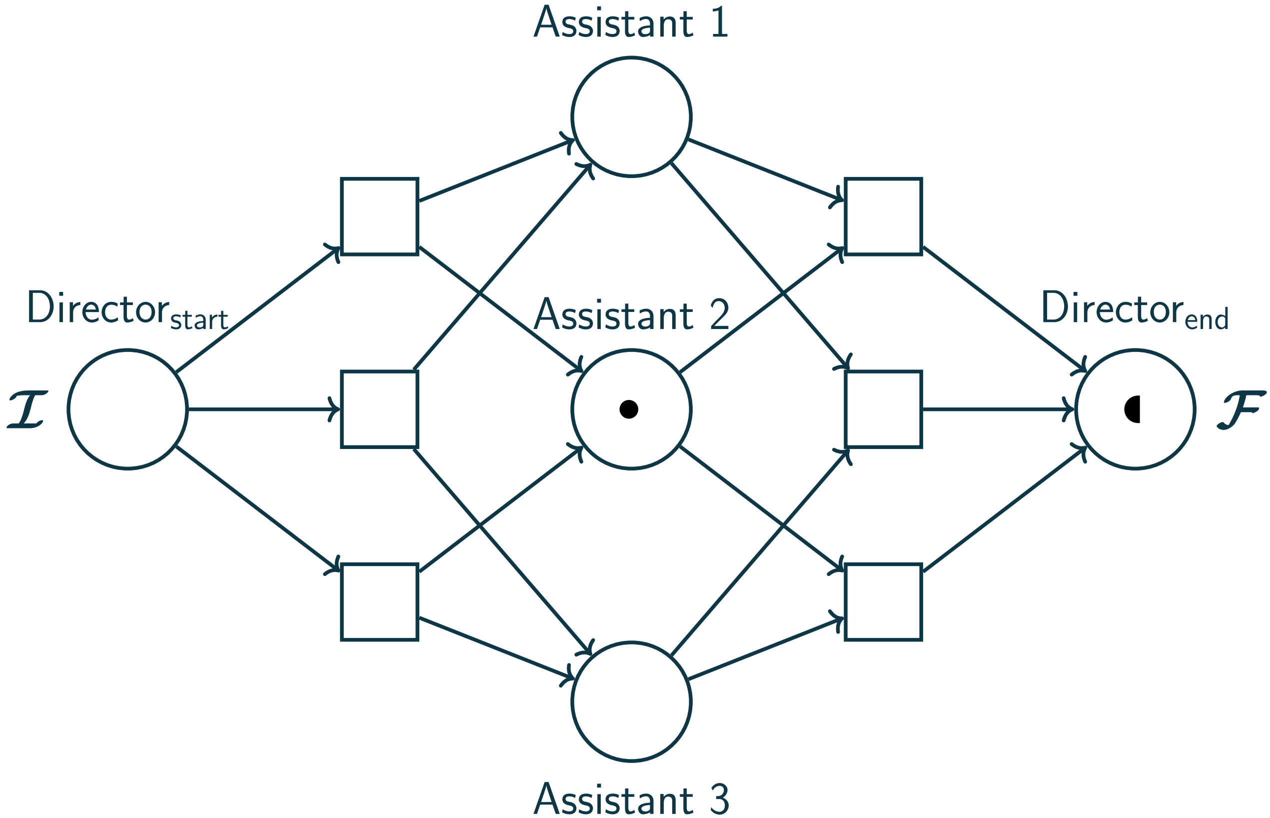
Relaxing generalised soundness

Continuous Reachability $\rightarrow_{\mathbb{Q}}$: Allow firing transitions partially



Relaxing generalised soundness

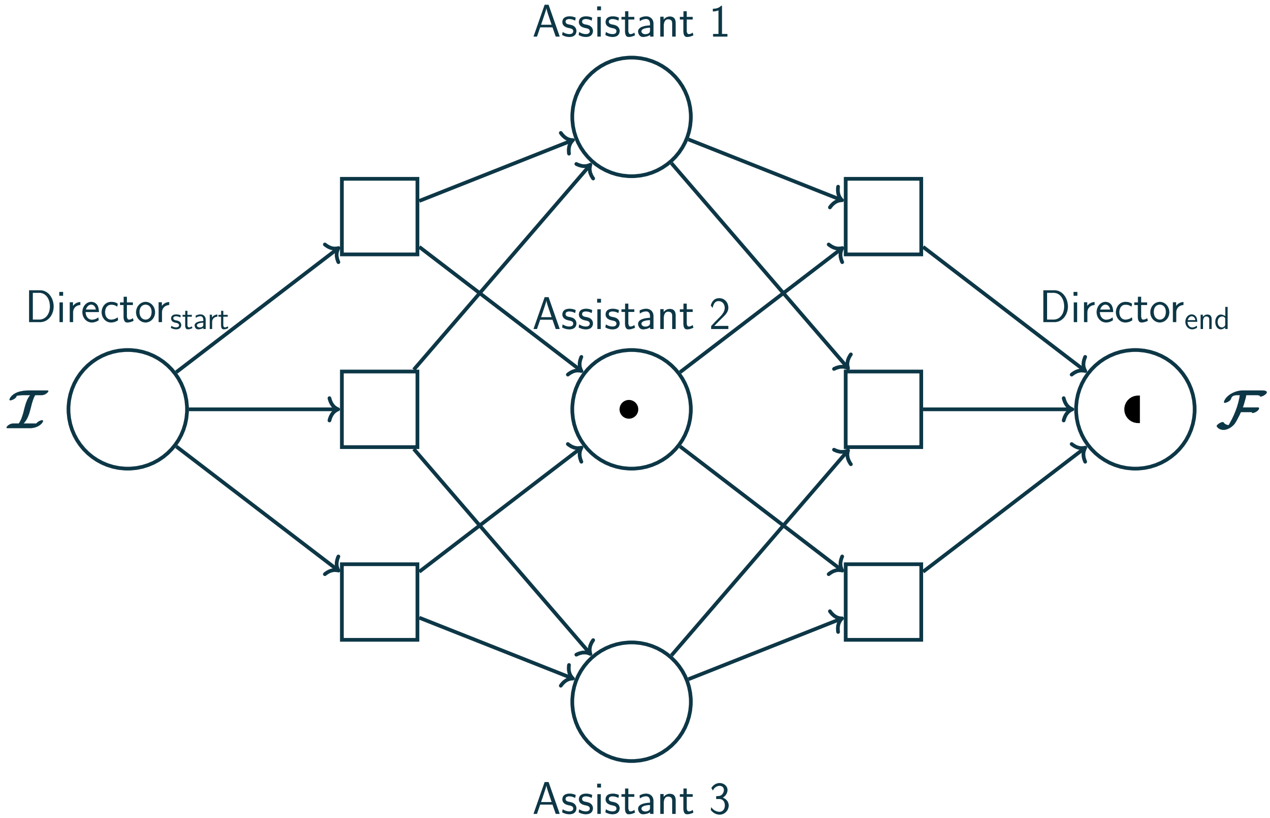
Continuous Reachability $\rightarrow_{\mathbb{Q}}$: Allow firing transitions partially



Continuous reachability approximates standard reachability!

Relaxing generalised soundness

Continuous Reachability $\rightarrow_{\mathbb{Q}}$: Allow firing transitions partially



Continuous reachability approximates standard reachability!

Advantage: Continuous reachability is in Ptime

Relaxing generalised soundness

Reachability:

$\exists \pi : m_s \xrightarrow{\pi} m_t$

Relaxing generalised soundness

Reachability:
$\exists \pi : m_s \xrightarrow{\pi} m_t$

$\Longrightarrow$

Continuous Reachability:
$\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

Relaxing generalised soundness

Reachability:
$\exists \pi : m_s \xrightarrow{\pi} m_t$

$\Longrightarrow$

Continuous Reachability:
$\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

1-**Soundness**:
$\forall \pi \exists \pi' : \{\mathcal{I} : 1\} \xrightarrow{\pi\pi'} \{\mathcal{F} : 1\}$

Relaxing generalised soundness

Reachability:
$$\exists \pi : m_s \xrightarrow{\pi} m_t$$

$\Longrightarrow$

Continuous Reachability:
$$\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$$

1-**Soundness**:
$$\forall \pi \exists \pi' : \{\mathcal{I} \colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F} \colon 1\}$$

Continuous Soundness:
$$\forall \pi \exists \pi' : \{\mathcal{I} \colon 1\} \xrightarrow{\pi\pi'}_{\mathbb{Q}} \{\mathcal{F} \colon 1\}$$

Relaxing generalised soundness

Reachability: $\exists\pi : m_s \xrightarrow{\pi} m_t$ $\Longrightarrow$ **Continuous Reachability**: $\exists\pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

1-**Soundness**: $\forall\pi\exists\pi' : \{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon 1\}$ $\not\Longrightarrow$ **Continuous Soundness**: $\forall\pi\exists\pi' : \{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'}_{\mathbb{Q}} \{\mathcal{F}\colon 1\}$

Relaxing generalised soundness

Reachability:
$\exists \pi : m_s \xrightarrow{\pi} m_t$

$\Longrightarrow$

Continuous Reachability:
$\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

1-**Soundness**:
$\forall \pi \exists \pi' : \{\mathcal{I} \colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F} \colon 1\}$

$\cancel{\Longrightarrow}$

Continuous Soundness:
$\forall \pi \exists \pi' : \{\mathcal{I} \colon 1\} \xrightarrow{\pi\pi'}_{\mathbb{Q}} \{\mathcal{F} \colon 1\}$

$\Longrightarrow$ [CAV'22]

Generalised Soundness:
$\forall k \forall \pi \exists \pi' : \{\mathcal{I} \colon k\} \xrightarrow{\pi\pi'} \{\mathcal{F} \colon k\}$

Relaxing generalised soundness

Reachability:
$\exists \pi : m_s \xrightarrow{\pi} m_t$

$\Longrightarrow$

Continuous Reachability:
$\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

1-**Soundness**:
$\forall \pi \exists \pi' : \{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon 1\}$

$\cancel{\Longrightarrow}$

Continuous Soundness:
$\forall \pi \exists \pi' : \{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'}_{\mathbb{Q}} \{\mathcal{F}\colon 1\}$

$\Longrightarrow$ [CAV'22]

Generalised Soundness:
$\forall k \forall \pi \exists \pi' : \{\mathcal{I}\colon k\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon k\}$

Generalised soundness has a continuous overapproximation

...contrary to many other $\forall\exists$ properties

Relaxing generalised soundness

Reachability:
$\exists \pi : m_s \xrightarrow{\pi} m_t$

$\Longrightarrow$

Continuous Reachability:
$\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

1-Soundness:
$\forall \pi \exists \pi' : \{\mathcal{I} \colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F} \colon 1\}$

$\not\Longrightarrow$

Continuous Soundness:
$\forall \pi \exists \pi' : \{\mathcal{I} \colon 1\} \xrightarrow{\pi\pi'}_{\mathbb{Q}} \{\mathcal{F} \colon 1\}$

$\Longrightarrow$ [CAV'22]

Generalised Soundness:
$\forall k \forall \pi \exists \pi' : \{\mathcal{I} \colon k\} \xrightarrow{\pi\pi'} \{\mathcal{F} \colon k\}$

Generalised soundness has a continuous overapproximation

...contrary to many other $\forall\exists$ properties

Liveness:
$\forall \pi \exists \pi' : m_s \xrightarrow{\pi\pi' t_{\text{live}}}$

$\not\Longrightarrow$

Relaxing generalised soundness

Reachability: $\exists \pi : m_s \xrightarrow{\pi} m_t$ $\quad\Longrightarrow\quad$ **Continuous Reachability**: $\exists \pi : m_s \xrightarrow{\pi}_{\mathbb{Q}} m_t$

1-**Soundness**: $\forall \pi \exists \pi' : \{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon 1\}$ $\quad\cancel{\Longrightarrow}\quad$ **Continuous Soundness**: $\forall \pi \exists \pi' : \{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'}_{\mathbb{Q}} \{\mathcal{F}\colon 1\}$

Generalised Soundness: $\forall k \forall \pi \exists \pi' : \{\mathcal{I}\colon k\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon k\}$ $\quad\Longrightarrow\quad$ **Continuous Soundness** [CAV'22]

Generalised soundness has a continuous overapproximation

...contrary to many other $\forall\exists$ properties

Liveness: $\forall \pi \exists \pi' : m_s \xrightarrow{\pi\pi' t_{\text{live}}}$ $\quad\cancel{\Longrightarrow}$

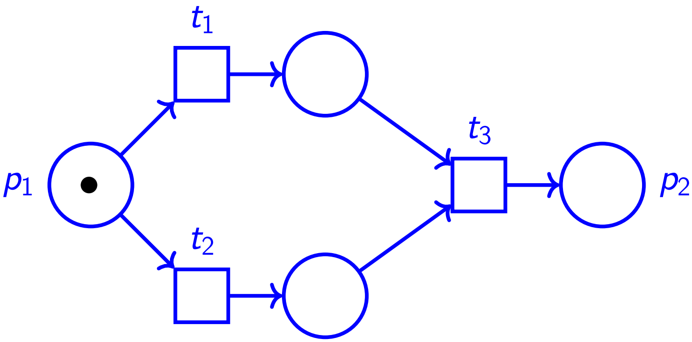
Home State: $\forall \pi \exists \pi' : m_s \xrightarrow{\pi\pi'} m_{\text{home}}$ $\quad\cancel{\Longrightarrow}$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?

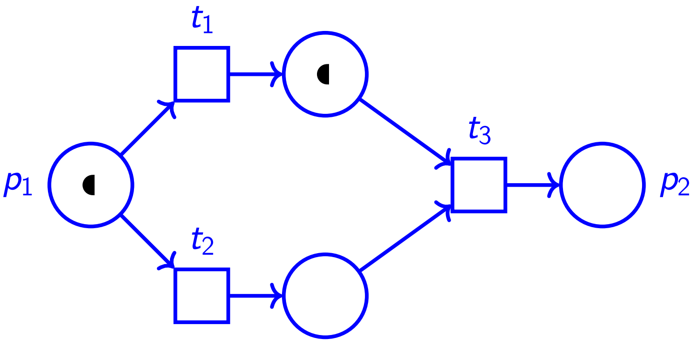
Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?



Generalised & Continuous Soundness

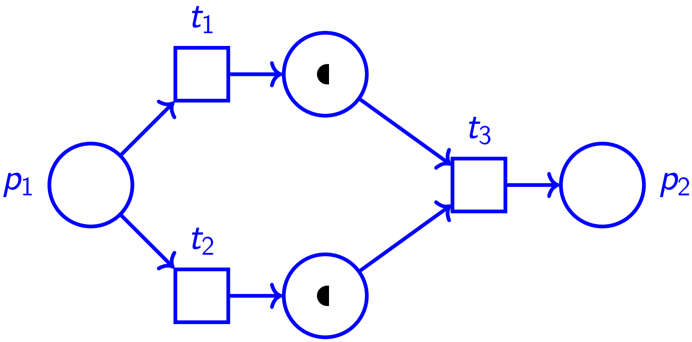
Why does generalised soundness require continuous soundness?



$^1/_2 t_1$

Generalised & Continuous Soundness

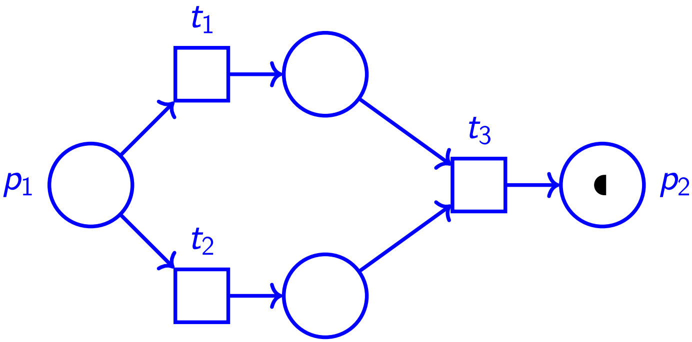
Why does generalised soundness require continuous soundness?



$^1/_2 t_1$ $^1/_2 t_2$

Generalised & Continuous Soundness

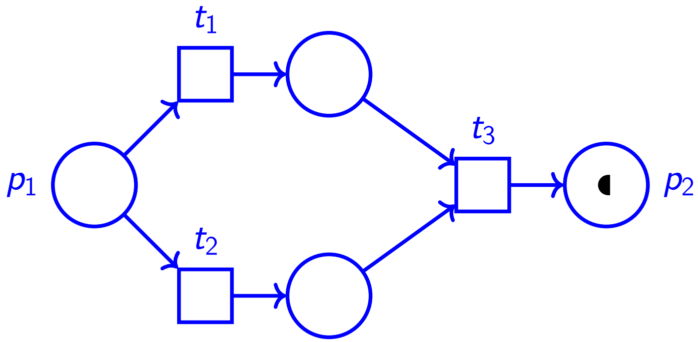
Why does generalised soundness require continuous soundness?



${}^1\!/_2 t_1$ ${}^1\!/_2 t_2$ ${}^1\!/_2 t_3$

Generalised & Continuous Soundness

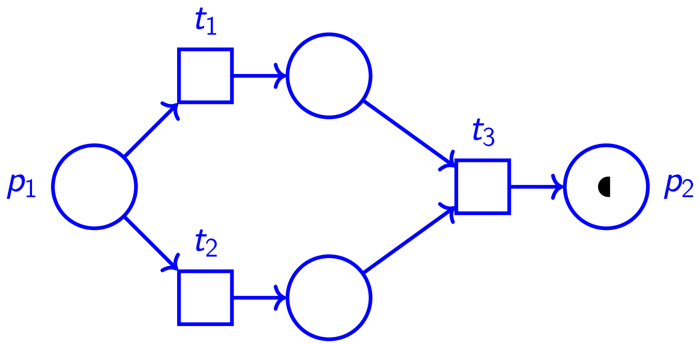
Why does generalised soundness require continuous soundness?



$$\{p_1 : 1\} \xrightarrow{{}^1\!/\!{}_2 t_1 \quad {}^1\!/\!{}_2 t_2 \quad {}^1\!/\!{}_2 t_3} \{p_2 : {}^1\!/\!{}_2\}$$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?

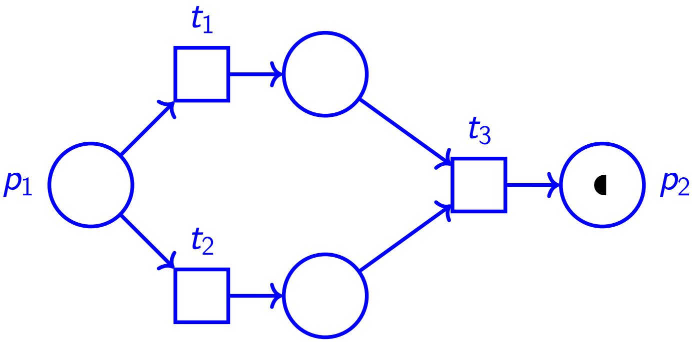


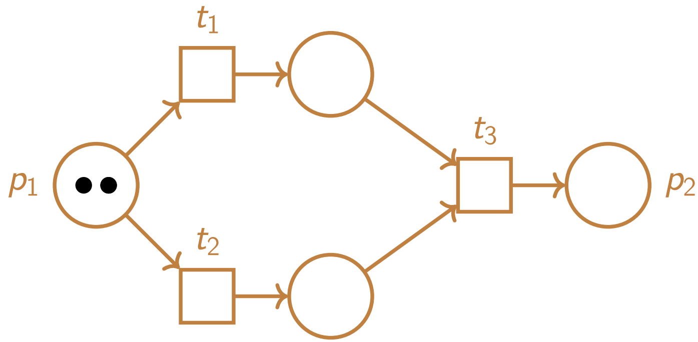
$$\{p_1 : 1\} \xrightarrow{\frac{1}{2}t_1 \ \ \frac{1}{2}t_2 \ \ \frac{1}{2}t_3} \{p_2 : \frac{1}{2}\}$$

Continuous reachability

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?



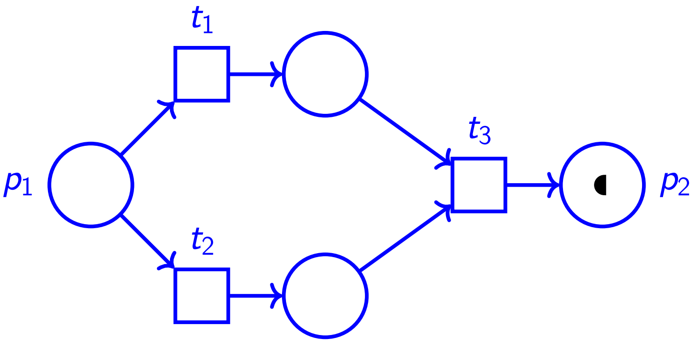


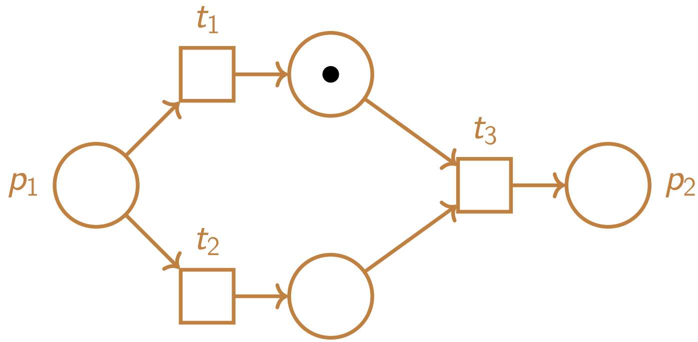
$$\{p_1 : 1\} \xrightarrow{\;1/2\, t_1 \quad 1/2\, t_2 \quad 1/2\, t_3\;} \{p_2 : 1/2\}$$

Continuous reachability

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





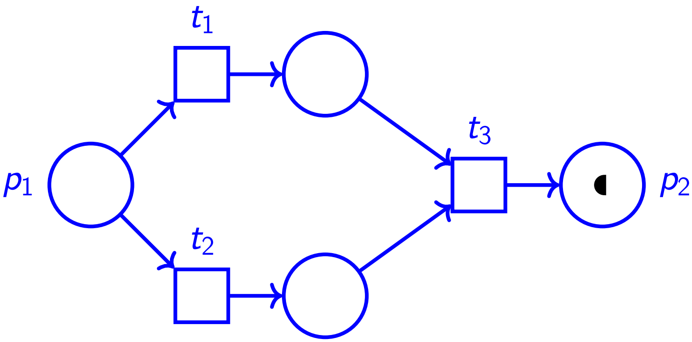
$$\{p_1 : 1\} \xrightarrow{{}^1/_2 t_1 \ \ {}^1/_2 t_2 \ \ {}^1/_2 t_3} \{p_2 : {}^1/_2\}$$

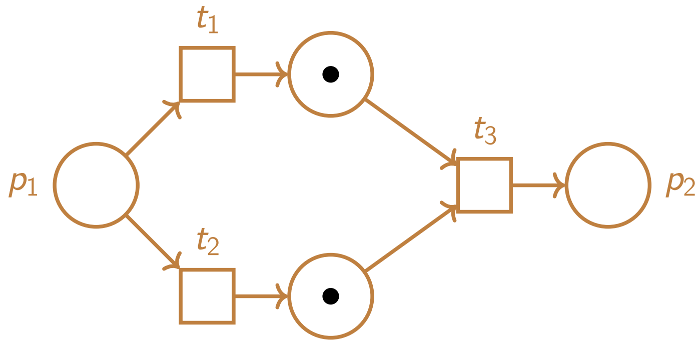
Continuous reachability

t_1

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





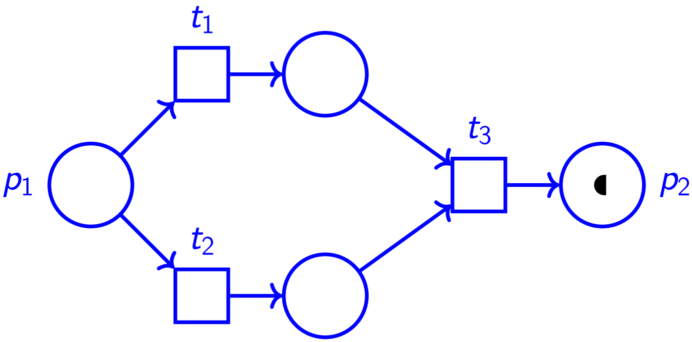
$$\{p_1 : 1\} \xrightarrow{1/2\, t_1 \;\; 1/2\, t_2 \;\; 1/2\, t_3} \{p_2 : 1/2\}$$

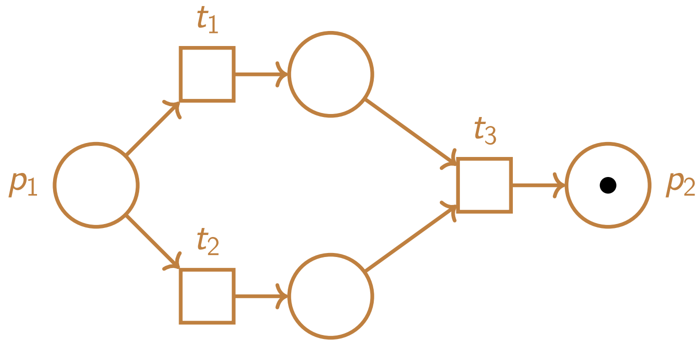
Continuous reachability

$t_1 \quad t_2$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





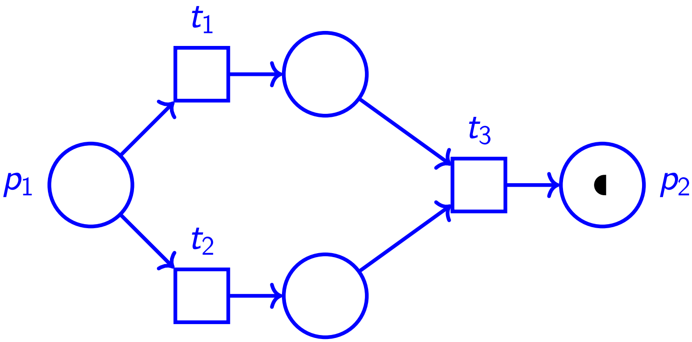
$$\{p_1 : 1\} \xrightarrow{\ {}^1\!/_2 t_1 \quad {}^1\!/_2 t_2 \quad {}^1\!/_2 t_3\ } \{p_2 : {}^1\!/_2\}$$

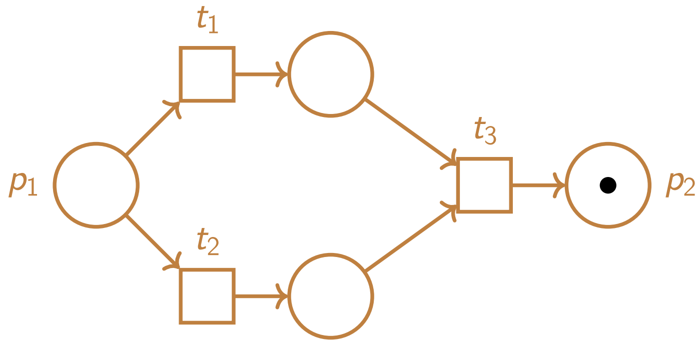
Continuous reachability

$t_1 \quad t_2 \quad t_3$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





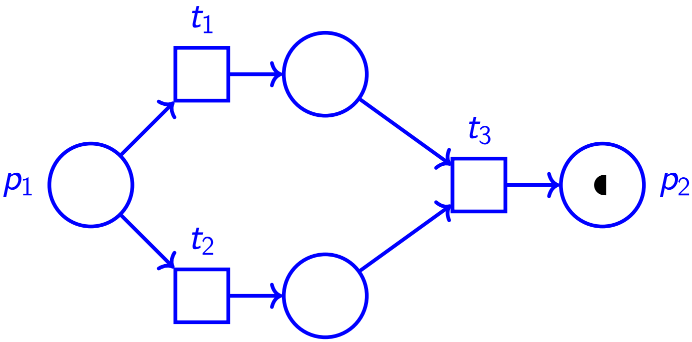
$\{p_1 : 1\} \xrightarrow{1/2 t_1 \;\; 1/2 t_2 \;\; 1/2 t_3} \{p_2 : 1/2\}$

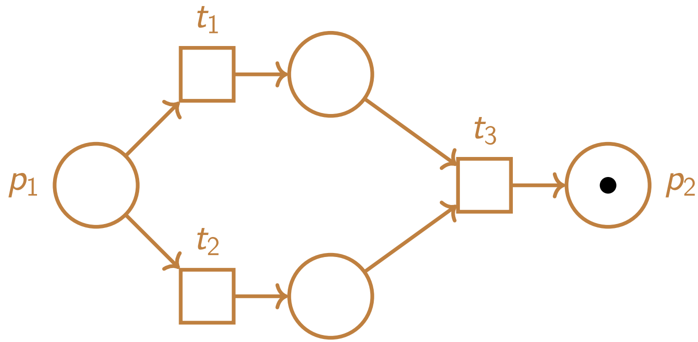
Continuous reachability

$\{p_1 : 2\} \xrightarrow{t_1 \;\; t_2 \;\; t_3} \{p_2 : 1\}$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





$$\{p_1 : 1\} \xrightarrow{{}^{1}\!/_{2}t_1 \quad {}^{1}\!/_{2}t_2 \quad {}^{1}\!/_{2}t_3} \{p_2 : {}^{1}\!/_{2}\}$$

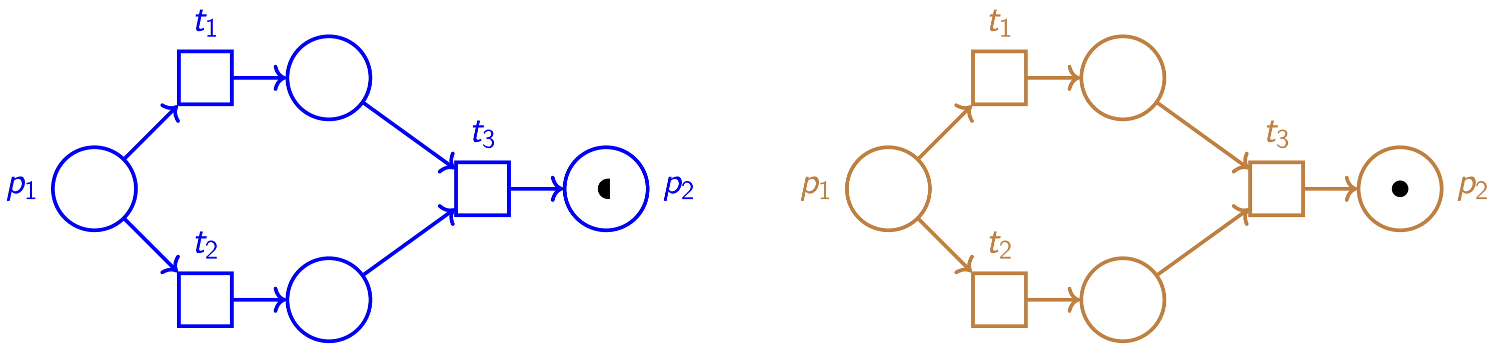
Continuous reachability

$$\{p_1 : 2\} \xrightarrow{t_1 \quad t_2 \quad t_3} \{p_2 : 1\}$$

Reachability with many tokens

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?

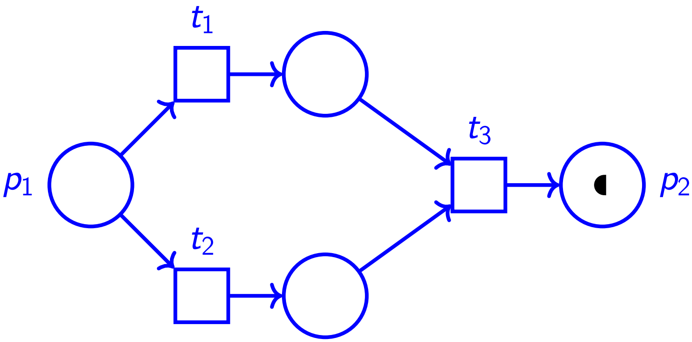


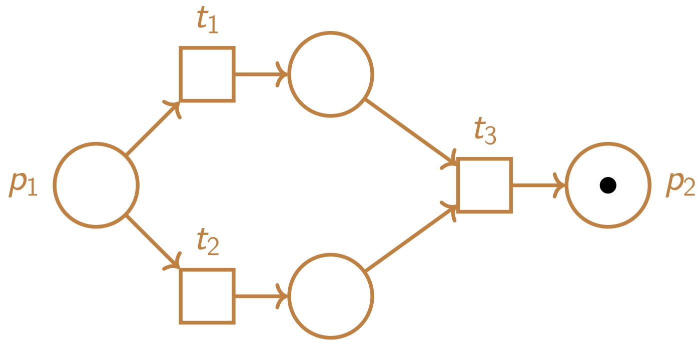
$$\{p_1 : 1\} \xrightarrow{{}^1\!/_2 t_1 \quad {}^1\!/_2 t_2 \quad {}^1\!/_2 t_3} \{p_2 : {}^1\!/_2\} \qquad \{p_1 : 2\} \xrightarrow{t_1 \quad t_2 \quad t_3} \{p_2 : 1\}$$

Continuous reachability $\Leftrightarrow$ Reachability with many tokens

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





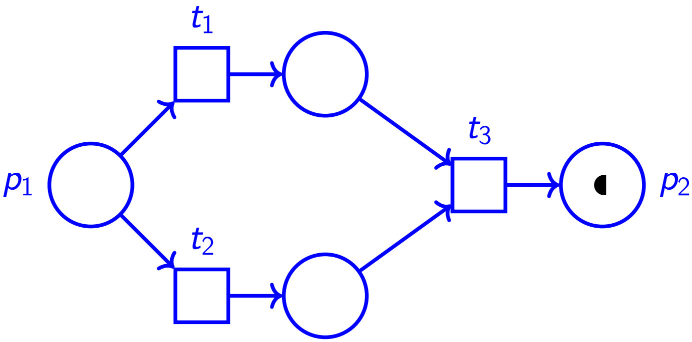
$\{p_1 : 1\} \xrightarrow{{}^1\!/_2 t_1 \quad {}^1\!/_2 t_2 \quad {}^1\!/_2 t_3} \{p_2 : {}^1\!/_2\}$ $\quad$ $\{p_1 : 2\} \xrightarrow{t_1 \quad t_2 \quad t_3} \{p_2 : 1\}$

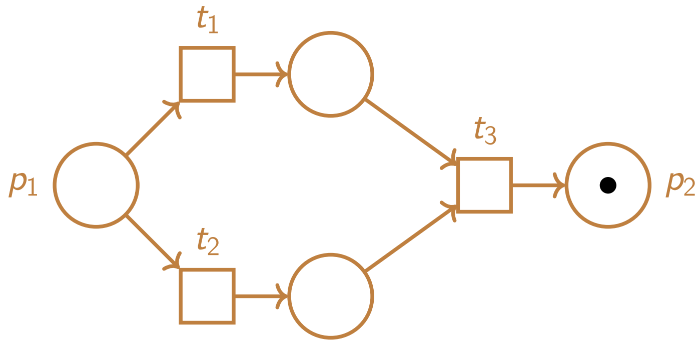
Continuous reachability $\Leftrightarrow$ Reachability with many tokens

$$\{\mathcal{I} : 1\} \rightarrow_{\mathbb{Q}} m \not\rightarrow_{\mathbb{Q}} \{\mathcal{F} : 1\}$$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





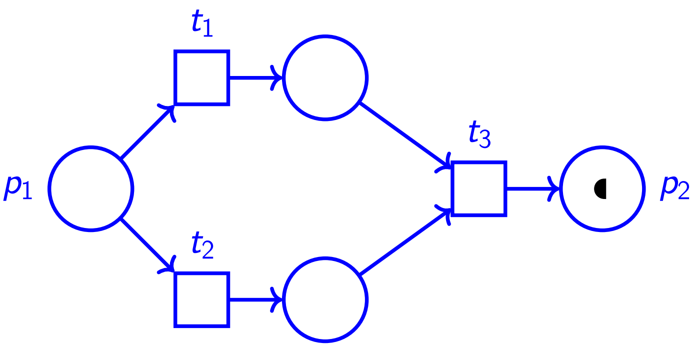
$\{p_1 : 1\} \xrightarrow{{}^1\!/_2 t_1 \quad {}^1\!/_2 t_2 \quad {}^1\!/_2 t_3} \{p_2 : {}^1\!/_2\}$ $\qquad$ $\{p_1 : 2\} \xrightarrow{t_1 \quad t_2 \quad t_3} \{p_2 : 1\}$

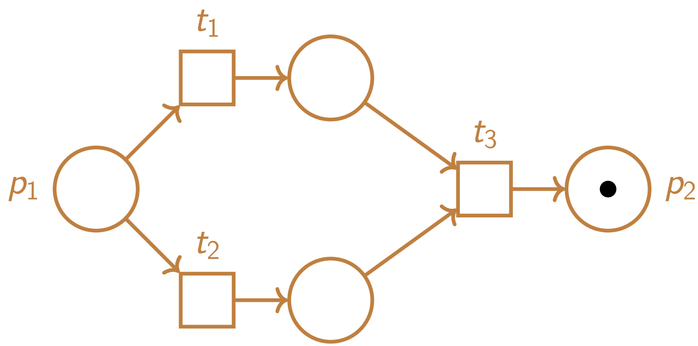
Continuous reachability $\Leftrightarrow$ Reachability with many tokens

$$\{\mathcal{I} : 1\} \rightarrow_{\mathbb{Q}} m \not\rightarrow_{\mathbb{Q}} \{\mathcal{F} : 1\} \qquad \exists\, k: \{\mathcal{I} : k\} \rightarrow m \not\rightarrow \{\mathcal{F} : k\}$$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





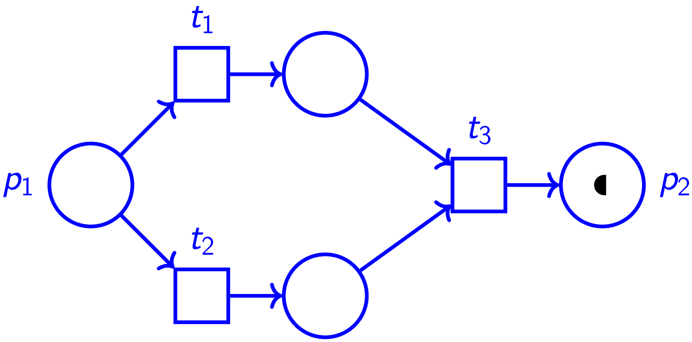
$$\{p_1 : 1\} \xrightarrow{{}^1\!/_2 t_1 \;\; {}^1\!/_2 t_2 \;\; {}^1\!/_2 t_3} \{p_2 : {}^1\!/_2\} \qquad \{p_1 : 2\} \xrightarrow{t_1 \;\; t_2 \;\; t_3} \{p_2 : 1\}$$

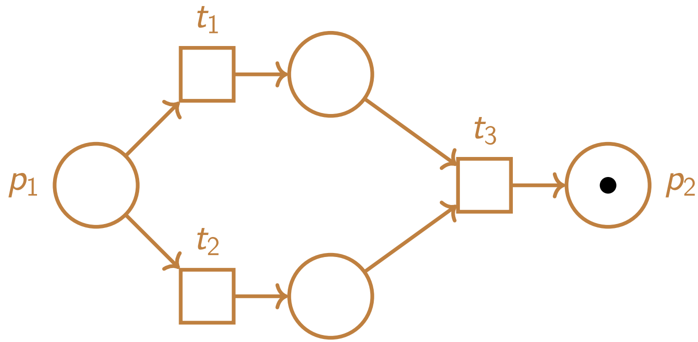
Continuous reachability $\Leftrightarrow$ Reachability with many tokens

$$\{\mathcal{I} : 1\} \to_{\mathbb{Q}} m \not\to_{\mathbb{Q}} \{\mathcal{F} : 1\} \implies \begin{array}{c} \exists\, k: \\ \{\mathcal{I} : k\} \to m \not\to \{\mathcal{F} : k\} \end{array}$$

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





$$\{p_1 : 1\} \xrightarrow{1/2 t_1 \quad 1/2 t_2 \quad 1/2 t_3} \{p_2 : 1/2\} \qquad \{p_1 : 2\} \xrightarrow{t_1 \quad t_2 \quad t_3} \{p_2 : 1\}$$

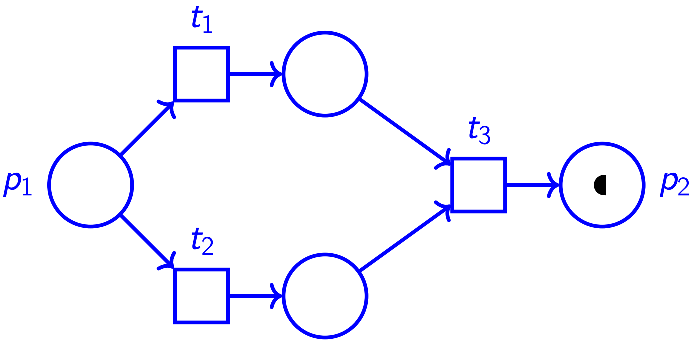
Continuous reachability $\Leftrightarrow$ Reachability with many tokens

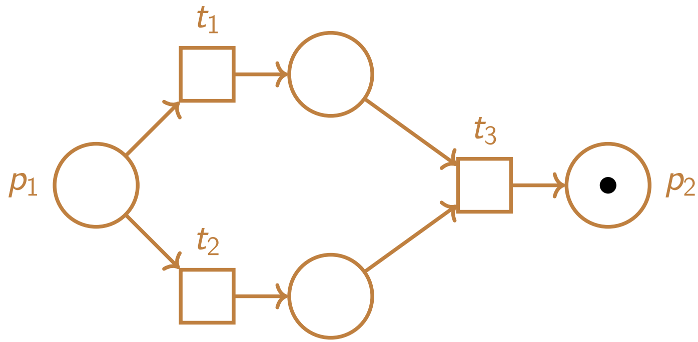
$$\{\mathcal{I} : 1\} \to_{\mathbb{Q}} m \not\to_{\mathbb{Q}} \{\mathcal{F} : 1\} \implies \exists\, k : \{\mathcal{I} : k\} \to m \not\to \{\mathcal{F} : k\}$$

Continuous unsound

Generalised & Continuous Soundness

Why does generalised soundness require continuous soundness?





$$\{p_1 : 1\} \xrightarrow{1/2\, t_1 \quad 1/2\, t_2 \quad 1/2\, t_3} \{p_2 : 1/2\} \qquad \{p_1 : 2\} \xrightarrow{t_1 \quad t_2 \quad t_3} \{p_2 : 1\}$$

Continuous reachability $\Leftrightarrow$ Reachability with many tokens

$$\{\mathcal{I} : 1\} \rightarrow_{\mathbb{Q}} m \not\rightarrow_{\mathbb{Q}} \{\mathcal{F} : 1\} \implies \overset{\exists\, k:}{\{\mathcal{I} : k\} \rightarrow m \not\rightarrow \{\mathcal{F} : k\}}$$

Continuous unsound Generalised unsound

Complexity of Continuous Soundness

Continuous Reachability: in **PTIME** [Fraca&Haddad, 2013]

Complexity of Continuous Soundness

Continuous Reachability: in **PTIME** [Fraca&Haddad, 2013]

Continuous *Inclusion*: **in coNP** [Blondin et al., 2017]

$\mathbb{Q}\text{-Reach}(N, m) \subseteq \mathbb{Q}\text{-Reach}(N', m')$

Complexity of Continuous Soundness

Continuous Reachability: in **PTIME** [Fraca&Haddad, 2013]

Continuous *Inclusion*: **in coNP** [Blondin et al., 2017]

$\mathbb{Q}\text{-Reach}(N, m) \subseteq \mathbb{Q}\text{-Reach}(N', m')$

Continuous Soundness: **coNP-complete** [CAV'22]

$\mathbb{Q}\text{-Reach}(N, \{\mathcal{I}\colon 1\}) \subseteq \mathbb{Q}\text{-Reach}(N^{\text{Reversed}}, \{\mathcal{F}\colon 1\})$

Continuous Soundness is a useful criterion

Benchmarks: 1976 industrial nets

Continuous Soundness is a useful criterion

Benchmarks: 1976 industrial nets

1334/1976 nets are continuous unsound!

Continuous Soundness is a useful criterion

Benchmarks: 1976 industrial nets

1334/1976 nets are continuous unsound!

Remaining nets are continuous sound
...and also generalised sound

Continuous Soundness is a useful criterion

Benchmarks: 1976 industrial nets

1334/1976 nets are continuous unsound!

Remaining nets are continuous sound
...and also generalised sound

Why is continuous soundness so accurate in practice?

Continuous Soundness is a useful criterion

Benchmarks: 1976 industrial nets

1334/1976 nets are continuous unsound!

Remaining nets are continuous sound
...and also generalised sound

Why is continuous soundness so accurate in practice?

Many instances are actually easy:
Free Choice Workflow Nets

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* 5.
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* 5.
  Soundness in Ptime, and all soundness variants are equivalent

Free-Choice Workflow Nets

Workflow nets with a restriction on transitions:

Free-Choice Workflow Nets

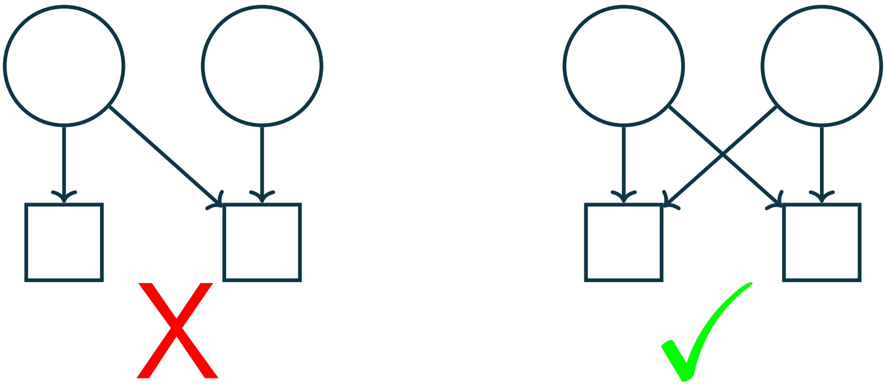
Workflow nets with a restriction on transitions:

Transitions that share an input place must share **all** input places

Free-Choice Workflow Nets

Workflow nets with a restriction on transitions:

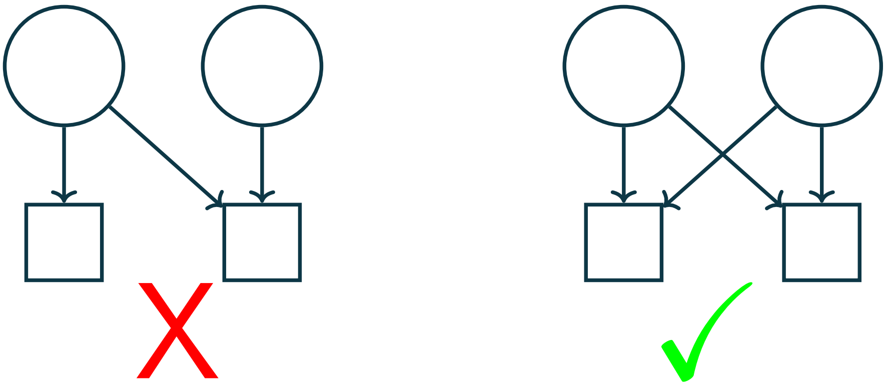
Transitions that share an input place must share **all** input places



Free-Choice Workflow Nets

Workflow nets with a restriction on transitions:

Transitions that share an input place must share **all** input places

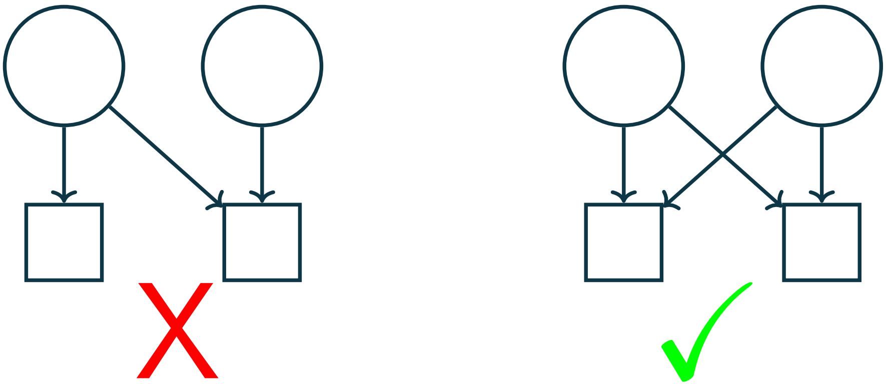


Soundness is in Ptime [van der Aalst, 1998]

Free-Choice Workflow Nets

Workflow nets with a restriction on transitions:

Transitions that share an input place must share **all** input places



Soundness is in Ptime [van der Aalst, 1998]

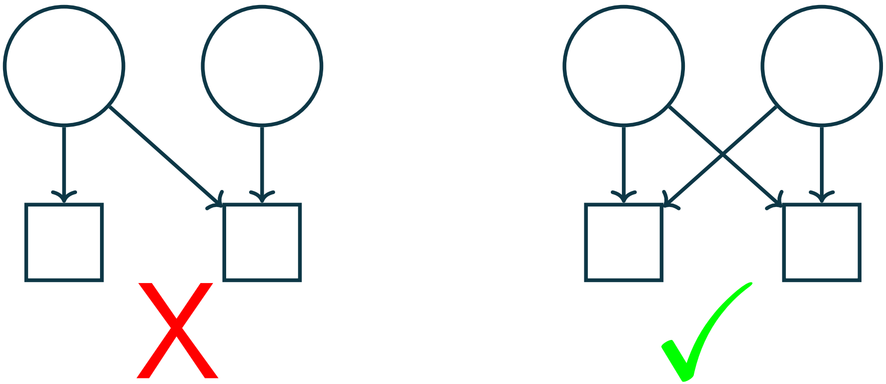
Soundness notions are equivalent:

1-Sound $\underset{\text{[Ping et al.,'04]}}{\equiv}$ Gen. Sound $\underset{\text{[CAV'22]}}{\equiv}$ Struct. Sound $\underset{\text{[CAV'22]}}{\equiv}$ Cont. Sound

Free-Choice Workflow Nets

Workflow nets with a restriction on transitions:

Transitions that share an input place must share **all** input places



Soundness is in Ptime [van der Aalst, 1998]

Soundness notions are equivalent:

1-Sound $\underset{\text{[Ping et al.,'04]}}{\equiv}$ Gen. Sound $\underset{\text{[CAV'22]}}{\equiv}$ Struct. Sound $\underset{\text{[CAV'22]}}{\equiv}$ Cont. Sound

Continuous soundness is **exact** on free-choice nets

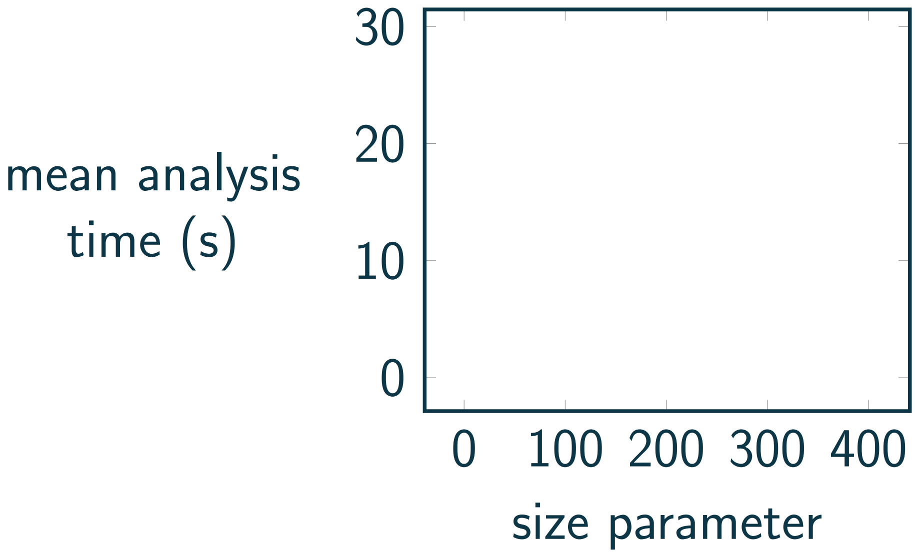
Continuous Soundness on Free Choice nets

Deciding soundness via:

Continuous Soundness vs State Space Exploration

Continuous Soundness on Free Choice nets

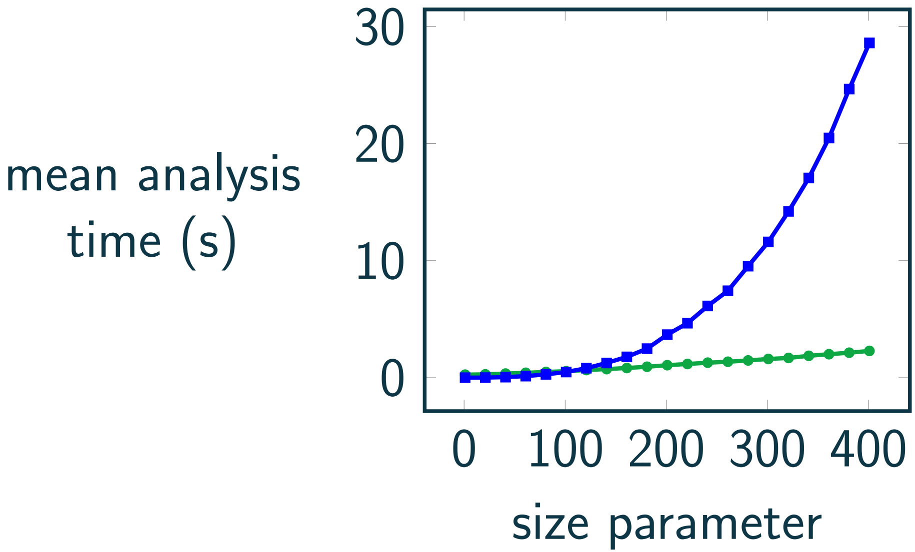
Deciding soundness via:
Continuous Soundness vs State Space Exploration



Continuous Soundness on Free Choice nets

Deciding soundness via:

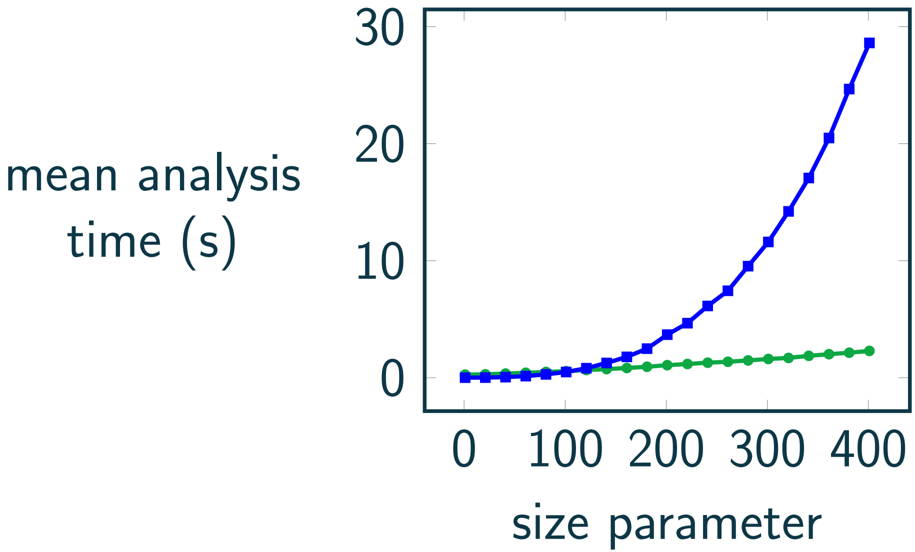
Continuous Soundness vs State Space Exploration



Continuous Soundness on Free Choice nets

Deciding soundness via:

Continuous Soundness vs State Space Exploration



Promising addition to existing techniques for **Free Choice nets**

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	1.
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	2.
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	3.

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* 4.
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* 5.
  Soundness in Ptime, and all soundness variants are equivalent

Checking soundness - complexity?

	known results	our work	
***k*-Soundness**	Decidable EXPSPACE-hard? [van der Aalst;'96, '97]	EXPSPACE-complete	**1.**
Generalised Soundness	Decidable [van Hee et al.;'04]	PSPACE-complete	**2.**
Structural Soundness	Decidable [Ţiplea, Marinescu;'04]	EXPSPACE-complete	**3.**

Exact algorithms are impractical in general; instead:

- Focus on semi-decision procedures - *Continuous Soundness* **4.**
  co-NP complete necessary condition for generalised soundness
- Focus on subclasses - *Free-Choice Workflow Nets* **5.**
  Soundness in Ptime, and all soundness variants are equivalent

Conclusion

Workflow nets formally model **processes**

Soundness is a widely used correctness condition

Variants: **Generalised Soundness**, **Structural Soundness**

Established exact complexities of soundness variants

Continuous soundness: necessary for gen. soundness
and equivalent to soundness variants on **Free-Choice nets**