Verifying generalised and structural soundness of workflow nets via relaxations

Philip Offtermatt

Joint work with
Michael Blondin and Filip Mazowiecki

Verifying generalised and structural soundness of workflow nets via relaxations

Philip Offtermatt

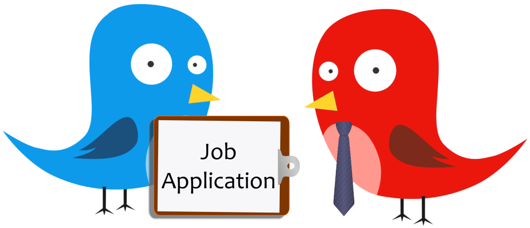
Joint work with
Michael Blondin and Filip Mazowiecki

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

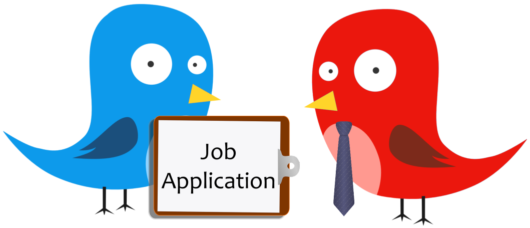
UNIVERSITAS VARSOVIENSIS

UNIVERSITÉ DE SHERBROOKE

Processes are everywhere!

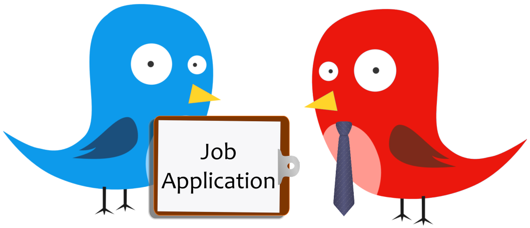


Processes are everywhere!



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

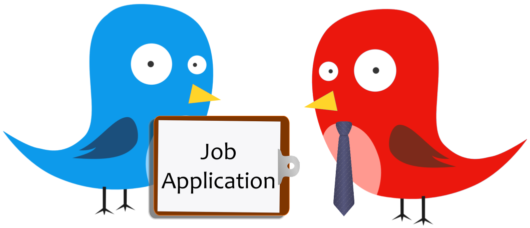
Processes are everywhere!



- 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!

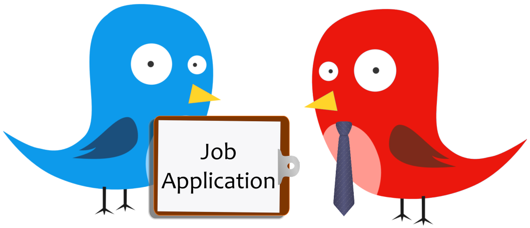


- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/ Recheck application

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

Can we handle applications faster?

Processes are everywhere!



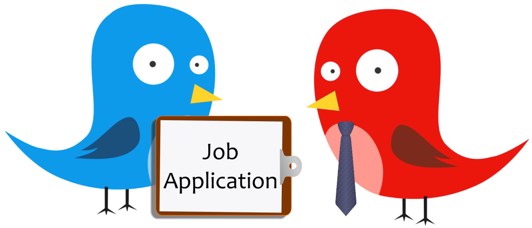
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/ Recheck application

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

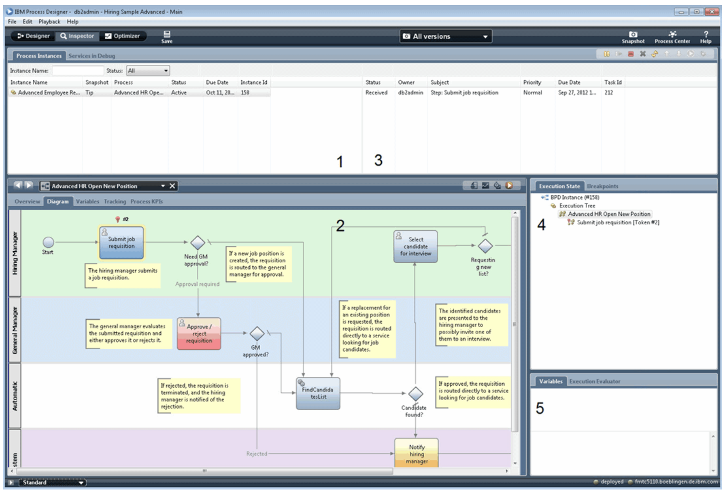
Can we handle applications faster?

Will every applicant hear back from us?

Processes are everywhere!



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

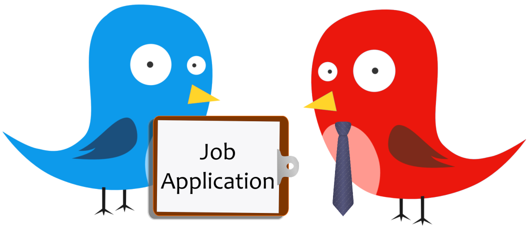


How many applicants will we need until we find a new hire?

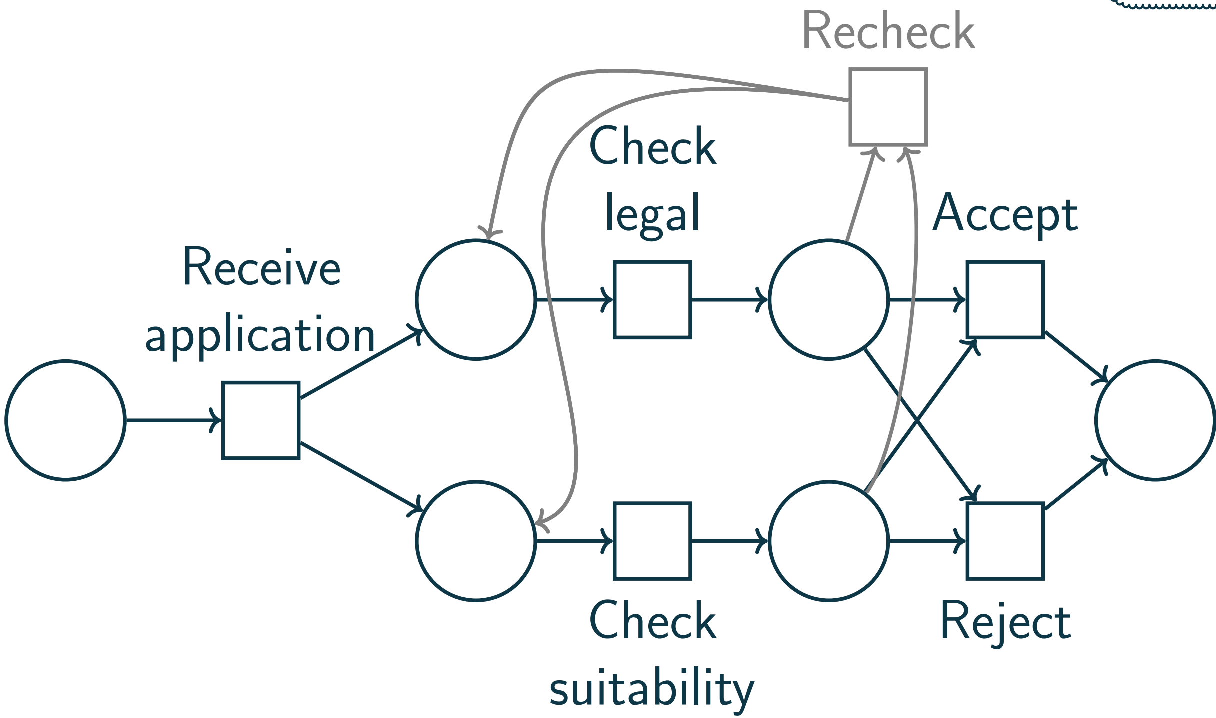
Can we handle applications faster?

Will every applicant hear back from us?

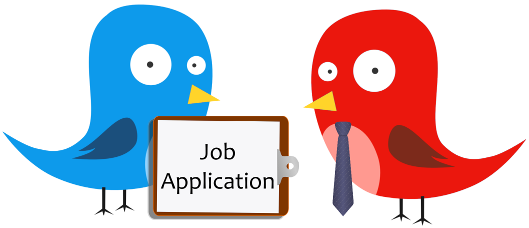
Formally modeling processes: Workflow nets



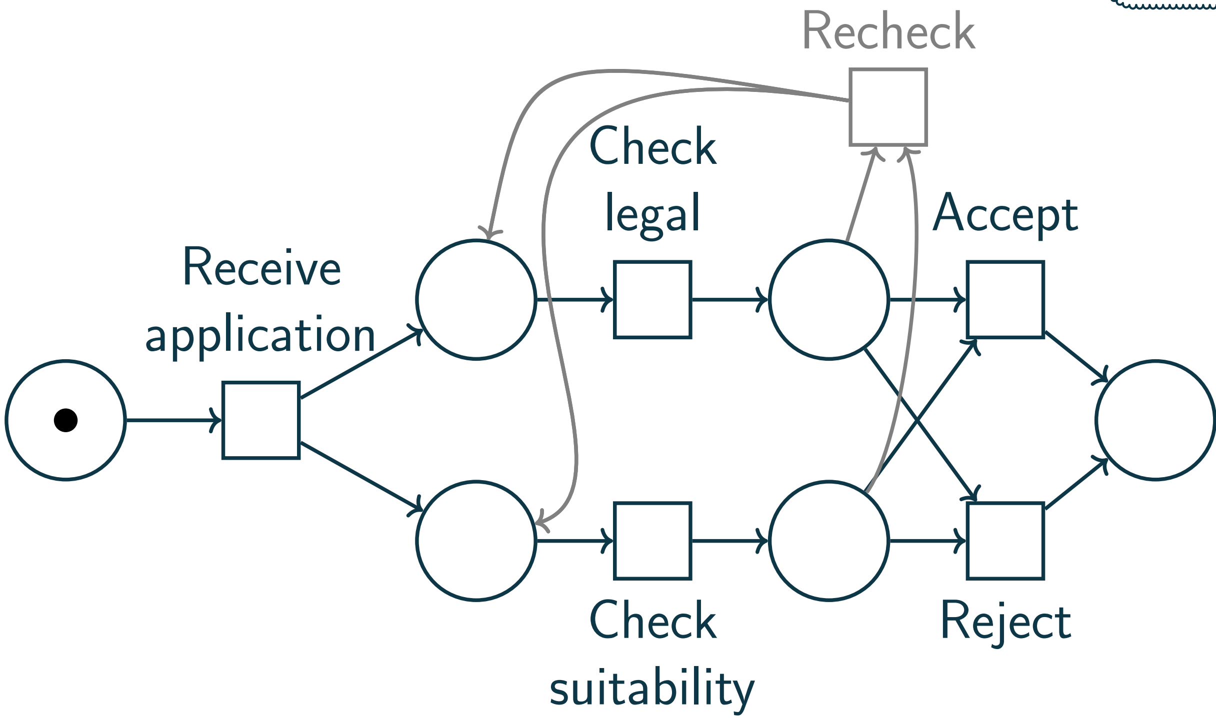
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



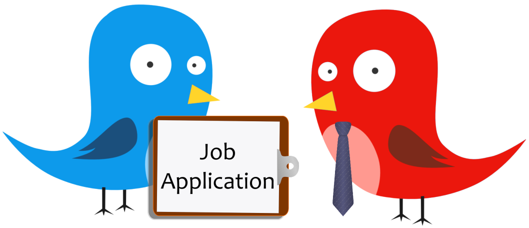
Formally modeling processes: Workflow nets



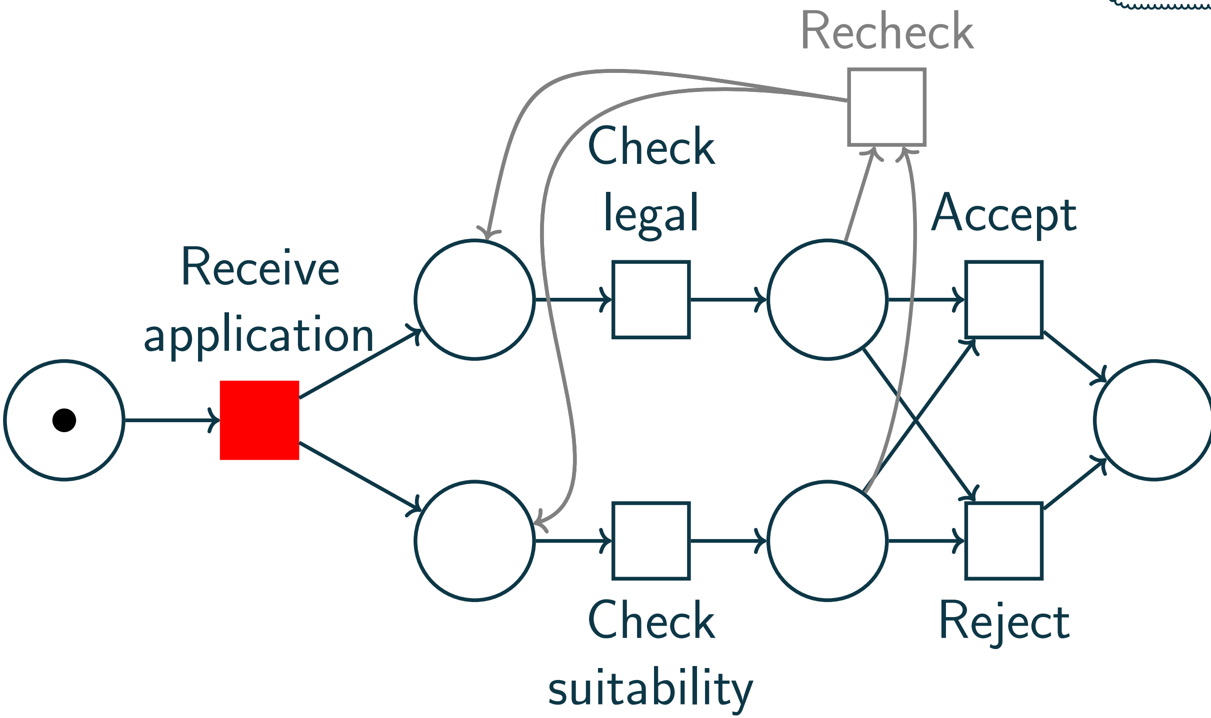
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



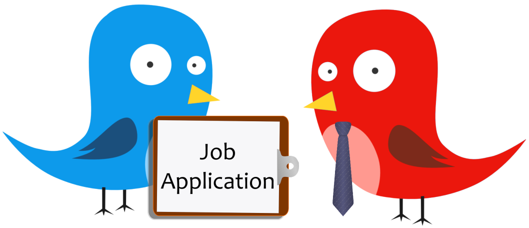
Formally modeling processes: Workflow nets



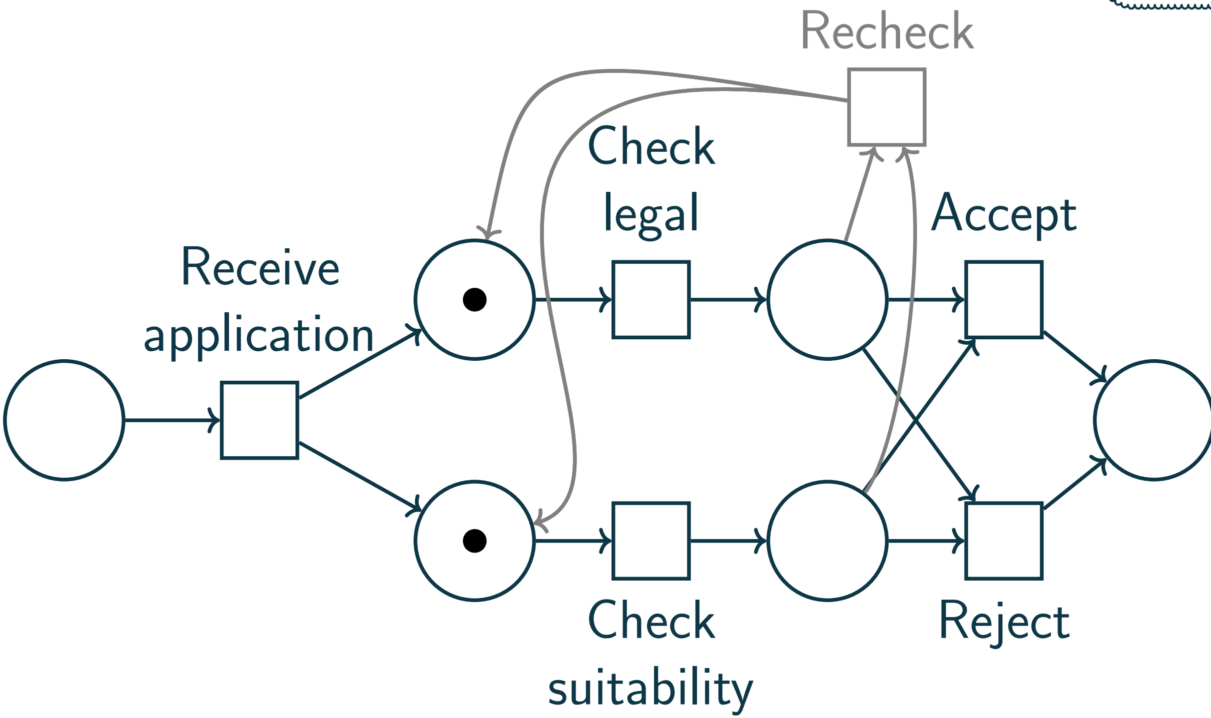
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



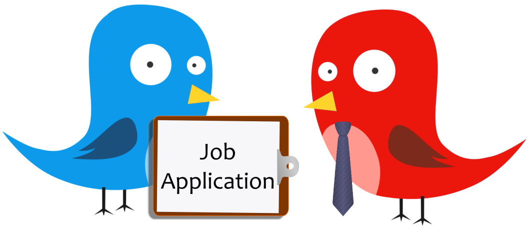
Formally modeling processes: Workflow nets



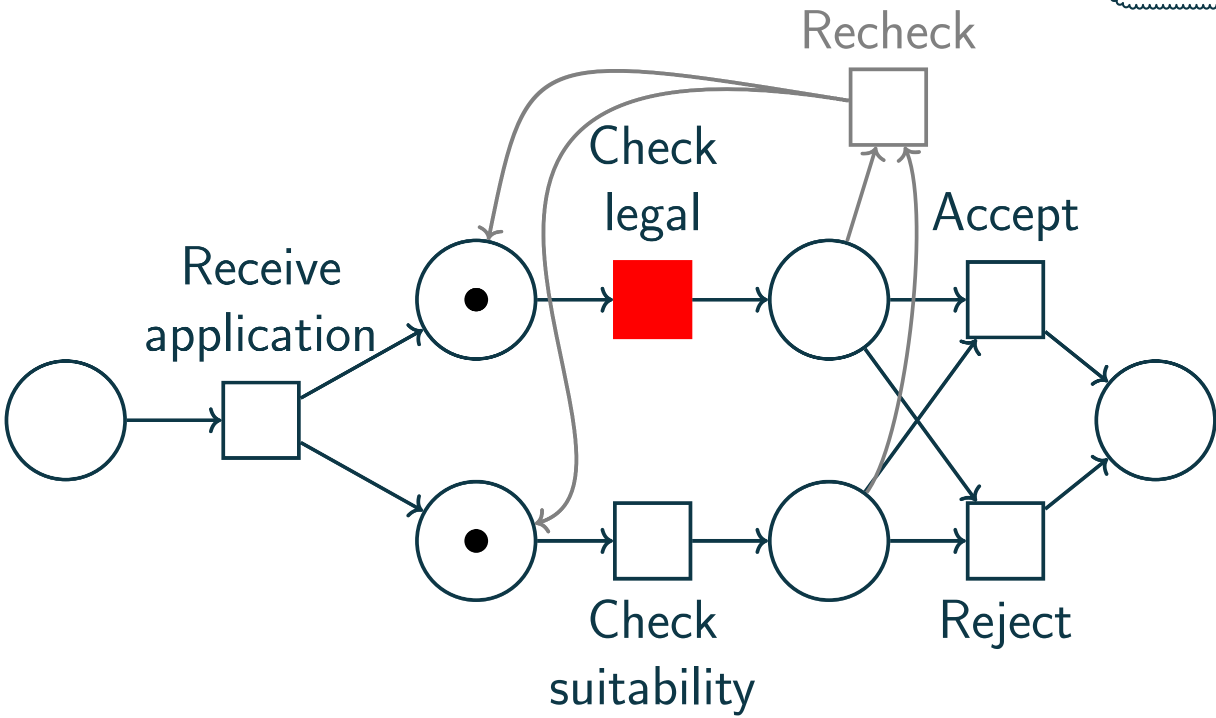
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



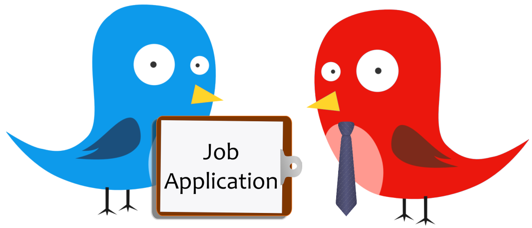
Formally modeling processes: Workflow nets



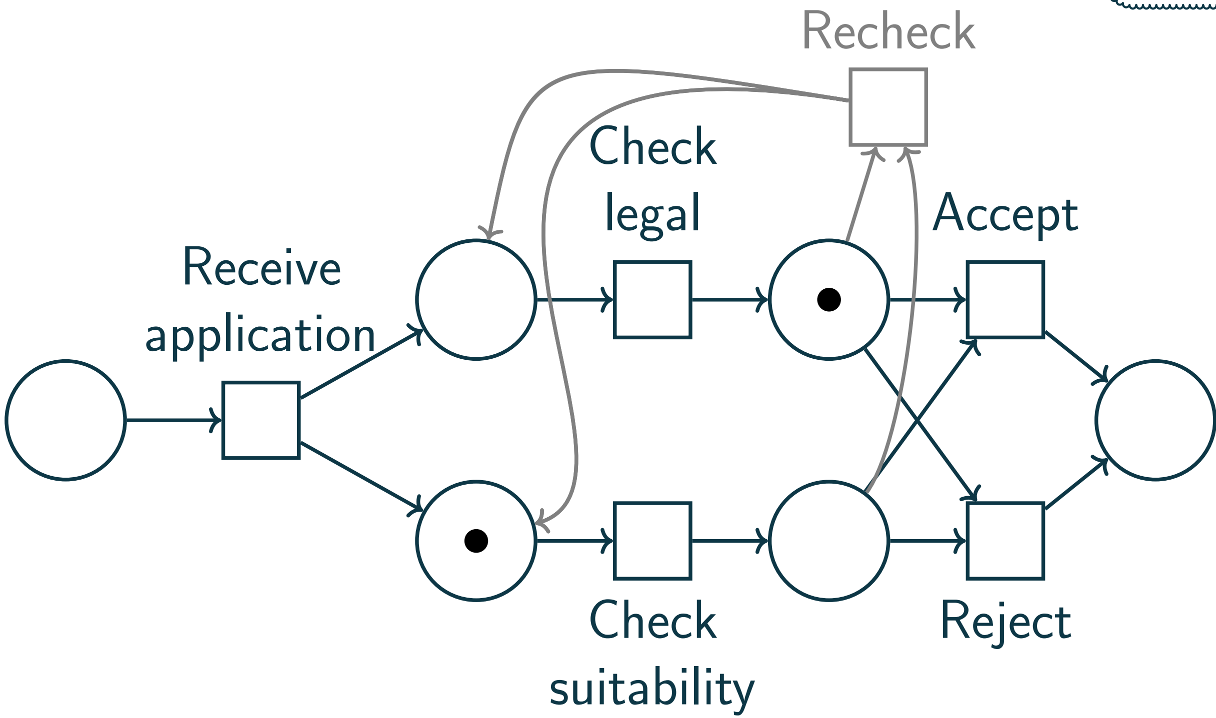
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



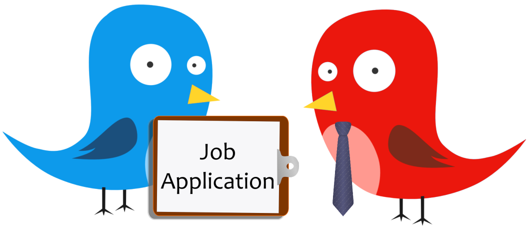
Formally modeling processes: Workflow nets



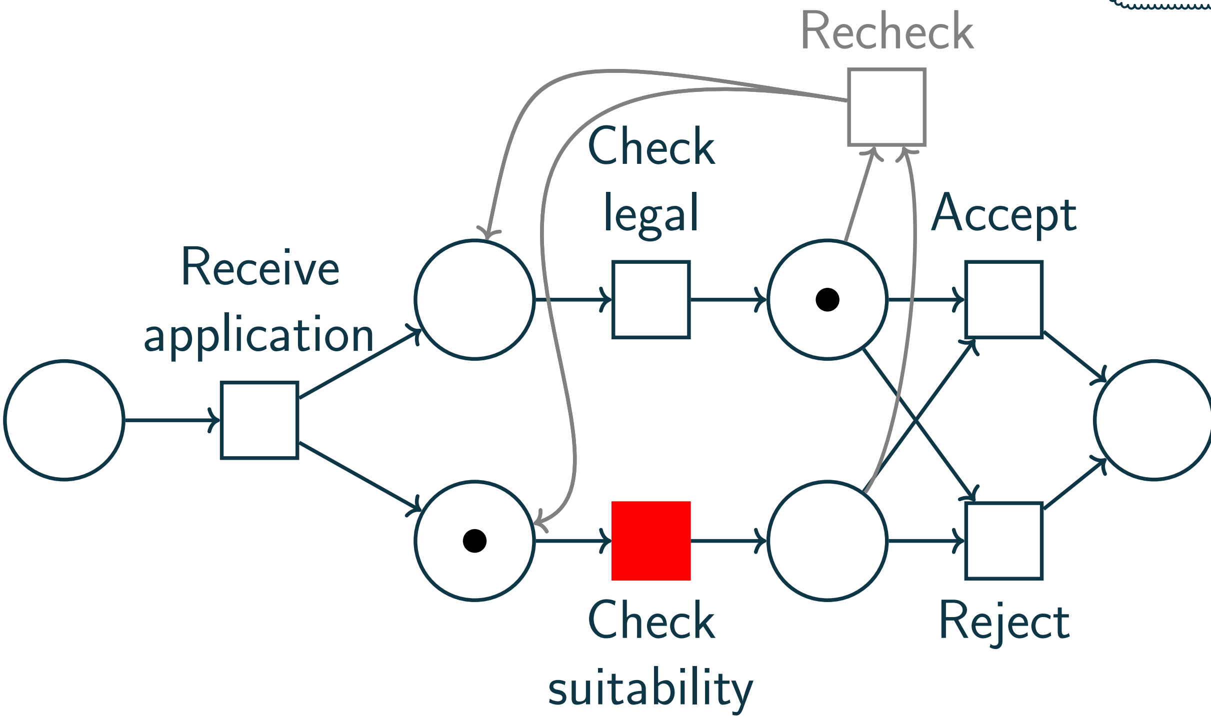
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



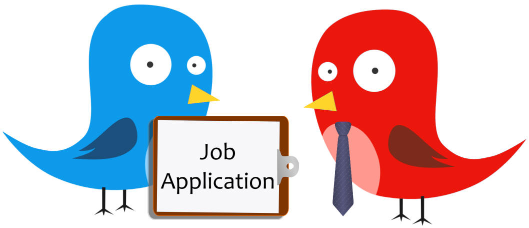
Formally modeling processes: Workflow nets



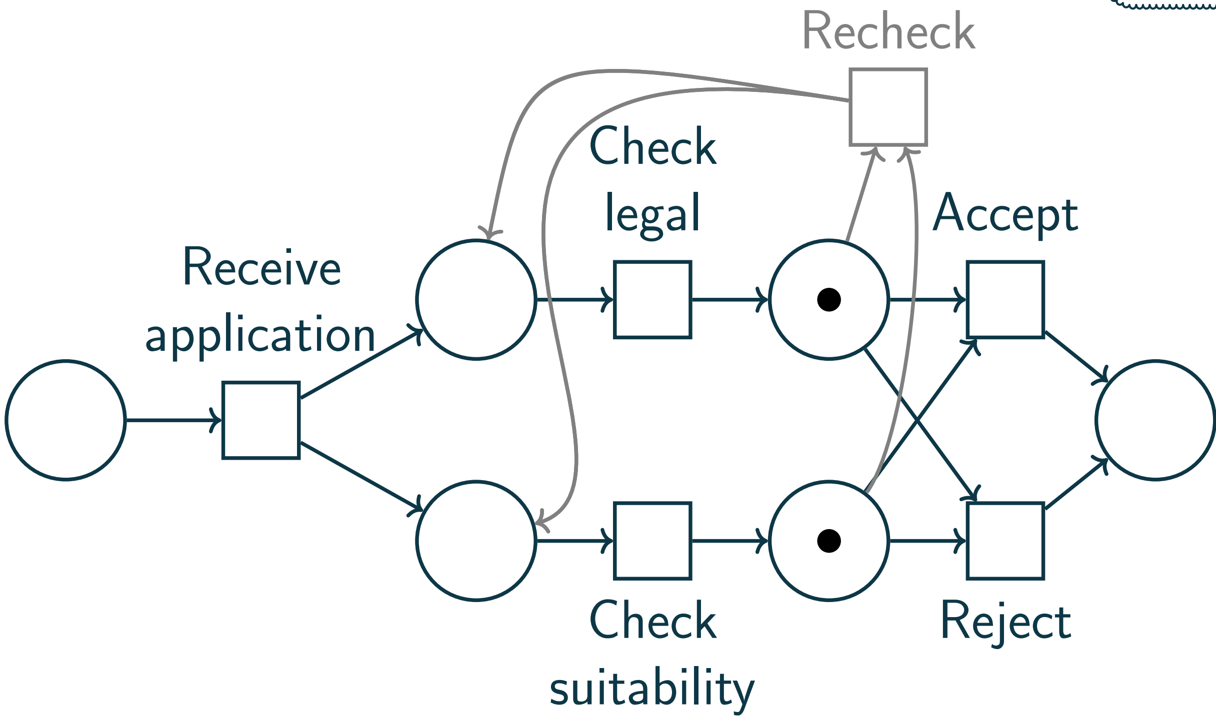
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



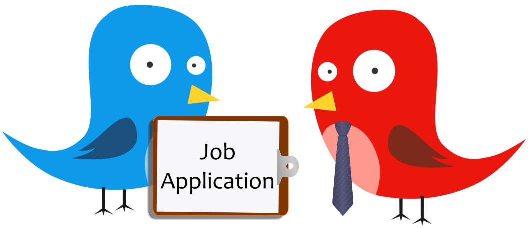
Formally modeling processes: Workflow nets



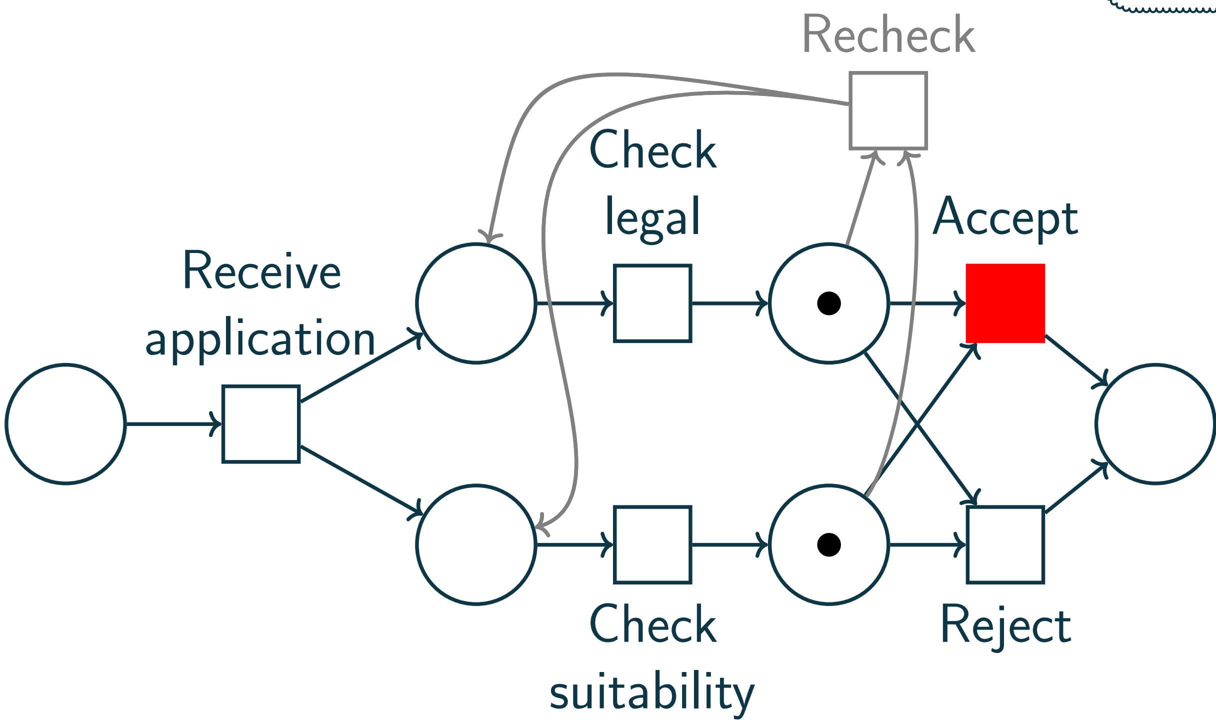
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



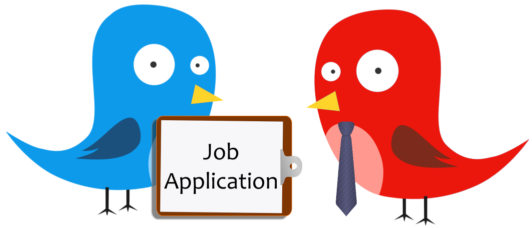
Formally modeling processes: Workflow nets



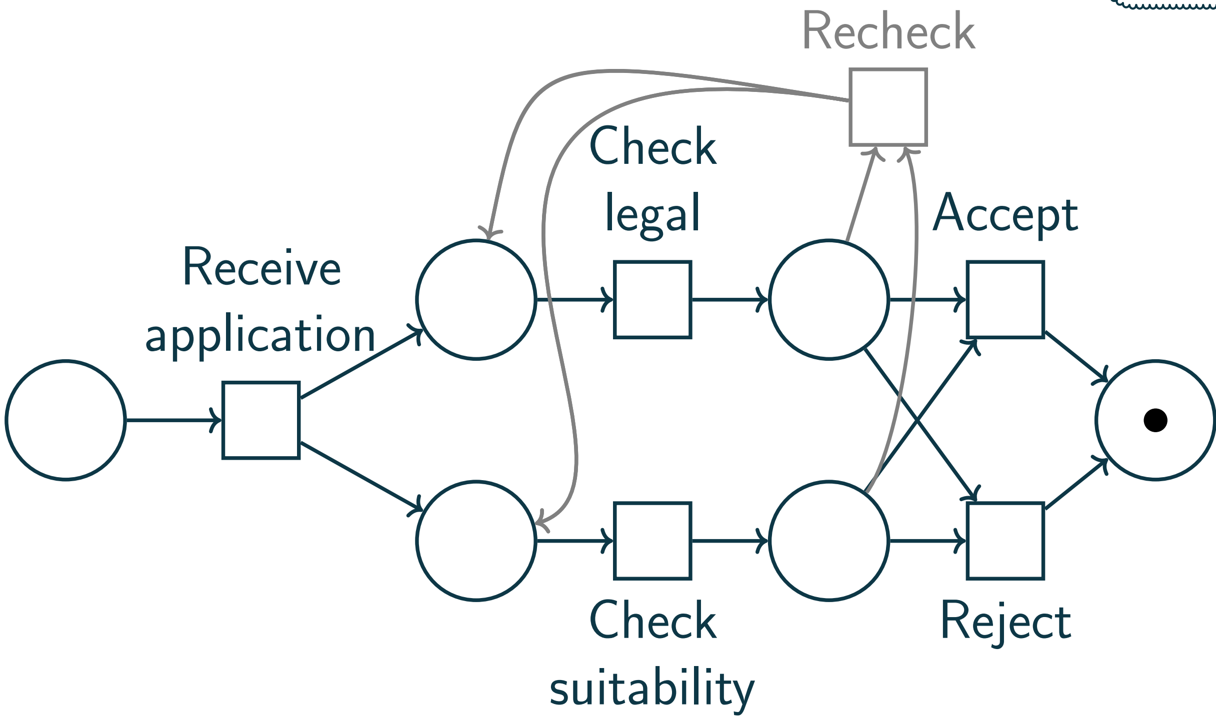
- Receive application
  - Check legal requirements
  - Check applicant suitability
- Decide: Accept/Reject/Recheck application



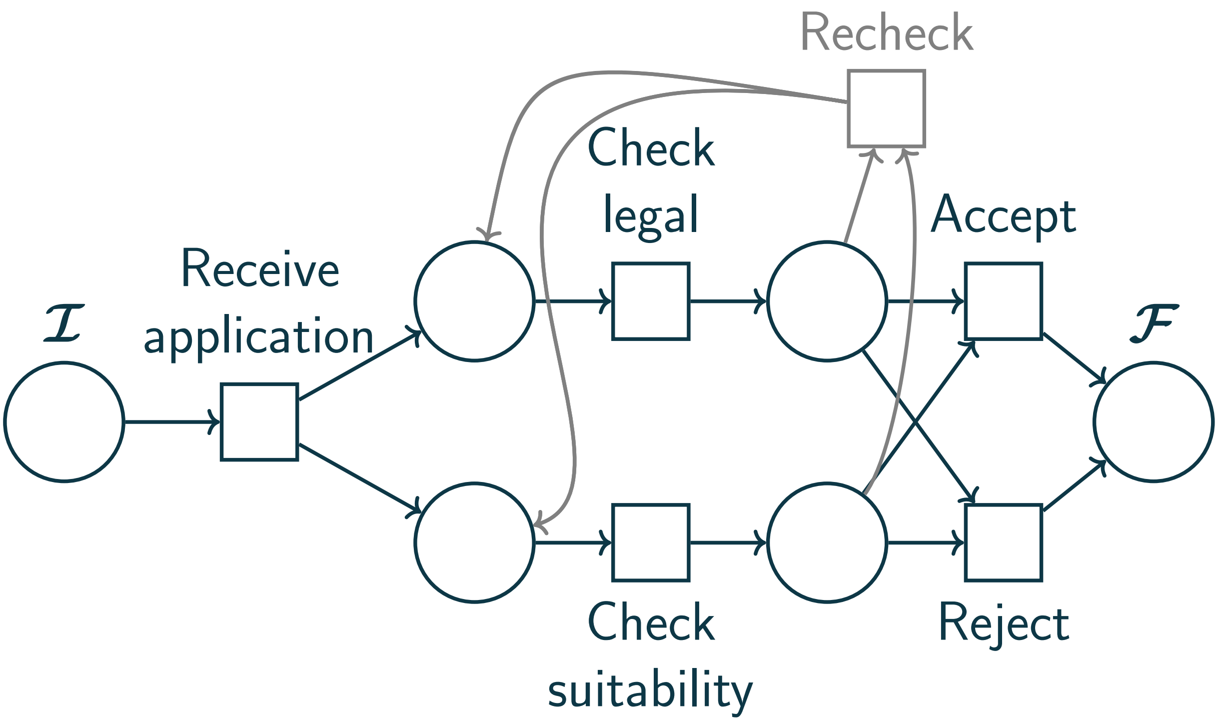
Formally modeling processes: Workflow nets



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

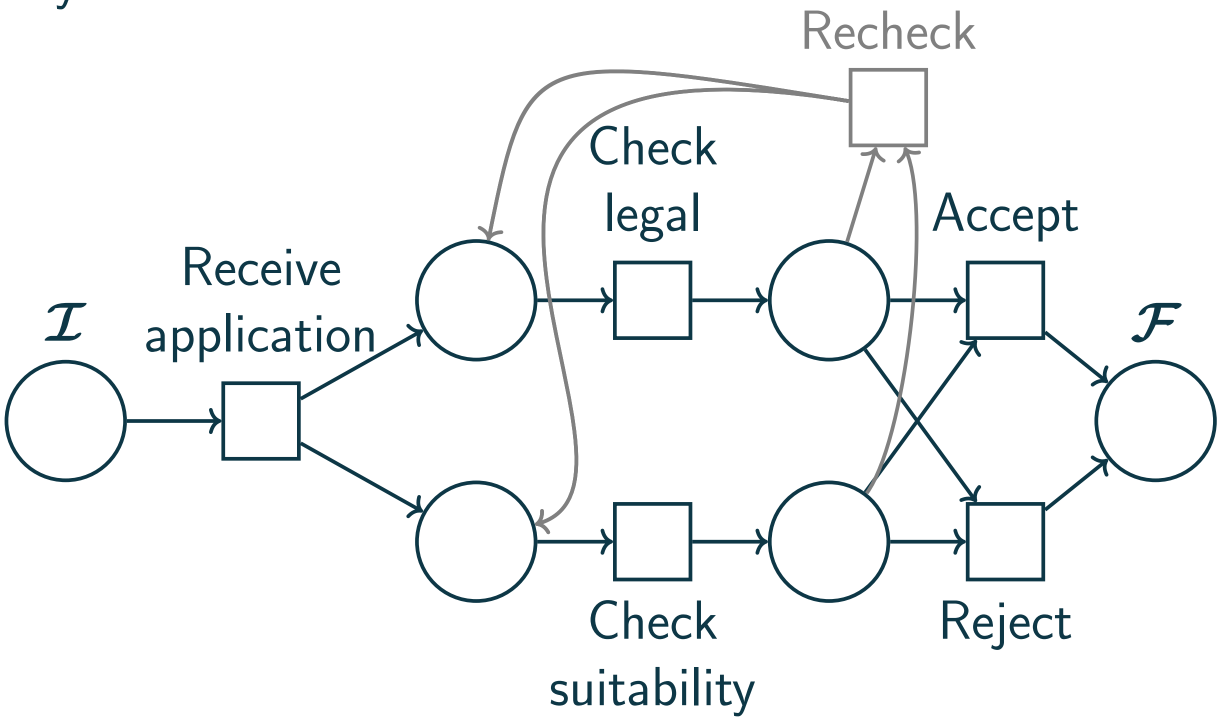


Correctness conditions for processes



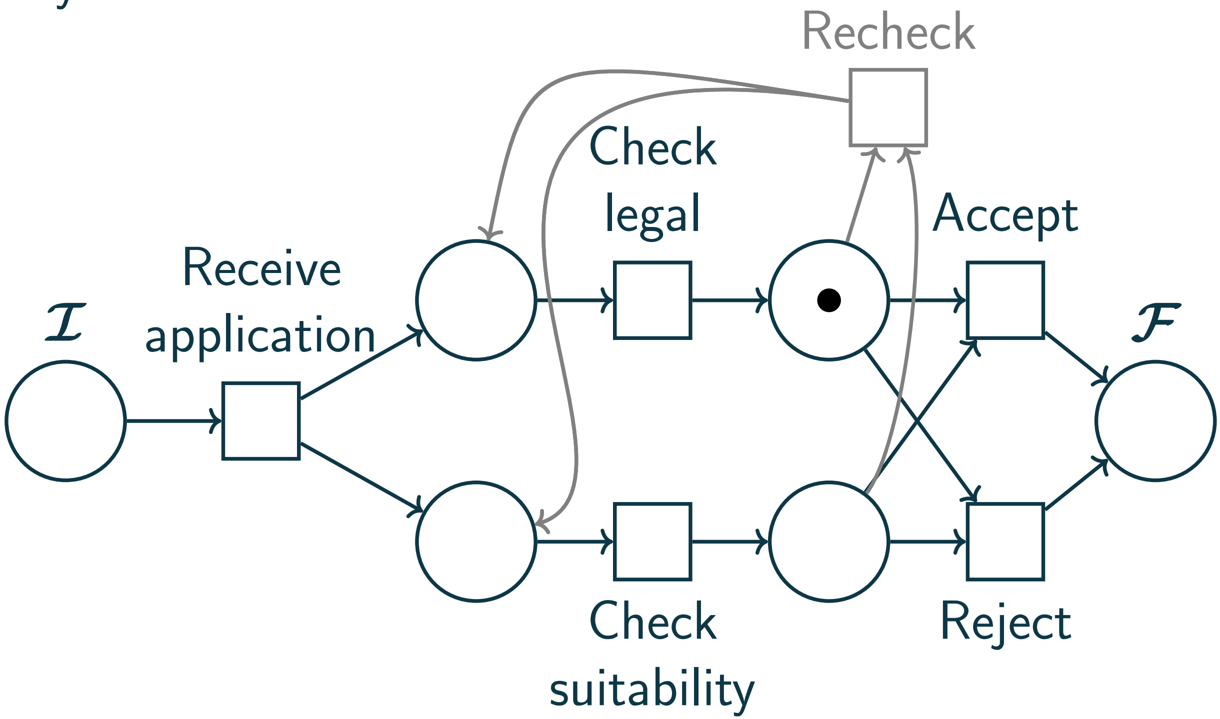
Correctness conditions for processes

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



Correctness conditions for processes

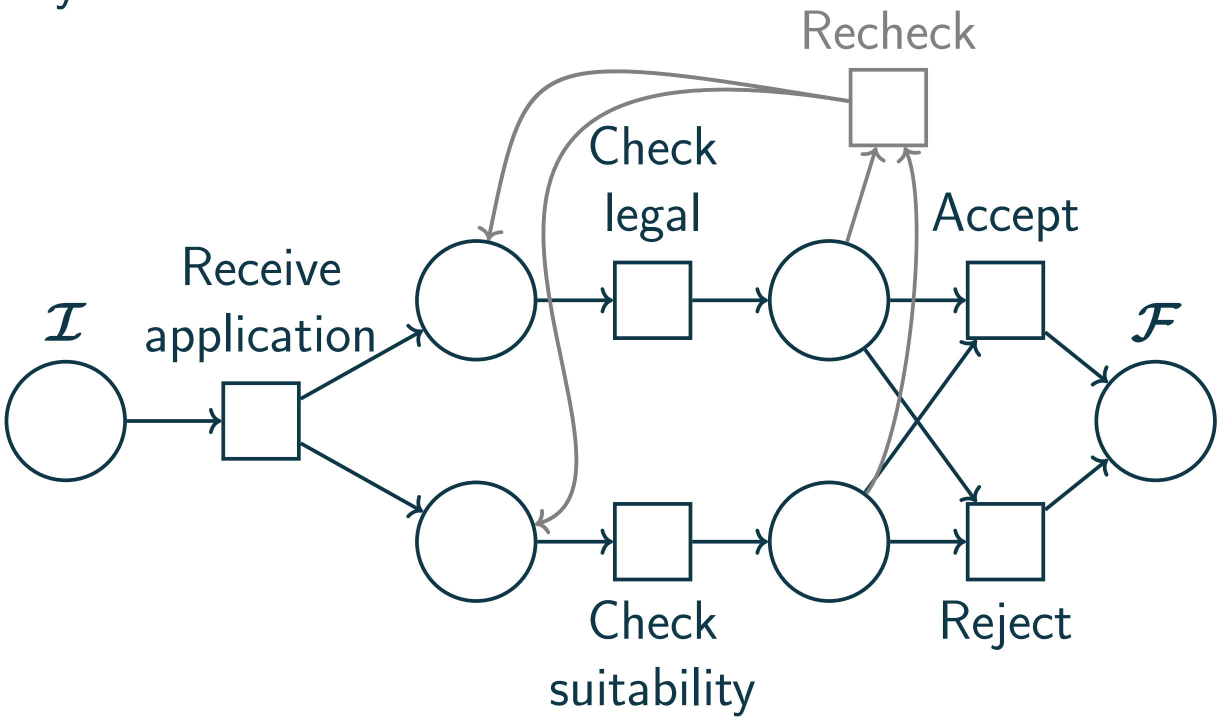
Option to complete:
We should be able to reach a
a marking that has tokens
only in $\mathcal{F}$



Correctness conditions for processes

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

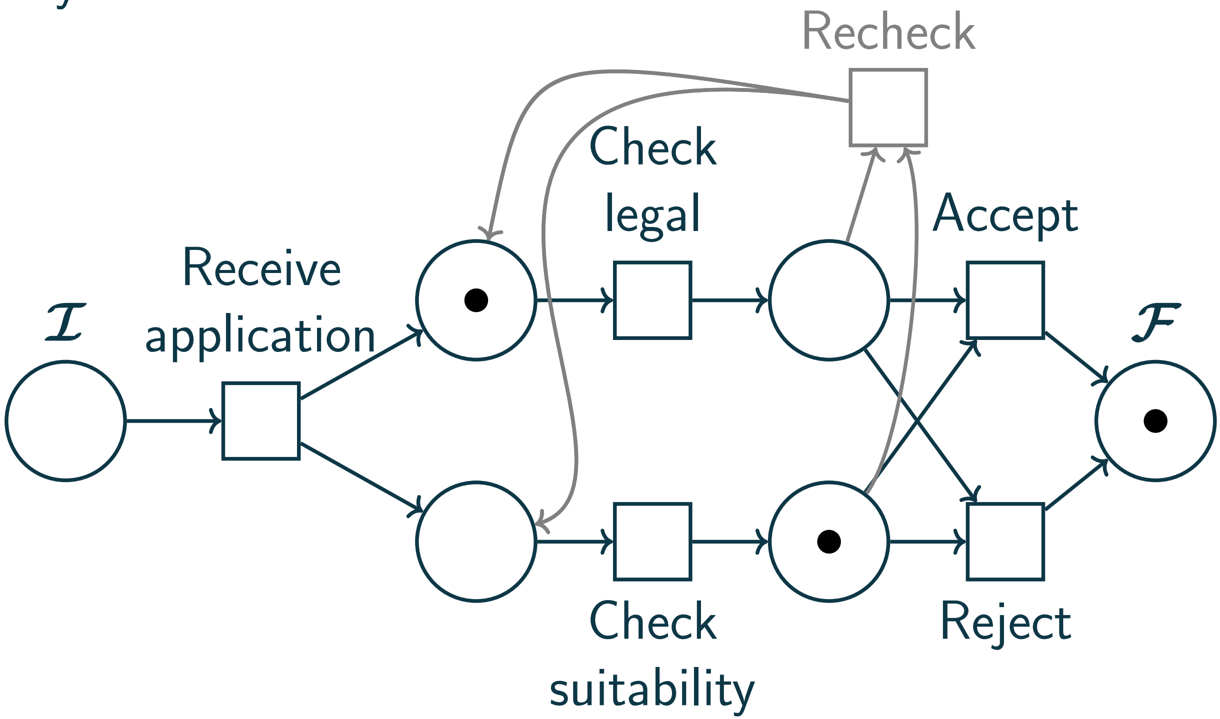
Proper completion:
When $\mathcal{F}$ is marked
the rest of the net is empty



Correctness conditions for processes

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

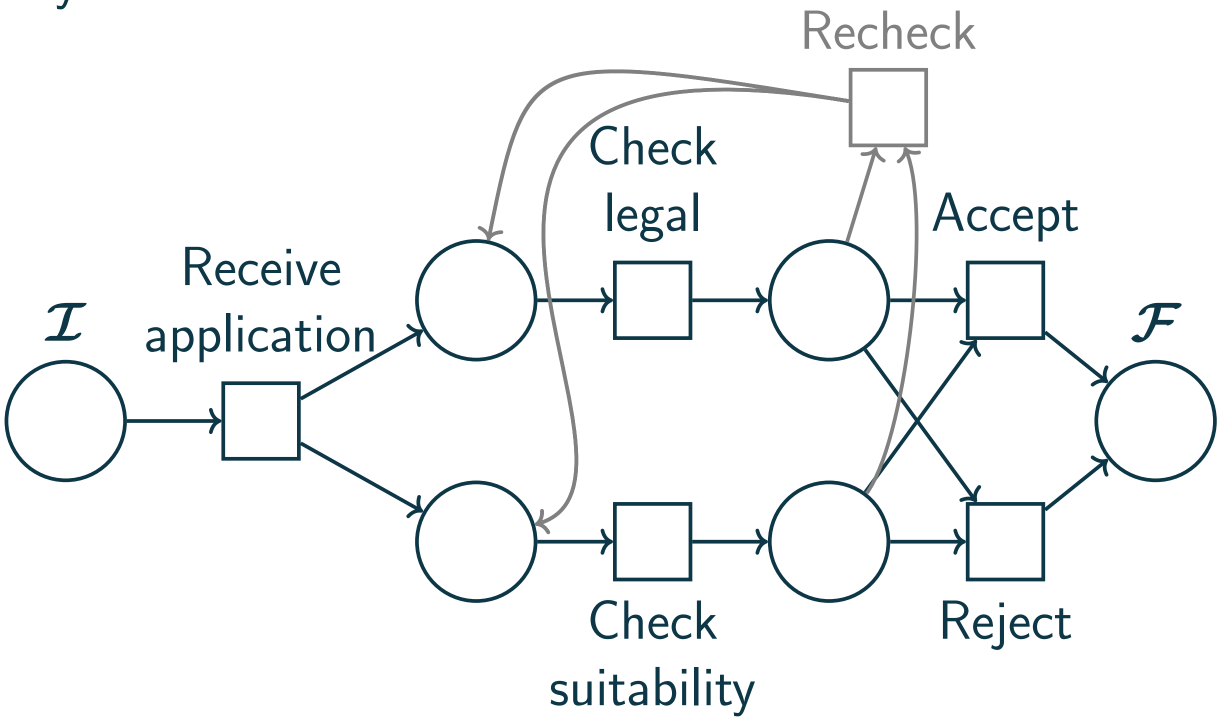
Proper completion:
When $\mathcal{F}$ is marked the rest of the net is empty



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

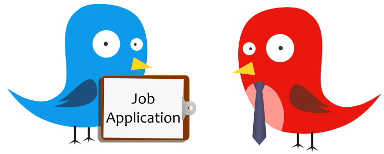


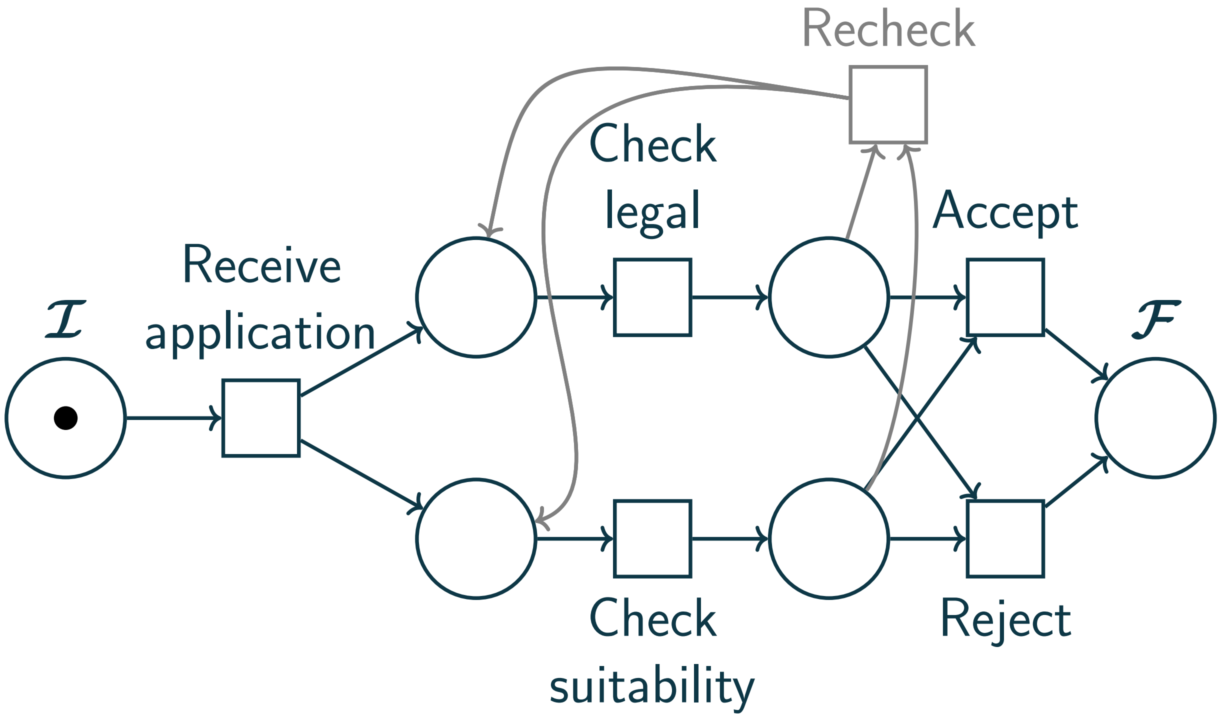
Can we condense these into a single condition?

A concise correctness condition

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

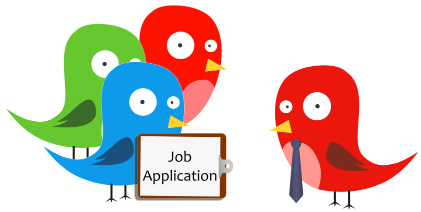
$\forall$ runs π $\exists$ run π' : $\{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon 1\}$



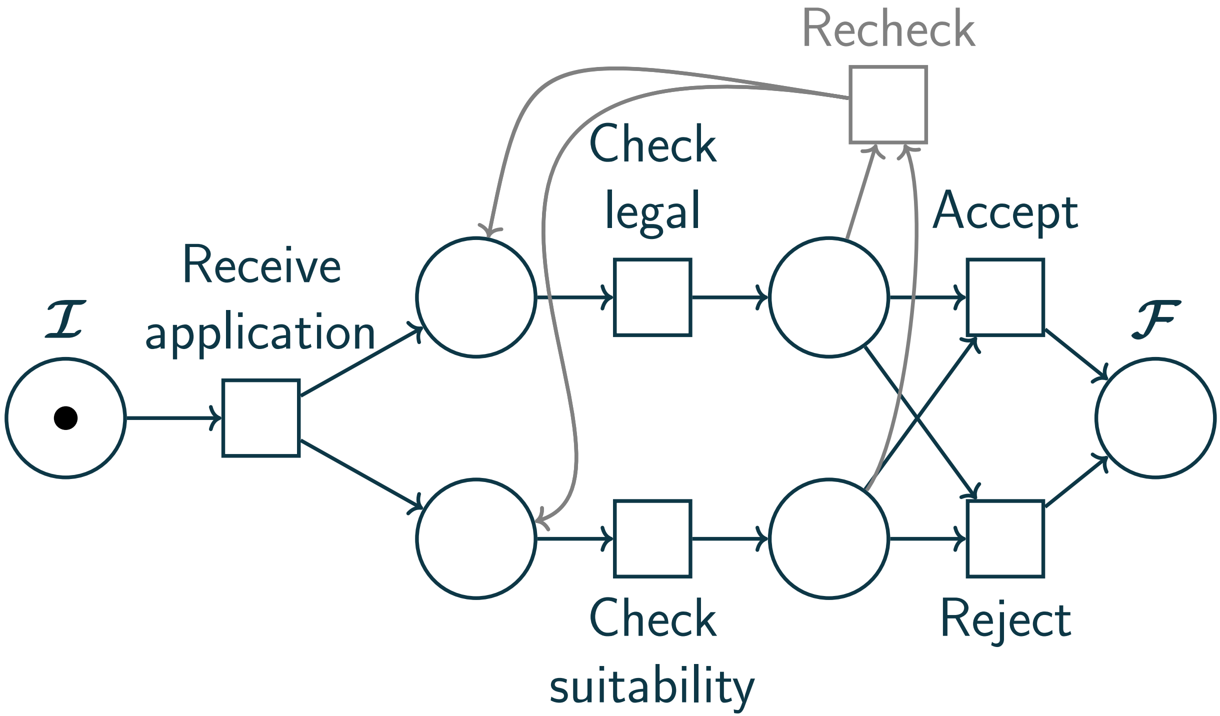


A concise correctness condition

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

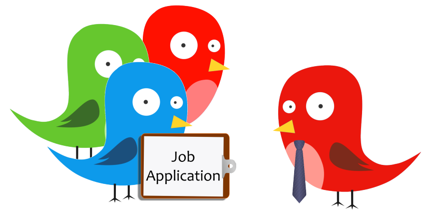


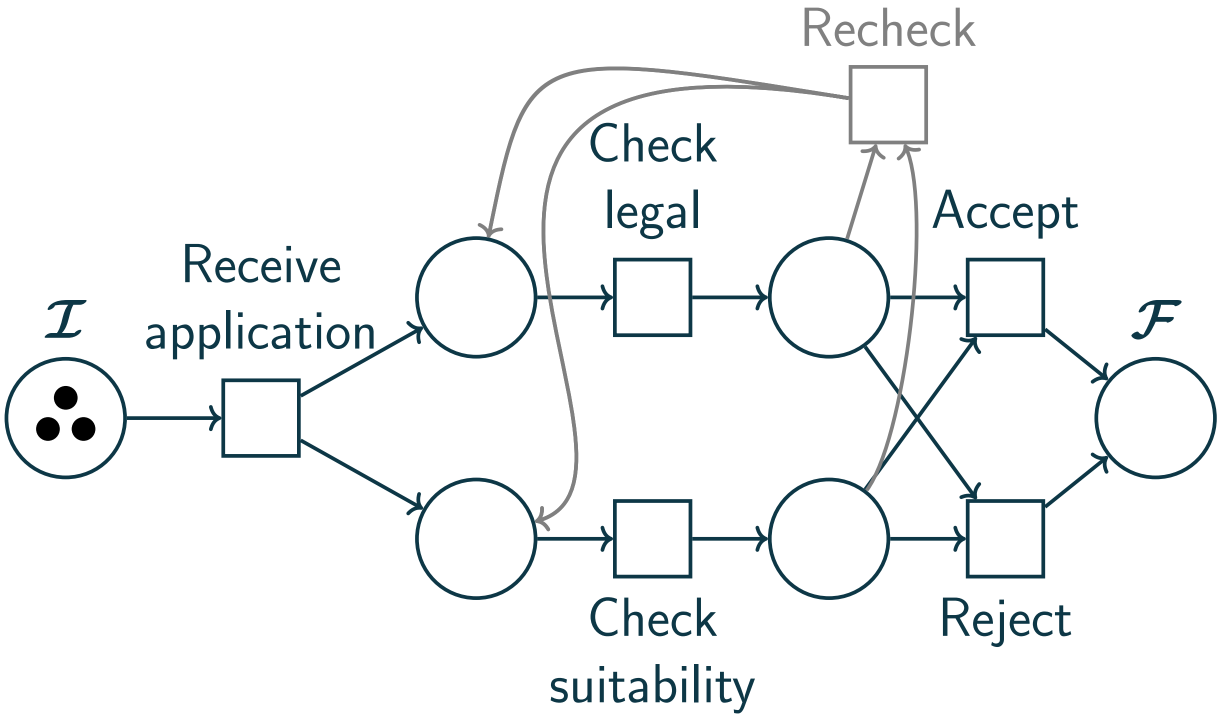
$\forall$ runs π $\exists$ run π' : $\{\mathcal{I}\colon 1\} \xrightarrow{\pi\pi'} \{\mathcal{F}\colon 1\}$



Extending soundness

***k*-soundness:**
From any marking reachable from $\{\mathcal{I}\colon {\color{red}k}\}$, the final marking $\{\mathcal{F}\colon {\color{red}k}\}$ 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:
$\forall k\colon k$-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:
$\forall k\colon$ k-sound

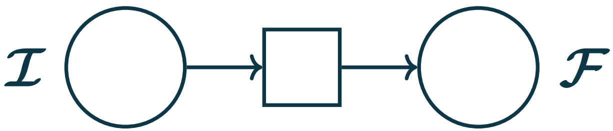
Structural soundness:
$\exists k\colon$ k-sound

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

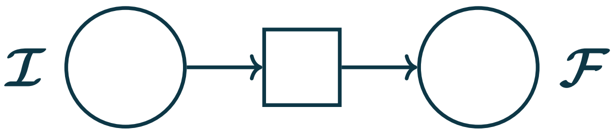


Variants of soundness

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

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

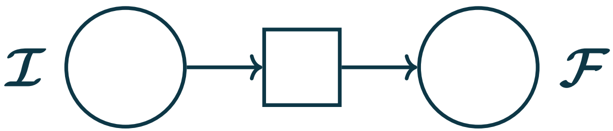
Structural soundness:
$\exists k$: k-sound



✓

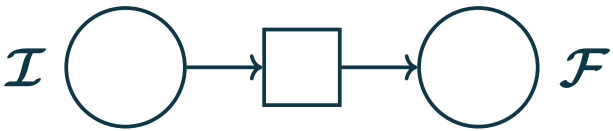
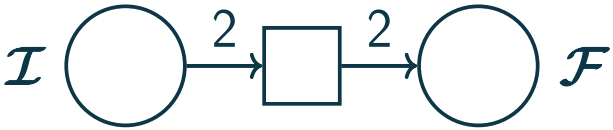
Variants of soundness

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
	✓	✓

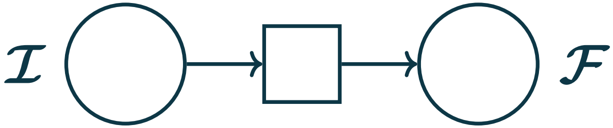
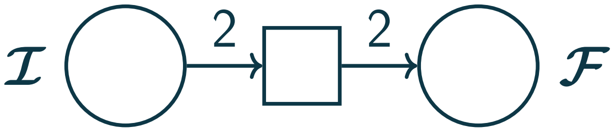
Variants of soundness

***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
	✓	✓
		

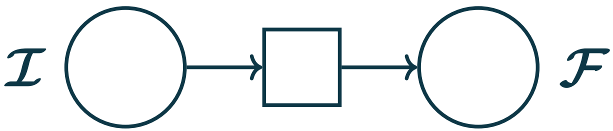
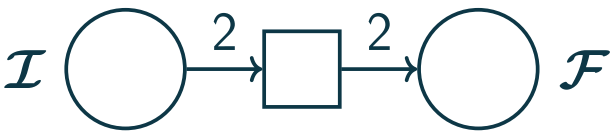
Variants of soundness

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

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

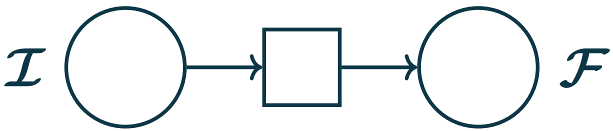
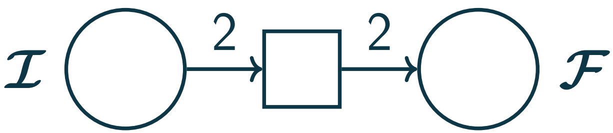
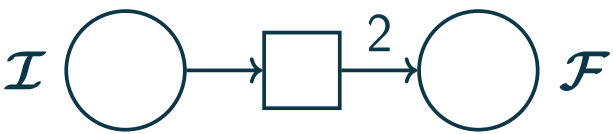
Variants of soundness

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

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

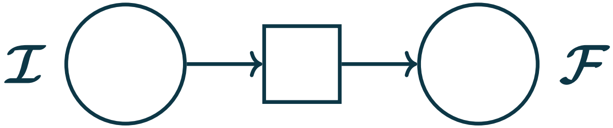
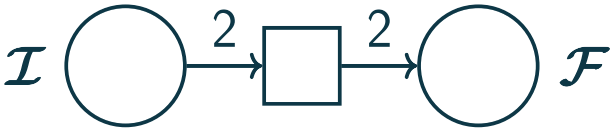
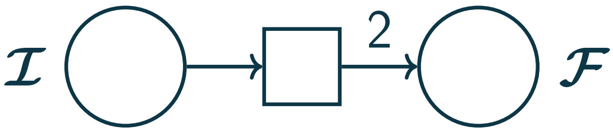
Variants of soundness

***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
		

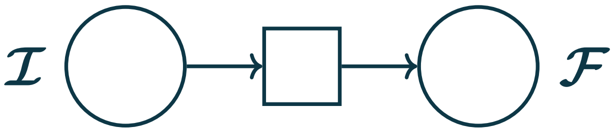
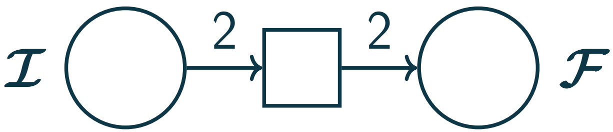
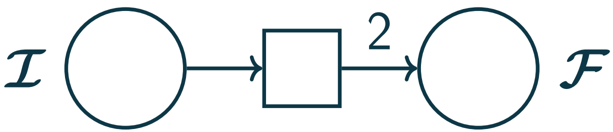
Variants of soundness

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
	✓	✓
	✗ Not 1-sound	✓ 2-sound
	✗	✗

Verifying soundness

Complexity

Generalised soundness	
Structural soundness	

Verifying soundness

	Complexity
Generalised soundness	PSPACE-complete
Structural soundness	

Verifying soundness

	Complexity
Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Verifying soundness

Complexity

Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

Verifying soundness

Complexity

Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Known incomplete algorithms:

Reduction rules

$$\begin{tikzcd} \boxed{\text{Net}} \arrow[d, "\text{Reduce}"] \\ \boxed{\text{Reduced net}} \end{tikzcd}$$

Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

Verifying soundness

Complexity

Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

Known incomplete algorithms:

Reduction rules

Net

Reduce

Reduced net

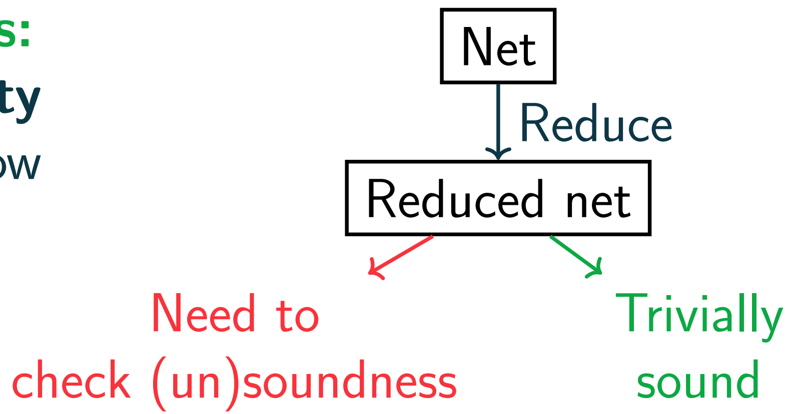
Trivially sound

Verifying soundness

	Complexity
Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Known incomplete algorithms:

Reduction rules



Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

Verifying soundness

Complexity

Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Our work

Integer unboundedness
Continuous soundness

Structural quasi-soundness via continuous reachability

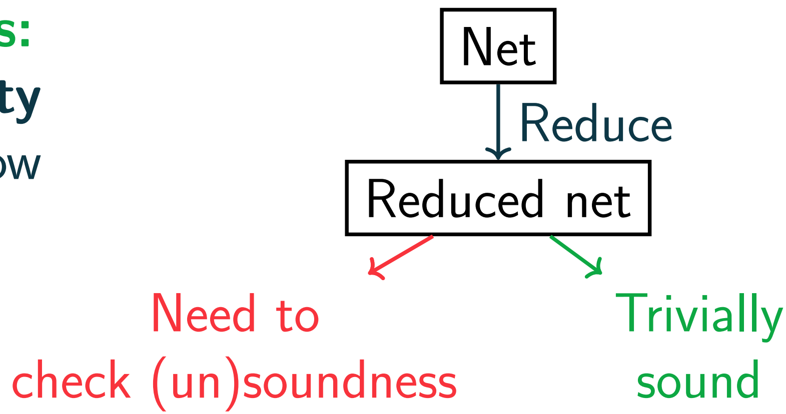
Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

Known incomplete algorithms:

Reduction rules



Verifying soundness

	Complexity
Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Our work

Integer unboundedness
Continuous soundness 1

Structural quasi-soundness via continuous reachability

Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

Known incomplete algorithms:

Reduction rules

Net
Reduce
Reduced net

Need to check (un)soundness

Trivially sound

Verifying soundness

Complexity

Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Known exact algorithms:

Use Petri net reachability

Exhaustive exploration is slow

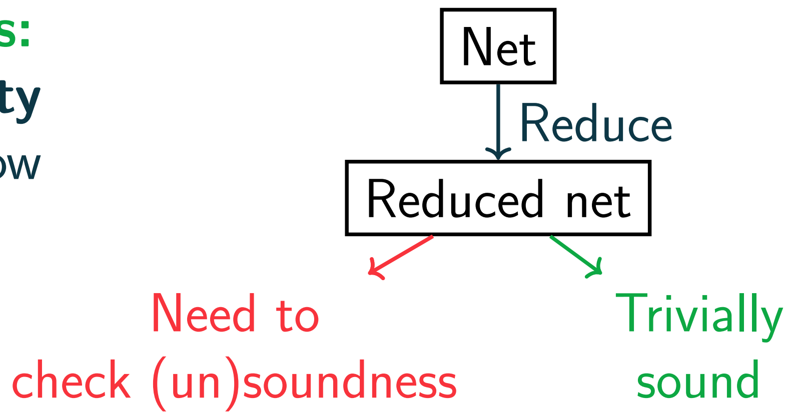
Our work

Integer unboundedness
Continuous soundness 1

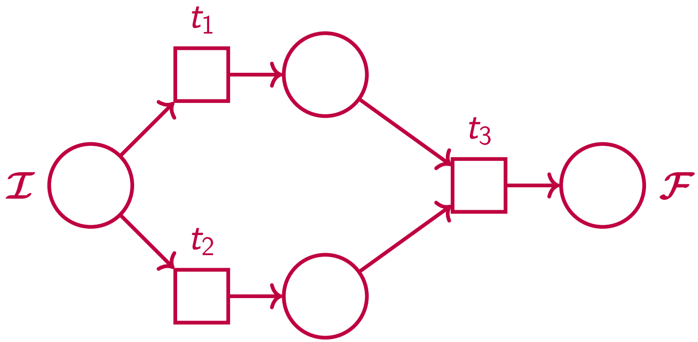
Structural quasi-soundness via continuous reachability 2

Known incomplete algorithms:

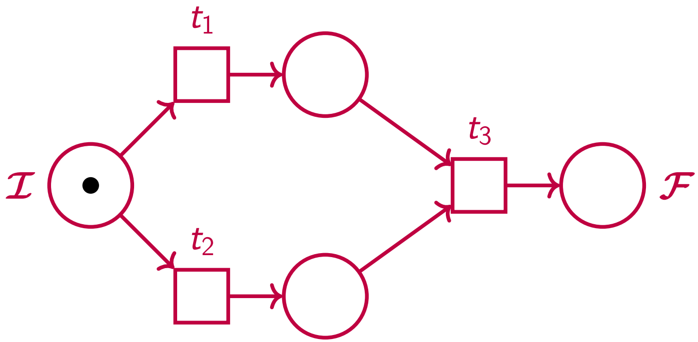
Reduction rules



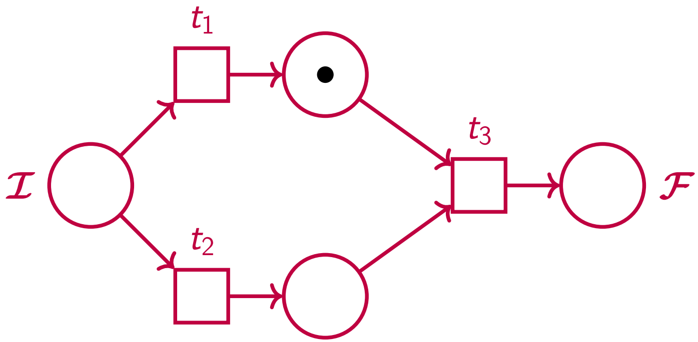
Reachability Relaxations



Reachability Relaxations

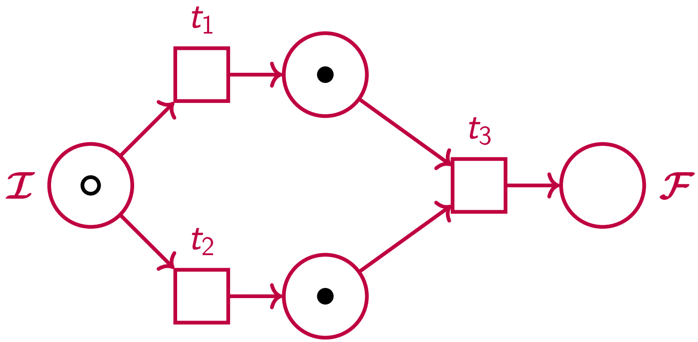


Reachability Relaxations



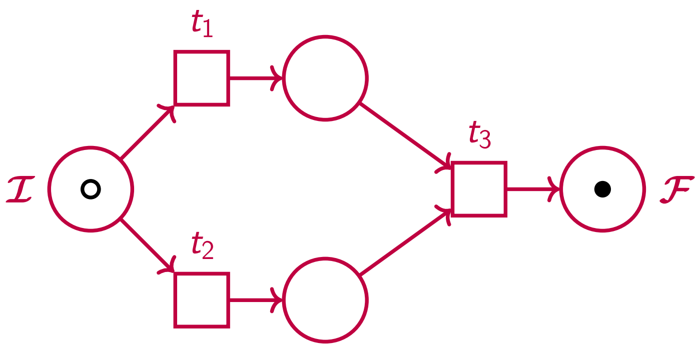
t_1

Reachability Relaxations



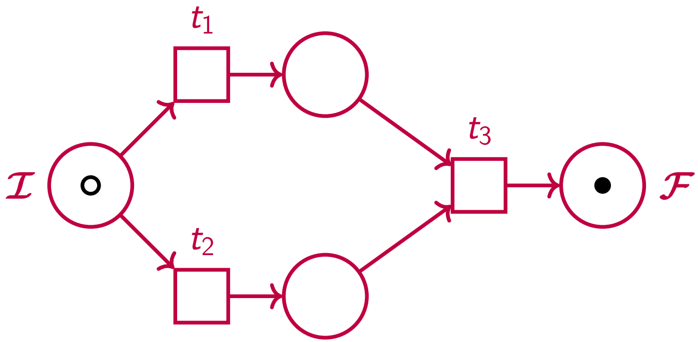
$t_1 \quad t_2$

Reachability Relaxations



$t_1 \quad t_2 \quad t_3$

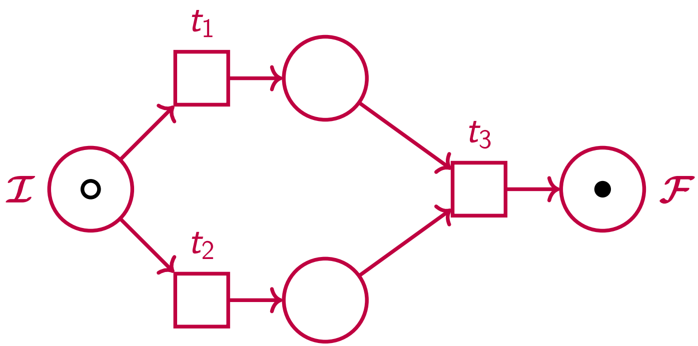
Reachability Relaxations

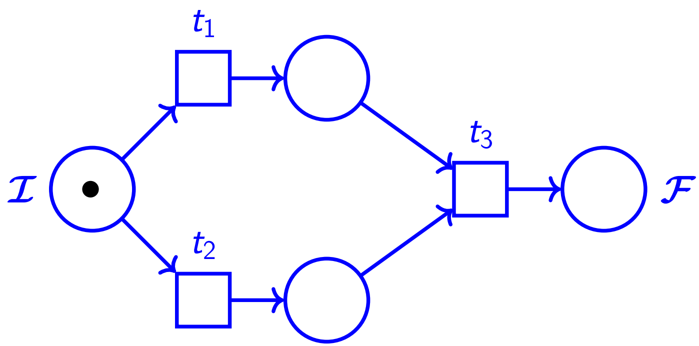


t_1 t_2 t_3

Integer semantics

Reachability Relaxations

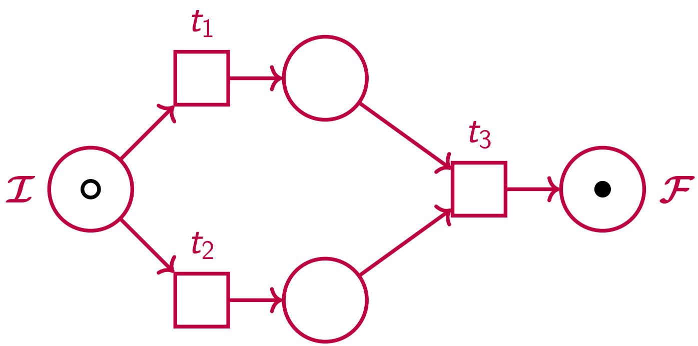




$t_1 \quad t_2 \quad t_3$

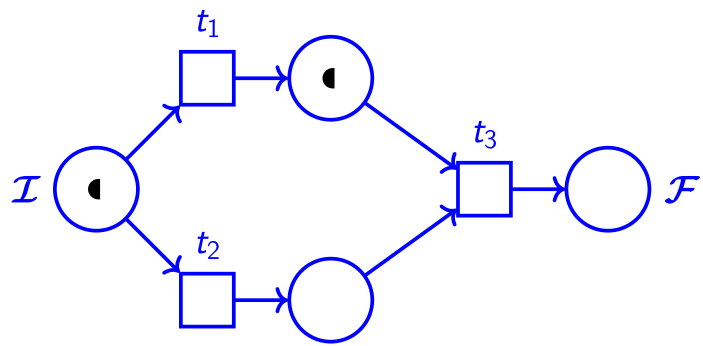
Integer semantics

Reachability Relaxations



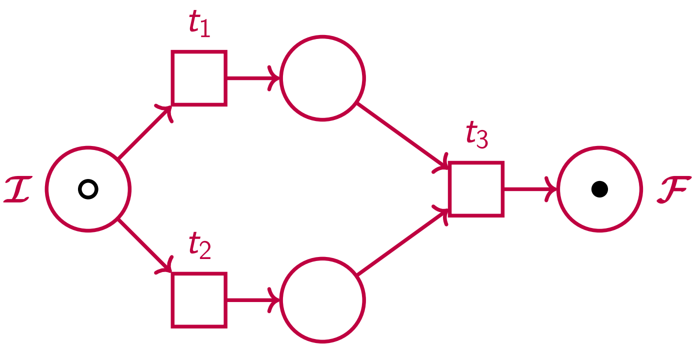
t_1 t_2 t_3

Integer semantics



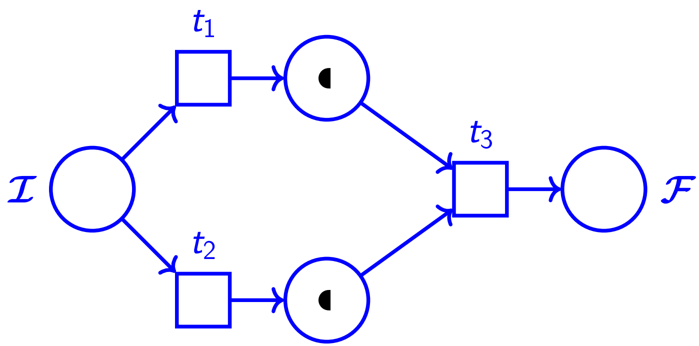
$1/2 t_1$

Reachability Relaxations



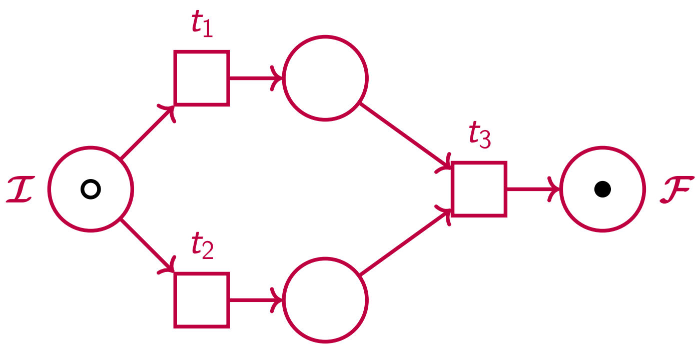
$t_1 \quad t_2 \quad t_3$

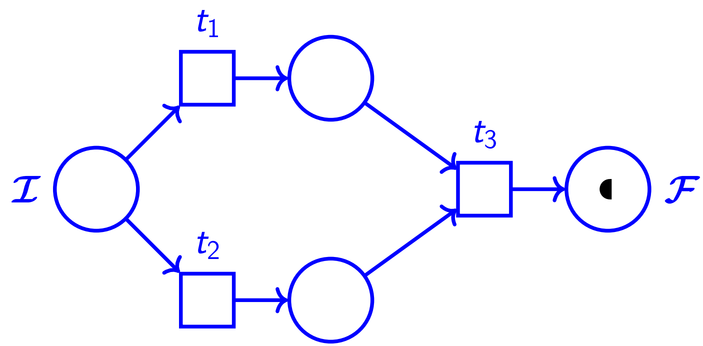
Integer semantics



$1/2 t_1 \quad 1/2 t_2$

Reachability Relaxations



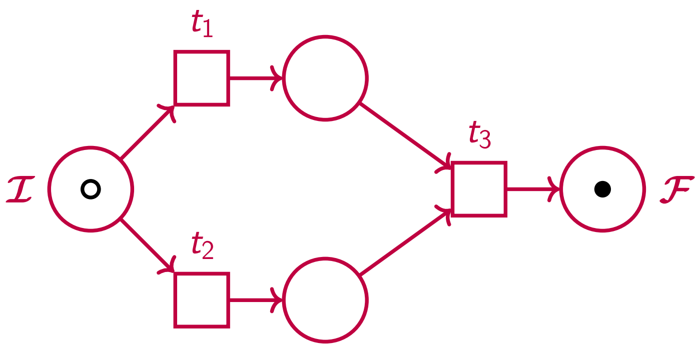


$t_1 \quad t_2 \quad t_3$

Integer semantics

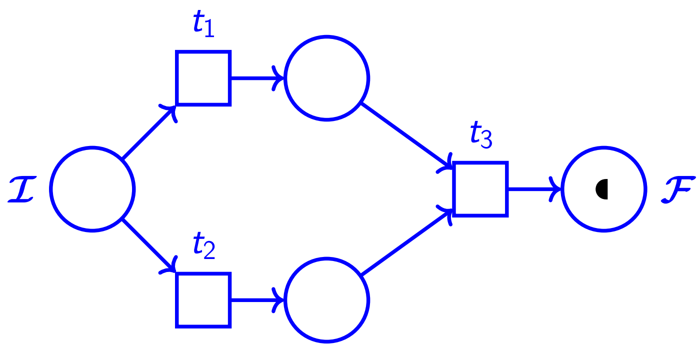
$1/2t_1 \quad 1/2t_2 \quad 1/2t_3$

Reachability Relaxations



$t_1 \quad t_2 \quad t_3$

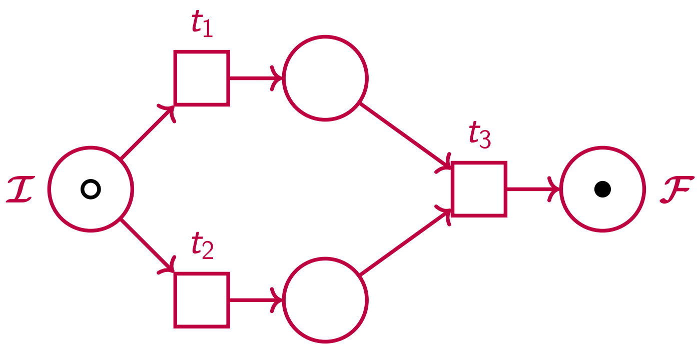
Integer semantics



$1/2t_1 \quad 1/2t_2 \quad 1/2t_3$

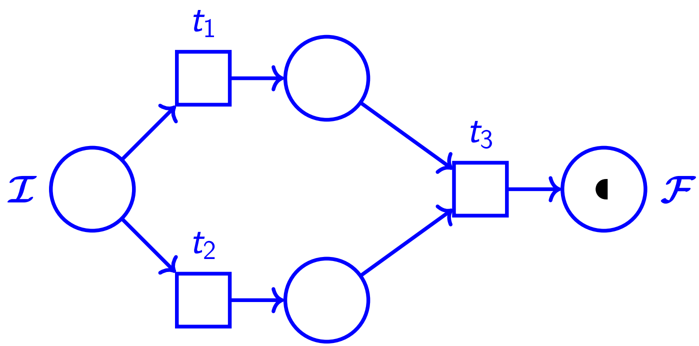
Continuous semantics

Reachability Relaxations



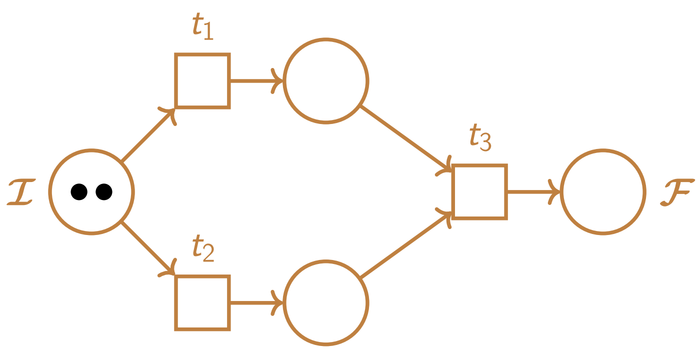
$t_1 \quad t_2 \quad t_3$

Integer semantics

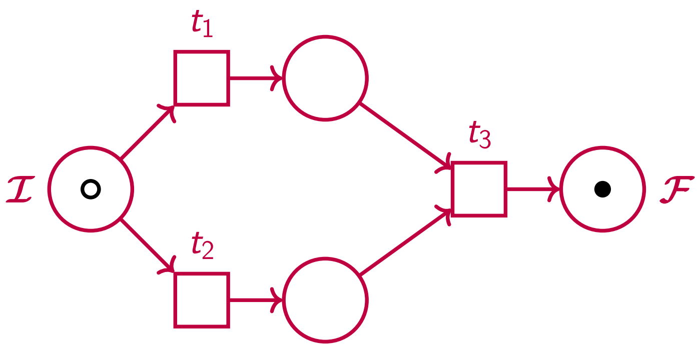


$1/2t_1 \quad 1/2t_2 \quad 1/2t_3$

Continuous semantics

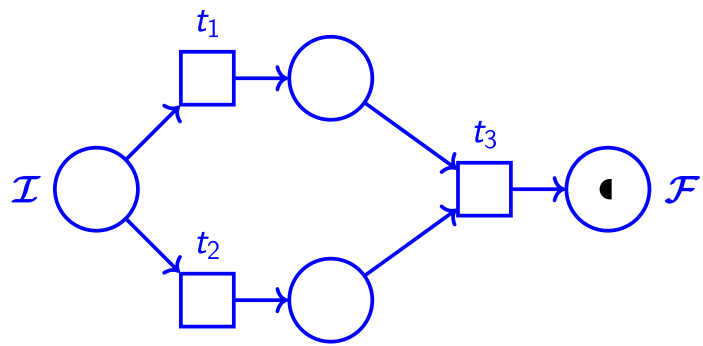


Reachability Relaxations



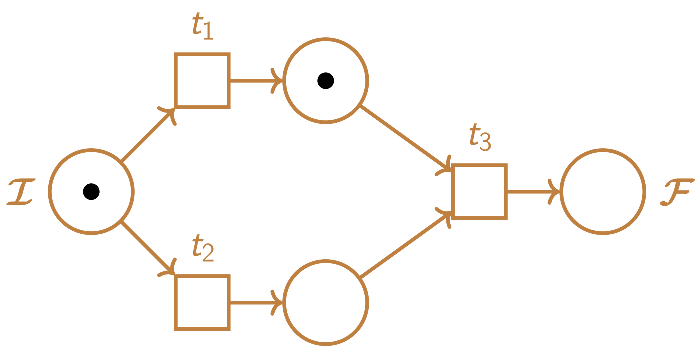
$t_1 \quad t_2 \quad t_3$

Integer semantics



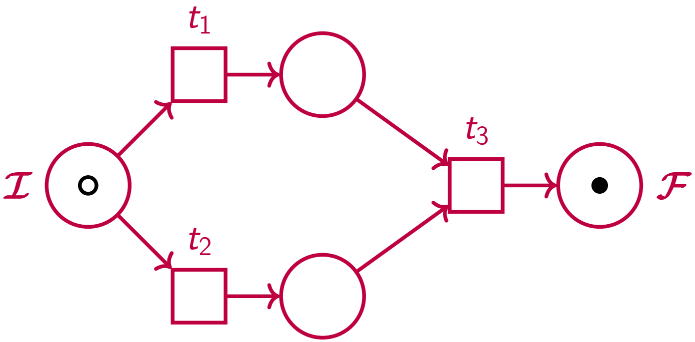
$1/2 t_1 \quad 1/2 t_2 \quad 1/2 t_3$

Continuous semantics



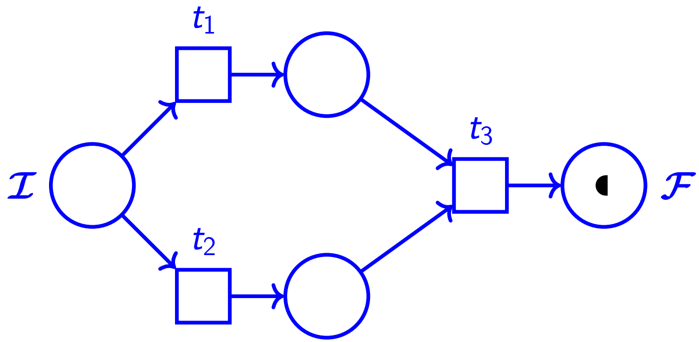
t_1

Reachability Relaxations



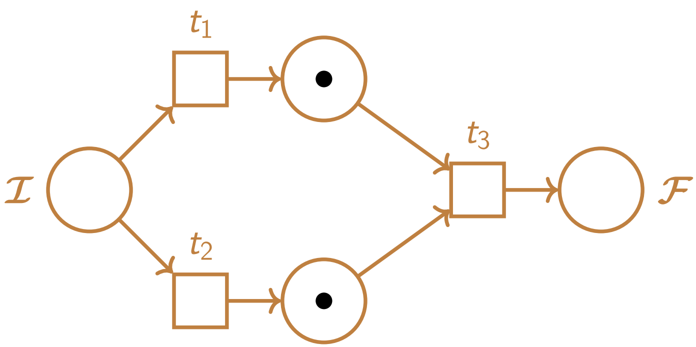
$t_1 \quad t_2 \quad t_3$

Integer semantics



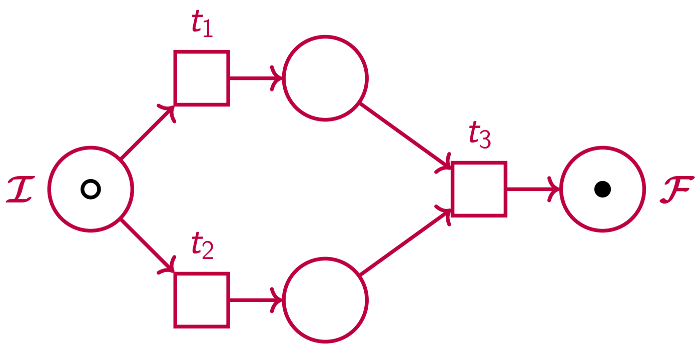
$1/2t_1 \quad 1/2t_2 \quad 1/2t_3$

Continuous semantics



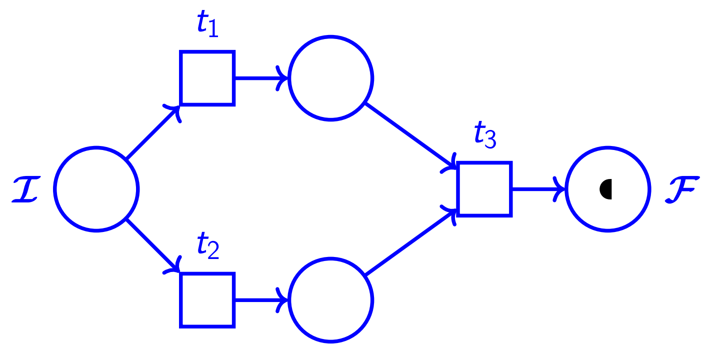
$t_1 \quad t_2$

Reachability Relaxations



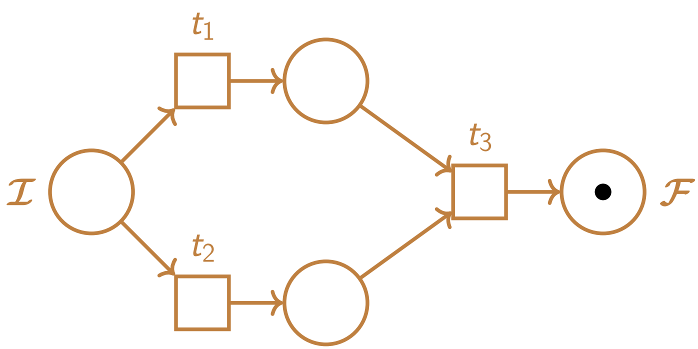
$t_1 \quad t_2 \quad t_3$

Integer semantics



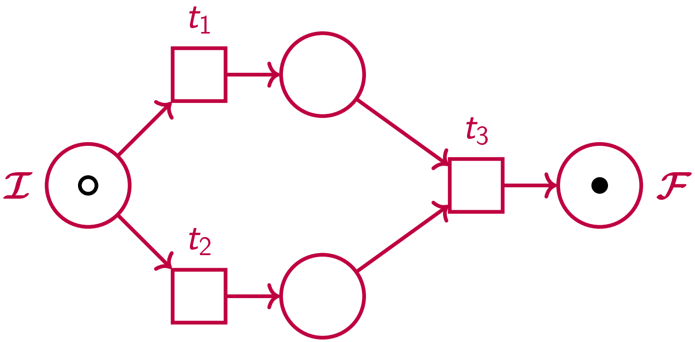
$1/2t_1 \quad 1/2t_2 \quad 1/2t_3$

Continuous semantics



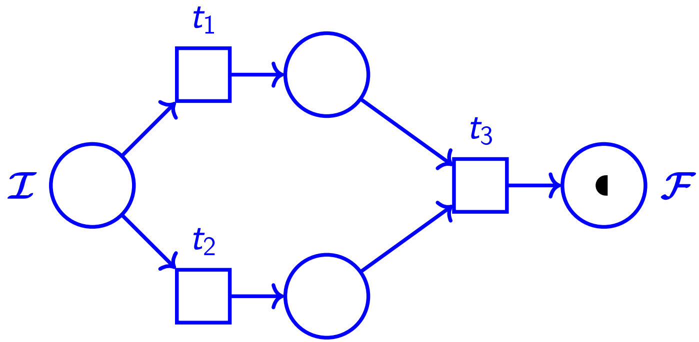
$t_1 \quad t_2 \quad t_3$

Reachability Relaxations



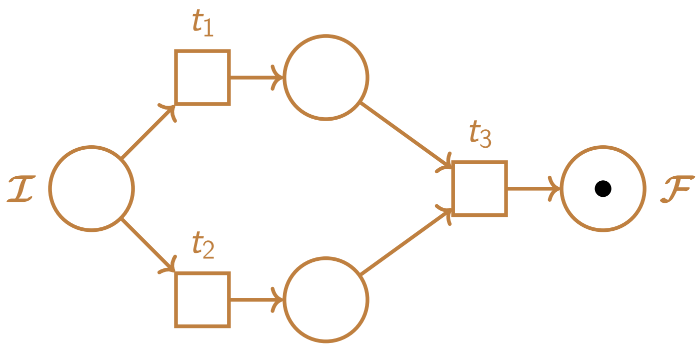
t_1 t_2 t_3

Integer semantics



$1/2t_1$ $1/2t_2$ $1/2t_3$

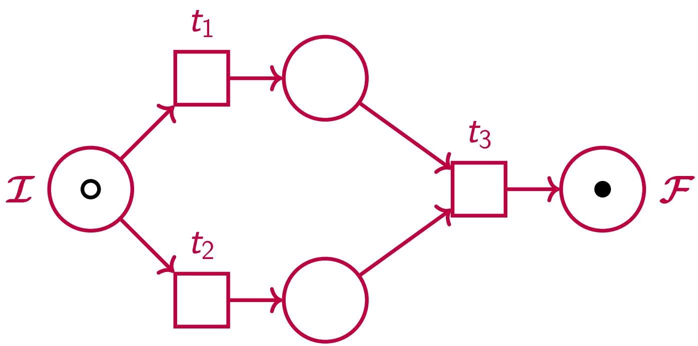
Continuous semantics



t_1 t_2 t_3

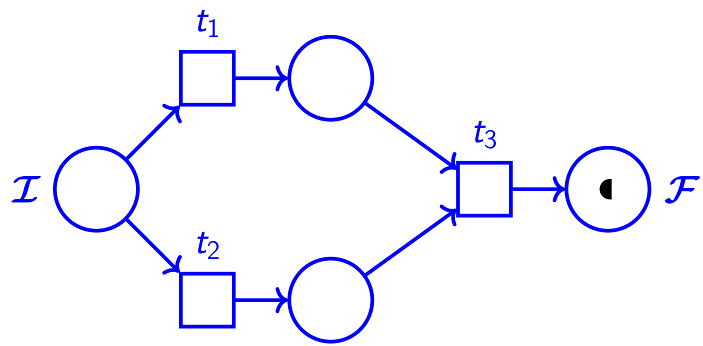
Reachability with many initial tokens

Reachability Relaxations



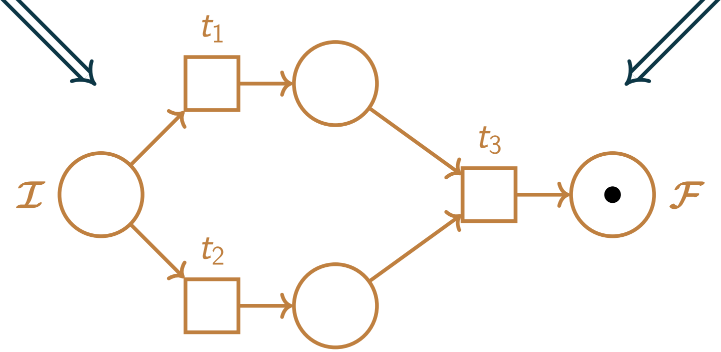
$t_1 \quad t_2 \quad t_3$

Integer semantics



$1/2 t_1 \quad 1/2 t_2 \quad 1/2 t_3$

Continuous semantics



$t_1 \quad t_2 \quad t_3$

Reachability with many initial tokens

Integer Semantics and Generalised Soundness

Integer Semantics and Generalised Soundness

Integer unboundedness:
$\{\mathcal{I}: 1\}$ reaches m reaches m' with $m' > m$
under **integer** semantics

Integer Semantics and Generalised Soundness

Integer unboundedness:

$\{\mathcal{I}\colon 1\}$ reaches m reaches m' with $m' > m$

under **integer** semantics

$$\boxed{\text{A net } N \text{ is integer unbounded} \implies N \text{ is not generalised sound}}$$

Integer Semantics and Generalised Soundness

Integer unboundedness:
$\{\mathcal{I} : 1\}$ reaches m reaches m' with $m' > m$
under **integer** semantics

$$\boxed{\text{A net } N \text{ is integer unbounded} \implies N \text{ is not generalised sound}}$$

Checking integer unboundedness:
PTIME via Integer Linear Programming

Continuous Semantics and Generalised Soundness

Continuous Semantics and Generalised Soundness

Continuous Soundness:

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

under **continuous** semantics

Continuous Semantics and Generalised Soundness

Continuous Soundness:
$\{\mathcal{I}\colon 1\}$ reaches $m \implies m$ reaches $\{\mathcal{F}\colon 1\}$
under **continuous** semantics

$$\boxed{\text{A net } N \text{ is not continuously sound} \implies N \text{ is not generalised sound}}$$

Continuous Semantics and Generalised Soundness

Continuous Soundness:
$\{\mathcal{I}\colon 1\}$ reaches $m \implies m$ reaches $\{\mathcal{F}\colon 1\}$
under **continuous** semantics

A net N is not continuously sound $\implies$ N is not generalised sound

Checking continuous soundness:
coNP-complete

Continuous Semantics and Generalised Soundness

Continuous Soundness:
$\{\mathcal{I}\colon 1\}$ reaches $m \implies m$ reaches $\{\mathcal{F}\colon 1\}$
under **continuous** semantics

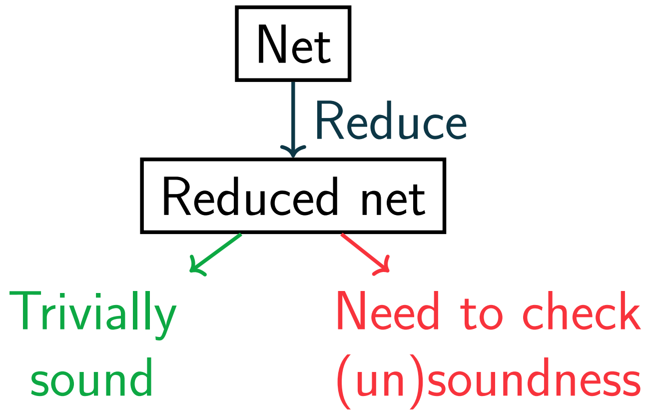
$\boxed{\text{A net } N \text{ is not continuously sound} \implies N \text{ is not generalised sound}}$

Checking continuous soundness:
coNP-complete

Bonus:
Continuous sound $\equiv$ Generalised sound $\equiv$ Structurally sound
on *Free-Choice Nets*

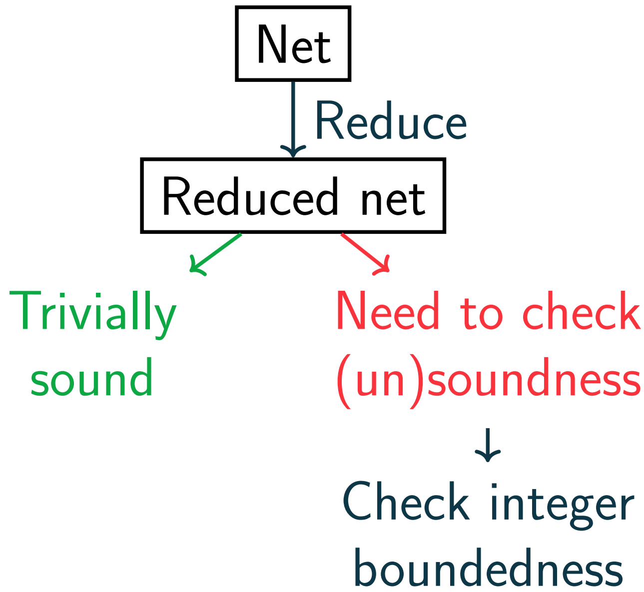
Generalised Soundness: Our approach

Reduction rules



Generalised Soundness: Our approach

Reduction rules



Generalised Soundness: Our approach

Reduction rules

Net

↓ Reduce

Reduced net

Trivially sound

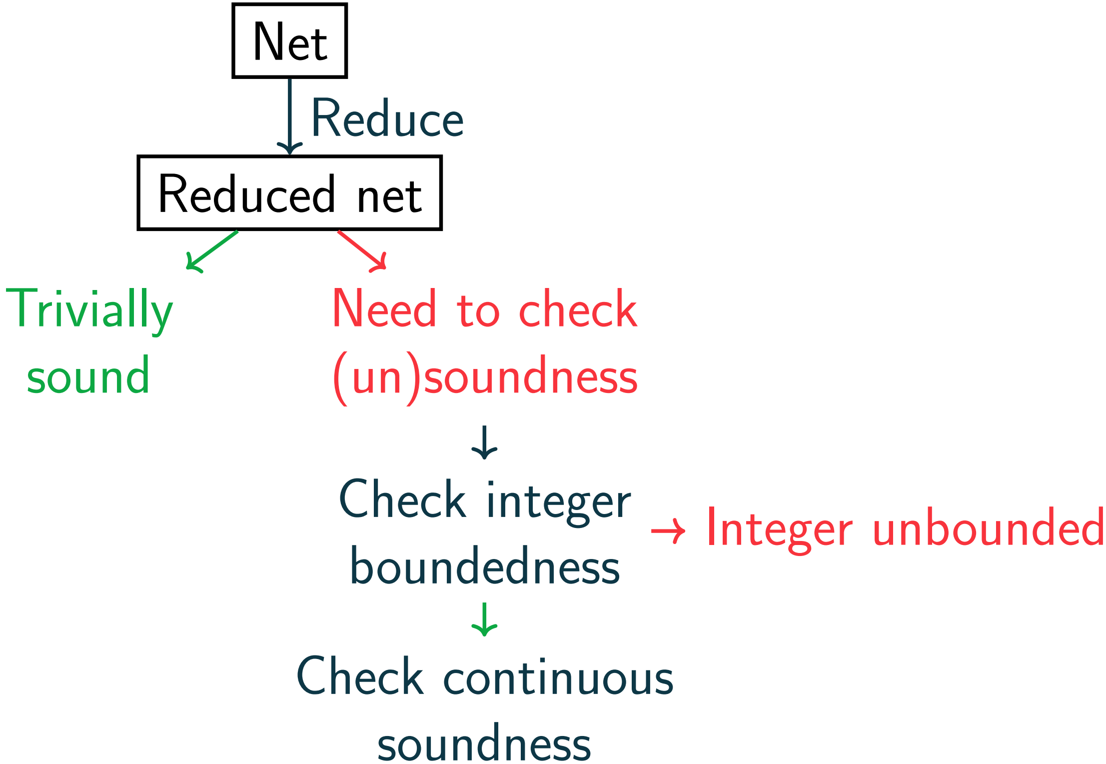
Need to check (un)soundness

↓

Check integer boundedness → Integer unbounded

Generalised Soundness: Our approach

Reduction rules



Generalised Soundness: Our approach

Reduction rules

Net

Reduce

Reduced net

Trivially sound

Need to check (un)soundness

Check integer boundedness → Integer unbounded

Check continuous soundness → Continuously unsound

Generalised Soundness: Our approach

Reduction rules

Net

Reduce

Reduced net

Trivially sound

Need to check (un)soundness

Check integer boundedness → Integer unbounded

Continuously sound ← Check continuous soundness → Continuously unsound

Exhaustive exploration needed

Generalised Soundness: Our approach

Reduction rules

Net

↓ Reduce

Reduced net

↙ Trivially sound

↘ Need to check (un)soundness

↓

Check integer boundedness → Integer unbounded

↓

Check continuous soundness

← Continuously sound

→ Continuously unsound

Exhaustive exploration needed

Benchmarks: 1976 industrial nets

Generalised Soundness: Our approach

Reduction rules

Net

↓ Reduce

Reduced net

642/1976 Trivially sound

Need to check (un)soundness

↓

Check integer boundedness → Integer unbounded

↓

Continuously sound ← Check continuous soundness → Continuously unsound

Exhaustive exploration needed

Benchmarks:
1976 industrial nets

Generalised Soundness: Our approach

Reduction rules

Net

↓ Reduce

Reduced net

642/1976 Trivially sound

Need to check (un)soundness

1111/1976

↓

Check integer boundedness → Integer unbounded

↓

Continuously sound ← Check continuous soundness → Continuously unsound

Exhaustive exploration needed

Benchmarks:
1976 industrial nets

Generalised Soundness: Our approach

Reduction rules

Net

↓ Reduce

Reduced net

642/1976 Trivially sound

Need to check (un)soundness

↓

Check integer boundedness → Integer unbounded

1111/1976

↓

Continuously sound ← Check continuous soundness → Continuously unsound

Exhaustive exploration needed

223/1976

Benchmarks:
1976 industrial nets

Verifying soundness

Complexity

Generalised soundness	PSPACE-complete
Structural soundness	EXPSPACE-complete

Integer unboundedness
Continuous soundness 1

Structural quasi-soundness via continuous reachability 2

Quasi-Soundness

k-**soundness:**

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

Quasi-Soundness

k-soundness:

$\{\mathcal{I}\colon k\}$ reaches m

$\implies$ m reaches $\{\mathcal{F}\colon k\}$

k-quasi-soundness:

$\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$

Quasi-Soundness

k-**soundness:**
$\{\mathcal{I}\colon k\}$ reaches m
$\implies m$ reaches $\{\mathcal{F}\colon k\}$

$\Longrightarrow$

k-**quasi-soundness:**
$\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$

Quasi-Soundness

k-**soundness:**
$\{\mathcal{I}\colon k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}\colon k\}$

$\Longrightarrow$

k-**quasi-soundness:**
$\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

Quasi-Soundness

k-soundness:
$\{\mathcal{I}: k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}: k\}$

$\implies$

k-quasi-soundness:
$\{\mathcal{I}: k\}$ reaches $\{\mathcal{F}: k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

N is structurally sound
$\implies$ N is k_N sound
[Tiplea & Marinescu, 2005]

Quasi-Soundness

k-soundness:
$\{\mathcal{I}: k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}: k\}$

$\Longrightarrow$

k-quasi-soundness:
$\{\mathcal{I}: k\}$ reaches $\{\mathcal{F}: k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

Net
$\downarrow$ Reduce
Reduced net
$\swarrow$
Trivially sound

N is structurally sound
$\implies$ N is k_N sound
[Tiplea & Marinescu, 2005]

Quasi-Soundness

k-soundness:
$\{\mathcal{I}: k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}: k\}$

$\implies$

k-quasi-soundness:
$\{\mathcal{I}: k\}$ reaches $\{\mathcal{F}: k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

Net

Reduce

Reduced net

Trivially sound

Find k_N

N is structurally sound
$\implies$ N is k_N sound
[Tiplea & Marinescu, 2005]

Quasi-Soundness

k-soundness:
$\{\mathcal{I}: k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}: k\}$

$\Longrightarrow$

k-quasi-soundness:
$\{\mathcal{I}: k\}$ reaches $\{\mathcal{F}: k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

N is structurally sound
$\implies$ N is k_N sound
[Tiplea & Marinescu, 2005]

Net $\xrightarrow{\text{Reduce}}$ Reduced net

Reduced net $\to$ Trivially sound

Reduced net $\to$ Find k_N $\to$ None $\implies$ not struct. sound

Quasi-Soundness

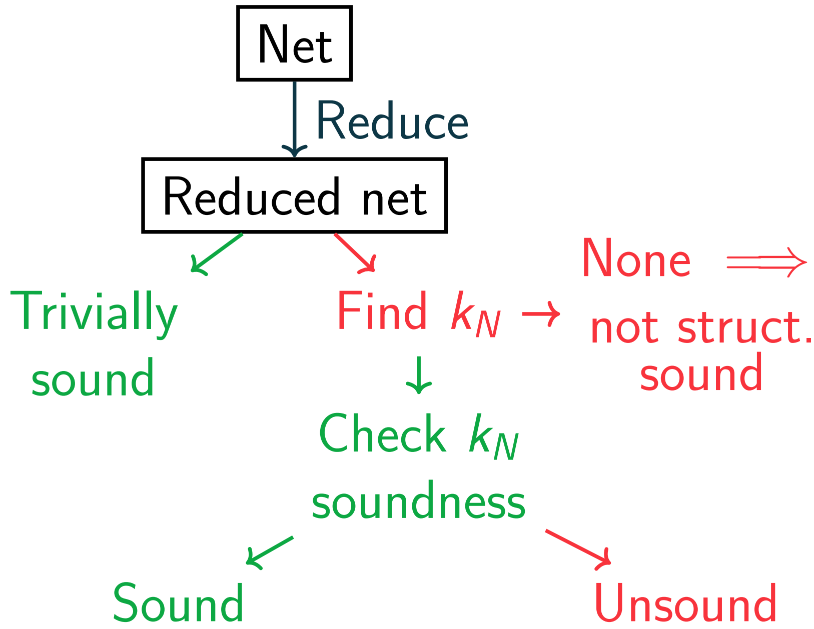
k-soundness:
$\{\mathcal{I}\colon k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}\colon k\}$

$\Longrightarrow$

k-quasi-soundness:
$\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

N is structurally sound
$\implies$ N is k_N sound
[Tiplea & Marinescu, 2005]



Quasi-Soundness

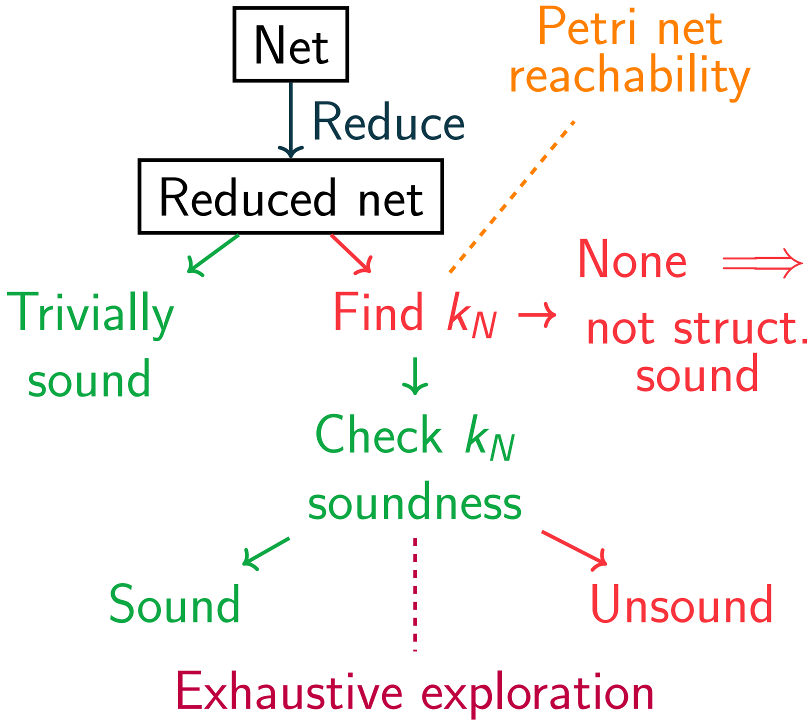
k-**soundness:**
$\{\mathcal{I}: k\}$ reaches m
$\implies m$ reaches $\{\mathcal{F}: k\}$

$\Longrightarrow$

k-**quasi-soundness:**
$\{\mathcal{I}: k\}$ reaches $\{\mathcal{F}: k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

N is structurally sound
$\implies N$ is k_N sound
[Tiplea & Marinescu, 2005]



Quasi-Soundness

k-soundness:
$\{\mathcal{I}: k\}$ reaches m
$\implies$ m reaches $\{\mathcal{F}: k\}$

k-quasi-soundness:
$\{\mathcal{I}: k\}$ reaches $\{\mathcal{F}: k\}$

$\mathbf{k_N}$:
min so that N
is k-quasi-sound

N is structurally sound
$\implies$ N is k_N sound
[Tiplea & Marinescu, 2005]

Our approach:
Continuous reachability

Petri net
reachability

Net

Reduce

Reduced net

Trivially
sound

Find k_N $\to$

None $\implies$
not struct.
sound

Check k_N
soundness

Sound

Unsound

Exhaustive exploration

Quasi-Soundness via Continuous Reachability

k_N exists $\equiv$ $\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$ under the **continuous semantics**

Quasi-Soundness via Continuous Reachability

k_N exists $\equiv$ $\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$ under the **continuous semantics**

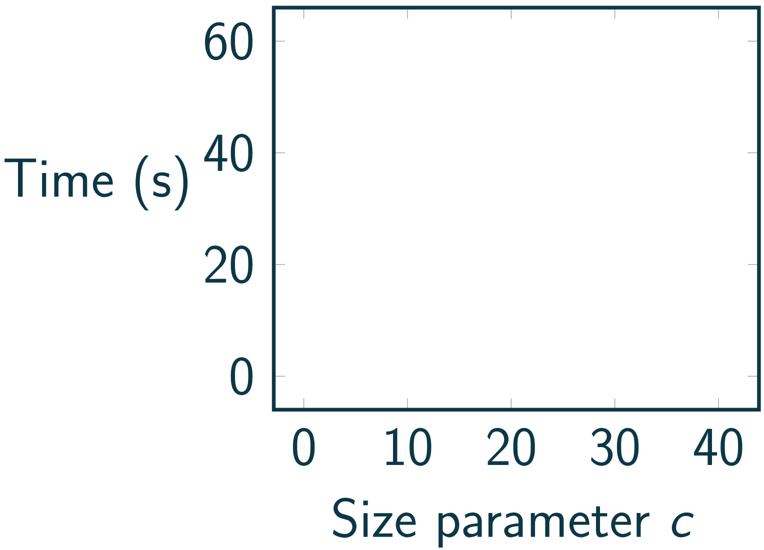
Approach:
Implement continuous semantics into SMT

Quasi-Soundness via Continuous Reachability

k_N exists $\equiv$ $\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$ under the **continuous semantics**

Approach:

Implement continuous semantics into SMT

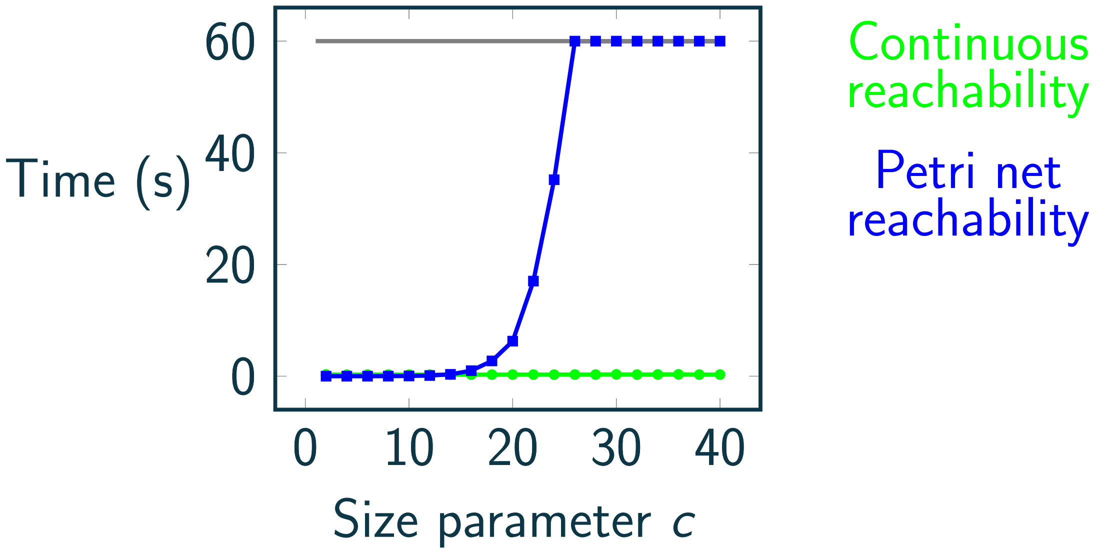


Quasi-Soundness via Continuous Reachability

k_N exists $\equiv$ $\{\mathcal{I}\colon k\}$ reaches $\{\mathcal{F}\colon k\}$ under the **continuous semantics**

Approach:

Implement continuous semantics into SMT



Conclusion

Workflow nets are a model for business processes

Generalised and **structural** soundness are challenging to verify

Approximations can provide helpful heuristics

Conclusion

Workflow nets are a model for business processes

Generalised and **structural** soundness are challenging to verify

Approximations can provide helpful heuristics

Implementation: **FastForward for Soundness**

`https://doi.org/10.6084/m9.figshare.19721674.v2`

CAV Artifact Evaluation Available

CAV Artifact Evaluation Reusable

Conclusion

Workflow nets are a model for business processes

Generalised and **structural** soundness are challenging to verify

Approximations can provide helpful heuristics

Implementation: **FastForward for Soundness**

`https://doi.org/10.6084/m9.figshare.19721674.v2`

Thanks!