On Petri nets with Hierarchical Special Arcs

S Akshay\textsuperscript{1}, S. Chakraborty\textsuperscript{1}, Ankush Das\textsuperscript{2}, Vishal Jagannath\textsuperscript{1}, Sai Sandeep\textsuperscript{1}

CONCUR, Berlin
7 Sept 2017

1: Dept of CSE, IIT Bombay,
2: CMU, USA
Preliminaries
Petri nets

- Petri net (PN) is a tuple \((P, T, F, M_0)\),
  - \(P\) is set of places, \(T\) is set of transitions,
  - \(M_0 : P \to \mathbb{N}\) is the initial marking and
  - \(F : (P \times T) \cup (T \times P) \to \mathbb{N}\) is the flow relation.
- usual definitions: marking \(M : P \to \mathbb{N}\), firability, runs...
Petri nets

- Petri net (PN) is a tuple \((P, T, F, M_0)\),
  - \(P\) is set of places, \(T\) is set of transitions,
  - \(M_0 : P \rightarrow \mathbb{N}\) is the initial marking and
  - \(F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}\) is the flow relation.
- usual definitions: marking \(M : P \rightarrow \mathbb{N}\), firability, runs...
- \(\leq\) is component-wise order over markings
Decision Problems

Definition
Given a Petri net \( N = (P, T, F, M_0) \),

- **Termination (or TERM):** Does there exist an infinite run from marking \( M_0 \)?
- **Reachability (or REACH):** Given a marking \( M \), is there a run from \( M_0 \) which reaches \( M \)?
- **Coverability (or COVER):** Given a marking \( M \), is there a marking \( M' \geq M \) which is reachable from \( M_0 \)?
Decision Problems

Definition

Given a Petri net $N = (P, T, F, M_0)$,

- **Termination** (or **TERM**): Does there exist an infinite run from marking $M_0$?
- **Reachability** (or **REACH**): Given a marking $M$, is there a run from $M_0$ which reaches $M$?
- **Coverability** (or **COVER**): Given a marking $M$, is there a marking $M' \geq M$ which is reachable from $M_0$?
- **Deadlock-freeness** (or **DLFREE**): Does there exist a marking $M$ reachable from $M_0$, such that no transition is firable at $M$?
- **(Place-)Boundedness**: Does some (a given) place get unboundedly many tokens?
We can add a few special arcs into Petri nets.

- Inhibitor arcs
We can add a few special arcs into Petri nets.

- Inhibitor arcs
- Reset arcs
• We can add a few special arcs into Petri nets.
  • Inhibitor arcs
  • Reset arcs
  • Transfer arcs
We can add a few special arcs into Petri nets.

- Inhibitor arcs
- Reset arcs
- Transfer arcs

Redefine flow

\[ F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N} \cup \{I, R\} \cup \{S_p \mid p \in P\} \]
• Inhibitors are zero-tests

• Petri nets with 2 inhibitors model 2-counter machines.
• Inhibitors are zero-tests
• Petri nets with 2 inhibitors model 2-counter machines.
• One way to deal with this: impose hierarchy on places [Rei08].
  • A total order $\sqsupseteq$ on $P$ such that
    \[ \forall (p, t) \in P \times T, \quad F(p, t) \in I \implies (\forall q \sqsubseteq p, \ F(q, t) \in I) \]
• Inhibitors are zero-tests
• Petri nets with 2 inhibitors model 2-counter machines.
• One way to deal with this: impose hierarchy on places [Rei08].
  • A total order $\sqsubseteq$ on $P$ such that
    $\forall (p, t) \in P \times T$, $F(p, t) \in I \implies (\forall q \sqsubseteq p, F(q, t) \in I)$

\[
\begin{array}{c}
p_1 \\
\text{Tr}^P \\
p_2 \\
p_3 \\
\text{Reset} \\
p
\end{array}
\]
Adding priorities to job scheduling!
Adding priorities to job scheduling!
Adding priorities to job scheduling!

- The case of a single inhibitor arc/transition is an interesting and well-studied subcase!
State of the art: What is known about these problems?
State of the art

<table>
<thead>
<tr>
<th>Term</th>
<th>Cover</th>
<th>Reach</th>
<th>DLFree</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>✓ [May84, Ler12]</td>
</tr>
<tr>
<td>R/T-PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>× [DFS98]</td>
</tr>
<tr>
<td>I-PN</td>
<td>× [Min67]</td>
<td>× [Min67]</td>
<td>× [Min67]</td>
</tr>
<tr>
<td>HIPN</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
</tr>
<tr>
<td>R+HIPN</td>
<td></td>
<td>× [[DFS98], Thm 4]</td>
<td>× [[DFS98], Thm 4]</td>
</tr>
<tr>
<td>T+HIPN</td>
<td></td>
<td>× [[DFS98], Thm 4]</td>
<td>× [[DFS98], Thm 4]</td>
</tr>
</tbody>
</table>

Questions:
- What happens when resets/transfers are added to HIPN?
- Understanding the boundary of decidability and undecidability...
- Can we “weaken” the notion of Hierarchy?
### State of the art

<table>
<thead>
<tr>
<th>Term</th>
<th>Cover</th>
<th>Reach</th>
<th>DLFree</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>✓ [May84, Ler12]</td>
</tr>
<tr>
<td>R/T-PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>X [DFS98]</td>
</tr>
<tr>
<td>HIPN</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
</tr>
<tr>
<td>R+HIPN</td>
<td></td>
<td></td>
<td>X[[DFS98], Thm 4]</td>
</tr>
<tr>
<td>T+HIPN</td>
<td></td>
<td></td>
<td>X[[DFS98], Thm 4]</td>
</tr>
</tbody>
</table>

**Questions:**

- What happens when resets/transfers are added to HIPN?
## State of the art

<table>
<thead>
<tr>
<th>Term</th>
<th>Cover</th>
<th>Reach</th>
<th>DLTFree</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>✓ [May84, Ler12]</td>
</tr>
<tr>
<td>R/T-PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>✓ [DFS98]</td>
</tr>
<tr>
<td>I-PN</td>
<td>✓ [Min67]</td>
<td>✓ [Min67]</td>
<td>✓ [Min67]</td>
</tr>
<tr>
<td>HIPN</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
</tr>
<tr>
<td>R+HIPN</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
</tr>
<tr>
<td>T+HIPN</td>
<td>✓ [DFS98], Thm 4</td>
<td>✓ [DFS98], Thm 4</td>
<td>✓ [DFS98], Thm 4</td>
</tr>
</tbody>
</table>

### Questions:

- What happens when resets/transfers are added to HIPN?
  - Understanding the boundary of decidability and undecidability...
### State of the art

<table>
<thead>
<tr>
<th>Term</th>
<th>Cover</th>
<th>Reach</th>
<th>DLFree</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>✓ [May84, Ler12]</td>
</tr>
<tr>
<td>R/T-PN</td>
<td>✓ (see [FS01])</td>
<td>✓ (see [FS01])</td>
<td>✗ [DFS98]</td>
</tr>
<tr>
<td>I-PN</td>
<td>✗ [Min67]</td>
<td>✓ [Min67]</td>
<td>✗ [Min67]</td>
</tr>
<tr>
<td>HIPN</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
<td>✓ [Rei08, Bon13]</td>
</tr>
<tr>
<td>R+HIPN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T+HIPN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Questions:**

- What happens when resets/transfers are added to HIPN?
  - Understanding the boundary of decidability and undecidability...
- Can we “weaken” the notion of Hierarchy?
### Questions:

- What happens when resets/transfers are added to HIPN?
  - Understanding the boundary of decidability and undecidability...
- (4.) “Weakening” Hierarchy in HIPN using resets and transfers.
Part 1: Termination in R+HIPN
• Difficulty: The traditional Finite Reachability Tree (FRT) doesn’t work for R+HIPN due to inhibitor arcs.

Idea
Modify the definition of FRT (specifically the subsumption condition), to allow inhibitor arcs.
Termination in R+HIPN

- Difficulty: The traditional Finite Reachability Tree (FRT) doesn’t work for R+HIPN due to inhibitor arcs.
- Usual idea: Explore all runs. If the net terminates, then this is a decision procedure. Else, stop when a marking is “subsumed” (which must happen thanks to WQO)!
Termination in \( R+\text{HIPN} \)

- Difficulty: The traditional Finite Reachability Tree (FRT) doesn’t work for \( R+\text{HIPN} \) due to inhibitor arcs.

- Usual idea: Explore all runs. If the net terminates, then this is a decision procedure. Else, stop when a marking is “subsumed” (which must happen thanks to WQO)!

- Subsumption: If \( M_1 \leq M_2 \), then we can stop (witness for nontermination), as we can repeat this.

In the presence of inhibitors, this is not true!
Termination in R+HIPN

- Difficulty: The traditional Finite Reachability Tree (FRT) doesn’t work for R+HIPN due to inhibitor arcs.
- Usual idea: Explore all runs. If the net terminates, then this is a decision procedure. Else, stop when a marking is “subsumed” (which must happen thanks to WQO)!
- Subsumption: If $M_1 \leq M_2$, then we can stop (witness for nontermination), as we can repeat this.

In the presence of inhibitors, this is not true!

**Idea**

Modify the definition of FRT (specifically the subsumption condition), to allow inhibitor arcs.
Theorem

Checking termination in R+HIPN is decidable.

Proof sketch/intuition:

- For any place $p \in P$, we define the index of the place $p$ ($\text{Index}(p)$) as the number of places $q \in P$ such that $q \sqsubseteq p$.
Theorem

Checking termination in R+HIPN is decidable.

Proof sketch/intuition:

- For any place \( p \in P \), we define the index of the place \( p \) \((\text{Index}(p))\) as the number of places \( q \in P \) such that \( q \sqsubseteq p \).
- For \( i \in \mathbb{N} \), \( M_1 \) and \( M_2 \) are \( i \)-Compatible (denoted \( \text{Compat}_i(M_1, M_2) \)) if
  \[
  \forall p \in P \ \text{Index}(p) \leq i \implies M_1(p) = M_2(p)
  \]
Termination in R+HIPN

Theorem

Checking termination in R+HIPN is decidable.

Proof sketch/intuition:

- For any place \( p \in P \), we define the index of the place \( p \) (\( \text{Index}(p) \)) as the number of places \( q \in P \) such that \( q \subseteq p \).
- For \( i \in \mathbb{N} \), \( M_1 \) and \( M_2 \) are \( i \)-Compatible (denoted \( \text{Compat}_i(M_1, M_2) \)) if
  \[
  \forall p \in P \ \text{Index}(p) \leq i \implies M_1(p) = M_2(p)
  \]
- For any transition \( t \in T \), its index is defined as
  \[
  \text{Index}(t) = \max_{F(p,t)=1} \text{Index}(p)
  \]

By convention, if there is no such place, then \( \text{Index}(t) = 0 \).
Definition (Modified subsumption)

Consider a run $M_2 \xrightarrow{\rho} M_1$. Let $t^* = \text{argmax}_{t \in \rho} \text{Index}(t)$.

$$\text{Subsume}(M_2, M_1, \rho) = M_2 \leq M_1 \land (\text{Compat}_{\text{Index}(t^*)}(M_1, M_2))$$
Definition (Modified subsumption)

Consider a run $M_2 \xrightarrow{\rho} M_1$. Let $t^* = \arg\max_{t \in \rho} \text{Index}(t)$.

$$\text{Subsume}(M_2, M_1, \rho) = M_2 \leq M_1 \land \left( \text{Compat}_{\text{Index}(t^*)}(M_1, M_2) \right)$$

Then, we can show

- This must happen if $\exists$ non-terminating run (thanks to WQO).
Termination in R+HIPN

Definition (Modified subsumption)

Consider a run $M_2 \xrightarrow{\rho} M_1$. Let $t^* = \arg\max_{t \in \rho} \text{Index}(t)$.

$$\text{Subsume}(M_2, M_1, \rho) = M_2 \leq M_1 \land \left( \text{Compat}_{\text{Index}(t^*)}(M_1, M_2) \right)$$

Then, we can show

- This must happen if $\exists$ non-terminating run (thanks to WQO).
- Also, if it happens, there is a non-terminating run.
  - Let $M_1 \leq M_2$, $i \in \mathbb{N}$, $\text{Compat}_i(M_1, M_2)$. Then for any run $\rho$ over $T_i = \{t | t \in T \land \text{Index}(t) \leq i\}$, if $M_1 \xrightarrow{\rho} M_1'$, then $M_2 \xrightarrow{\rho} M_2'$, where $M_1' \leq M_2'$ and $\text{Compat}_i(M_1', M_2')$. 
Definition (Modified subsumption)

Consider a run $M_2 \xrightarrow{\rho} M_1$. Let $t^* = \arg\max_{t \in \rho} \text{Index}(t)$.

$$\text{Subsume}(M_2, M_1, \rho) = M_2 \leq M_1 \land \left( \text{Compat}_{\text{Index}(t^*)}(M_1, M_2) \right)$$

Then, we can show

- This must happen if $\exists$ non-terminating run (thanks to WQO).
- Also, if it happens, there is a non-terminating run.
  - Let $M_1 \leq M_2$, $i \in \mathbb{N}$, $\text{Compat}_i(M_1, M_2)$. Then for any run $\rho$ over $T_i = \{ t \mid t \in T \land \text{Index}(t) \leq i \}$, if $M_1 \xrightarrow{\rho} M'_1$, then $M_2 \xrightarrow{\rho} M'_2$, where $M'_1 \leq M'_2$ and $\text{Compat}_i(M'_1, M'_2)$.

From this and effectivity, we get our result.
Part 2: Moving on to transfer arcs
But first – A detour to program termination!
Basic undecidability result – Turing 1936

Termination of a generic program with a loop is undecidable:

\[
\text{while (conditions) \{commands\}}
\]
Termination of linear loop programs

Basic undecidability result – Turing 1936
Termination of a generic program with a loop is undecidable:

\[
\text{while } (\text{conditions}) \ {\text{commands}}
\]

But now, let us consider a much simpler case:

An initialized homogeneous linear program
\[
\vec{x} := \vec{b}; \ \text{while } (\vec{c}^T \vec{x} > \vec{0}) \ {\vec{x} := A\vec{x}}
\]
Termination of linear loop programs

Basic undecidability result – Turing 1936
Termination of a generic program with a loop is undecidable:

\[ \text{while } (\text{conditions}) \{ \text{commands} \} \]

But now, let us consider a much simpler case:

An initialized homogeneous linear program
\[ \vec{x} := \vec{b}; \quad \text{while } (\vec{c}^T \vec{x} > \vec{0}) \{ \vec{x} := A\vec{x} \} \]

Termination problem for simple linear programs
Does an instance of the above program i.e., \(\langle \vec{b}; \vec{c}; A \rangle\), terminate?
Termination of linear loop programs

**Basic undecidability result – Turing 1936**
Termination of a generic program with a loop is undecidable:

\[
\text{while (conditions)} \{ \text{commands} \}
\]

But now, let us consider a much simpler case:

**An initialized homogeneous linear program**
\[
\vec{x} := \vec{b}; \text{ while } (\vec{c}^T \vec{x} > \vec{0}) \{ \vec{x} := A\vec{x} \}
\]

**Termination problem for simple linear programs**
Does an instance of the above program i.e., \(\langle \vec{b}; \vec{c}; A \rangle\), terminate?

This problem is also called the positivity problem!
Termination of linear loop programs

Basic undecidability result – Turing 1936
Termination of a generic program with a loop is undecidable:

\[
\textbf{while } (\text{conditions}) \quad \{ \text{commands} \}
\]

But now, let us consider a much simpler case:

An initialized homogeneous linear program
\[
\vec{x} := \vec{b}; \quad \textbf{while } (\vec{c}^T \vec{x} > \vec{0}) \quad \{ \vec{x} := A\vec{x} \}
\]

Termination problem for simple linear programs
Does an instance of the above program i.e., \( \langle \vec{b}; \vec{c}; A \rangle \), terminate?

This problem is also called the positivity problem!
– rewrite as \( \forall n \geq 0, \text{ is } \vec{c}^T \cdot A^n \cdot \vec{b} > 0? \)
Decidability of the Positivity problem

- Decidability of Skolem/Positivity for 2,3,4... in 1981, ’85, ’05, ’06, ’09 by various authors.

- In 2014, Ouaknine and Worrell showed the best known result:
  - positivity of order $\leq 5$ is decidable with complexity $\text{coNP}^{PP^{PP^{PP}}}$.  
  - decidability for order 6 would imply major breakthroughs in analytic number theory (Diophantine approx of transcendental numbers).
Decidability of Skolem/Positivity for 2, 3, 4... in 1981, ’85, ’05, ’06, ’09 by various authors.

In 2014, Ouaknine and Worrell showed the best known result:

- positivity of order $\leq 5$ is decidable with complexity $\text{coNP}^{PP^{PP^{PP}}}$. 
- decidability for order 6 would imply major breakthroughs in analytic number theory (Diophantine approx of transcendental numbers).

**Bottomline:** The general problem is still wide open!
Question
Can you model program termination with Petri nets?

Simulating a program
Consider the following while loop program
\[ v = v_0; \text{while} (v \geq 0) v = Mv. \]
• Clearly, this program is non-terminating iff \[ M^k v_0 \geq 0 \text{ for all } k. \]
• We construct a net \( N \) which simulates the program, i.e., terminates iff the program does.
Question
Can you model program termination with Petri nets?

Theorem
Program termination/positivity reduces to termination of Petri nets with one transfer and one inhibitor arc!
Question
Can you model program termination with Petri nets?

Theorem
Program termination/positivity reduces to termination of Petri nets with one transfer and one inhibitor arc!

Simulating a program
Consider the following while loop program
\[ v = v_0; \text{ while } (v \geq 0) \ v = Mv. \]

- Clearly, this program is non-terminating iff \( M^k v_0 \geq 0 \) for all \( k \).
- We construct a net \( N \) which simulates the program, i.e., terminates iff the program does.
Consider

\[ M = \begin{bmatrix}
1 & -4 & 7 \\
2 & -5 & -8 \\
-3 & -6 & 9
\end{bmatrix} \]
Reduction from Positivity to T+HIPN

\[
\begin{bmatrix}
1 & -4 & 7 \\
2 & -5 & -8 \\
-3 & -6 & 9
\end{bmatrix}
\]
Reduction from Positivity to T+HIPN

\[
\begin{bmatrix}
1 & -4 & 7 \\
2 & -5 & -8 \\
-3 & -6 & 9 \\
\end{bmatrix}
\times
\begin{bmatrix}
5 \\
6 \\
7 \\
\end{bmatrix}
\]

E.g: First entry of col vec \( Mv = 5(1) + 6(-4) + 7(7) \)
Reduction from Positivity to T+HIPN
Initial marking assigns \((v_0)_i\), to place \(u_i\), and \(\sum_{1 \leq i \leq n}(\sum_{1 \leq j \leq n}|M_{ji}|)(v_0)_i\) tokens to \(G\), all others 0.

**Lemma:** \(\exists\) a non-term run in \(N\) iff \(M^k v_0 \geq 0 \ \forall k \in \mathbb{N}\).
Links to program termination

- We do not have a two-way reduction... so termination for T+HIPN could still be undecidable. (Open problem 1)
Links to program termination

- We do not have a two-way reduction... so termination for T+HIPN could still be undecidable. (Open problem 1)
- Can we reduce positivity to termination of R+HIPN?
• We do not have a two-way reduction... so termination for T+HIPN could still be undecidable. (Open problem 1)
• Can we reduce positivity to termination of R+HIPN? (Open problem 2) :P
Links to program termination

- We do not have a two-way reduction... so termination for T+HIPN could still be undecidable. (Open problem 1)
- Can we reduce positivity to termination of R+HIPN? (Open problem 2) :P
- If not, what about other problems? Reachability is already undecidable.
We do not have a two-way reduction... so termination for T+HIPN could still be undecidable. (Open problem 1)

Can we reduce positivity to termination of R+HIPN? (Open problem 2) :P

If not, what about other problems? Reachability is already undecidable.
What about coverability?
Theorem

Coverability is undecidable for Petri nets with 2 resets and 1 inhibitor arc.
Theorem

Coverability is undecidable for Petri nets with 2 resets and 1 inhibitor arc.
<table>
<thead>
<tr>
<th>Term</th>
<th>Cover</th>
<th>Reach</th>
<th>DLFRee</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>✓ (see [FS01]) ✓ (see [FS01]) ✓ (May84, Ler12) ✓ [CEP95, Hac74]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R/T-PN</td>
<td>✓ (see [FS01]) ✓ (see [FS01]) × [DFS98] × [MIn67]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I-PN</td>
<td>× [Min67] × [Min67]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIPN</td>
<td>✓ [Rei08, Bon13] ✓ [Rei08, Bon13] ✓ [Rei08, Bon13]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R+HIPN</td>
<td>✓ [DFS98, Thm 4] ✓ [Red.frm [DFS98], Thm 4]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T+HIPN</td>
<td>× [DFS98, Thm 4] × [Red.frm [DFS98], Thm 4]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(see paper!)
Part 3: “Weakening” Hierarchy?
Adding resets/transfers within hierarchy
Definition of HIPN

A total order $\sqsubseteq$ on $P$ such that

$$\forall (p, t) \in P \times T, \ F(p, t) \in I \implies (\forall q \sqsubseteq p, \ F(q, t) \in I).$$
Adding resets and transfers within Hierarchy

Definition of HIPN

A total order $\sqsubseteq$ on $P$ such that
$$\forall (p, t) \in P \times T, \ F(p, t) \in I \implies (\forall q \sqsubseteq p, \ F(q, t) \in I).$$

What if we change this to:
Definition of HIRPN: A seemingly larger class!

A total order $\sqsubseteq$ on $P$ such that
\[
\forall (p, t) \in P \times T, \quad F(p, t) \in I \implies (\forall q \sqsubseteq p, \ F(q, t) \in (I \lor R)).
\]
Definition of HIRPN: A seemingly larger class!

A total order $\sqsubseteq$ on $P$ such that
\[
\forall (p, t) \in P \times T, \ F(p, t) \in I \implies (\forall q \sqsubseteq p, \ F(q, t) \in (I \lor R)).
\]

Reset

- This is not a HIPN (or a R+HIPN), but it is a HIRPN!
Adding resets and transfers within Hierarchy

Definition of HIRPN: A seemingly larger class!
A total order \( \sqsubseteq \) on \( P \) such that
\[
\forall (p, t) \in P \times T, \quad F(p, t) \in I \implies (\forall q \sqsubseteq p, \quad F(q, t) \in (I \lor R))
\]

- This is not a HIPN (or a R+HIPN), but it is a HIRPN!
- A R+HIPN which is not a HIRPN.
Adding resets and transfers within Hierarchy

**Definition of HIRPN**: A seemingly larger class!

A total order \( \sqsubseteq \) on \( P \) such that

\[
\forall (p, t) \in P \times T, \quad F(p, t) \in I \implies (\forall q \sqsubseteq p, \quad F(q, t) \in (I \lor R)).
\]

- This is not a HIPN (or a R+HIPN), but it is a HIRPN!
- A R+HIPN which is not a HIRPN.
- Can do the same with transfers...
Theorem

HIRPNs are still easy: Can reduce to HIPNs, which preserving reachability. Hence obtain decidability of properties.
Theorem

HIRPNs are still easy: Can reduce to HIPNs, which preserving reachability. Hence obtain decidability of properties.
Theorem
HIRPNs are still easy: Can reduce to HIPNs, which preserving reachability. Hence obtain decidability of properties.

Theorem
Hierarchy is useless with transfers: i.e., HITPNs have same properties as $T+\text{HIPNs}$. 
Conclusion
## Results: Summary

**Table 1:** Results for all other extensions are subsumed by these results. Can add boundedness column too!
• Reducing the number of counters.
• What about complexity?
• Coverability for Petri nets with 1 reset and 1 inhibitor arc (without hierarchy)?
• An approach towards the positivity/Skolem problem via WSTS?
References


