# Complexity of the Soundness Problem of Bounded Workflow Nets

**GuanJun LIU, Jun SUN, Yang LIU, JinSong DONG**

Singapore University of Technology and Design

National University of Singapore

# Outline

- **Introduction to WF-nets and WF-nets with reset arcs (reWF-nets)**
- **NP-hardness of the soundness problem of WF-nets**
- **PSPACE-hardness of the soundness problem of reWF-nets**

# Outline

- **Introduction to WF-nets and WF-nets with reset arcs (reWF-nets)**
- **NP-hardness of the soundness problem of WF-nets**
- **PSPACE-hardness of the soundness problem of reWF-nets**

# Introduction to WF-nets

**Definition** (*WF-nets [Aalst et al]*): A net $N = (P, T, F)$ is a workflow net (WF-net) if:

1. $N$ has two special places $i \in P$ (source place) and $o \in P$ (sink place) such that ${}^{\bullet}i = \emptyset$ and $o^{\bullet} = \emptyset$; and

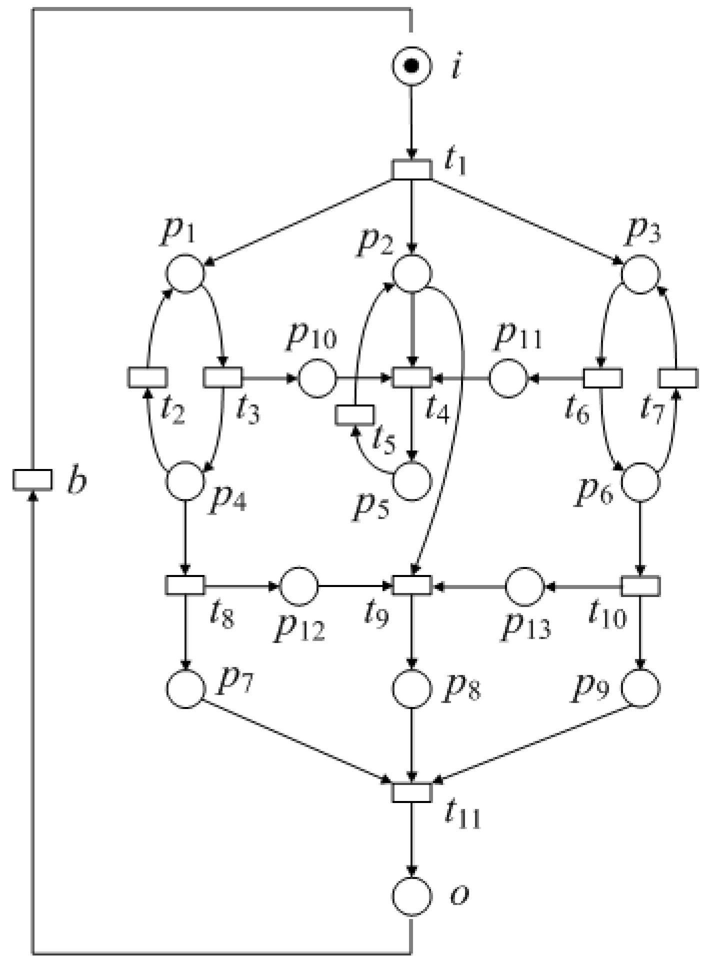
2. $N^E = (P, T \cup \{b\}, F \cup \{(b, i), (o, b)\})$ is strongly connected.

**Definition** (*Soundness of WF-nets [Aalst et al]*): A WF-net $N = (P, T, F)$ is sound if:

1. $\forall M \in R(N, M_0)$: $M_d \in R(N, M)$; and

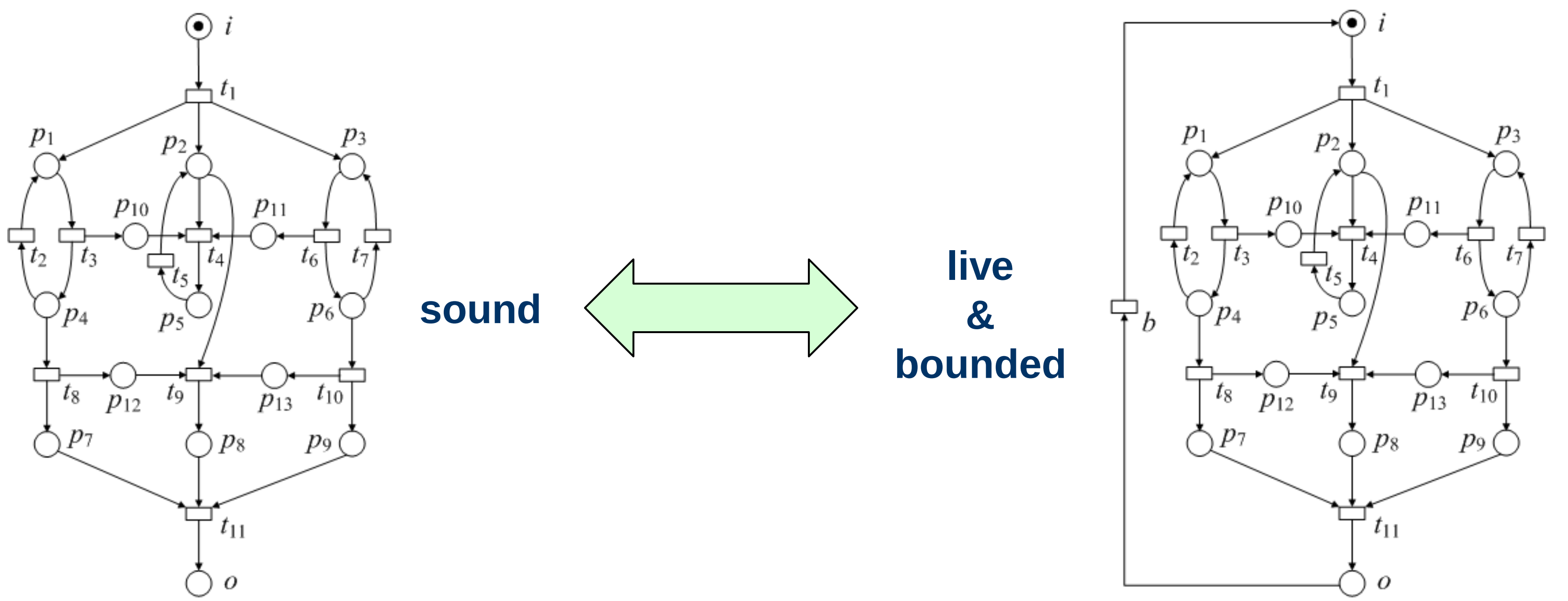
2. $\forall t \in T, \exists M \in R(N, M_0)$: $M[t\rangle$.

where $M_0 = i$ and $M_d = o$.



# Introduction to WF-nets

**Theorem** (*[Aalst et al]*): Let $N = (P, T, F)$ be a WF-net, $N^E = (P, T \cup \{b\}, F \cup \{(b, i), (o, b)\})$, and $M_0 = i$. Then, $N$ is sound if and only if $(N^E, M_0)$ is live and bounded.




**Corollary**: Let $N = (P, T, F)$ be a WF-net, and $(N^E, M_0) = (P, T \cup \{b\}, F \cup \{(b, i), (o, b)\}, i)$ be bounded. Then, $N$ is sound if and only if $(N^E, M_0)$ is live.

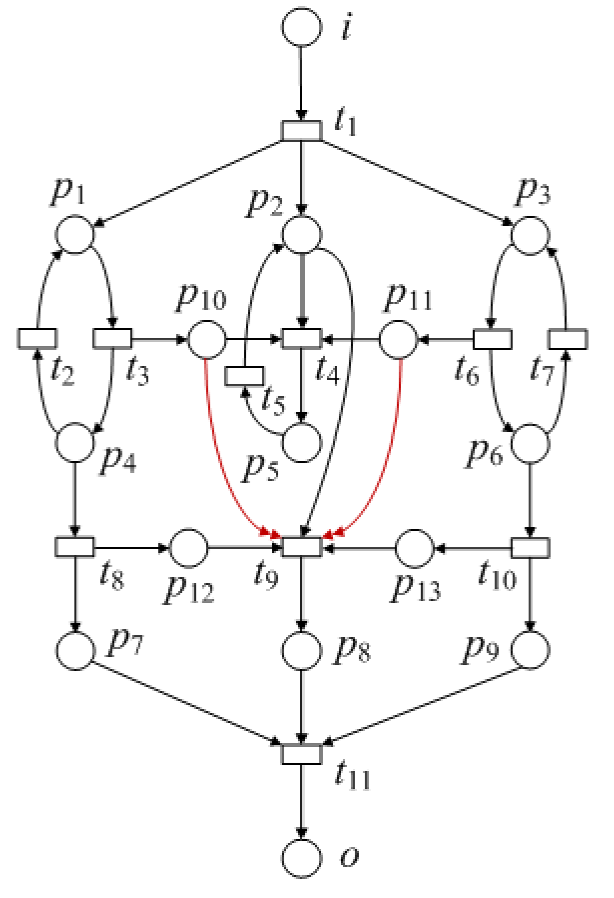
# Introduction to reWF-nets

**Definition** (*reWF-nets [Aalst et al]*): A 4-tuple $N = (P, T, F, R)$ *is* a workflow net with reset arcs (reWF-net) if:

1. $(P, T, F)$ is a WF-net; and

2. $R \subseteq [P \setminus \{o\} \times T]$ is the set of reset arcs.

**Definition**: Transition $t$ is **enabled** at $M$ if $\forall p \in {}^{\bullet}t$: $M(p) > 0$. **Firing** an enabled transition $t$ produces a new marking $M'$ such that $M(p) = 0$ if $p \in {}^{\circ}t$; $M'(p) = M(p) - 1$ if $p \notin {}^{\circ}t \wedge p \in {}^{\bullet}t \setminus t^{\bullet}$; $M'(p) = M(p) + 1$ if $p \notin {}^{\circ}t \wedge p \in t^{\bullet} \setminus {}^{\bullet}t$; and $M'(p) = M(p)$ otherwise.



# Introduction to reWF-nets

**Definition** (*Soundness of reWF-nets [Aalst et al]*): An reWF-net $N = (P, T, F, R)$ is sound if:

1. $\forall M \in R(N, M_0)$: $M_d \in R(N, M)$; and

2. $\forall t \in T, \exists M \in R(N, M_0)$: $M[t\rangle$.

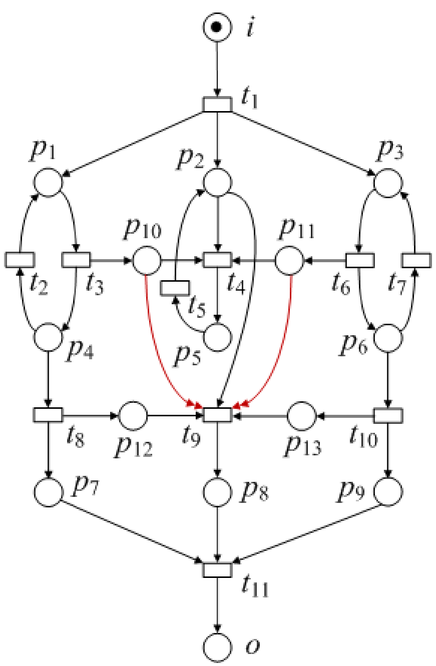
where $M_0 = i$ and $M_d = o$.

**Theorem** (*[Aalst et al]*): The soundness problem of reWF-nets is undecidable.

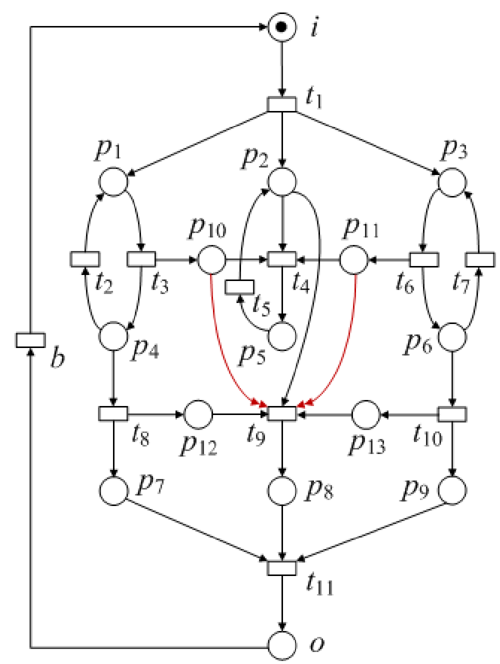
If the trivial extension of an reWF-net, $(N^E, M_0) = (P, T \cup \{b\}, F \cup \{(b, i), (o, b)\}, R, i)$, is bounded, then its soundness problem is decidable by its reachability graph.

# Introduction to reWF-nets

**Theorem** : Let $N = (P, T, F, R)$ be an reWF-net, and $(N^E, M_0) = (P, T \cup \{b\}, F \cup \{(b, i), (o, b)\}, R, i)$ be bounded. Then, $N$ is sound if and only if $(N^E, M_0)$ is live. (note: "only if" is proven by [Aalst])




**sound** ⇔ **live & bounded**



# Outline

- **Introduction to WF-nets and WF-nets with reset arcs (reWF-nets)**
- **NP-hardness of the soundness problem of WF-nets**
- **PSPACE-hardness of the soundness problem of reWF-nets**

# NP-hardness of soundness of WF-nets

For each expression of disjunctive normal form (DNF) in which each term has three literals,

$$H = D_1 \vee D_2 \vee \ldots \vee D_m = (l_{1,1} \wedge l_{1,2} \wedge l_{1,3}) \vee (l_{2,1} \wedge l_{2,2} \wedge l_{2,3}) \vee \ldots \vee (l_{m,1} \wedge l_{m,2} \wedge l_{m,3})$$

we can construct a WF-nets (in polynomial time) by which we can compute if the value of the DNF expression is true.

# NP-hardness of soundness of WF-nets

**Step1**: assign values to variables.

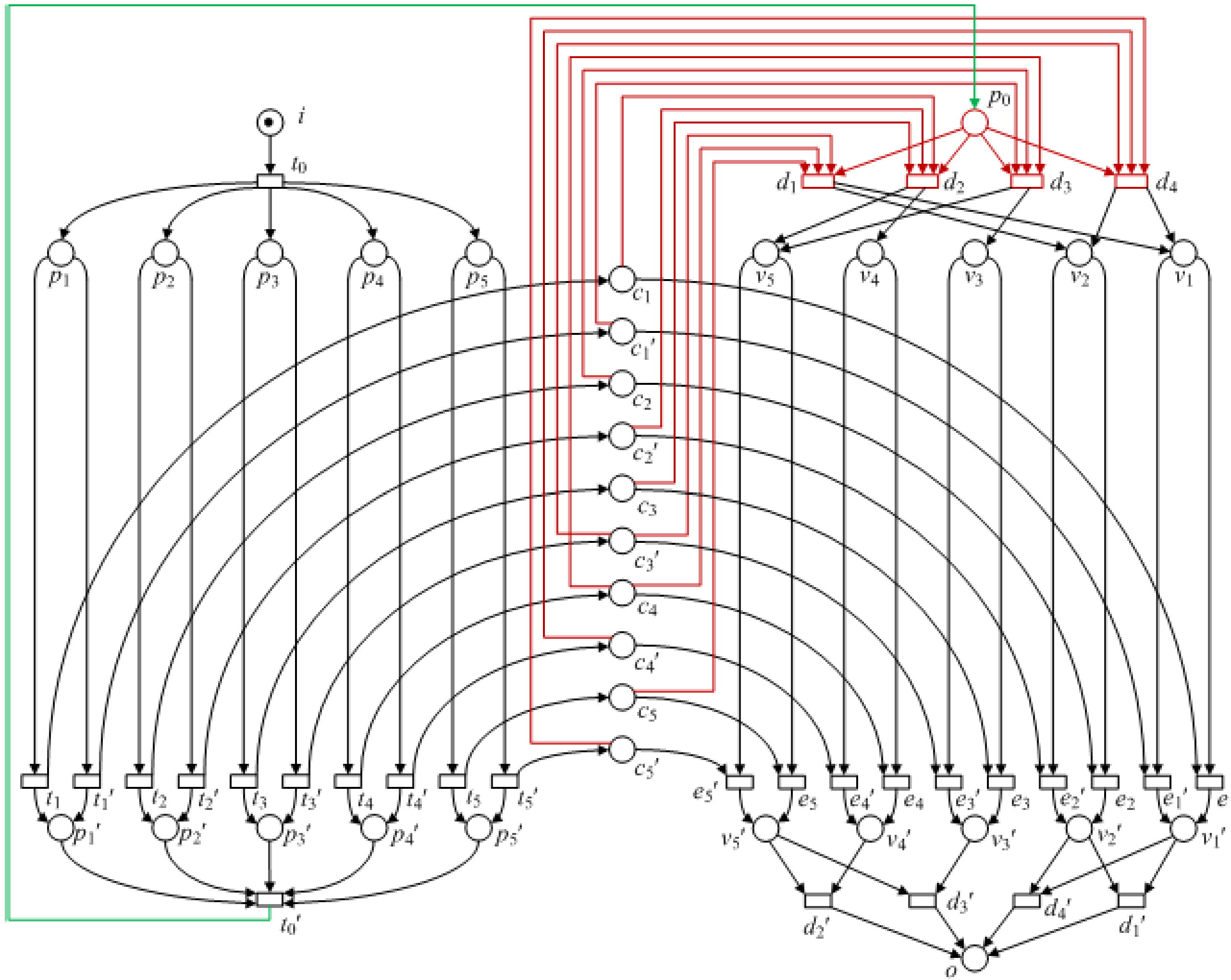
$c_i \sim x_i$

$c_i' \sim \neg x_i$

**Step2**: decide if the assignment make the DNF expression true. If it is true, there is (only) one transition $d_i$ that can be fired.

$d_i \sim D_i$

**Step3**: remove the remainder tokens.



$$H = (\neg x_3 \wedge x_4 \wedge x_5) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge x_4) \vee (\neg x_3 \wedge \neg x_4 \wedge \neg x_5)$$

# NP-hardness of soundness of WF-nets

**Lemma**: The trivial extension of the constructed WF-net is live if and only if $H = 1$ for each assignment of variables.

**Lemma**: The trivial extension of the constructed WF-net is bounded at the initial marking $M_0 = i$.

**Theorem**: The problem of soundness of WF-nets is co-NP-hard.

# Outline

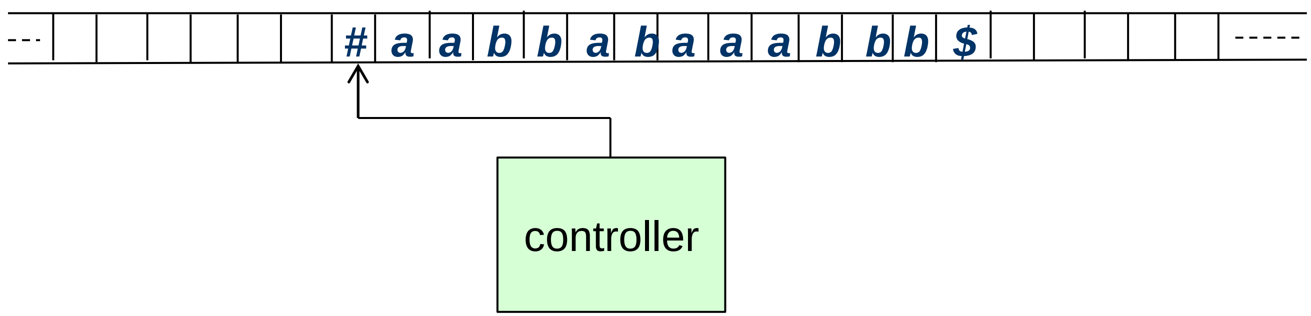
- **Introduction to WF-nets and WF-nets with reset (reWF-nets)**
- **NP-hardness of the soundness problem of WF-nets**
- **PSPACE-hardness of the soundness problem of reWF-nets**

# PSPACE-hardness of soundness of reWF-nets

For each Linear Bounded Automata (LBA) with an input string, we can always construct an reWF-net (in polynomial time) by which we can decide whether the LBA accepts this input string.

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$

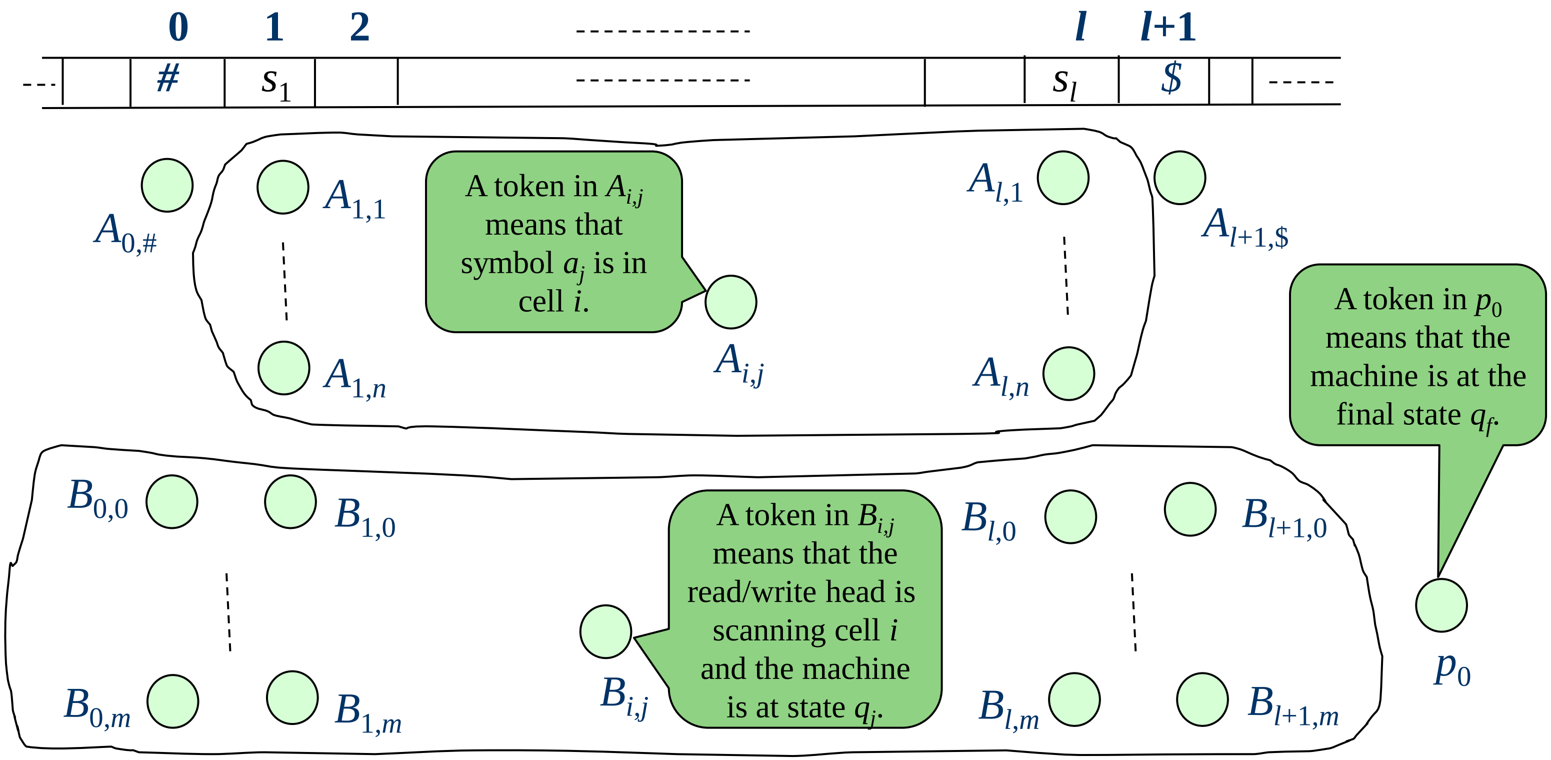
- $Q = \{q_0, q_1, \ldots, q_m, q_f\}$ — set of states, initial state $q_0$, final state $q_f$
- $\Gamma = \{a_1, \ldots, a_n\}$ — tape alphabet
- $\Sigma \subseteq \Gamma$ — input alphabet
- $\Delta \subseteq Q \times \Gamma \times \{R, L\} \times Q \times \Gamma$ — set of transitions
- $\#$ — left bound symbol
- $\$$ — right bound symbol



# PSPACE-hardness of soundness of reWF-nets

**Step1**: use places to represent the tape information, machine state, and read/write head position.

Let $Q=\{q_0, q_1, \ldots, q_m, q_f\}$, $m \geq 0$, $\Gamma=\{a_1, a_2, \ldots, a_n\}$, $n > 0$, $|S| = l$, and cells storing $\#S\$$ be labelled 0, 1, …, $l$, and $l+1$, respectively.



# PSPACE-hardness of soundness of reWF-nets

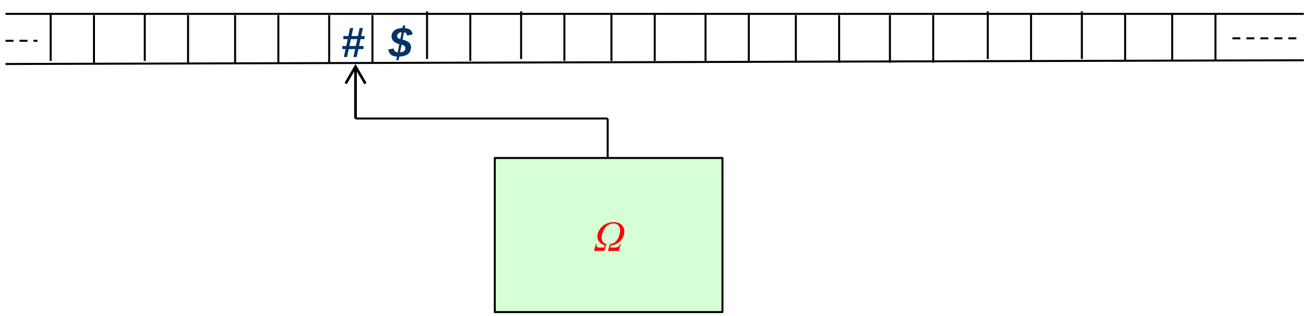
$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$

– $Q = \{q_0, q_1, q_2, q_3, q_f\}$

– $\Gamma = \{a, b, X\}$

– $\Sigma = \{a, b\}$

– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X),$
$(q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a),$
$(q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$



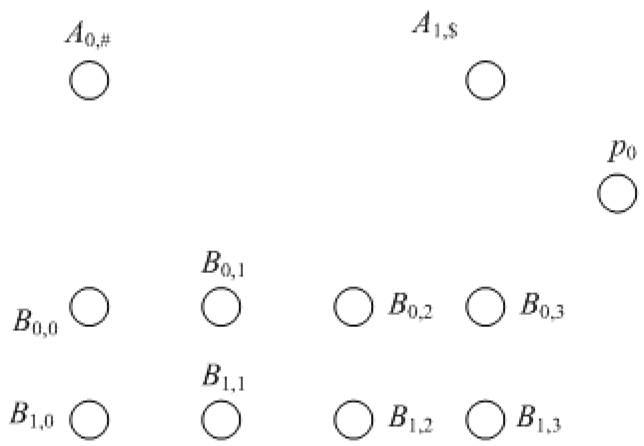

For example: the above LBA with the empty string as its input.

Note: the LBA produce the language $\{a^{i_1}b^{i_1}a^{i_2}b^{i_2}\ldots a^{i_m}b^{i_m} \mid i_1, i_2, \ldots, i_m, m \in \mathrm{N}\}$

# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$
– $\Gamma = \{a, b, X\}$
– $\Sigma = \{a, b\}$
– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X),$
$(q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a),$
$(q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

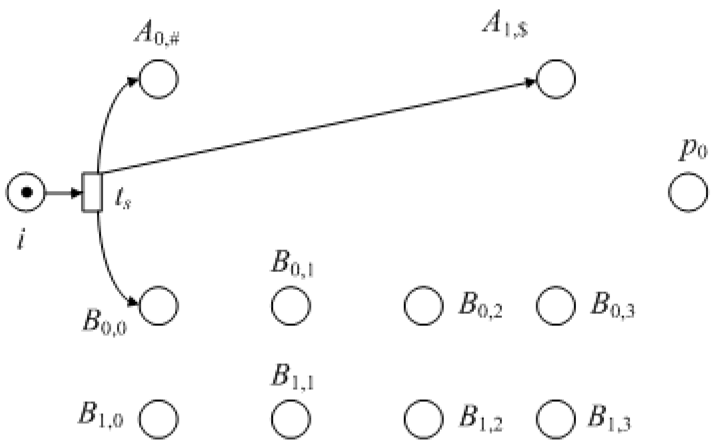
**Step1**: use places to represent tape information, machine state, & read/write head position.



# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$
– $\Gamma = \{a, b, X\}$
– $\Sigma = \{a, b\}$
– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X),$
$(q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a),$
$(q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

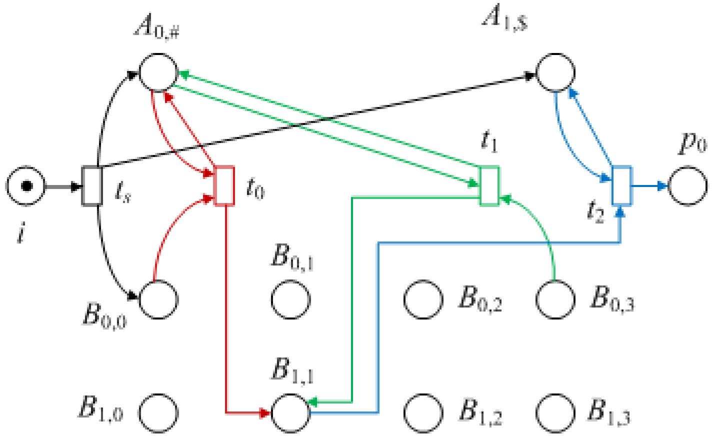
**Step2**: use a net transition to produce the machine's initial configuration.



# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$
– $\Gamma = \{a, b, X\}$
– $\Sigma = \{a, b\}$
– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X), (q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a), (q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

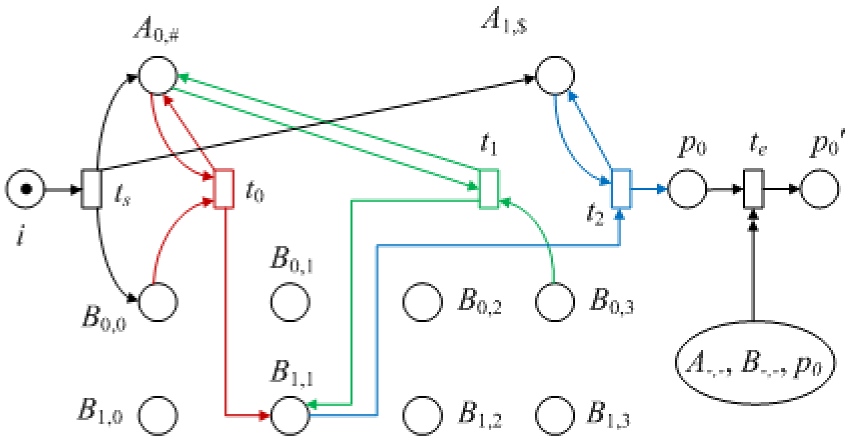
**Step3**: use net transitions to model machine transitions.



# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$
– $\Gamma = \{a, b, X\}$
– $\Sigma = \{a, b\}$
– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X), (q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a), (q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

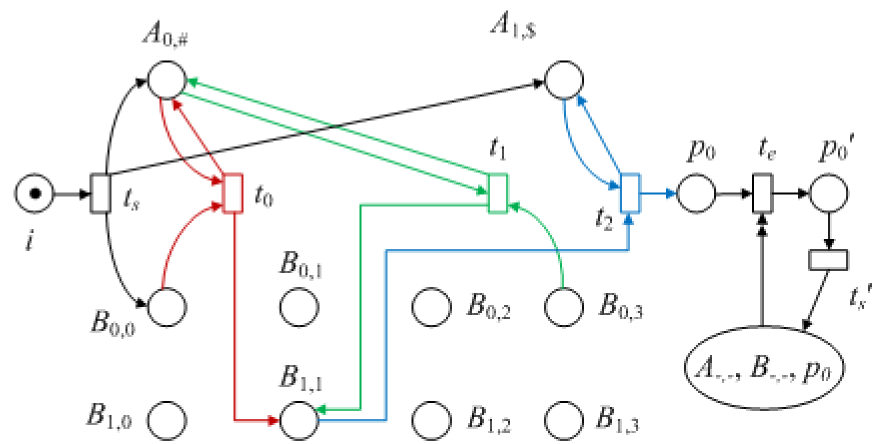
**Step4**: use a net transition, associating with reset arcs, to remove remainder tokens.



# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$
– $\Gamma = \{a, b, X\}$
– $\Sigma = \{a, b\}$
– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X),$
$(q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a),$
$(q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

**Step5**: use a net transition to input tokens in order to make each net transition have a friable right.



# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$

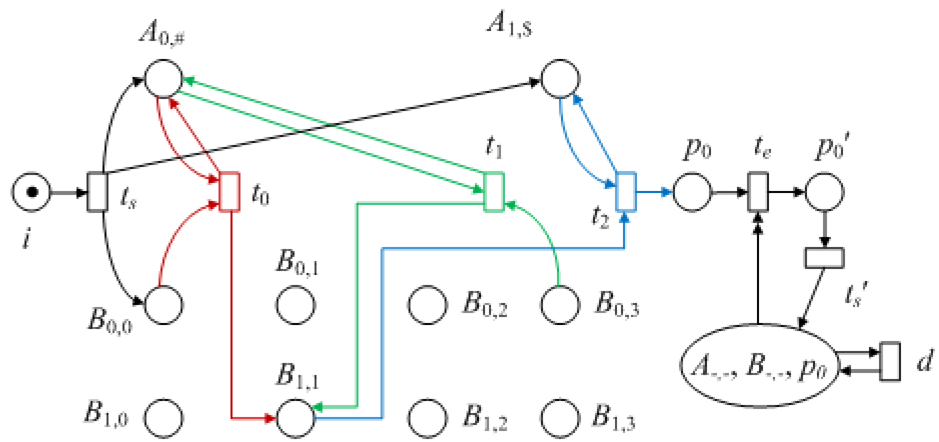
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$

– $\Gamma = \{a, b, X\}$

– $\Sigma = \{a, b\}$

– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X), (q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a), (q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

**Step6**: use a net transition to connect with each place by a self-loop in order to make the net be strongly connected.



# PSPACE-hardness of soundness of reWF-nets

$\Omega = (Q, \Gamma, \Sigma, \Delta, q_0, q_f, \#, \$)$

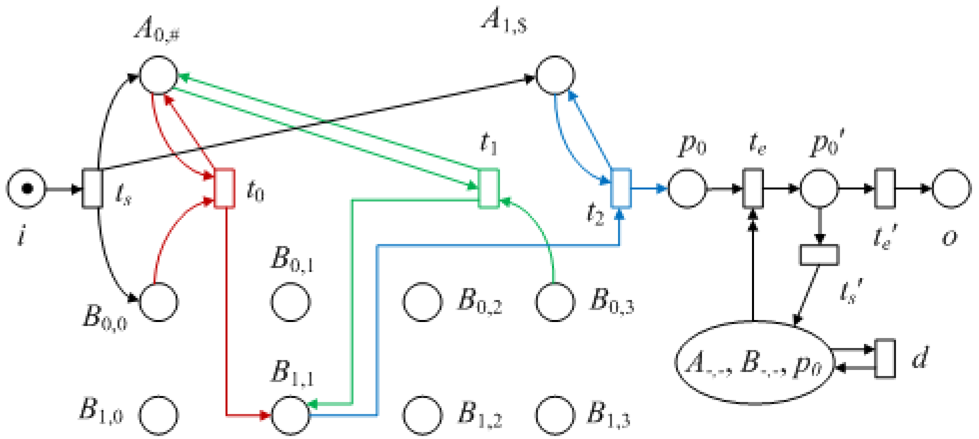
– $Q = \{q_0, q_1, q_2, q_3, q_f\}$

– $\Gamma = \{a, b, X\}$

– $\Sigma = \{a, b\}$

– $\Delta = \{(q_0, \#, R, q_1, \#), (q_1, \$, L, q_f, \$), (q_1, X, R, q_1, X), (q_1, a, R, q_2, X), (q_2, a, R, q_2, a), (q_2, X, R, q_2, X), (q_2, b, L, q_3, X), (q_3, a, L, q_3, a), (q_3, X, L, q_3, X), (q_3, \#, R, q_1, \#)\}$

**Step7**: finally, use a net transition to finish the whole computation.



# PSPACE-hardness of soundness of reWF-nets

**Lemma**: The LBA accepts the input string iff the trivial extention of the constructed reWF-net is live.

**Lemma**: The trivial extention of the constructed reWF-net is bounded.

**Theorem**: The soundness problem of reWF-nets is PSPACE-hard.

# Thanks！