Vérification de contraintes temporelles
pour des systèmes de contrôle commande
à l'aide des réseaux de Petri

Verification of temporal constraints
for control command systems
by means of Petri nets

R. Valette, F. Girault, L.A. Künzle, B. Pradin Chézalviel, J. Cardoso

LAAS-CNRS, F-31077 Toulouse Cedex 4
CURRENT STATE OF THE ART

- Analysis of "good" properties - consistency and completeness
- p-invariants - correctness of resource allocation (mut. ex.)
- Reachability graph generation - Atomata based proofs, Temporal logics

- Informal reasonings directly on the Petri net structure, paths

Why are these reasonings still necessary? Is it possible to formalize them?
TRUTH AND TIME

Propositions are true with respect to a time interval

Three cases (at least) can be encountered:

- Eternal truths
- Transitory truths
- Consumable resources
TRUTH AND TIME - Eternal truth

"humans are mortal" \(\Rightarrow\) P1
"Socrates is a human" \(\Rightarrow\) P2
\(\Rightarrow\) "Socrates is mortal" \(\Rightarrow\) P3

\(\Rightarrow\) P1, P2 \(\Rightarrow\) P3 or P1 \(\land\) P2 \(\Rightarrow\) P3

Two states of knowledge within reasoning:
P1, P2 \(\Rightarrow\) P1, P2, P3

Monotony is always verified
Required for validation, correctness (OK \(\forall\) M)
TRUTH AND TIME - Transitory truth

"The weather is fine" $\Rightarrow$ P1
"I am in Montréal" $\Rightarrow$ P2
$\Rightarrow$ "I visit the city" $\Rightarrow$ P3

P1, P2 $\Rightarrow$ P3 but a storm broke out and stop the visit

Three states of knowledge within reasoning:
P1, P2 $\Rightarrow$ P1, P2, P3 $\mid$ storm $\triangleright$ P2

Monotony is only verified on some time intervals
Required for property for a state ($OK \exists M$)
TRUTH AND TIME - Consumable resource

"Machine $m$ is idle" $\Rightarrow$ P1
"Part $p$ is ready" $\Rightarrow$ P2
$\Rightarrow$ "$p$ is machined on $m$" $\Rightarrow$ P3

P1, P2 $\Rightarrow$ P3 but $m$ no longer idle and $p$ no longer ready

Two states of knowledge within reasoning:
P1, P2 $\Rightarrow$ P3

Monotony is never verified
Required for sequences correctness (OK $M \rightarrow M'$)
Resources as Logical Propositions - Counting

"2 Machines are idle"  ⇒  2.P1
"Part p  is ready"  ⇒  P2
⇒ "p  is machined on a machine "  ⇒  P3
"1 machine remains idle"  ⇒  1.P1

P1, P2 ⇒ P3  is transformed into:
2.P1, P2 ⇒ P1, P3  which differs from P1, P2 ⇒ P1, P2, P3

Monotony is transformed into LINEARITY
Consuming implies Counting
Resources as Logical Propositions - Negation

"1 Machine is idle" \(\Rightarrow\) P1

"0 Machine is idle" \(\Rightarrow\) ? (absence of any resource)

"2 Machines are idle" \(\Rightarrow\) 2.P1

"I need one idle Machine" \(\Rightarrow\) not(P1)

P1 and not(P1) are not contradictory
It is possible to have an idle machine and to need one more in order to machine two parts

Classical negation is lost with eternal truth
Consuming is transformed into producing and vice-versa
Resources as Logical Propositions - Decision

\[ a: 1 \text{ machine free} \\
\; b: 1 \text{ machine allocated for op. } "b" \\
\; c: 1 \text{ machine allocated for op. } "c" \]

\[(a \rightarrow b), (a \rightarrow c) \models (a \rightarrow b \land c) \text{ is valid but INCORRECT (classical)} \]
\[(a \rightarrow o b), (a \rightarrow o c) \models (a \rightarrow o b \otimes c) \text{ is NOT valid (ll)} \]
\[(a \rightarrow o b), (a \rightarrow o c) \models (2.a \rightarrow o b \otimes c) \text{ is valid and correct (ll)} \]

Consuming enables representing decision (action) within a unique logical world (in a valid formula)

It is not required to operate in 3 different worlds
Resources as Logical Propositions - Linear Logic Motivations

J.Y. Girard has defined the linear logic in order to be consistent with resources.

Started from sequent calculus (knowledge transformation i.e. reachability) not ADDING new rules REMOVING the rules which were inconsistent with resources.

Consistent with the observation of any Petri net addict: Consuming tokens results in simple net structures "Testing" token counts of places results in problems (elementary loops which are not actual cons. and prod.)
Petri nets and Linear Logic - Translation of Petri net into ll

Net structure
- $t_1: a \otimes d \rightarrow o c$
- $t_2: b \otimes c \rightarrow o e \otimes f$
- $t_3: e \rightarrow o d \otimes g$

A sequence
- $t_1, t_2, t_3 \vdash a \otimes b \otimes d \rightarrow o f \otimes g \otimes d$

Its firing
- $M_1, (a \otimes b \otimes d \rightarrow o f \otimes g \otimes d) \vdash M_2$

A sequent which is an eternal truth
which can be used in a classical log. reasoning
Petri nets and Linear Logic - Sequence characterization

Sequent for $t_1 \ t_2 \ t_3$ differs from that for $t_2 \ t_3 \ t_1$ although $\mathbb{L}$ is commutative

$t_1, \ t_2, \ t_3 \vdash a \otimes b \otimes d \leftarrow o \ f \otimes g \otimes d$

$t_1, \ t_2, \ t_3 \vdash a \otimes b \otimes c \leftarrow o \ f \otimes g \otimes c$

More precise than characteristic vector
Indeed a generalization of $Pre$ and $Post$ for sequences
Petri nets and Linear Logic - Firing the sequence

Conditions for the sequent being valid

\[ M_1, (a \otimes b \otimes d \rightarrow o f \otimes g \otimes d) \rightarrow M_2 \]

\[ M_1 = (a \otimes b \otimes d) \otimes C \]
\[ M_2 = (f \otimes g \otimes d) \otimes C \]

C is the context = any token distribution

From sequence => Conditions on markings (states)
Independent from initial marking - from reachability graph
Petri nets and Linear Logic - Expliciting time

\[ M_1, (a \otimes b \otimes d \rightarrow o f \otimes g \otimes d) \rightarrow M_2 \]

If sequent obtained by concatenation (cut rule)
\[ \theta(t_1 t_2 t_3) = \theta(t_1) + \theta(t_2) + \theta(t_3) \]

\[ \theta(t_1 t_2 t_3) \] is the duration required to transform \( M_1 \) into \( M_2 \) whatever \( C \)

Independent from initial marking - from reachability graph
Petri nets and Linear Logic - True concurrency

Characterization of traces (set of sequences)
Union of $Pre$ and $Post$

$t_1, t_2 \vdash a \rightarrow o \ c$
$t_3, t_4 \vdash d \rightarrow o \ f$
$(t_1 \ t_2) || (t_3 \ t_4) : t_1, t_2, t_3, t_4 \vdash a \otimes d \rightarrow o \ c \otimes f$

$\theta(t_1 \ t_2) = \theta(t_1) + \theta(t_2)$
$\theta(t_3 \ t_4) = \theta(t_3) + \theta(t_4)$
$\theta((t_1 \ t_2) || (t_3 \ t_4)) = \max((\theta(t_1) + \theta(t_2)), (\theta(t_3) + \theta(t_4)))$

Takes into account true concurrency
Conclusion

- Petri Net + Linear Logic => Logical formalization of intuitive reasonings on PN graphical structure

- No state enumeration (reachability graph)
  - Independent from initial marking
  (does not depend on the number of parts)
  - Based on the concept of sequences
  (state transformations, not states)

- Only reachability (possibility) (M2 from M1) no necessity
  - No negation (no proof by contradiction)