Algebraic Petri nets

Didier Buchs

Université de Genève

30 avril 2009
Algebraic Petri Nets

- Informal introduction
Algebraic Petri Nets

- Informal introduction
- APN Specification
Algebraic Petri Nets

- Informal introduction
- APN Specification
- Behavioral axioms
Algebraic Petri Nets

- Informal introduction
- APN Specification
- Behavioral axioms
- Examples
Algebraic Petri Nets

- Informal introduction
- APN Specification
- Behavioral axioms
- Examples
- Petri Net interpretation
Algebraic Petri Nets

- Informal introduction
- APN Specification
- Behavioral axioms
- Examples
- Petri Net interpretation
- Semantics
Algebraic Petri Nets

- Informal introduction
- APN Specification
- Behavioral axioms
- Examples
- Petri Net interpretation
- Semantics
- Equivalences
Algebraic Petri Nets

- Informal introduction
- APN Specification
- Behavioral axioms
- Examples
- Petri Net interpretation
- Semantics
- Equivalences
- Equivalences wrt Algebras
Elementary example

- Adds all token from place *numbers* and put the result into place *result*.
- Expected kind of verification : CTL Example :
  \[ AF\{\text{numbers} = \emptyset\} \] : Eventually all tokens from *numbers* will be consumed. Namely the process terminates.
Formal and Mathematical basis
Formal and Mathematical basis

- Petri Nets + Algebraic abstract data types
  - Structure = Petri Net
  - Tokens = Algebraic values
  - Pre/Post Conditions = Algebraic terms attached to flow relation
  - Conditions
Formal and Mathematical basis

- Petri Nets + Algebraic abstract data types
  - Structure = Petri Net
  - Tokens = Algebraic values
  - Pre/Post Conditions = Algebraic terms attached to flow relation
  - Conditions

- Semantics as transition systems (labelled)
  - True Concurrency described by step semantics
  - Not finite branching
  - Label as transition name and variable binding
Formal and Mathematical basis (2)
Formal and Mathematical basis(2)

- Observational equivalences (Bisimulation)
  - Equivalence relation
  - Symmetric simulation
  - Hide internal states
  - Preserve branching
Observational equivalences (Bisimulation)
- Equivalence relation
- Symmetric simulation
- Hide internal states
- Preserve branching

Verification of properties
- Property expression based on temporal logic
- Model Checking algorithms
Observational equivalences (Bisimulation)
- Equivalence relation
- Symmetric simulation
- Hide internal states
- Preserve branching

Verification of properties
- Property expression based on temporal logic
- Model Checking algorithms

Implementation notions:
\[ SP \models P \iff (\text{Sem}(SP) \Leftrightarrow \text{Sem}(P)|_{SP}) \]
Definition (Algebraic Petri net specification)

An *algebraic Petri net specification* is defined as
\( N - SPEC = \langle Spec, T, P, X, AX \rangle \), where:

- \( Spec = \langle \Sigma, X', E \rangle \) is an algebraic specification extended in \( \langle [\Sigma], X', E \rangle \), where \( \Sigma = \langle S, F \rangle \).
- \( T \) is the set of transition names.
- \( P \) is the set of place names, and we define a function \( \tau : P \rightarrow S \) which associates a sort to each place.
- \( AX \) is a set of axioms defined below.
Multiset of tokens

A place is a multiset of tokens, in order to model this bags of values are defined.
Given $\Sigma = \langle S, F \rangle$, we define new sorts and operations such as:

$S' = S \cup \{[s] | \forall s \in S\}$ and $F' = F \cup \{\epsilon : \rightarrow [s], - : s \rightarrow [s], - + : [s], [s] \rightarrow [s] | \forall s \in S\}$
forming the extended signature $[\Sigma] = \langle S', F' \rangle$

An extended algebra $[A]$ is naturally given to any $A \in Alg(\Sigma)$ that is:

$A[s] = \{ f : A_s \rightarrow \mathbb{N} \}$

$\epsilon_A(a) := 0$, $\forall a \in A$,

$-_A(a) = f$ s.t. $f(a_1) = 1$ if $a = a_1$, 0 otherwise, $\forall a_1 \in A$

$f_1 +_A f_2 = f_3$ s.t. $f_3(a) = f_1(a) + f_2(a)$, $\forall a \in A$
Definition (Algebraic Petri net axioms)

Given an algebraic Petri net $\text{N-SPEC} = \langle \text{Spec}, T, P, X, AX \rangle$, an axiom of $AX$ is a 4-tuple $\langle t, Cond, In, Out \rangle$, s.t. :

- $t \in T$ is the transition name for which the axiom is defined.
- $Cond \subseteq T_{\Sigma, X} \times T_{\Sigma, X}$ is a set of equalities attached to transition name $t$ for this axiom, which are satisfied iff all the relations $a = b$ of $Cond$ are satisfied.
- $In = (In_p)_{p \in P}$ is a $P$-sorted set of terms, s.t. $\forall p \in P, In_p \in (T_{\Sigma, X})[\tau(p)]$ is the label of the arc from place $p$ to transition $t$.
- $Out = (Out_p)_{p \in P}$ is a $P$-sorted set of terms, s.t. $\forall p \in P, Out_p \in (T_{\Sigma, X})[\tau(p)]$ is the label of the arc from transition $t$ to place $p$. 
Syntactic description of the example

\[
\text{sum: } (c > 0) = \text{true } \Rightarrow \\
\text{numbers n, total s, counter c } \rightarrow \\
\text{total s + n, counter c -1;}
\]

\[
\text{end: } \text{counter 0, total s } \rightarrow \\
\text{result s;}
\]

Which is equivalent to :

\[
\text{end: } (c=0) = \text{true } \Rightarrow \\
\text{counter c, total s } \rightarrow \\
\text{result s;}
\]
Definition (Markings, initial marking, initially marked algebraic Petri net)

Given an algebra $A$, we define the set of markings $M^A = \{(S_p)_{p \in P}\}$ with $S_p \in [A][\tau(p)], p \in P$.

We call initial marking $m_{\text{init}}$ a $P$-sorted set of terms of $(T[\Sigma, \emptyset])_{\tau(p)}, p \in P$.

Finally, an initially marked algebraic Petri net $N - \text{SPEC}$ is a couple $\langle N - \text{SPEC}, m_{\text{init}} \rangle$. 
Petri Net interpretation

Instantiation of variables accordingly to the algebraic semantic domain.

\[
\text{sum: } (c > 0) = \text{true} \Rightarrow 
\begin{align*}
\text{numbers } n, \text{ total } s, \text{ counter } c &\rightarrow \\
\text{total } s + n, \text{ counter } c -1;
\end{align*}
\]
Events and transition systems

**Definition (Events)**

Given a \( N - SPEC = \langle Spec, T, P, X, AX \rangle \), \( Spec = \langle \Sigma, X', AX' \rangle \) and \( A \) a \( \Sigma - algebra \). The events are \( E = T \times \{ I : X \rightarrow A \} \) i.e. its transition name and interpretation.

**Definition (Transition system)**

A transition system with labels is a relation : \( State \times Label \times State \)

A transition : \( e \in Label, x \in State \) et \( y \in State \)

\((x, e, y) \in State \times Label \times State \) will be noted \( x \xrightarrow{e} y \)

In the following, depending on the intent of the semantics, the set \( T \) or the events \( E \) will be used as labels in the transition systems.
Definition of APN transition system

Definition (Transition system)

Given a model $A$ of a specification $Spec$, and an algebraic Petri net specification $N$-SPEC $= \langle Spec, T, P, X, AX \rangle$, a transition system over $A$ and $N$-SPEC is a set $TS_A(N$-SPEC $) \subseteq M^A \times T \times M^A$. Its elements, called state transitions, are noted $m \xrightarrow{t} m'$, where $m, m' \in M^A$, $t \in T$. We have : $(m \xrightarrow{t} m') \in TS_A(N$-SPEC $)$ iff there exists an interpretation $\sigma$ s.t. :

- $\exists \langle t, Cond, In, Out \rangle \in AX$ choice of axiom
- $\forall p \in P, \llbracket In_p \rrbracket^A_\sigma \subseteq m(p)$
- $\forall p \in P, m'(p) + \llbracket In_p \rrbracket^A_\sigma = m(p) + \llbracket Out_p \rrbracket^A_\sigma$. 

Didier Buchs

Algebraic Petri nets
Definition (Initially marked transition system)

Let \( \langle N\text{-SPEC}, m_{\text{init}} \rangle \) be an initially marked algebraic Petri net. Then an "initially marked transition system" 
\[ TS_A(\langle N\text{-SPEC}, m_{\text{init}} \rangle) \subseteq TS_A(N\text{-SPEC}) \]  
is the set of state transitions of \( TS_A(N\text{-SPEC}) \) transitively connected to the marking \( \left[ m_{\text{init}} \right]_A^\sigma \) by the transition relation (\( \sigma \) being an interpretation).
Transition system of the example
Observations on the transition system

- Atomicity of the transition firing
- Extension to step semantics
  - Extend to multiset of events
  - Conjunction of conditions
  - Sum of pre/post conditions
- Labelling with/without interpretation
  - $x \overset{t,l}{\rightarrow} y$
  - $x \overset{t}{\rightarrow} y$
Implicit vs explicit conditions

\[
\text{end: counter } 0, \text{ total } s \rightarrow \\
\text{result } s;
\]

Which is equivalent to:

\[
\text{end: } (c=0) \Rightarrow \\
\text{counter } c, \text{ total } s \rightarrow \\
\text{result } s;
\]

There is a normal form, where axioms are written with only variables in the pre-conditions. In conditions and post-condition terms are allowed.
free vs bounded variables

bound:

numbers n ... ->

...;

is bounded if it appears in the pre-condition.

free:

... ->

result s;

is free if it appears only in the post-condition (and not indirectly linked by conditions).
These variables are universally quantified, leading to potentially infinite branching of the TS.
**Definition (Strong Bisimulation)**

A *strong bisimulation* between two transition systems $TS_1$, $TS_2$ is the relation $R \subseteq \text{State}(TS_1) \times \text{State}(TS_2)$ such that :

- if $st_1 \mathrel{R} st_2$ and $st_1 \xrightarrow{e} st'_1 \in TS_1$ then there is $st_2 \xrightarrow{e} st'_2 \in TS_2$ such that $st'_1 \mathrel{R} st'_2$;
- if $st_1 \mathrel{R} st_2$ and $st_2 \xrightarrow{e} st'_2 \in TS_2$ then there is $st_1 \xrightarrow{e} st'_1 \in TS_1$ such that $st'_1 \mathrel{R} st'_2$;
- $st_1^{\text{init}} \mathrel{R} st_2^{\text{init}}$

We say that $TS_1$ and $TS_2$ are *strongly bisimilar*, if there exists such a non-empty relation $R$, and we denote this by $TS_1 \iff TS_2$. 

---

**Didier Buchs**

Algebraic Petri nets
Exemple of bisimulation

Masking of hidden choice:

Diagram:

- States: s0, s1, s2, s3, s4, r0, r1, r2, r3
- Transitions: a, b

- s0 transitions to s1 with a, to r0 with a
- s1 transitions to s3 with b
- s2 transitions to s4 with b
- r0 transitions to r1 with a
- r1 transitions to r2 and r3 with b
- s3 and s4 are not connected to any other states
- r2 and r3 are not connected to any other states
Exemple of bisimulation

Discrimination of choices:

![Diagram showing discrimination of choices](image-url)
Remarks on bisimulation

The bisimulation has properties such as:

- it abstracts states
- it is more discriminant than trace equivalence
- it is less constraining than isomorphism
- it is fully abstract wrt branching temporal logics (CTL)\(^1\)

The effect of the change of algebra (for instance \(A, B \in \text{Alg}(\text{spec})\)) on equivalence, is also linked to observations of the algebras.

\(A =? = B \Rightarrow (TS_A(SP) \Leftrightarrow TS_B(SP))\)\(^2\)

Use of bisimulation to define an implementation relation:

\(SP \models P \Leftrightarrow (TS(SP) \Leftrightarrow TS(P)|_{SP})\)

1. Modulo well chosen atomic propositions
2. The equivalence relation \(=? =\) should take into account similar effect of the \(\subseteq\), cond constraint of the semantic rule. In the general case, unfortunately, isomorphism between algebra is required. On a limited set of possible expression better constraint can apply.
Summary

- We have defined an abstract syntax for algebraic Petri Nets\(^3\)
- The semantics of APN was defined as transition system\(^4\)
- Equivalence of similar systems is defined by bisimulation.
- Implementation notions based on bisimulation have been proposed

---

3. a concrete one is defined for the COOPNBuilder tool
4. an abstract syntax and semantics is defined as ISO/IEC 15909-1 :2004 standard for High Level Nets