Finite and Infinite Model Checking of Dual Transition Petri Net Models

Mauricio Varea, Bashir Al-Hashimi, and Michael Leuschel
Overview

- Target: Embedded Systems
- Model: Dual Transition Petri Nets
- Verification:
  - Finite State Model Checking
  - Infinite State Model Checking
- Concluding Remarks.
“A modern car may have up to 70 Embedded Systems”
[P. Thoma, proc. of DATE’99]
Dual Transition Petri Nets

- Complexity & Heterogeneity are key issues

DATA + CONTROL

DTPN Model

Datapath

Controller

Data Flow

Control Flow
Dual Transition Petri Nets

- **Structure**
  - Places → State / Storage
  - Transitions → Control and Data Flow

- **Behaviour**
  - Complex Marking
  - Tokens and “values”

\[
\mu : P \rightarrow N \quad \mu : P \rightarrow C
\]

\[
\alpha_i \cdot e^{i \cdot \frac{\pi}{2} \cdot \gamma}
\]
Dual Transition Petri Nets

Example: Multiplier

\[
\begin{align*}
q_1 & \rightarrow a \\
q_3 & \rightarrow b \\
p_1 & \rightarrow t_1 \\
p_7 & \rightarrow t_4 \\
p_4 & \rightarrow t_2 \\
p_6 & \rightarrow t_5
\end{align*}
\]

Mauricio Varea, University of Southampton, UK
Verification

- Finite State Model Checking → SMV
- Infinite State Model Checking → CLP
Finite State Model Checking

module ctrl_transition(p,preset,postset,g){
  en : boolean;
  en := &{ (preset[i] > 0) ->
     (p[i].phase >= preset[i]) : i = 1..(NN)}
    & g;

  for(i=1;i<=(NN);i=i+1)
    if (en) pp[i] := (p[i].phase - preset[i] + postset[i])
       mod Kc;
    else pp[i] := p[i].phase;
}

module data_transition(p,preset,postset,ctrl,h,k,dl){
  en : boolean; en := (k in ctrl) & (~dl);

  for(i=1;i<=(NN);i=i+1)
    sum[i] := (p[i].modulus * preset[i]) mod Kd;
  for(i=1;i<=(NN);i=i+1)
    if (en) pp[i] := ( postset[i] * ((+sum) + h) ) mod Kd;
    else pp[i] := p[i].modulus;
}

typedef place;

module guard(op,sgn,val);

Finite State Model Checking

- LTL / CTL
  - Logic Operators
  - Temporal Operators
  - Path Quantifier

- Tool used: Cadence SMV
  - Non-deterministic scheduler
  - Deadlock free fairness constraint
Finite State Model Checking

Example: 4-bit Ring Counter

\[
G(p[1]=1 \rightarrow X p[1]=2) \\
G(p[1]=2 \rightarrow X p[1]=4) \\
G(p[1]=4 \rightarrow X p[1]=8) \\
G(p[1]=8 \rightarrow F p[1]=1)
\]
Finite State Model Checking

- State Space Exploration

![Graph showing time vs. capacity with Data Domain and Control Domain lines.](image)
Infinite State Model Checking

- Nondeterministic Finite Automaton = Infinite State System
- Verification of Infinite State Systems is UNDECIDABLE, therefore, restrictions are needed in order to make it DECIDABLE
- CLP is ideal for NP-hard problems based on constraints
Concluding Remarks

- DTPN models have a very tight control and data flow representation, which is suitable for Embedded Systems design.

- safe DTPN models $\rightarrow$ SMV
- k-bounded DTPN models $\rightarrow$ CLP
- unbounded DTPN models $\rightarrow$ CLP

Future Work:
- Applicability to Hw/Sw Co-Synthesis.