CIS 525,
SOFTWARE DEVELOPMENT OF PARALLEL AND DISTRIBUTED SYSTEMS
Dr. Boleslaw Mikolajczak
email: bmikolajczak@umassd.edu

LECTURE NOTES

LECTURE 1/2: INTRODUCTION TO CONDITION/EVENT NETS
1. Condition/Event Nets; basic properties and examples: **example** of four seasons of the year and several modifications, **example** producer/consumer problem with unbounded buffer, **example** of modeling logic circuits, **examples** of nonsequential programs, representation of facts (logical expressions) on C/E nets
2. Some examples of system modeling using C/E nets:
- **example** of access rights of six processors to memory with exclusive WRITE, and concurrent READ
- **example** of organization of the borrowing process in a public library; multiple refinement.

Homework #1; analysis of elementary properties of C/E nets.

LECTURE 3/4: C/E AND P/T NETS IN DIFFERENT AREAS OF COMPUTER SCIENCE
1. Additional examples of C/E and P/T nets: **example** of gas pump system with deadlock, **example** of concrete production process, **example** of five philosophers problem, **example** of sender/receiver system, **example** of instruction execution cycle in a processor, **example** instruction pipelining in pipelined processor, **example** of subnet of discrete mathematics, **example** of fault-tolerant systems and their synthesis using PNs, **example** of p-process and q-process system with three types of different resources, **example** of synthesis of digital control system from PNs (hardware and software implementation), **example** of railway road intersection.

LECTURE 5/6: DYNAMIC PROPERTIES OF C/E NETS
1. *Contact-free* C/E nets: what does it mean to have contact situation, algorithm of designing contact-free C/E net from arbitrary C/E net, net, dual net, subnet, *Case graphs* of C/E nets, Serialization of concurrent events, *Matrix representation* of C/E nets and P/T nets.

Homework # 2: control system with two wagons/its reachability graph and hardware/software implementation of C/E nets and P/T nets.

LECTURE 7/8: DESIGN/CPN SOFTWARE PACKAGE ANALYSIS
1. General characteristic of the *Design/CPN* software package: place inscriptions, event inscriptions, arc inscriptions - bindings, transition guards
2. **Example:** Hierarchical version of the five philosophers problem using DESIGN/CPN.
3. Behavioral subclasses of the C/E nets and systems: sequential vs. concurrent, deterministic vs. nondeterministic, conflict vs. conflict-free, conflict set, confusion, increasing confusion, decreasing confusion, confusion = concurrency+conflict.
4. **Example:** DESIGN/CPN version of the producer/consumer problem with bounded and unbounded buffer.

LECTURE 9/10: OCCURRENCE SETS AND OCCURRENCE GRAPHS OF CP-NETS
1. **Computer demonstration** of the p-processes and q-processes problem with three different resources: experimental presentation of deadlock, liveness for concurrent systems.
2. **Example** of distributed database system using DESIGN/CPN: analysis of the model, place invariants, proving deadlock and liveness using place invariants.
3. *Occurrence graph* (OG): occurrence element, occurrence set, firing strategies - fairness and scheduling algorithms.

LECTURE 11/12: DESIGN/CPN SIMULATOR AS DEBUGGER OF PARALLEL PROGRAMS
1. What can be proved by formal analysis of CP nets and what can be achieved by simulation.
2. Comments on the meaning of nondeterminism in parallel processing: don't know nondeterminism, don't care nondeterminism, conflict nondeterminism.
3. Requirements of a parallel debugger (what kind of major functions a parallel debugger should satisfy).
4. **Example:** Proving system properties of Readers/Writers problem using place invariants.
5. Features of good *parallel debugger*: detecting and avoiding deadlocks, detecting and controling conflicts (nondeterminism); fairness of conflict resolution schedules, prediction of all potential states of the concurrent system, peachability issue between arbitrary two system states.

LECTURE 13/14: ANALYSIS OF CONCURRENCY IN C/E NETS
1. Processes of C/E nets: definition, properties, examples.
2. Relation of concurrency and linear relationship in C/E nets; slices, cuts, linear segments, ^0B, B^0,^-B, B^+.

Project #1; modeling of **lift system** using P/T nets and CP nets.

LECTURE 15/16: SEMANTICS OF TIMED PETRI NETS AND MODELING USING CP-NETS
1. What are responsive systems? (real-time, fault-tolerant, parallel/distributed)
2. Two main techniques to represent time in PNs (time represented on transitions, on places)
3. **Example** of *timed* PNs (apples reselling process). *Weak and Strong Semantics* in Timed Petri nets.

4. **Example:** ADA program segment modeling using *Timed PNs.*
5. **Example** of chemical reaction with three compounds using *Timed Petri net.*
6. Algorithm how to transform timed PNs with places into timed PNs with transitions. Refiring of transitions in Timed PNs.
7. Timed PNs with duration and delay on transitions. Time in PN/CP-nets; how simulation with time proceeds using DESIGN/CPN software?
8. Factors having influence on size of timestamps (current model time, color of the CPN variables, reference variables, input files)
9. **Example** of modeling the Boston-Cape Cod ferry system using DESIGN/CPN timing facilities with McIntosh computer simulation. Performance evaluation of system modeled with timing facilities.

LECTURE 17/18: MODELING OF FLEXIBLE MANUFACTURING SYSTEMS USING HIERARCHICAL CP-NETS

1. *Hierarchy* highlights (hiding, reusability, freedom of decision). *Hierarchical CP-nets* and their constructs: general characterization of hierarchical CP-nets, static page instances: prime pages, semantics of hierarchical substitution of places (by substitution of places we understand ADTs recursive application of this rule), semantics of *hierarchical substitution* of transitions, dynamic *page instances:* semantics of *hierarchical invocation* of transitions, global *fusion of places*.
2. **Example** of factory assembly line modeled with hierarchical CP-nets. Folding is applicable when one has identical objects, but less suited for asymmetrical arrangements.

Homework #3: *Synchronic distance* of C/E nets and properties of the C/E nets.
Homework #4: Hierarchical *Colored Petri nets* and their behavioral properties.

LECTURE 19/20: SYNCHRONIZATION OF EVENTS IN PETRI NETS

1. *Synchronic distance* of two sets of events in C/E nets.
2. Algorithm to compute synchronic distance for C/E nets using distance of events and variance.
3. Practical method of computing *synchronic distance* for concurrent, sequential and cyclic systems.
4. Processes and their compositions; elementary processes.
5. Ambiguity of synchronic distance interpretations from concurrency point of view.

LECTURE 21/22: MODELING OF PHONE SYSTEM USING HIERARCHICAL CP-NETS

1. Assumptions: *phone system* as perceived by the user, no timeouts, no special services.
2. Analysis of *subnets:* Establishing Connection, Breaking Connection by the Recipient, Breaking Connection by the Sender, Proving system properties using *place invariants*: what happens when user calls himself, what happens when someone calls a number which has Established Connection, what happens when Established Connection is broken by the Recipient
3. Computer *simulation of the phone system* using DESIGN/CPN.

Project # 2; extension of given **phone system** using: directory service, police service, emergency service, multi-area service. Proving properties of system using place invariants.

LECTURE 23/24: MODELING OF CACHE COHERENCY PROTOCOLS FOR MULTIPROCESSORS WITH DISTRIBUTED CACHES USING HIERARCHICAL CP-NETS

1. ARCHITECTURES OF MULTIPROCESSORS WITH PRIVATE CACHES AND INTERCONNECTION NETWORKS
2. TOP-DOWN DESCRIPTION AND MODELING OF THE ARCHIBALD'S COHERENCY PROTOCOL USING Pr/T NETS.
3. DESCRIPTION OF COMPLEX SYNCHRONIZATION OF THE ARCHIBALD'S PROTOCOL.
4. DERIVATION OF INVARIANTS DIRECTLY FROM THE Pr/T NET WITHOUT UNFOLDING IT.
5. BEHAVIORAL PROPERTIES OF CACHE COHERENCE PROTOCOL ARE STUDIED (CORRECTNESS, FAIRNESS, BOUNDS FOR THE HARDWARE RESOURCES).

LECTURE 25/26: HOW TO COMPUTE PLACE INVARIANTS OF CP-NETS ?

1. *Equational/matrix representations* of CP nets.
2. *Well-formedness* and *soundedness* of matrix representation of CP-nets.
3. Four methods of matrix representation reduction (similar in nature to *Gaussian elimination* method)
4. *Reachability trees* for CP-nets: theoretical backgrounds, *five dining philosophers* problem
4. **Three examples** of CP nets reduction and invariant computation: *distributed database system*, *public phone system*, *hierarchical version of the five philosopher's system.*

LECTURE 27/28: STUDENT ORAL PRESENTATIONS OF RESEARCH PAPERS:

1. F. Pinto: *Specification of Pascal Semantics Using Petri Nets*
2. T. Gessl: *Modeling and Specification of Communication Protocols using Numerical Petri Nets*
3. Ke Tao: *Modeling and Specification of Prolog Logic Programs Using Place/Transition and Predicate/Transition Petri Nets*
4. H. Trivedi: *Object-Oriented Programming and Software Engineering Using Hierarchical Petri Nets*

FINAL EXAM: Analysis of CP net modeling cache coherency protocols for multiprocessor systems.