(INFORMAL)
INTRODUCTION
INTO
PETRI NETS
MODELLING

PETRI NETS, BASICS 1

(1) NODES

places

transitions

“passive elements”
“active elements”
“chem. compounds”
“chem. reactions”

(2) ARCS

preconditions

postconditions

action
(3) **TOKENS**

(moving objects, vehicles, work pieces, control flow pointer, dates, ..., *units of substances* (e.g., Mol), ...)

- condition is not fulfilled
- condition is (one times) fulfilled
- condition is n times fulfilled

(4) **MARKING**

(system state, *substance distribution*)

How many tokens are on each place?

- initial marking

(5) **FLOW OF TOKENS**

- an action *may* happen, if
  
  - all preconditions are fulfilled (corresponding to the arc weights);

- if an action happens, then
  
  - tokens are removed from all preconditions (corresponding to the arc weights), and
  
  - tokens are added to all postconditions (corresponding to the arc weights);

- an action happens (firing of a transition)
  
  - atomic
  
  - time-less
FIRING RULE, EXAMPLE 1

FIRING RULE, EXAMPLE 2

t1 fires

t1 fires
EXAMPLE 1, CHEMICAL REACTION EQUATIONS

- FOR LIGHT-INDUCED PHOSPHORYLATION
  \[ 2 \text{NAD}^+ + 2 \text{H}_2\text{O} \rightarrow 2 \text{NADH} + 2 \text{H}^+ + \text{O}_2 \]

- FROM THE PHOTOSYNTHESIS
  \[ 2 \text{CO}_2 + \text{H}_2\text{S} + 2 \text{H}_2\text{O} \rightarrow 2 (\text{CH}_2\text{O}) + \text{H}_2\text{SO}_4 \]

EXAMPLE 2, PRODUCER/CONSUMER, UNBOUNDED

- SYSTEM WITHOUT ARC WEIGHTS

- SYSTEM WITH ARC WEIGHTS
EXAMPLE 3, PRODUCER/CONSUMER, BOUNDED

SYSTEM WITHOUT ARC WEIGHTS

PRODUCER
- ready_to_send
- send
- produce
- ready_to_produce

CONSUMER
- ready_to_consume
- receive
- consume

SYSTEM WITH ARC WEIGHTS

PRODUCER
- ready_to_send
- send
- produce
- ready_to_produce

CONSUMER
- ready_to_consume
- receive
- consume

EXAMPLE 4, TRAVEL PREPARATION

next_vacation
- begin_preparations
- begin_preparations
- end_preparations
- repeat

accommodation?
- transport_means?
- luggage?

transport_means!
- accommodation!

transport_means?
- accommodation!

transport_means?
- accommodation!

next_vacation
begin_preparations
end_preparations
repeat
TYPICAL BASIC STRUCTURES 0

- FORWARD BRANCHING

- BACKWARD BRANCHING

TYPICAL BASIC STRUCTURES 1

- NECESSARY CONDITION

- SUFFICIENT CONDITION

- CAUSAL RELATION
  
  "X HAS TO HAPPEN BEFORE Y"
TYPICAL BASIC STRUCTURES 2

- SEQUENCE OF ACTIONS

- BRANCHING / ALTERNATIVES

- REPETITION

TYPICAL BASIC STRUCTURES 3

- CONCURRENCY

- COMMUNICATION / SYNCHRONISATION
TYPICAL BASIC STRUCTURES 4

- STATIC CONFLICT

- DYNAMIC CONFLICT

TYPICAL BASIC STRUCTURES 5

- FREE OF DYNAMIC CONFLICTS, EX. 1

- FREE OF DYNAMIC CONFLICTS, EX. 2
CONFUSION

- concurrency and conflict overlap
  - \( t1 \not\# t2 \) and \( t2 \not\# t3 \),
  - but \( t1 \) concurrent to \( t3 \)

- **case 1**: \( t1 < t3 \)
  - conflict \( t2 \# t3 \) disappears,
  - firing of \( t3 \) does not involve a conflict decision

- **case 2**: \( t3 < t1 \)
  - conflict \( t2 \# t3 \) exists,
  - firing of \( t3 \) involves a conflict decision

- the interleaving sequences of concurrency may encounter different amount of decisions

- an observer outside of the system does not know whether a decision took place or not