METRO :
Mapping Specification Notations to Analysis Tools

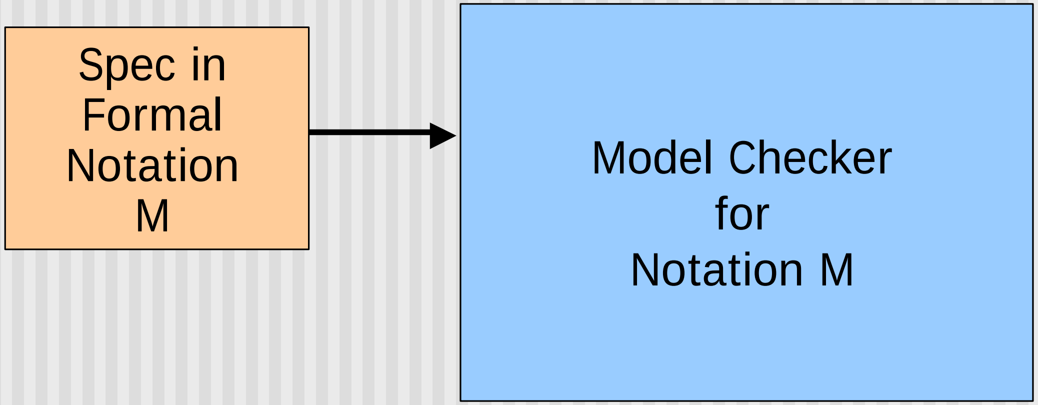
Jianwei Niu   Yun Lu   Joanne Atlee   Nancy Day

{jniu, y4lu, jmatlee, nday}@uwaterloo.ca

WATFORM

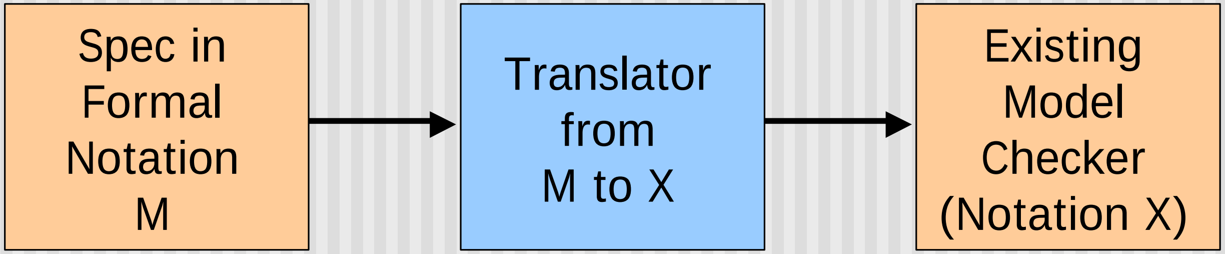
University of Waterloo

Current Approaches for Constructing Analysis Tools



Focus on model-based specification notations
(e.g., statecharts variants, process algebras)

Current Approaches for Constructing Analysis Tools



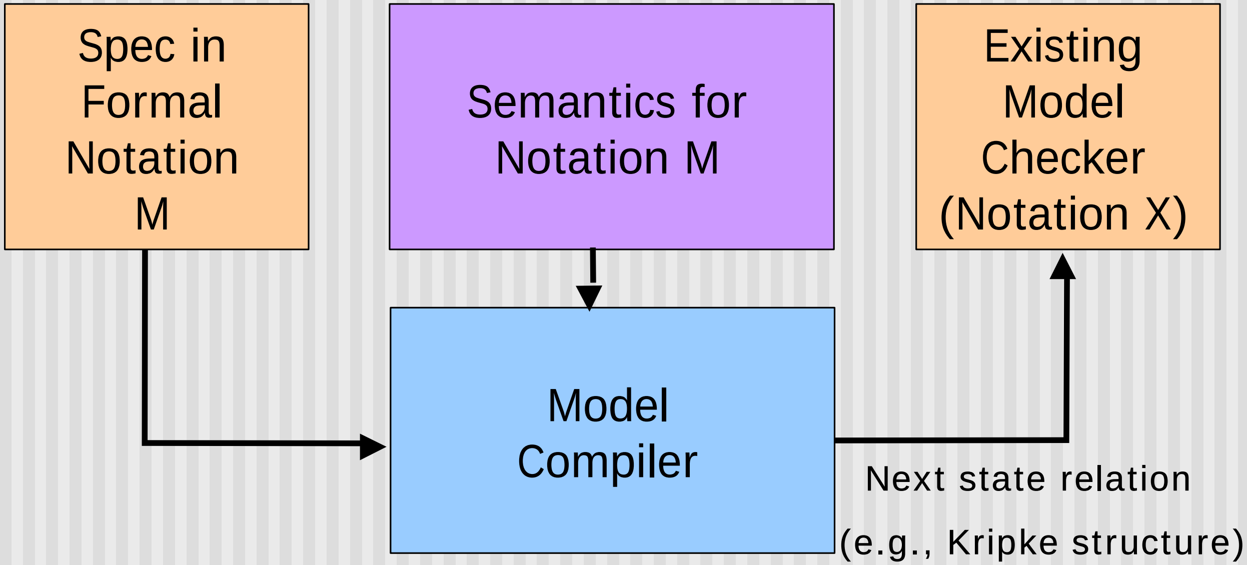
Spec in Formal Notation M

Translator from M to X

Existing Model Checker (Notation X)

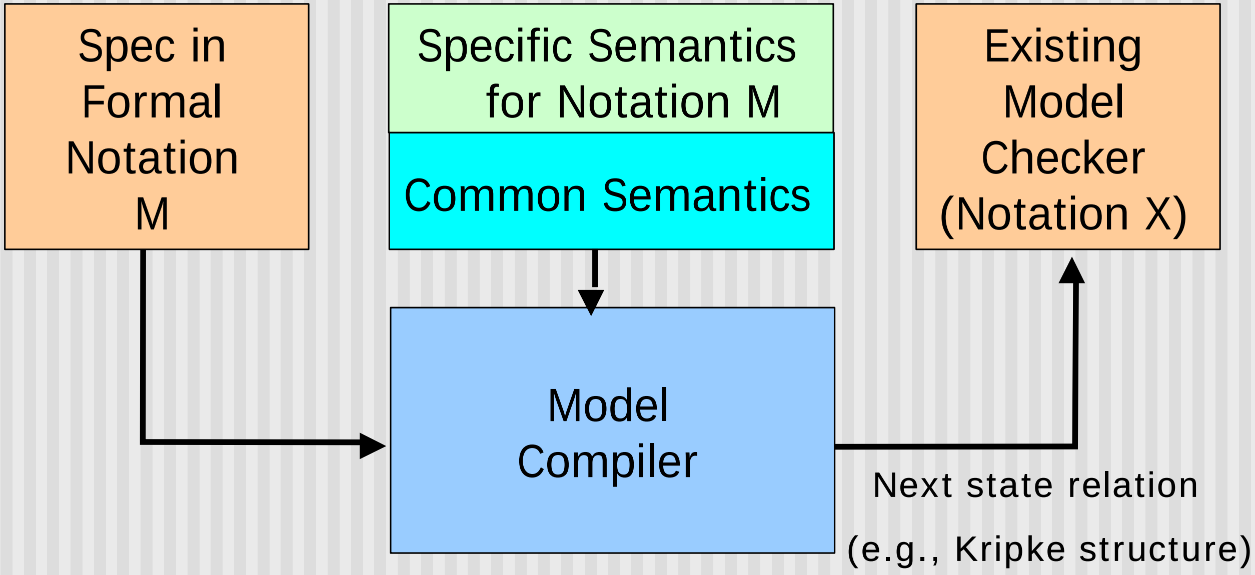
It is difficult to maintain a customized analysis tool or a translator when the notation evolves

Current Approaches for Constructing Analysis Tools

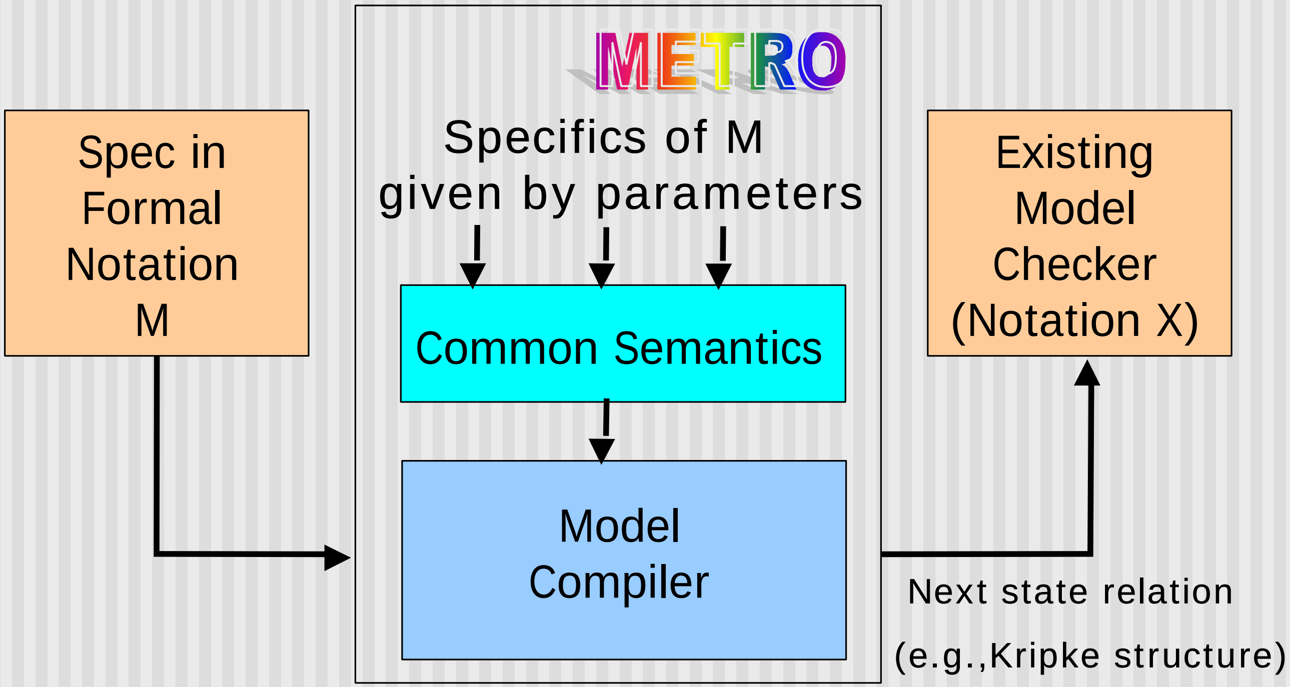


It is hard for users to write the semantics of notation in a semantic description language

Our Approach for Mapping Specification Notations to Analysis Tools


Spec in Formal Notation M
Specific Semantics for Notation M
Common Semantics
Existing Model Checker (Notation X)
Model Compiler
Next state relation
(e.g., Kripke structure)

Our Approach for Mapping Specification Notations to Analysis Tools



METRO

Spec in Formal Notation M

Specifics of M given by parameters

Common Semantics

Model Compiler

Existing Model Checker (Notation X)

Next state relation

(e.g.,Kripke structure)

Outline

- Hierarchical Transition Systems
- Template semantics
  - Step semantics
  - Composition operators
- METRO --- mapping notations to analyzers

Computation Model

- Hierarchical Transition Systems (HTS) with
  - States and state hierarchy
  - Internal events
  - External events
  - Variables
  - Transitions

    <source_state, trig_event, condition, action, destination_state>
- No concurrency

  concurrency introduced when composing multiple HTSs

Semantics of HTS --- Snapshots

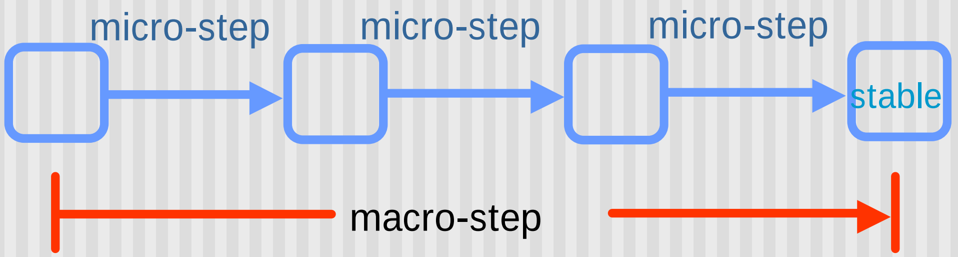
- Snapshot: observable point in execution

Basic Elements	current states current internal events current variable values generated external events	
Auxiliary Elements	auxiliary states auxiliary internal events auxiliary variable values auxiliary external events	used to determine which transitions are enabled

Semantics of HTS --- Steps

- Operational semantics: a relation over pairs of snapshots (steps)
- Micro-steps: execute a single transition
- Macro-steps: execute a sequence of micro-steps until reach a stable snapshot

stable snapshot : no transitions are enabled in the snapshot



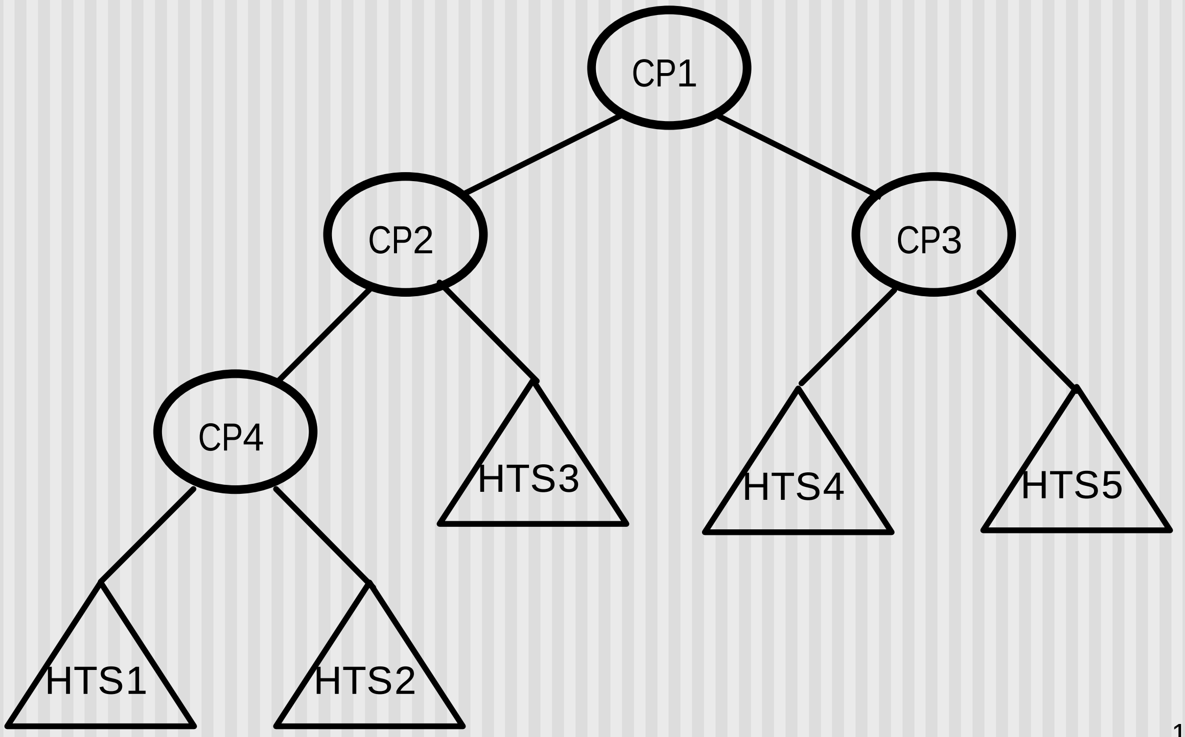
Template Semantics of HTS

Common semantics	Template parameters
enabled transitions	enabling states enabling events enabling variable values
apply	change state generate events change variable values
init	initialize state info initialize event info initialize variable info

Outline

- Hierarchical Transition Systems
- Template semantics
  - Step semantics
  - Composition operators
- METRO --- mapping notations to analyzers

Composition Operators



Semantics of Composition Operators

- Represent concurrency, communication, and synchronization
- Constrain
  - Which components to execute
  - When to transfer control between components
  - How to exchange events and data
- Composition at micro-step and macro-step levels

Seven Composition Operators

- Interleaving
- Parallel
- Synchronization
  - Environmental
  - Rendezvous
- Sequence
- Choice
- Interrupt

Template Semantics for Specification Notations

- Instantiation of the template semantics
  - Define parameters
  - Choose composition operators
- Descriptions of notations’ semantics using our template semantics
  - CCS, CSP, LOTOS
  - Statecharts variants

    Harel’s, Pnueli & Shalev’s, RSML, STATEMATE
  - SCR
  - SDL
  - Petri Nets

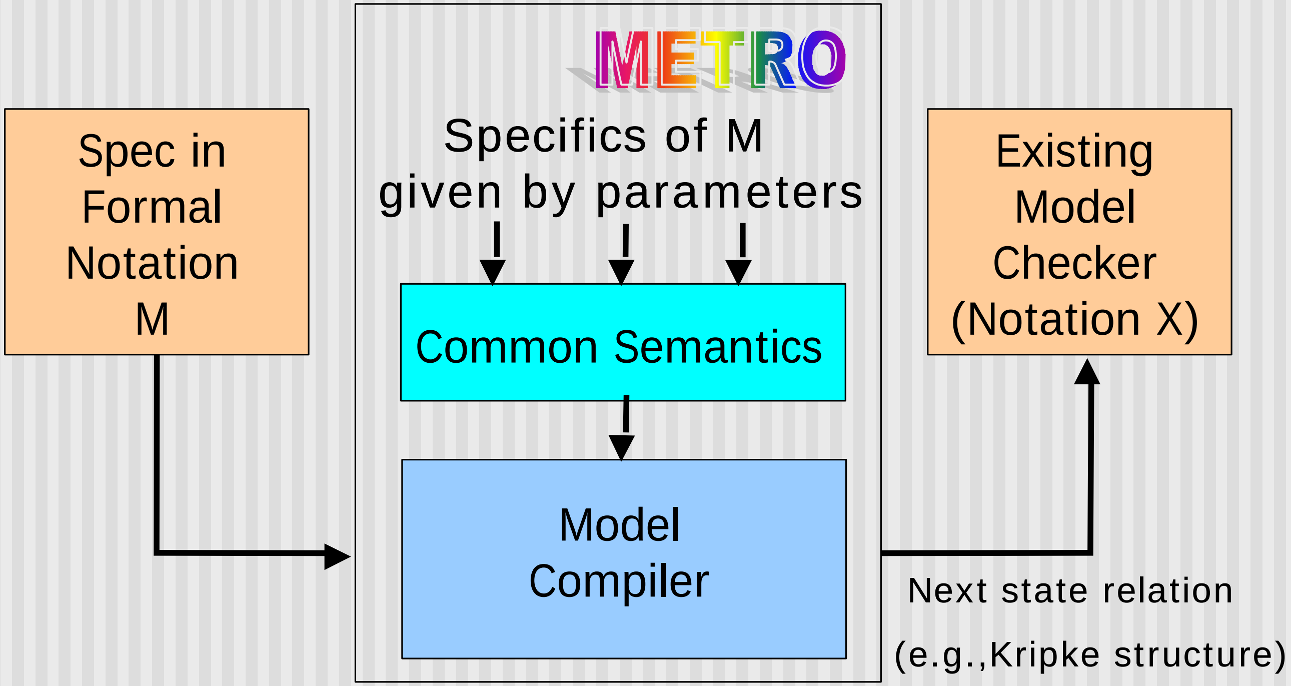
Outline

- Hierarchical Transition Systems
- Template semantics
  - Step semantics
  - Composition operators
- METRO --- mapping notations to analyzers

METRO

- Our template-based semantics framework, METRO, can be used to generate effectively a transition-relation, which then can be used as an input to formal analysis tools
- We are implementing METRO in higher-order logic
  - We have implemented a slice of METRO to handle
    - basic transition systems
    - hierarchical machines

Mapping Specification Notations to Analysis Tools



Spec in Formal Notation M

METRO

Specifics of M given by parameters

Common Semantics

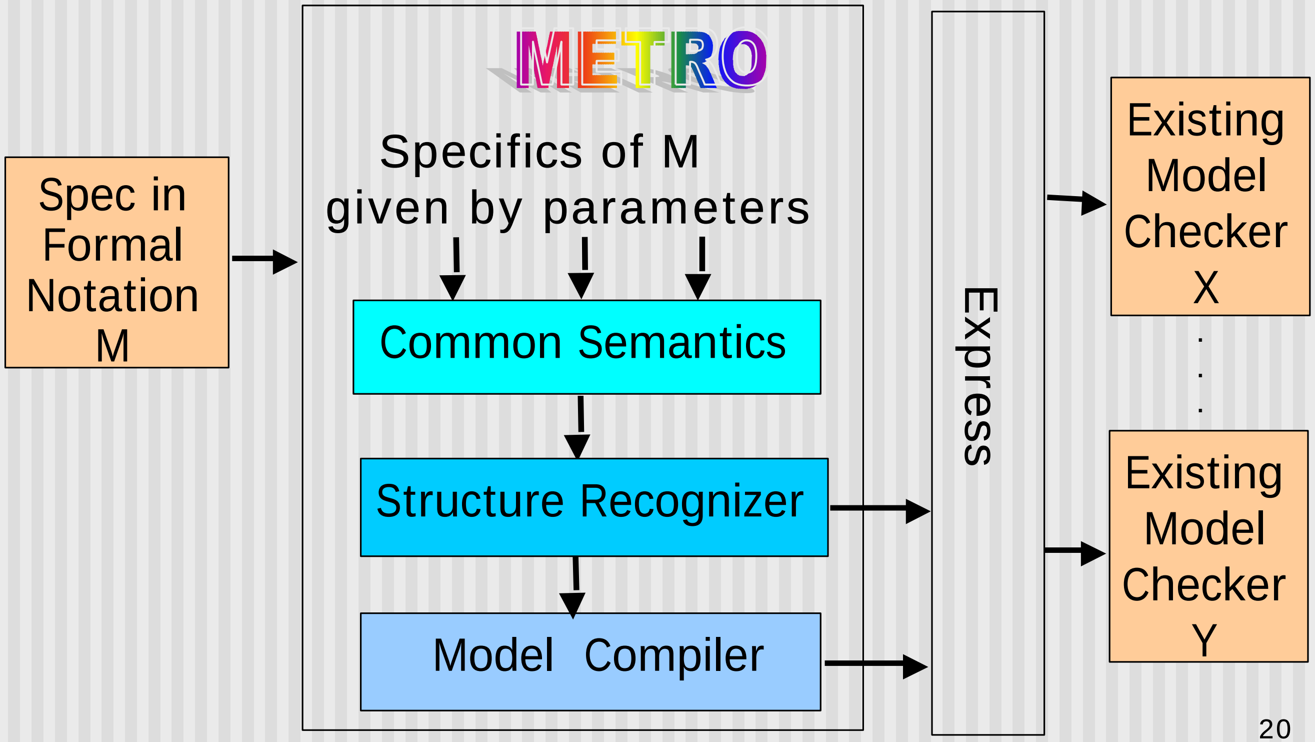
Model Compiler

Existing Model Checker (Notation X)

Next state relation

(e.g.,Kripke structure)

Optimization



Spec in Formal Notation M

METRO

Specifics of M given by parameters

Common Semantics

Structure Recognizer

Model Compiler

Express

Existing Model Checker X

. . .

Existing Model Checker Y

Current Results

- We have developed template semantics to capture the common semantics and specify a notation's distinct behaviours as parameters (FSE'02)
  - template semantics separate step-semantics from composition operators

- We have defined the semantics of specification notations as instantiations of our template (e.g., statecharts variants, SCR, SDL) (RE'03)
  - understand and compare the semantics of notations

Future Work

- Complete the implementation of METRO
  - seven composition and concurrency operators
- Optimization of the mapping from specification notations to analysis tools
- Case studies to validate the approach

Questions?