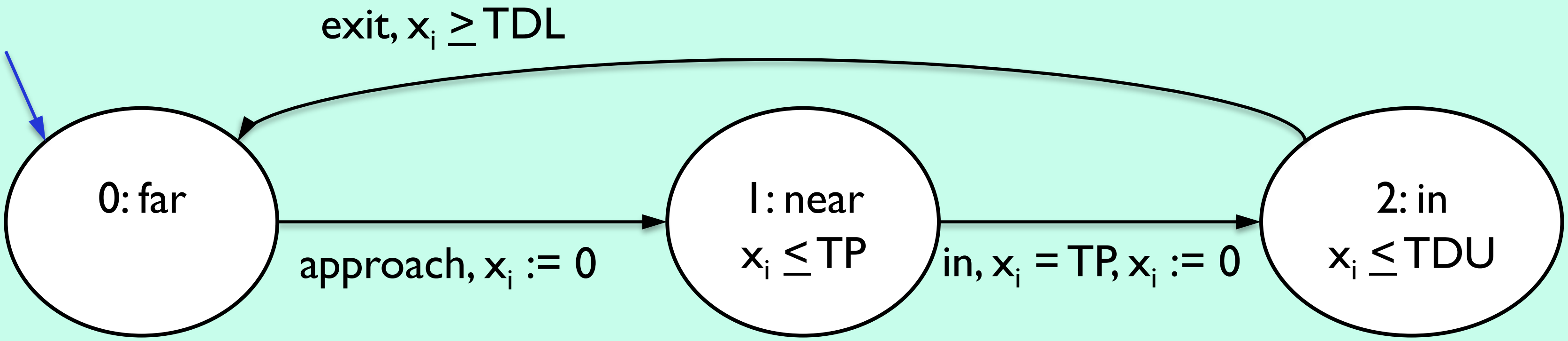
Analyzing Data Structure Choices for On-The-Fly Real Time Model Checking

Peter Fontana and Rance Cleaveland

University of Maryland, College Park

Work In Progress, April 28, 2011

Real-Time Model Checking



TCTL (Invalid): $AF_{<\infty}[\text{near} \lor \text{in}]$

TCTL (Valid): $AG_{<\infty}[\text{near} \rightarrow AF_{\leq TP+TDU}[\text{far}]]$

Background

- Timed Automata model checkers
  - *UPPAAL, RED, KRONOS*
  - Restricted sets of properties
- Predicate Equation Systems (PES) [Zhang, Cleaveland, 2005]
  - First order logic with fixpoint formulae
  - General framework for on-the-fly model checking

On-The-Fly Model Checking

- Goal-directed proof construction
- Uses circularity to detect fixpoints
- For timed automata:
  - *Clock zones* represent sets of states concisely
  - Clock zone data structures important for performance

Goals

- Investigate the impact of clock zone data structures of on-the-fly model checking performance
- **Context:** use PES engine to model check a *subset* of *SIMULINK*

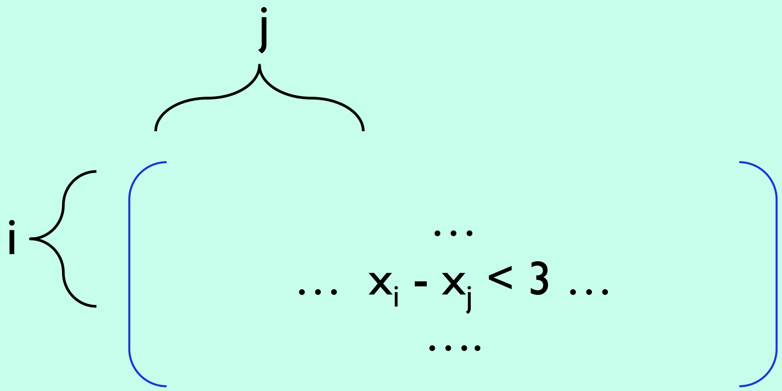
Clock Zones

- **Example:** $x_1 = 2 \wedge x_2 < 3 \wedge x_1 - x_3 \leq 1$
- Clock Zone = *convex* set of clock constraints
- **Definition:**

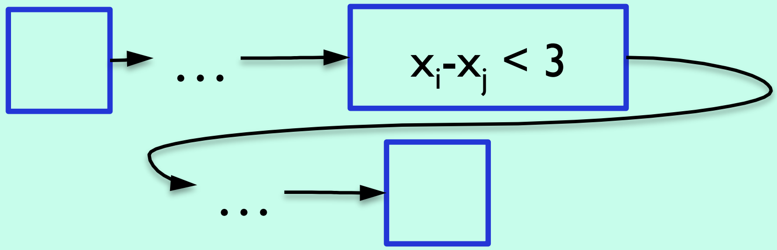
  $\mathbf{z :=} x < c \mid x > c \mid x \leq c \mid x \geq c \mid x - y < c$
  $\mid x - y > c \mid x - y \leq c \mid x - y \geq c \mid z_1 \wedge z_2$

Clock Zone Implementations

- DBM: Matrix (Difference Bound Matrix)



- CRDZone: Linked list, nodes in lexicographical order (omit implicit nodes)



Experiment

- **Purpose:** Analyze performance of DBM, CRDZone on PES-based on-the-fly model checking
- **Hypothesis:** The CRDZone will improve ***time*** and space performance
- **Setup:**
  - Replace DBM with CRDZone in model checker
  - Compare time, space on various benchmarks

Benchmark Suite

- A: valid specification, correct system
- B: invalid specification, correct system
- C: valid specification, buggy system
- 21 model-checker invocations per category

Preliminary Data Analysis

- Compare **paired differences** between DBM and CRDZone
- **Conclusions:**
  - CRDZone performs slightly faster for majority
  - Huge variation

Statistic	**DBM – CRDzone (time - s)**	**DBM – CRDZone (space – MB)**
#Benchmark	37	37
Mean	0.42	-104.0
Standard Deviation	1001.40	298.2
95% CI (Mean)	-333.67 – 334.10	-203.4 – (-4.6)
P-Value for Mean $\neq 0$	0.999	0.033
Median	7.21	-0.5
P-Value for Median $\neq 0$	0.012	0.157

Future Work

- Expand checkable specification range
- Continue optimizing code for performance
- Further uses for PES Engine
    - SIMULINK
    - Vacuity checking