Local Reasoning in the Verification of Parameterized Systems

Swen Jacobs

with Johannes Faber and Viorica Sofronie-Stokkermans

Mar 17 2008

What is this about?

- Local Reasoning

  We know how to instantiate quantified formulae

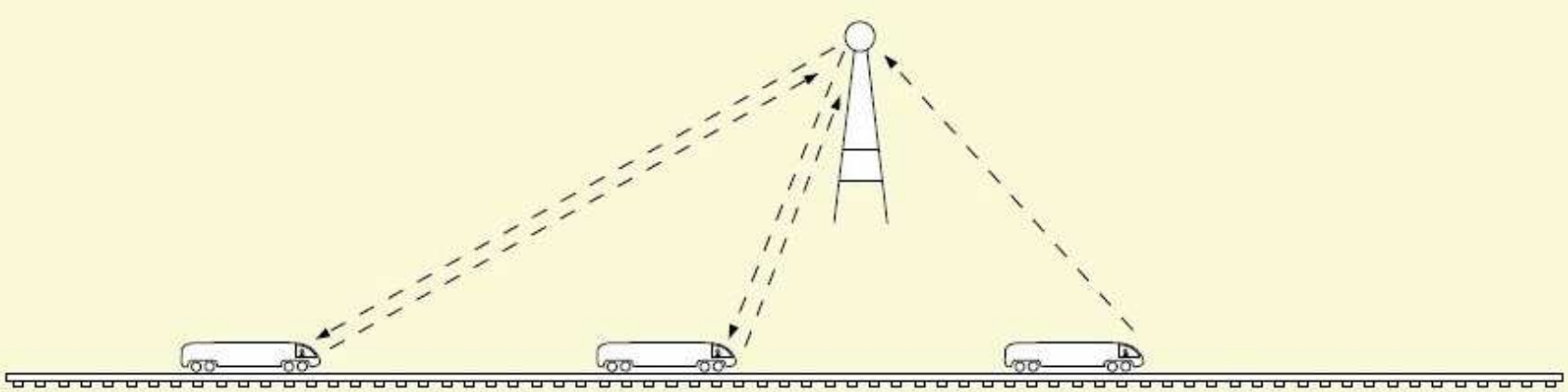
- Verification

  We want to prove properties of a formally specified system

- Parameterized Systems

  Parameters are number of (identical) components or constant values which influence behaviour of the system

Case Study: European Train Control System



Time between updates:	$\Delta_t > 0$	$\mathbb{R}$
Minimum and maximum speed:	$0 \leq \mathit{min} < \mathit{max}$	$\mathbb{R}$
Minimum secure distance:	*alarm*	$\mathbb{R}$
Number of trains:	$n > 0$	$\mathbb{N}$
Train positions:	*pos*	$\mathbb{N} \rightarrow \mathbb{R}$

System Behaviour

Update rules, e.g.:

(F2) $\forall i\ (0 < i < n\ \wedge\ pos(i-1) > 0\ \wedge\ pos(i-1) - pos(i) \geq alarm$
$\rightarrow pos(i) + \Delta_t * \min \leq pos'(i) \leq pos(i) + \Delta_t * \max)$

In general,

$$\forall i\ \phi[pos] \rightarrow \psi[pos'(i)]$$

s.t. ϕ are mutually exclusive for a given i

and ψ is either $t_1 \leq pos'(i) \leq t_2$ or $pos'(i) = t$.

Verification

Monotonicity:

$$Mon(pos): \quad \forall i, j: \ 0 \leq i < j < n \rightarrow pos(j) < pos(i)$$

Safety condition: $Mon(pos) \cup Update \models Mon(pos')$,

or $Mon(pos) \cup Update \cup \neg Mon(pos') \models \bot$

Locality of Mon(pos) and Update allows us to reduce problem to the ground fragment of $\mathbb{R} \cup \mathbb{N} \cup EUF$, for which efficient solvers exist.

Extensions (I)

Consider a variable number of up to n trains:

Function pos

1	2	3	4	5	6	7	8	9				
p1	p2	p3	p4	p5	p6	p7	p8	p9				

Update rules: $\forall i : first \leq i \leq last \wedge \phi_j \rightarrow t_1 \leq pos(i) \leq t_2$

[*PDPAR*06, *ENTCS*07] Applications of Hierarchical Reasoning in the Verification of Complex Systems

Extensions (II)

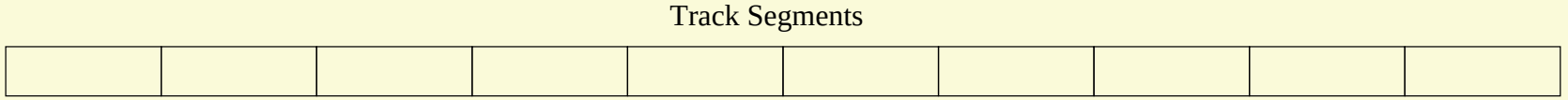
In addition to normal operation, also consider emergencies:

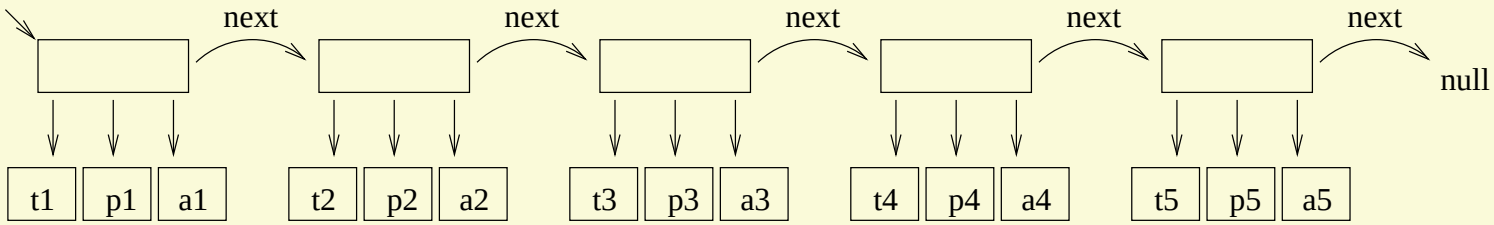
- separate update rules for emergency case
- consider braking distances
- using formal specification in CSP-OZ-DC

[*IFM*07] Verifying CSP-OZ-DC specifications with complex data types and timing parameters

Current & Future Work (I)

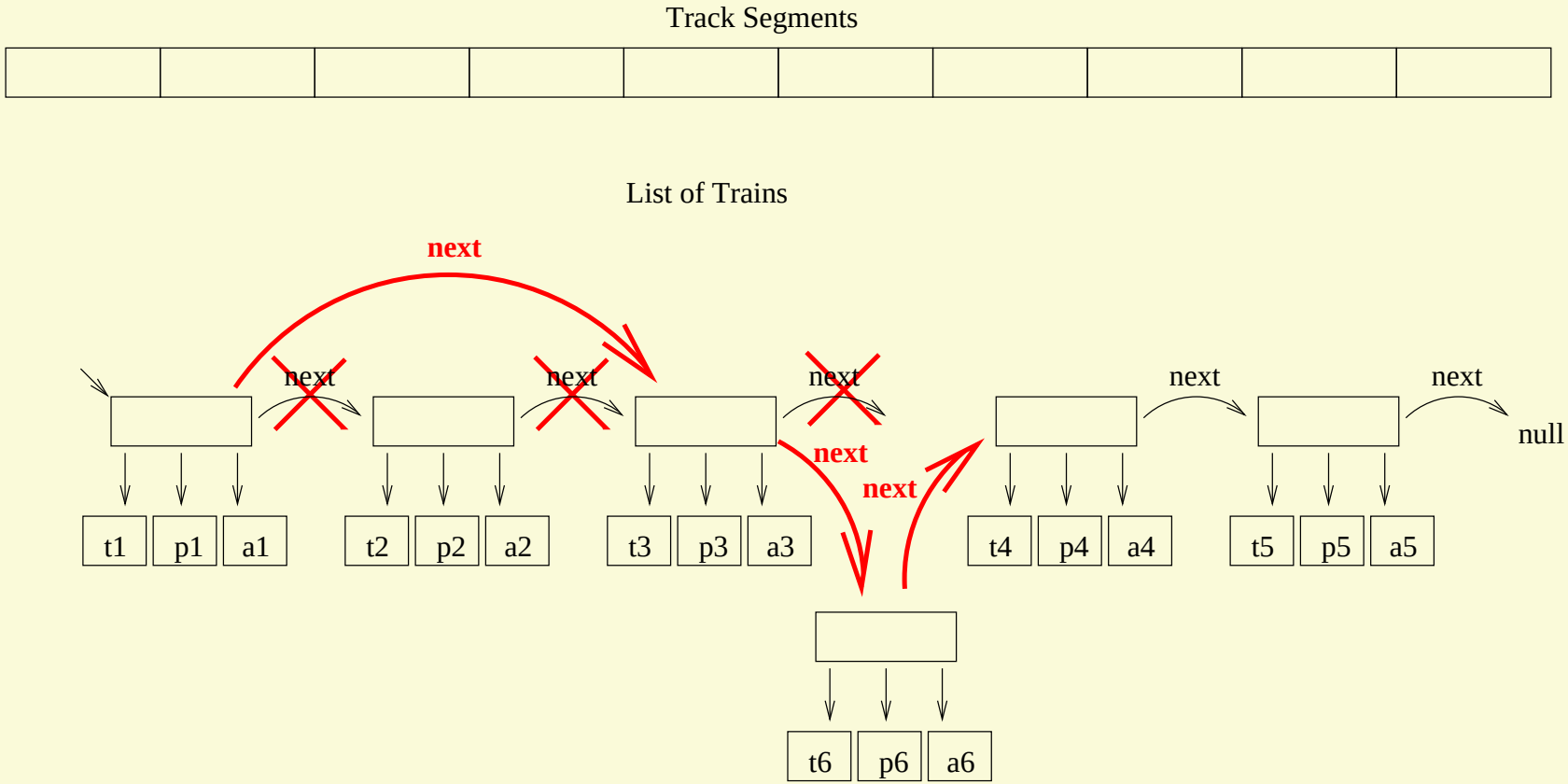
Pointer data structure (see [*TACAS*08]) and track segments:





Current & Future Work (I)

Pointer data structure (see [*TACAS*08]) and track segments:



Current & Future Work (II)

Complex track topology: