Safety vs. Liveness

Safety Properties

- Something (bad) *never happens*.
- e.g. invariance properties

Liveness Properties

- Something (good) *will happen*
- e.g. termination, progress
- Liveness properties are *essential*.

Systems Development using Event-B

M_0 — refined by → M_1 ··········· refined by ··········→ M_n

satisfies ϕ_0 satisfies ϕ_1 satisfies ϕ_n

- $\phi_0, \phi_1, \ldots, \phi_n$: safety properties.

Unit-B = UNITY + Event-B

- Developments using Unit-B are *guided by both safety and liveness requirements*.

scheduling assumptions

+ + +

M_0 — refined by → M_1 ··········· refined by ··········→ M

\+ preserves *liveness* \+ *preserves liveness*

satisfies ϕ_0 satisfies ϕ_1 sat

\+ + +

progress properties

Traces and the Language of Temporal Logic

A trace σ is a (finite or infinite) sequence of states

$$\sigma = s_0, s_1, s_2, s_3, \ldots$$

- A (basic) state formula P is any *first-order logic formula*,
- The basic formulae can be extended by combining the Boolean operators $(\neg, \wedge, \vee, \Rightarrow)$ with *temporal operators*:

always: $\Box\,\phi$ $s_0 \to s_1 \to s_2 \to s_3$ - - - - ; ϕ ϕ ϕ ϕ

eventually: $\Diamond\,\phi$ $s_0 \to s_1 \to s_2 \to s_3$ - - - - ; $\neg\phi$ $\neg\phi$ ϕ $\neg\phi$

until: $\phi_1 \,\mathcal{U}\, \phi_2$ $s_0 \to s_1 \to s_2 \to s_3$ - - - - ; ϕ_1 ϕ_1 ϕ_1 ϕ_2

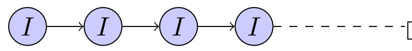
Unit-B Models. *Guarded* and *Scheduled* Events

e

any t **where**
  $G.t.v$
during
  $C.t.v$
upon
  $F.t.v$
then
  $S.t.v.v'$
end

- Execution of e.t corresponds to a formula $act.(\mathrm{e}.t)$.

$\Box I$



- $C.t.v$: *coarse-schedule*.
- $F.t.v$: *fine-schedule*.
- Healthiness condition:

$$C.t.v \land F.t.v \;\Rightarrow\; G.t.v$$

Liveness (Scheduling) Assumption

If $C.t.v$ holds infinitely long and $F.t.v$ holds infinitely often then eventually $\mathsf{e}.t$ is executed when $F.t.v$ holds.

$$sched.(\mathsf{e}.t) \;=\; \Box(\Box C \land \Box \Diamond F \;\Rightarrow\; \Diamond(F \land act.(\mathsf{e}.t)))$$

Schedules vs. Fairness

$\mathsf{e} \mathrel{\widehat{=}}$ **any** t **where** $G.t.v$ **during** $C.t.v$ **upon** $F.t.v$ **then** … **end**

- Schedules are a *generalisation* of weak- and strong-fairness.
- Weak-fairness: If e is *enabled infinitely long* then e eventually occurs.
  - Let C be G and F be $\top$.
- Strong-fairness: If e is *enabled infinitely often* then e eventually occurs.
  - Let F be G and C be $\top$.

Conventions

$\mathsf{e} \mathrel{\widehat{=}}$ **any** t **where** … **during** $C.t.v$ **upon** $F.t.v$ **then** … **end**

- *Unscheduled* events (without **during** and **upon**): C *is* $\bot$
- When only **during** is present (no **upon**), F *is* $\top$.
- When only **upon** is present (no **during**), C *is* $\top$.

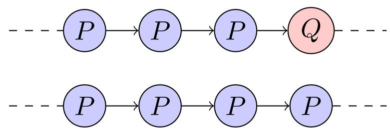
Safety Properties

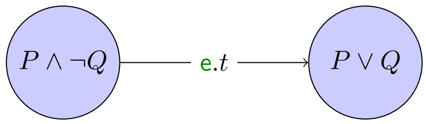
- *Invariance* properties:
- *Unless* properties: P **un** Q
  - Prove: For *every event* $\mathsf{e}.t$ in $\mathbf{M}$

$\Box(P \Rightarrow P\,\mathcal{U}\,Q)$

or

$\Box(P \Rightarrow \Box P)$





Liveness Properties

- *Progress* properties

$$P \leadsto Q \;\;\widehat{=}\;\; \Box(P \;\Rightarrow\; \Diamond Q)$$

- Some important rules

$$(P \Rightarrow Q) \;\;\Rightarrow\;\; (P \leadsto Q)$$

(Implication)

$$(P \leadsto Q) \land (Q \leadsto R) \;\;\Rightarrow\;\; (P \leadsto R)$$

(Transitivity)

A Signal Control System

SAF 1 There is at most one train on each block

LIVE 2 Each train in the network eventually leaves

Refinement Strategy

Model 0 To model trains in the network, focus on *LIVE 2*

Ref. 1 To introduce the network topology

Ref. 2 To take into account *SAF 1*

Ref. 3 To introduce signals and derive a specification for the controller

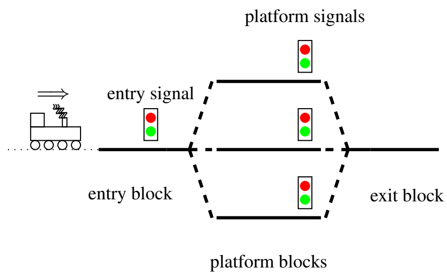
A Signal Control System. The Initial Model

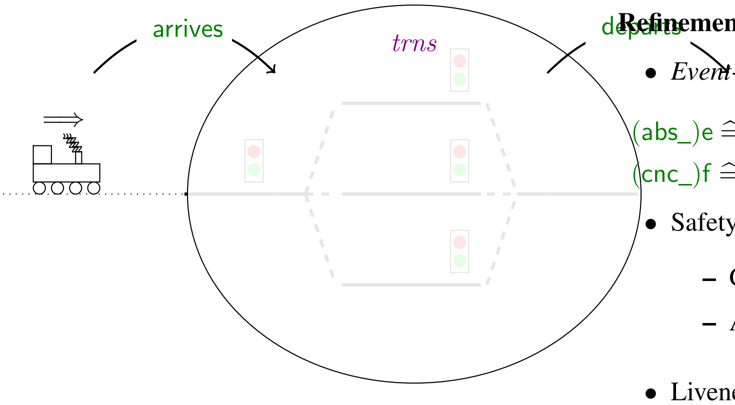
Sketch

LIVE 2 Each *train in the network* eventually leaves

variables : $trns$

invariants : $trns \subseteq TRN$





Refinement

- *Event-based* reasoning.

(abs_)e $\widehat{=}$ **any** t **where** G **during** C **upon** F **then** S **end**
(cnc_)f $\widehat{=}$ **any** t **where** H **during** D **upon** E **then** R **end**

- Safety:
  - Guard strengthening: $H \Rightarrow G$
  - Action strengthening: $R \Rightarrow S$
- Liveness:
  - Scheduling assumptions strengthening.
  - Schedules weakening:

$$(\Box C \ \wedge \ \Box\Diamond F) \quad \Rightarrow \quad \Diamond(\Box D \ \wedge \ \Box\Diamond E) \qquad \text{(REF\_LIVE)}$$

Transient Properties

Theorem 1 (Implementing $P \rightsquigarrow \neg P$). $\mathbf{M}$ *satisfies* $P \rightsquigarrow \neg P$ *if there exists an event in* $\mathbf{M}$

e $\widehat{=}$ **any** t **where** $G.t.v$ **during** $C.t.v$ **upon** $F.t.v$ **then** $S.t.v.v'$ **end**

such that

$$\Box(P \Rightarrow C)\ , \qquad \text{(SCH)}$$

$$C \rightsquigarrow F\ , \qquad \text{(PRG)}$$

$$P \wedge C \wedge F \xrightarrow{\text{e}.t} \neg P \qquad \text{(NEG)}$$

- Note: general progress properties can be proved using the *induction* or *ensure* rules.

Schedules Weakening
Practical Rules

$$(\Box C \wedge \Box\Diamond F) \quad \Rightarrow \quad \Diamond(\Box D \wedge \Box\Diamond E) \qquad \text{(REF\_LIVE)}$$

Practical rules

- Coarse-schedule following

$$C \wedge F \ \rightsquigarrow \ D \qquad \text{(C\_FLW)}$$

- Coarse-schedule stabilising

$$D \ \mathbf{un} \ \neg C \qquad \text{(C\_STB)}$$

- Fine-schedule following

$$C \wedge F \ \rightsquigarrow \ E \qquad \text{(F\_FLW)}$$

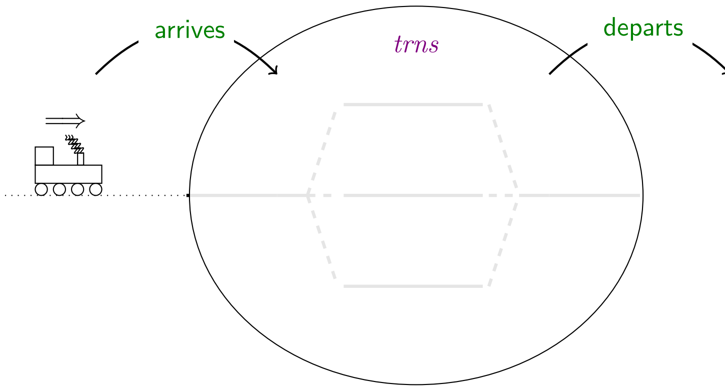
A Signal Control System. The Initial Model
Properties

departs
  any t **where**
    $t \in TRN$
  during
    $t \in trns$
  then
    $trns := trns \setminus \{t\}$
  end

prg0_1 : $t \in trns \ \rightsquigarrow \ t \notin trns$

- (SCH) is trivial.
- No fine-schedule (F is $\top$) hence (PRG) is trivial.
- The event falsifies $t \in trns$ (NEG)

A Signal Control System. The First Refinement
The State



inv1_1 : $loc \in trns \rightarrow BLK$

A Signal Control System. The First Refinement

Refinement of departs

(abs_)departs
any t **where**
$t \in TRN$
during
$t \in trns$
then
$trns := trns \setminus \{t\}$
end

(cnc_)departs
any t **where**
$t \in trns \wedge loc.t = Exit$
during
$t \in trns \wedge loc.t = Exit$
then
$trns := trns \setminus \{t\}$
$loc := \{t\} \lhd loc$
end

- Guard and action strengthening are trivial.

- Coarse-schedule following (amongst others):

$$t \in trns \;\leadsto\; t \in trns \wedge loc.t = Exit \quad (\text{prg1\_1})$$

A Signal Control System. The First Refinement

New Event moveout

moveout
any t **where**
$t \in trns \wedge loc.t \in PLF$
during
$t \in trns \wedge loc.t \in PLF$
then
$loc.t := Exit$
end

A Signal Control System. The Second Refinement

The State

SAF 1 There is at most one train on each block

invariants :
$$\forall t_1, t_2 \cdot t_1 \in trns \wedge t_2 \in trns \wedge loc.t_1 = loc.t_2 \Rightarrow t_1 = t_2$$

A Signal Control System. The Second Refinement

Refinement of moveout

(abs_)moveout
any t **where**
$t \in trns \wedge loc.t \in PLF$
during
$t \in trns \wedge loc.t \in PLF$
then
$loc.t := Exit$
end

(cnc_)moveout
any t **where**
$t \in trns \wedge loc.t \in PLF \wedge$
$Exit \notin \operatorname{ran}.loc$
during
$t \in trns \wedge loc.t \in PLF$
upon
$Exit \notin \operatorname{ran}.loc$
then
$loc.t := Exit$
end

- Neither weak- nor strong-fairness is satisfactory.
  - Weak-fairness requires $Exit$ to be free infinitely long.
  - Strong-fairness is too strong assumption.

Summary

The Unit-B Modelling Method

- Guarded and *scheduled* events.
- Reasoning about *liveness (progress) properties*.
- *Refinement* preserving safety and liveness properties.
- Developments are *guided by safety and liveness requirements*.

Future Work

- Decomposition / Composition
- Tool support

Refinement

The UNITY way vs. the Event-B way

- UNITY: Refines the *formulae*.

$$\overbrace{\phi \Leftarrow \phi_1 \Leftarrow \ldots \Leftarrow \phi_n}^{\text{Refinement}} \quad \underbrace{\phi_n \;\dashv\; \mathbf{M}}_{\text{Translation}}$$

– Cons: *Hard to understand* the choice of refinement.

- Event-B: Refines *transition systems*.

$$\underbrace{\phi \quad \dashv \quad \mathbf{M}_0}_{\text{Verification}}\!\!\!\!\!\!\overbrace{\phantom{\mathbf{M}_0}\text{ refined by } \mathbf{M}_1 \ldots \text{ refined by } \mathbf{M}}^{\text{Refinement}}$$

– Cons: No support for *liveness properties*.

Execution of Unit-B Models

$$ex.\mathbf{M} \;=\; saf.\mathbf{M} \wedge live.\mathbf{M} \tag{1}$$

$$saf.\mathbf{M} \;=\; init.\mathbf{M} \wedge \Box\, step.\mathbf{M} \tag{2}$$

$$step.\mathbf{M} \;=\; (\exists \mathsf{e}.t \in \mathbf{M} \cdot act.(\mathsf{e}.t)) \;\vee\; \textsc{skip} \tag{3}$$

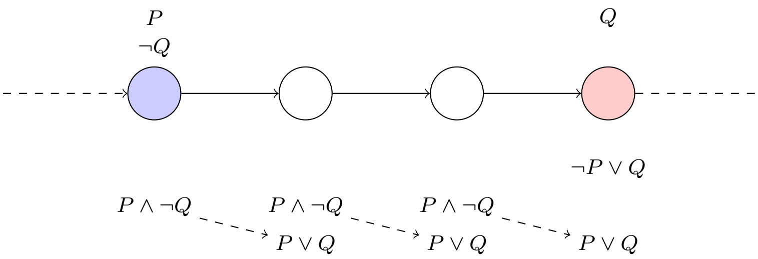
$$live.\mathbf{M} \;=\; \forall \mathsf{e}.t \in \mathbf{M} \cdot sched.(\mathsf{e}.t) \tag{4}$$

$$sched.(\mathsf{e}.t) \;=\; \Box(\Box C \;\wedge\; \Box\Diamond F \;\Rightarrow\; \Diamond(F \wedge act.(\mathsf{e}.t))) \tag{5}$$

The Ensure Rule

Theorem 2 (The ensure-rule)**.** *For all state predicates p and q,*

$$(P \;\textbf{\textit{un}}\; Q) \;\wedge\; ((P \wedge \neg Q) \leadsto (\neg P \vee Q)) \quad\Rightarrow\quad (P \leadsto Q) \tag{ENS}$$



The specification of the controller

ctrl_platform
  any p **where**
    $p \in PLF \wedge p \in \operatorname{ran}.loc \wedge Exit \notin \operatorname{ran}.loc \wedge$
    $\forall q \cdot q \in PLF \Rightarrow sgn.q = RD$
  during
    $p \in PLF \wedge p \in \operatorname{ran}.loc \wedge sgn.p = RD$
  upon
    $Exit \notin \operatorname{ran}(loc) \wedge \forall q \cdot q \in PLF \wedge q \neq p \Rightarrow sgn.q = RD$
  then
    $sgn.p := GR$
  end