Real-Time Systems

Lecture 08: DC Implementables

2013-05-28

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lectures:

- (Un)decidability results for fragments of DC in discrete and continuous time.

This Lecture:

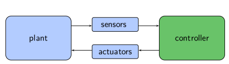
- **Educational Objectives:** Capabilities for following tasks/questions.
  - What does this standard forms mean? Give a satisfying interpretation.
  - What are implementables? What is a control automaton?
  - Please specify (and prove correct) a controller which satisfies this requirement.
- **Content:**
  - DC Standard Forms
  - Control Automata
  - DC Implementables
  - Example

DC Implementables

Requirements vs. Implementations

- **Problem:** in general, a DC requirement doesn't tell **how** to achieve it, how to build a controller/write a program which ensures it.
- What a controller (clearly) can do is:
  - consider inputs now,
  - change (local) state, or
  - wait,
  - set outputs now.

  (But not, e.g., consider future inputs now.)



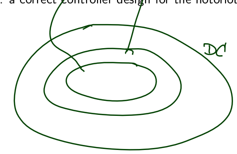
- So, if we have
  - a DC requirement 'Req',
  - a description 'Impl' in DC, which "uses" **just these** operations,

  then
  - proving correctness amounts to proving $\models_0$ Impl $\implies$ Req **(in DC)**
  - and we (more or less) know how to program (the correct) 'Impl' in a PLC language, or in C on a real-time OS, or or or...

Approach: Control Automata and DC Implementables

Plan:

- Introduce **DC Standard Forms**
- Introduce **Control Automata**
- Introduce **DC Implementables** as subset of **DC Standard Forms**
- Example: a correct controller design for the notorious Gas Burner



DC Standard Forms: Followed-by

no $\int$, no ℓ

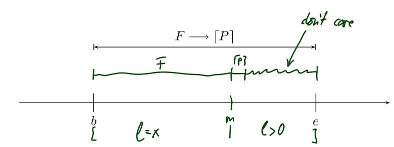
In the following: F is a DC **formula**, P a **state assertion**, θ a **rigid term**.

- **Followed-by:**

$$F \longrightarrow \lceil P \rceil :\iff \neg\Diamond(F\,;\lceil\neg P\rceil) \iff \Box\neg(F\,;\lceil\neg P\rceil)$$

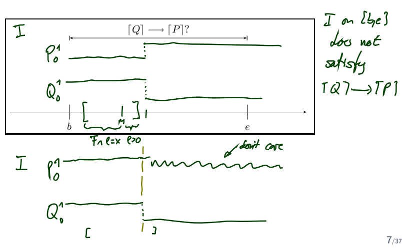
in other symbols

$$\forall x \bullet \Box((F \land \ell = x)\,;\ell > 0 \implies (F \land \ell = x)\,;\lceil P \rceil\,; true)$$



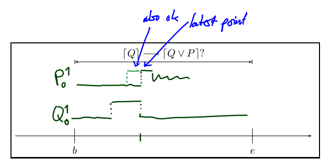
DC Standard Forms: Followed-by Examples

$$\forall x \bullet \Box((F \wedge \ell = x) ; \ell > 0 \implies (F \wedge \ell = x) ; \lceil P \rceil ; true)$$



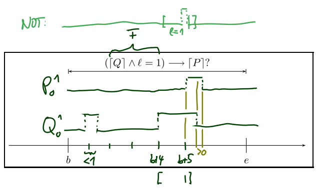
DC Standard Forms: Followed-by Examples

$$\forall x \bullet \Box((F \wedge \ell = x) ; \ell > 0 \implies (F \wedge \ell = x) ; \lceil P \rceil ; true)$$



DC Standard Forms: Followed-by Examples

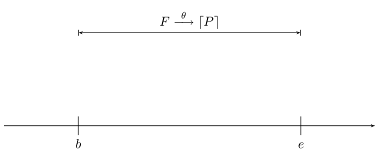
$$\forall x \bullet \Box((F \wedge \ell = x) ; \ell > 0 \implies (F \wedge \ell = x) ; \lceil P \rceil ; true)$$



DC Standard Forms: (Timed) leads-to

- **(Timed) leads-to:** rigid!

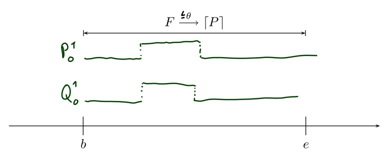
$$F \xrightarrow{\theta} \lceil P \rceil :\iff (F \wedge \ell = \theta) \longrightarrow \lceil P \rceil$$



DC Standard Forms: (Timed) up-to

- **(Timed) up-to:**

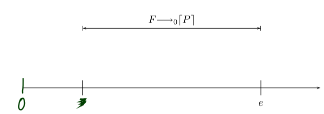
$$F \xrightarrow{\leq\theta} \lceil P \rceil :\iff (F \wedge \ell \leq \theta) \longrightarrow \lceil P \rceil$$



DC Standard Forms: Initialisation

- **Followed-by-initially:**

$$F \longrightarrow_0 \lceil P \rceil :\iff \neg(F ; \lceil \neg P \rceil)$$



- **(Timed) up-to-initially:**

$$F \xrightarrow{\leq\theta}_0 \lceil P \rceil :\iff (F \wedge \ell \leq \theta) \longrightarrow_0 \lceil P \rceil$$

- **Initialisation:**

$$\sqcap \vee (\lceil P \rceil ; true)$$

Control Automata

- Let $X_1, \dots, X_k$ be k state variables ranging over **finite** domains $\mathcal{D}(X_1), \dots, \mathcal{D}(X_k)$.
- With a DC formula 'Impl' ranging over $X_1, \dots, X_k$ we have a **system of k control automata**.
- 'Impl' is typically a conjunction of **DC implementables**.
- A state assertion of the form

$$X_i = d_i, \quad d_i \in \mathcal{D}(X_i),$$

  which constrains the values of X_i, is called **basic phase** of X_i.
- A **phase** of X_i is a Boolean combination of basic phases of X_i.
- **Abbreviations:**
  - Write X_i instead of $X_i = 1$, if X_i is Boolean.
  - Write d_i instead of $X_i = d_i$, if $\mathcal{D}(X_i)$ is disjoint from $\mathcal{D}(X_j)$, $i \neq j$.

Control Automata: Example

Model of Gas Burner controller as a system of four control automata:

- H Boolean, representing **heat request**, (input)
- F Boolean, representing **flame**, (input)
- C with $\mathcal{D}(C) = \{\text{idle}, \text{purge}, \text{ignite}, \text{burn}\}$, representing the (status of the) **controller**, (local)
- G Boolean, representing **gas valve**. (output)

reads

writes

- **Basic phase** of C:

$$C = \text{purge} \qquad \text{(or only: purge)}$$

- **Phase** of C:

$$\text{purge} \vee \text{idle}$$

DC Implementables

- DC Implementables are special patterns of DC Standard Forms (due to A.P. Ravn).
- Within one pattern,
  - $\pi, \pi_1, \dots, \pi_n$, $n \geq 0$, denote **phases** of **the same** state variable X_i,
  - φ denotes a state assertion not depending on X_i.
- θ denotes a **rigid** term.

- **Initialisation**:

$$\lceil\,\rceil \vee \lceil \pi \rceil \,;\, true$$

- **Sequencing**:

$$\lceil \pi \rceil \longrightarrow \lceil \pi \vee \pi_1 \vee \dots \vee \pi_n \rceil$$

- **Progress**:

$$\lceil \pi \rceil \xrightarrow{\theta} \lceil \neg\pi \rceil$$

- **Synchronisation**:

$$\lceil \pi \wedge \varphi \rceil \xrightarrow{\theta} \lceil \neg\pi \rceil$$

DC Implementables Cont'd

- **Bounded Stability**:

$$\underbrace{(\lceil \neg\pi \rceil \,;\, \lceil \pi \wedge \varphi \rceil)}_{F} \xrightarrow{\leq\theta} \lceil \underbrace{\pi \vee \pi_1 \vee \dots \vee \pi_n}_{P} \rceil$$

- **Unbounded Stability**:

$$\lceil \neg\pi \rceil \,;\, \lceil \pi \wedge \varphi \rceil \longrightarrow \lceil \pi \vee \pi_1 \vee \dots \vee \pi_n \rceil$$

- **Bounded initial stability**:

$$\lceil \pi \wedge \varphi \rceil \xrightarrow{\leq\theta}_0 \lceil \pi \vee \pi_1 \vee \dots \vee \pi_n \rceil$$

- **Unbounded initial stability**:

$$\lceil \pi \wedge \varphi \rceil \longrightarrow_0 \lceil \pi \vee \pi_1 \vee \dots \vee \pi_n \rceil$$

Specification by DC Implementables

- Let $X_1, \dots, X_k$ be a system of k control automata.
- Let 'Impl' be a conjunction of **DC implementables**.
- Then 'Impl' **specifies** all interpretations $\mathcal{I}$ of $X_1, \dots, X_k$ and all valuations $\mathcal{V}$ such that

$$\mathcal{I}, \mathcal{V} \models_0 \text{Impl}$$

- Hmm: And what does this have to do with controllers...?

Example: Gas Burner

Recall: Control Automata

Model of Gas Burner controller as a system of four control automata:

- H : Boolean,
  representing **heat request**, (input)
- F : Boolean,
  representing **flame**, (input)
- C with $\mathcal{D}(C) = \{\text{idle}, \text{purge}, \text{ignite}, \text{burn}\}$,
  representing the **controller**, (local)
- G : Boolean,
  representing **gas valve**. (output)

Gas Burner Controller Specification

$\lceil\rceil \vee \lceil \text{idle} \rceil \,;\, true, \quad \lceil\rceil \vee \lceil \neg H \rceil \,;\, true, \quad \lceil\rceil \vee \lceil \neg F \rceil \,;\, true, \quad \lceil\rceil \vee \lceil \neg G \rceil \,;\, true$	(Init-1 - 4)
$\lceil \text{idle} \rceil \longrightarrow \lceil \text{idle} \vee \text{purge} \rceil$	(Seq-1)
$\lceil \text{purge} \rceil \longrightarrow \lceil \text{purge} \vee \text{ignite} \rceil$	(Seq-2)
$\lceil \text{ignite} \rceil \longrightarrow \lceil \text{ignite} \vee \text{burn} \rceil$	(Seq-3)
$\lceil \text{burn} \rceil \longrightarrow \lceil \text{burn} \vee \text{idle} \rceil$	(Seq-4)
$\lceil \text{purge} \rceil \xrightarrow{30+\varepsilon} \lceil \neg\text{purge} \rceil$	(Prog-1)
$\lceil \text{ignite} \rceil \xrightarrow{0.5+\varepsilon} \lceil \neg\text{ignite} \rceil$	(Prog-2)
$\lceil \text{idle} \wedge H \rceil \xrightarrow{\varepsilon} \lceil \neg\text{idle} \rceil$	(Syn-1)
$\lceil \text{burn} \wedge (\neg H \vee \neg F) \rceil \xrightarrow{\varepsilon} \lceil \neg\text{burn} \rceil$	(Syn-2)
$\lceil G \wedge (\text{idle} \vee \text{purge}) \rceil \xrightarrow{\varepsilon} \lceil \neg G \rceil$	(Syn-3)
$\lceil \neg G \wedge (\text{ignite} \vee \text{burn}) \rceil \xrightarrow{\varepsilon} \lceil G \rceil$	(Syn-4)
$\lceil \neg\text{idle} \rceil \,;\, \lceil \text{idle} \wedge \neg H \rceil \longrightarrow \lceil \text{idle} \rceil$	(Stab-1)
$\lceil \text{idle} \wedge \neg H \rceil \longrightarrow_0 \lceil \text{idle} \rceil$	(Stab-1-init)
$\lceil \neg\text{purge} \rceil \,;\, \lceil \text{purge} \rceil \xrightarrow{\leq 30} \lceil \text{purge} \rceil$	(Stab-2)
$\lceil \neg\text{ignite} \rceil \,;\, \lceil \text{ignite} \rceil \xrightarrow{\leq 0.5} \lceil \text{ignite} \rceil$	(Stab-3)
$\lceil \neg\text{burn} \rceil \,;\, \lceil \text{burn} \wedge H \wedge F \rceil \longrightarrow \lceil \text{burn} \rceil$	(Stab-4)
$\lceil F \rceil \,;\, \lceil \neg F \wedge \neg\text{ignite} \rceil \longrightarrow \lceil \neg F \rceil$	(Stab-5)
$\lceil \neg F \wedge \neg\text{ignite} \rceil \longrightarrow_0 \lceil \neg F \rceil$	(Stab-5-init)
$\lceil G \rceil \,;\, \lceil \neg G \wedge (\text{idle} \vee \text{purge}) \rceil \longrightarrow \lceil \neg G \rceil$	(Stab-6)
$\lceil \neg G \wedge (\text{idle} \vee \text{purge}) \rceil \longrightarrow_0 \lceil \neg G \rceil$	(Stab-6-init)
$\lceil \neg G \rceil \,;\, \lceil G \wedge (\text{ignite} \vee \text{burn}) \rceil \longrightarrow \lceil G \rceil$	(Stab-7)

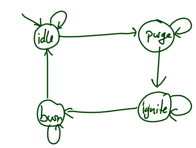
timing

input effect

link output to ctrl. phases

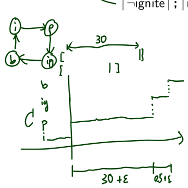
Gas Burner Controller Specification: Untimed

$\lceil\rceil \vee \lceil \text{idle} \rceil \,;\, true$	(Init-1)
$\lceil \text{idle} \rceil \longrightarrow \lceil \text{idle} \vee \text{purge} \rceil$	(Seq-1)
$\lceil \text{purge} \rceil \longrightarrow \lceil \text{purge} \vee \text{ignite} \rceil$	(Seq-2)
$\lceil \text{ignite} \rceil \longrightarrow \lceil \text{ignite} \vee \text{burn} \rceil$	(Seq-3)
$\lceil \text{burn} \rceil \longrightarrow \lceil \text{burn} \vee \text{idle} \rceil$	(Seq-4)



Gas Burner Controller Specification: Timing

$\lceil \text{purge} \rceil \xrightarrow{30+\varepsilon} \lceil \neg\text{purge} \rceil$	(Prog-1)
$\lceil \text{ignite} \rceil \xrightarrow{0.5+\varepsilon} \lceil \neg\text{ignite} \rceil$	(Prog-2)
$\lceil \neg\text{purge} \rceil \,;\, \lceil \text{purge} \rceil \xrightarrow{\leq 30} \lceil \text{purge} \rceil$	(Stab-2)
$\lceil \neg\text{ignite} \rceil \,;\, \lceil \text{ignite} \rceil \xrightarrow{\leq 0.5} \lceil \text{ignite} \rceil$	(Stab-3)



Gas Burner Controller Specification: Outputs

$\lceil \underbrace{G}_{\pi} \wedge \underbrace{(\text{idle} \vee \text{purge})}_{\varphi} \rceil \xrightarrow{\varepsilon} \lceil \neg G \rceil$	(Syn-3)
$\lceil \neg G \wedge (\text{ignite} \vee \text{burn}) \rceil \xrightarrow{\varepsilon} \lceil G \rceil$	(Syn-4)
$\lceil G \rceil \,;\, \lceil \neg G \wedge (\text{idle} \vee \text{purge}) \rceil \longrightarrow \lceil \neg G \rceil$	(Stab-6)
$\lceil \neg G \wedge (\text{idle} \vee \text{purge}) \rceil \longrightarrow_0 \lceil \neg G \rceil$	(Stab-6-init)
$\lceil \neg G \rceil \,;\, \lceil G \wedge (\text{ignite} \vee \text{burn}) \rceil \longrightarrow \lceil G \rceil$	(Stab-7)
$\lceil\rceil \vee \lceil \neg G \rceil ; true$	(Init-4)

π'

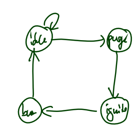
valve close/open after ε in the latest

valve stays closed/open

Gas Burner Controller Specification: Inputs

if H is "on" for less than ε, we need not change phase

$\lceil \text{idle} \wedge H \rceil \xrightarrow{\varepsilon} \lceil \neg\text{idle} \rceil$	(Syn-1)
$\lceil \text{burn} \wedge (\neg H \vee \neg F) \rceil \xrightarrow{\varepsilon} \lceil \neg\text{burn} \rceil$	(Syn-2)
$\lceil \neg\text{idle} \rceil \,;\, \lceil \text{idle} \wedge \neg H \rceil \longrightarrow \lceil \text{idle} \rceil$	(Stab-1)
$\lceil \text{idle} \wedge \neg H \rceil \longrightarrow_0 \lceil \text{idle} \rceil$	(Stab-1-init)
$\lceil \neg\text{burn} \rceil \,;\, \lceil \text{burn} \wedge H \wedge F \rceil \longrightarrow \lceil \text{burn} \rceil$	(Stab-4)



Gas Burner Controller Specification: Assumptions

$$\lceil\rceil \vee \lceil \neg H \rceil \,;\, true \qquad \text{(Init-2)}$$

$$\lceil\rceil \vee \lceil \neg F \rceil \,;\, true \qquad \text{(Init-3)}$$

$$\cancel{\lceil\rceil \vee \lceil \neg G \rceil \,;\, true \qquad \text{(Init-4)}}$$

$$\lceil F \rceil \,;\, \lceil \neg F \wedge \neg\mathsf{ignite} \rceil \longrightarrow \lceil \neg F \rceil \qquad \text{(Stab-5)}$$

$$\lceil \neg F \wedge \neg\mathsf{ignite} \rceil \longrightarrow_0 \lceil \neg F \rceil \qquad \text{(Stab-5-init)}$$

no spontaneous flame

Gas Burner Controller Correctness Proof

$$\text{GB-Ctrl} := \text{Init-1} \wedge \dots \wedge \text{Stab-7} \wedge \varepsilon > 0$$

Recall:

$$\text{Req} :\iff \Box(\ell \geq 60 \implies 20 \cdot \textstyle\int L \leq \ell)$$

and (cf. [Olderog and Dierks, 2008])

$$\models \text{Req-1} \implies \text{Req}$$

for the **simplified**

$$\text{Req-1} := \Box(\ell \leq 30 \implies \textstyle\int L \leq 1).$$

Here we show

$$\models \text{GB-Ctrl} \wedge A(\varepsilon) \implies \text{Req-1}.$$

Lemma 3.15

$$\models \underbrace{\text{GB-Ctrl}}_{[c,d]} \implies \Box \begin{pmatrix} & (\lceil \mathsf{idle} \rceil \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \mathsf{purge} \rceil \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \mathsf{ignite} \rceil \implies & \ell \leq 0.5 + \varepsilon) \\ \wedge & (\lceil \mathsf{burn} \rceil \implies & \int \neg F \leq 2\varepsilon) \end{pmatrix} \quad (*)$$

Proof: Let $\mathcal{I}$ be an interpretation, $\mathcal{V}$ a valuation, and $[c,d]$ an interval with $\mathcal{I}, \mathcal{V}, [c,d] \models$ GB-Ctrl. Let $[b,e] \subseteq [c,d]$.

- Case 1: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \mathsf{idle} \rceil$

$$\lceil G \wedge (\mathsf{idle} \vee \mathsf{purge}) \rceil \xrightarrow{\varepsilon} \lceil \neg G \rceil \qquad \text{(Syn-3)}$$

$$\lceil G \rceil \,;\, \lceil \neg G \wedge (\mathsf{idle} \vee \mathsf{purge}) \rceil \longrightarrow \lceil \neg G \rceil \qquad \text{(Stab-6)}$$

conclude

$$\mathcal{I}, \mathcal{V}, [b,e] \models \Box(\lceil G \rceil \implies \ell \leq \varepsilon) \wedge \neg\Diamond(\underbrace{\lceil G \rceil \,;\, \lceil \neg G \rceil \,;\, \lceil G \rceil}_{\text{gas valve doesn't open up again in idle phase}})$$

$(*)$

- Case 2: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \mathsf{purge} \rceil$ Analogously to case 1.

Lemma 3.15 Cont'd

$$\begin{array}{|rl|} \hline (\lceil \mathsf{idle} \rceil \implies & \int G \leq \varepsilon) \\ (\lceil \mathsf{purge} \rceil \implies & \int G \leq \varepsilon) \\ (\lceil \mathsf{ignite} \rceil \implies & \ell \leq 0.5 + \varepsilon) \\ (\lceil \mathsf{burn} \rceil \implies & \int \neg F \leq 2\varepsilon) \\ \hline \end{array} \quad (**)$$

- Case 3: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \mathsf{ignite} \rceil$

$$\lceil \mathsf{ignite} \rceil \xrightarrow{0.5+\varepsilon} \lceil \neg\mathsf{ignite} \rceil \qquad \text{(Prog-2)}$$

$$\mathcal{I}, \mathcal{V}, [b,e] \models \ell \leq 0.5 + \varepsilon$$

- Case 4: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \mathsf{burn} \rceil$

$$\lceil \mathsf{burn} \wedge (\neg H \vee \neg F) \rceil \xrightarrow{\varepsilon} \lceil \neg\mathsf{burn} \rceil \qquad \text{(Syn-2)}$$

$$\lceil F \rceil \,;\, \lceil \neg F \wedge \neg\mathsf{ignite} \rceil \longrightarrow \lceil \neg F \rceil \qquad \text{(Stab-5)}$$

$$\mathcal{I}, \mathcal{V}, [b,e] \models \Box(\lceil \neg F \rceil \implies \ell \leq \varepsilon) \wedge \neg\Diamond(\lceil F \rceil \,;\, \lceil \neg F \rceil \,;\, \lceil F \rceil)$$

$\lceil F \rceil$
$\lceil F \rceil ; \lceil \neg F \rceil$
$\lceil \neg F \rceil ; \lceil F \rceil ; \lceil \neg F \rceil$
$\lceil \neg F \rceil ; \lceil F \rceil$

$(**)$

Lemma 3.16

$$\models \exists\, \varepsilon \bullet \text{GB-Ctrl} \implies \underbrace{\Box(\ell \leq 30 \implies \textstyle\int L \leq 1)}_{\text{Req-1}}$$

Proof Sketch

Choose $\mathcal{I}, \mathcal{V}, [b,e]$ s.t. $\mathcal{I}, \mathcal{V}, [b,e] \models$ GB-Ctrl $\wedge\ \ell \leq 30$

Distinguish 5 cases:

$$\begin{array}{rll} \mathcal{I}, \mathcal{V}, [b,e] \models & \lceil\rceil & (0) \\ & \vee (\lceil \mathsf{idle} \rceil ; true \wedge \ell \leq 30) & (1) \\ & \vee (\lceil \mathsf{purge} \rceil ; true \wedge \ell \leq 30) & (2) \\ & \vee (\lceil \mathsf{ignite} \rceil ; true \wedge \ell \leq 30) & (3) \\ & \vee (\lceil \mathsf{burn} \rceil ; true \wedge \ell \leq 30) & (4) \end{array}$$

Lemma 3.16 Cont'd

- Case 0: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil\rceil$ ✓

- Case 1: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \mathsf{idle} \rceil \,;\, true \wedge \ell \leq 30$

$$\lceil \mathsf{idle} \rceil \longrightarrow \lceil \mathsf{idle} \vee \mathsf{purge} \rceil \qquad \text{(Seq-1)}$$

$$\lceil \neg\mathsf{purge} \rceil \,;\, \lceil \mathsf{purge} \rceil \xrightarrow{\leq 30} \lceil \mathsf{purge} \rceil \qquad \text{(Stab-2)}$$

$\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \mathsf{idle} \rceil \vee \lceil \mathsf{idle} \rceil ; \lceil \mathsf{purge} \rceil$

3.15 $\mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq \varepsilon \vee \int L \leq \varepsilon ; \int L \leq \varepsilon$

$\mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq 2\varepsilon$

Thus $\boxed{\varepsilon \leq 0.5}$ is sufficient for Req-1 in this case.

Lemma 3.16 Cont'd

- Case 2: $\mathcal{I}, \mathcal{V}, [b, e] \models \lceil \text{burn} \rceil \,;\, true \wedge \ell \leq 30$

$$\lceil \text{burn} \rceil \longrightarrow \lceil \text{burn} \vee \text{idle} \rceil \qquad \text{(Seq-4)}$$

$\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models (\lceil \text{burn} \rceil \vee \lceil \text{burn} \rceil ; \lceil \text{idle} \rceil ; true) \wedge \ell \leq 30$

3.15, (1) $\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models (\int L \leq 2\varepsilon \vee \int L \leq 2\varepsilon ; \int L \leq 2\varepsilon) \wedge \ell \leq 30$

$\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq 4\varepsilon$

Thus $\boxed{\varepsilon \leq 0.25}$ sufficient for Req-1.

Lemma 3.16 Cont'd

- Case 3: $\mathcal{I}, \mathcal{V}, [b, e] \models \lceil \text{ignite} \rceil \,;\, true \wedge \ell \leq 30$

$$\lceil \text{ignite} \rceil \longrightarrow \lceil \text{ignite} \vee \text{burn} \rceil \qquad \text{(Seq-3)}$$

$\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models (\lceil \text{ignite} \rceil \vee \lceil \text{ignite} \rceil ; \lceil \text{burn} \rceil ; true) \wedge \ell \leq 30$

3.5, (2) $\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq 0.5 + \varepsilon \vee (\int L \leq 0.5 + \varepsilon ; \int L \leq 4\varepsilon) \wedge \ell \leq 30$

$\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq 0.5 + 5\varepsilon$

So $\boxed{\varepsilon \leq 0.1}$ is sufficient for Req-1.

Lemma 3.16 Cont'd

- Case 4: $\mathcal{I}, \mathcal{V}, [b, e] \models \lceil \text{purge} \rceil \,;\, true \wedge \ell \leq 30$

$$\lceil \text{purge} \rceil \longrightarrow \lceil \text{purge} \vee \text{ignite} \rceil \qquad \text{(Seq-2)}$$

$\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models (\lceil \text{purge} \rceil \vee \lceil \text{purge} \rceil ; \lceil \text{ignite} \rceil ; true) \wedge \ell \leq 30$

3.15, (3) $\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq \varepsilon \vee (\int L \leq \varepsilon ; \int L \leq 0.5 + \varepsilon)$

$\hookrightarrow \mathcal{I}, \mathcal{V}, [b,e] \models \int L \leq 0.5 + 6\varepsilon$

Thus $\boxed{\varepsilon \leq \frac{1}{12}}$ is sufficient for Req-1 in this case.

Correctness Result

Theorem 3.17.

$$\models \left(\text{GB-Ctrl} \wedge \varepsilon \leq \frac{1}{12} \right) \implies \text{Req}$$

purge | ignite | burn | idle | purge

$\ell \geq 30$ | $\ell \geq 0.5$ | | | $\ell \geq 30$

$\int L \leq \varepsilon$ | $\int L \leq 0.5 + \varepsilon$ | $\int L \leq 2\varepsilon$ | $\int L \leq \varepsilon$ | $\int L \leq \varepsilon$

$\hookrightarrow \int L \leq 0.5 + 6\varepsilon$

$\hookrightarrow A(\varepsilon) := \varepsilon \leq \frac{1}{12}$

Discussion

- We used only

'Seq-1', 'Seq-2', 'Seq-3', 'Seq-4',
'Prog-2', 'Syn-2', 'Syn-3',
'Stab-2', 'Stab-5', 'Stab-6'.

What about

$$\text{Prog-1} = \lceil \text{purge} \rceil \xrightarrow{30+\varepsilon} \lceil \neg \text{purge} \rceil$$

for instance?

Nope, there is the requirement (not noted down) that the system does something finally, e.g. get the heating going on request.

References

References

[Olderog and Dierks, 2008] Olderog, E.-R. and Dierks, H. (2008). *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press.