Real-Time Systems

Lecture 07: DC Implementables

2014-06-03

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lectures:

- Semantical Correctness Proof

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

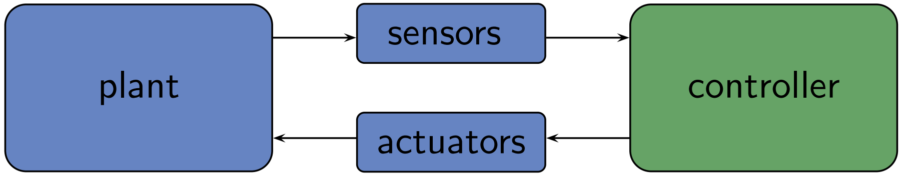
DC Implementables

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.

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.

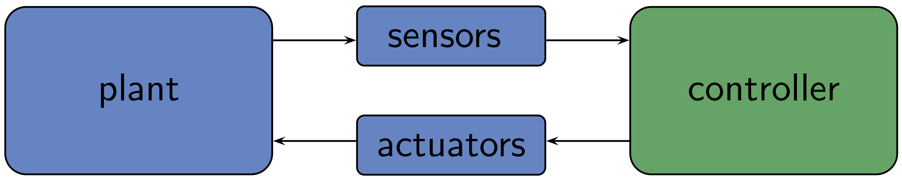


(But not, e.g., consider future inputs now.)

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.



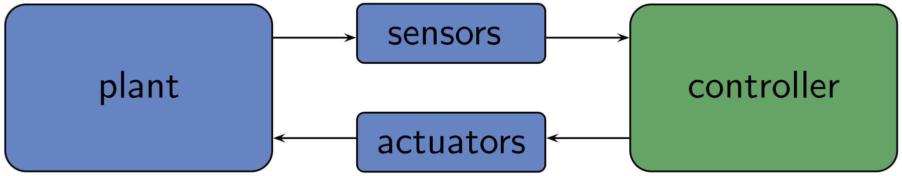
  (But not, e.g., consider future inputs now.)

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

  then

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.

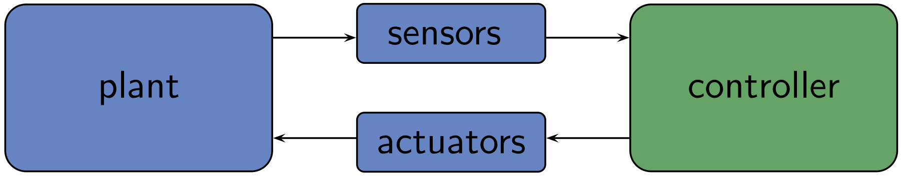
  

  (But not, e.g., consider future inputs now.)
- So, if we have
  - a DC requirement **'**Req**'**,
  - a description **'**Impl**'** in DC,
    which "uses" **just these** operations,

  then
  - proving correctness amounts to proving $\models_0$ Impl $\implies$ Req (**in DC**)

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 Impl'bles

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

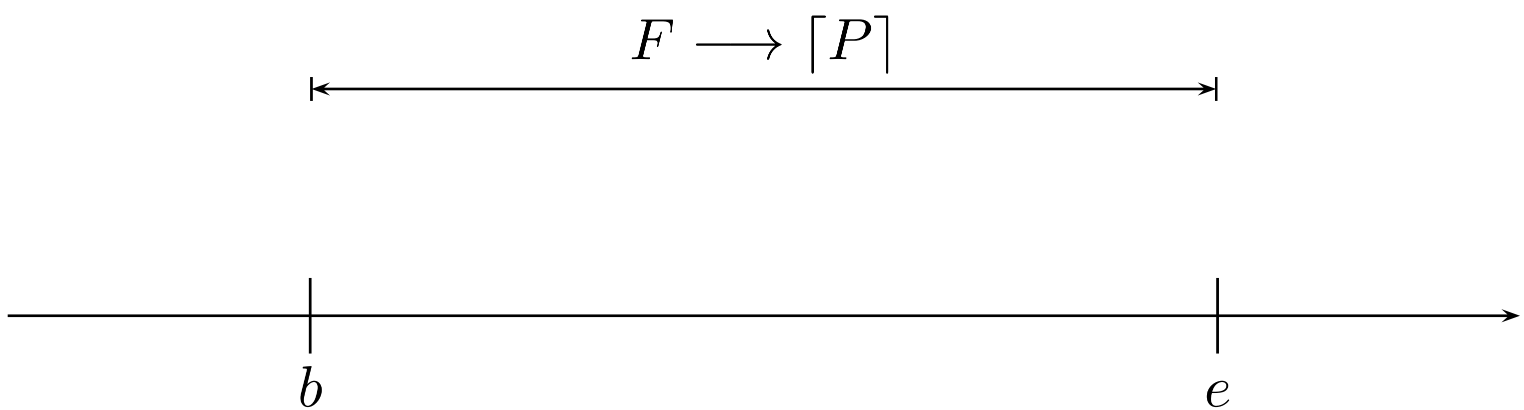
In the following: F is a DC **formula**, P a **state assertion**, θ a **rigid term**.

- **Followed-by:**

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

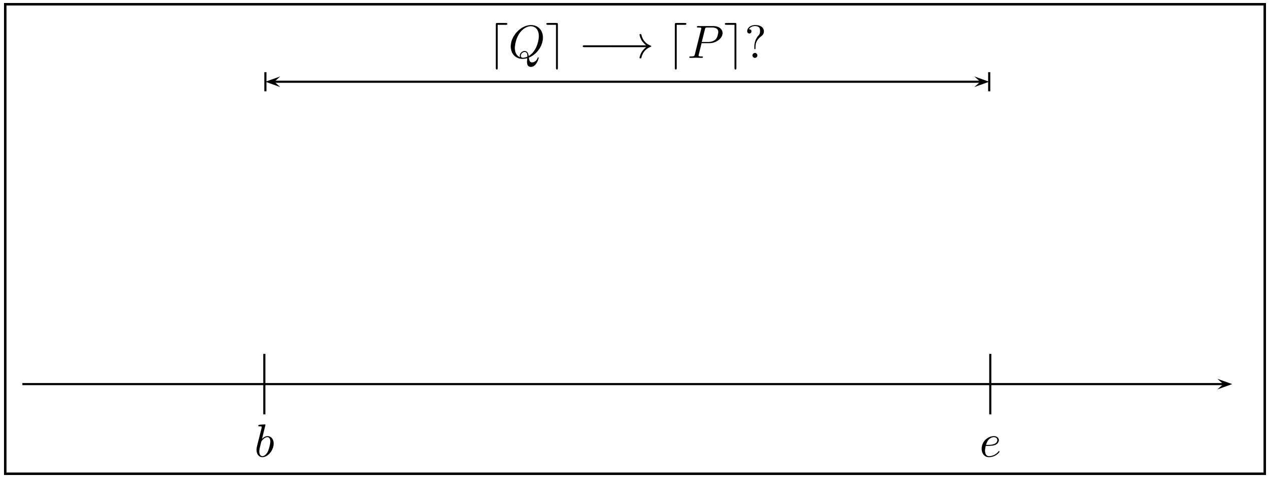
  in other symbols

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



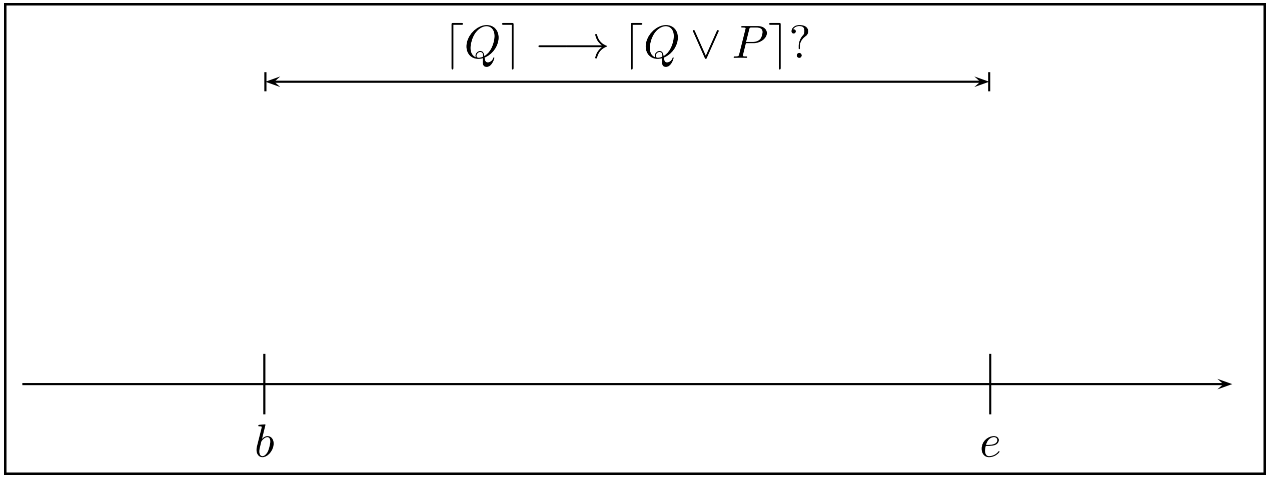
DC Standard Forms: Followed-by Examples

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



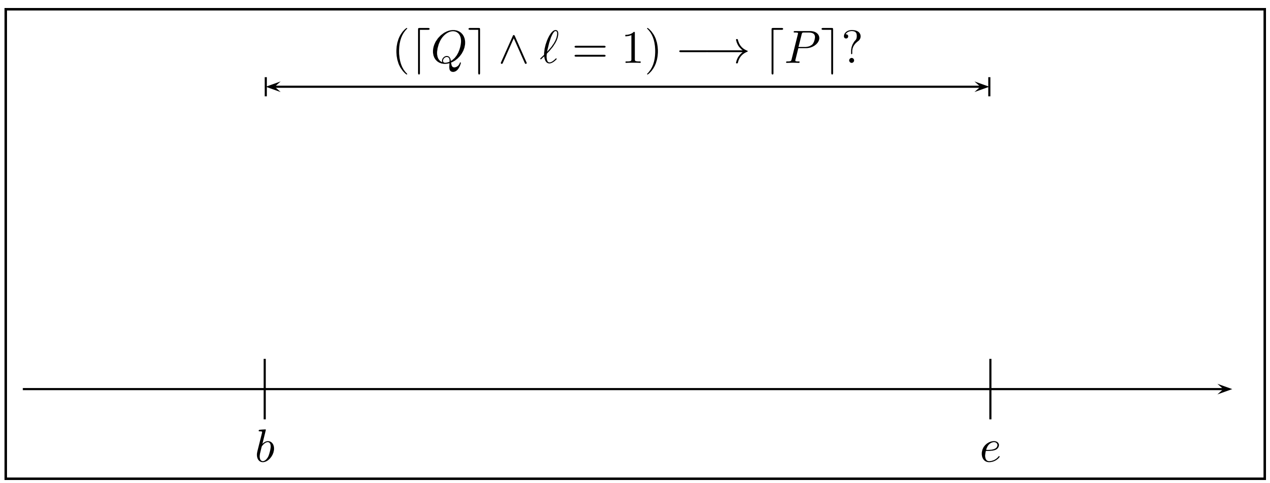
DC Standard Forms: Followed-by Examples

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



DC Standard Forms: Followed-by Examples

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

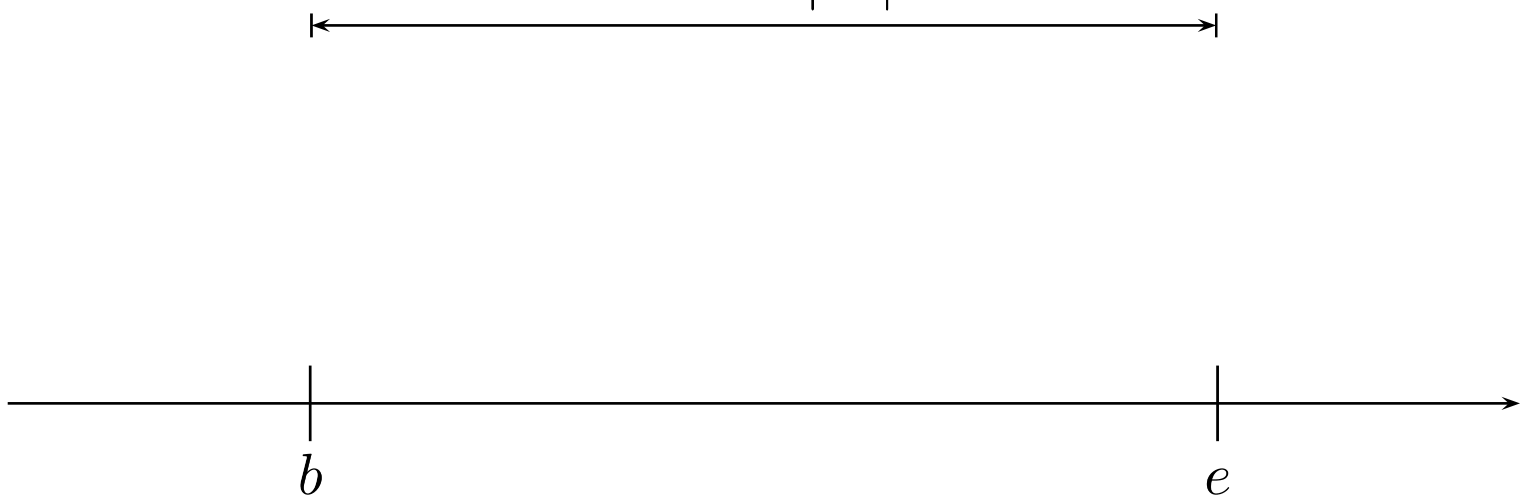


DC Standard Forms: (Timed) leads-to

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

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

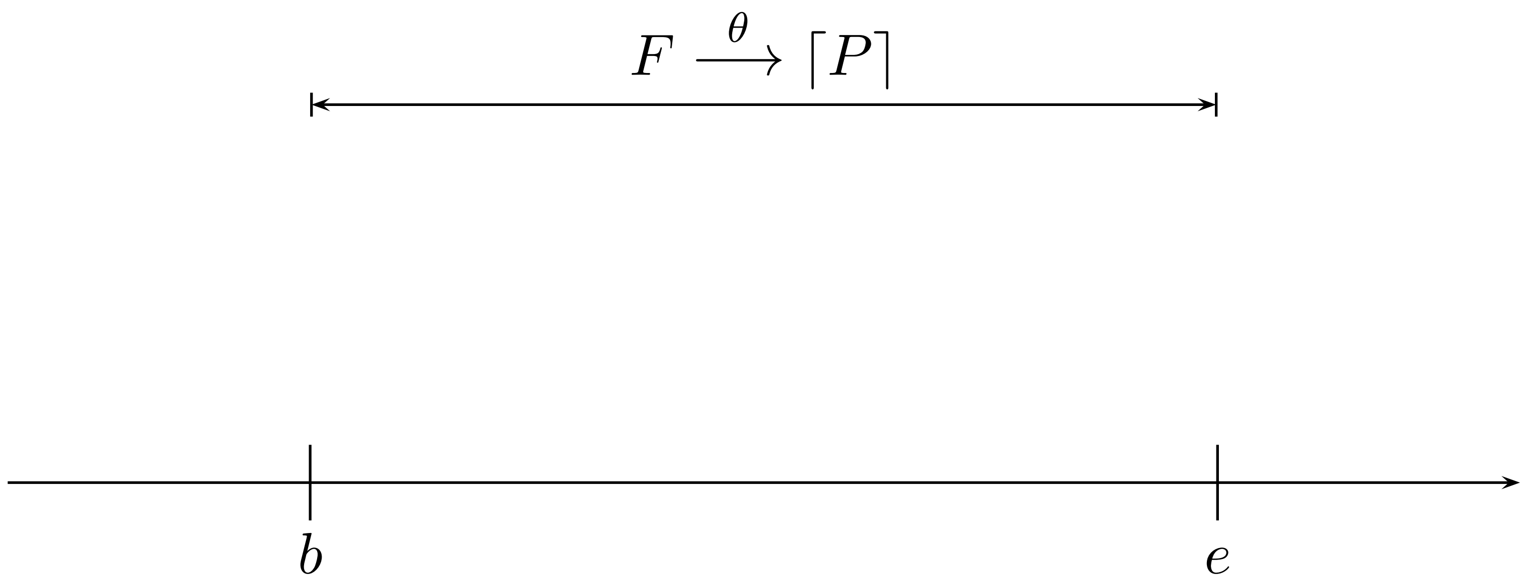
$$F \xrightarrow{\theta} \lceil P \rceil$$



DC Standard Forms: (Timed) up-to

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

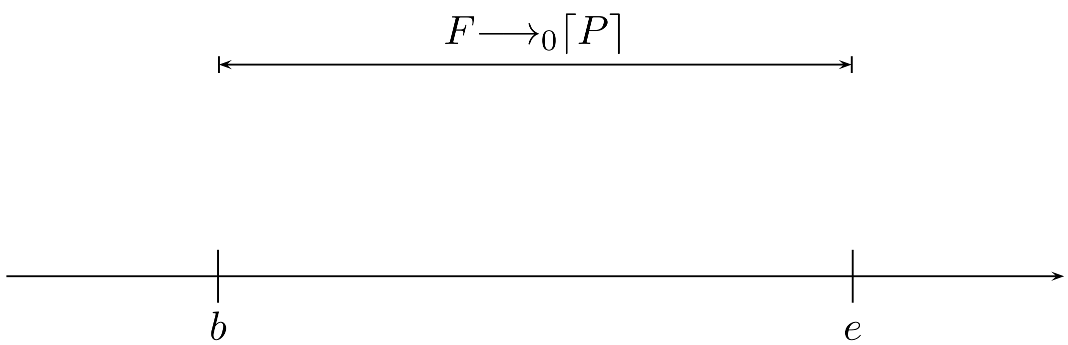
$$F \xrightarrow{\leq\theta} \lceil P \rceil :\iff (F \wedge \ell \leq \theta) \longrightarrow \lceil P \rceil$$



DC Standard Forms: Initialisation

- **Followed-by-initially:**

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



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

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

- **Initialisation:**

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

Control Automata

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

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)

- **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**:

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

- **Progress**:

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

DC Implementables Cont'd

- **Bounded Stability**:

$$\lceil \neg \pi \rceil \,;\, \lceil \pi \wedge \varphi \rceil \xrightarrow{\le\theta} \lceil \pi \vee \pi_1 \vee \cdots \vee \pi_n \rceil$$

- **Unbounded Stability**:

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

- **Bounded initial stability**:

$$\lceil \pi \wedge \varphi \rceil \xrightarrow{\le\theta}_0 \lceil \pi \vee \pi_1 \vee \cdots \vee \pi_n \rceil$$

- **Unbounded initial stability**:

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

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

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

- 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

$$\lceil\rceil \vee \lceil \mathsf{idle} \rceil \,;\, \mathit{true}, \quad \lceil\rceil \vee \lceil \neg H \rceil \,;\, \mathit{true}, \quad \lceil\rceil \vee \lceil \neg F \rceil \,;\, \mathit{true}, \quad \lceil\rceil \vee \lceil \neg G \rceil \,;\, \mathit{true} \qquad \text{(Init-1 - 4)}$$

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

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

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

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

$$\lceil \mathsf{purge} \rceil \xrightarrow{30+\varepsilon} \lceil \neg \mathsf{purge} \rceil \qquad \text{(Prog-1)}$$

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

$$\lceil \mathsf{idle} \wedge H \rceil \xrightarrow{\varepsilon} \lceil \neg \mathsf{idle} \rceil \qquad \text{(Syn-1)}$$

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

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

$$\lceil \neg G \wedge (\mathsf{ignite} \vee \mathsf{burn}) \rceil \xrightarrow{\varepsilon} \lceil G \rceil \qquad \text{(Syn-4)}$$

$$\lceil \neg \mathsf{idle} \rceil \,;\, \lceil \mathsf{idle} \wedge \neg H \rceil \longrightarrow \lceil \mathsf{idle} \rceil \qquad \text{(Stab-1)}$$

$$\lceil \mathsf{idle} \wedge \neg H \rceil \longrightarrow_0 \lceil \mathsf{idle} \rceil \qquad \text{(Stab-1-init)}$$

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

$$\lceil \neg \mathsf{ignite} \rceil \,;\, \lceil \mathsf{ignite} \rceil \xrightarrow{\leq 0.5} \lceil \mathsf{ignite} \rceil \qquad \text{(Stab-3)}$$

$$\lceil \neg \mathsf{burn} \rceil \,;\, \lceil \mathsf{burn} \wedge H \wedge F \rceil \longrightarrow \lceil \mathsf{burn} \rceil \qquad \text{(Stab-4)}$$

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

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

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

$$\lceil \neg G \wedge (\mathsf{idle} \vee \mathsf{purge}) \rceil \longrightarrow_0 \lceil \neg G \rceil \qquad \text{(Stab-6-init)}$$

$$\lceil \neg G \rceil \,;\, \lceil G \wedge (\mathsf{ignite} \vee \mathsf{burn}) \rceil \longrightarrow \lceil G \rceil \qquad \text{(Stab-7)}$$

Gas Burner Controller Specification: Untimed

$$\lceil\rceil \vee \lceil \mathsf{idle} \rceil \mathbin{;} \mathit{true} \tag{Init-1}$$

$$\lceil \mathsf{idle} \rceil \longrightarrow \lceil \mathsf{idle} \vee \mathsf{purge} \rceil \tag{Seq-1}$$

$$\lceil \mathsf{purge} \rceil \longrightarrow \lceil \mathsf{purge} \vee \mathsf{ignite} \rceil \tag{Seq-2}$$

$$\lceil \mathsf{ignite} \rceil \longrightarrow \lceil \mathsf{ignite} \vee \mathsf{burn} \rceil \tag{Seq-3}$$

$$\lceil \mathsf{burn} \rceil \longrightarrow \lceil \mathsf{burn} \vee \mathsf{idle} \rceil \tag{Seq-4}$$

Gas Burner Controller Specification: Timing

$$\lceil \mathsf{purge} \rceil \xrightarrow{30+\varepsilon} \lceil \neg \mathsf{purge} \rceil \qquad \text{(Prog-1)}$$

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

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

$$\lceil \neg \mathsf{ignite} \rceil \,;\, \lceil \mathsf{ignite} \rceil \xrightarrow{\leq 0.5} \lceil \mathsf{ignite} \rceil \qquad \text{(Stab-3)}$$

Gas Burner Controller Specification: Outputs

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

$$\lceil \neg G \wedge (\mathsf{ignite} \vee \mathsf{burn}) \rceil \xrightarrow{\varepsilon} \lceil G \rceil \tag{Syn-4}$$

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

$$\lceil \neg G \wedge (\mathsf{idle} \vee \mathsf{purge}) \rceil \longrightarrow_0 \lceil \neg G \rceil \tag{Stab-6-init}$$

$$\lceil \neg G \rceil \,;\, \lceil G \wedge (\mathsf{ignite} \vee \mathsf{burn}) \rceil \longrightarrow \lceil G \rceil \tag{Stab-7}$$

Gas Burner Controller Specification: Inputs

$$\lceil \mathsf{idle} \wedge H \rceil \xrightarrow{\varepsilon} \lceil \neg \mathsf{idle} \rceil \qquad \text{(Syn-1)}$$

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

$$\lceil \neg \mathsf{idle} \rceil \,;\, \lceil \mathsf{idle} \wedge \neg H \rceil \longrightarrow \lceil \mathsf{idle} \rceil \qquad \text{(Stab-1)}$$

$$\lceil \mathsf{idle} \wedge \neg H \rceil \longrightarrow_0 \lceil \mathsf{idle} \rceil \qquad \text{(Stab-1-init)}$$

$$\lceil \neg \mathsf{burn} \rceil \,;\, \lceil \mathsf{burn} \wedge H \wedge F \rceil \longrightarrow \lceil \mathsf{burn} \rceil \qquad \text{(Stab-4)}$$

Gas Burner Controller Specification: Assumptions

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

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

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

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

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

Gas Burner Controller Correctness Proof

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

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

$$\models \mathsf{GB\text{-}Ctrl} \implies \Box \left(\begin{array}{rrcl} & (\lceil \mathsf{idle} \rceil & \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \mathsf{purge} \rceil & \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \mathsf{ignite} \rceil & \implies & \ell \leq 0.5 + \varepsilon) \\ \wedge & (\lceil \mathsf{burn} \rceil & \implies & \int \neg F \leq 2\varepsilon) \end{array} \right)$$

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 $[b, e] \subseteq [c, d]$.

Lemma 3.15

$$\models \text{GB-Ctrl} \implies \Box \begin{pmatrix} & (\lceil \text{idle} \rceil \implies \int G \leq \varepsilon) \\ \wedge & (\lceil \text{purge} \rceil \implies \int G \leq \varepsilon) \\ \wedge & (\lceil \text{ignite} \rceil \implies \ell \leq 0.5 + \varepsilon) \\ \wedge & (\lceil \text{burn} \rceil \implies \int \neg F \leq 2\varepsilon) \end{pmatrix}$$

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: $\mathcal{I}, \mathcal{V}, [b, e] \models \lceil \text{idle} \rceil$

– 07 – 2014-06-03 – Sexa –

Lemma 3.15

$$\models \textsf{GB-Ctrl} \implies \Box \begin{pmatrix} & (\lceil \textsf{idle} \rceil \implies \int G \leq \varepsilon) \\ \wedge & (\lceil \textsf{purge} \rceil \implies \int G \leq \varepsilon) \\ \wedge & (\lceil \textsf{ignite} \rceil \implies \ell \leq 0.5 + \varepsilon) \\ \wedge & (\lceil \textsf{burn} \rceil \implies \int \neg F \leq 2\varepsilon) \end{pmatrix}$$

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: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \textsf{idle} \rceil$

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

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

Lemma 3.15

$$\models \text{GB-Ctrl} \implies \Box \left(\begin{array}{rrcl} & (\lceil \text{idle} \rceil & \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \text{purge} \rceil & \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \text{ignite} \rceil & \implies & \ell \leq 0.5 + \varepsilon) \\ \wedge & (\lceil \text{burn} \rceil & \implies & \int \neg F \leq 2\varepsilon) \end{array} \right)$$

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: $\mathcal{I}, \mathcal{V}, [b,e] \models \lceil \text{idle} \rceil$

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

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

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

Lemma 3.15

$$\models \text{GB-Ctrl} \implies \Box \begin{pmatrix} & (\lceil \mathsf{idle} \rceil \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \mathsf{purge} \rceil \implies & \int G \leq \varepsilon) \\ \wedge & (\lceil \mathsf{ignite} \rceil \implies & \ell \leq 0.5 + \varepsilon) \\ \wedge & (\lceil \mathsf{burn} \rceil \implies & \int \neg F \leq 2\varepsilon) \end{pmatrix}$$

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

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

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

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

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

Lemma 3.15 Cont'd

$$
\begin{aligned}
(\lceil \mathsf{idle} \rceil &\implies \textstyle\int G \leq \varepsilon) \\
(\lceil \mathsf{purge} \rceil &\implies \textstyle\int G \leq \varepsilon) \\
(\lceil \mathsf{ignite} \rceil &\implies \ell \leq 0.5 + \varepsilon) \\
(\lceil \mathsf{burn} \rceil &\implies \textstyle\int \neg F \leq 2\varepsilon)
\end{aligned}
$$

Lemma 3.15 Cont'd

$$
\begin{aligned}
(\lceil \mathsf{idle} \rceil &\implies \textstyle\int G \leq \varepsilon) \\
(\lceil \mathsf{purge} \rceil &\implies \textstyle\int G \leq \varepsilon) \\
(\lceil \mathsf{ignite} \rceil &\implies \ell \leq 0.5 + \varepsilon) \\
(\lceil \mathsf{burn} \rceil &\implies \textstyle\int \neg F \leq 2\varepsilon)
\end{aligned}
$$

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

Lemma 3.15 Cont'd

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

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

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

Lemma 3.15 Cont'd

$$(\lceil \mathsf{idle} \rceil \implies \textstyle\int G \leq \varepsilon)$$
$$(\lceil \mathsf{purge} \rceil \implies \textstyle\int G \leq \varepsilon)$$
$$(\lceil \mathsf{ignite} \rceil \implies \ell \leq 0.5 + \varepsilon)$$
$$(\lceil \mathsf{burn} \rceil \implies \textstyle\int \neg F \leq 2\varepsilon)$$

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

Lemma 3.15 Cont'd

$$
\begin{array}{rcr}
(\lceil \mathsf{idle} \rceil & \Longrightarrow & \int G \leq \varepsilon) \\
(\lceil \mathsf{purge} \rceil & \Longrightarrow & \int G \leq \varepsilon) \\
(\lceil \mathsf{ignite} \rceil & \Longrightarrow & \ell \leq 0.5 + \varepsilon) \\
(\lceil \mathsf{burn} \rceil & \Longrightarrow & \int \neg F \leq 2\varepsilon)
\end{array}
$$

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

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

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

Lemma 3.15 Cont'd

$$\begin{aligned} (\lceil \mathsf{idle} \rceil &\implies \textstyle\int G \leq \varepsilon) \\ (\lceil \mathsf{purge} \rceil &\implies \textstyle\int G \leq \varepsilon) \\ (\lceil \mathsf{ignite} \rceil &\implies \ell \leq 0.5 + \varepsilon) \\ (\lceil \mathsf{burn} \rceil &\implies \textstyle\int \neg F \leq 2\varepsilon) \end{aligned}$$

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

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

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

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

Lemma 3.16

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

Lemma 3.16 Cont'd

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

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

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

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

Lemma 3.16 Cont'd

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

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

Lemma 3.16 Cont'd

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

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

Lemma 3.16 Cont'd

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

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

Correctness Result

Theorem 3.17.

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

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?

References

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