

# *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*

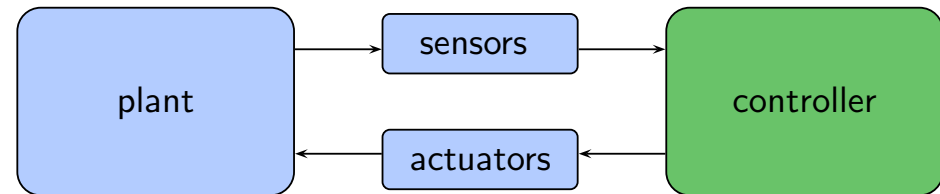
# Requirements vs. Implementations

- **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

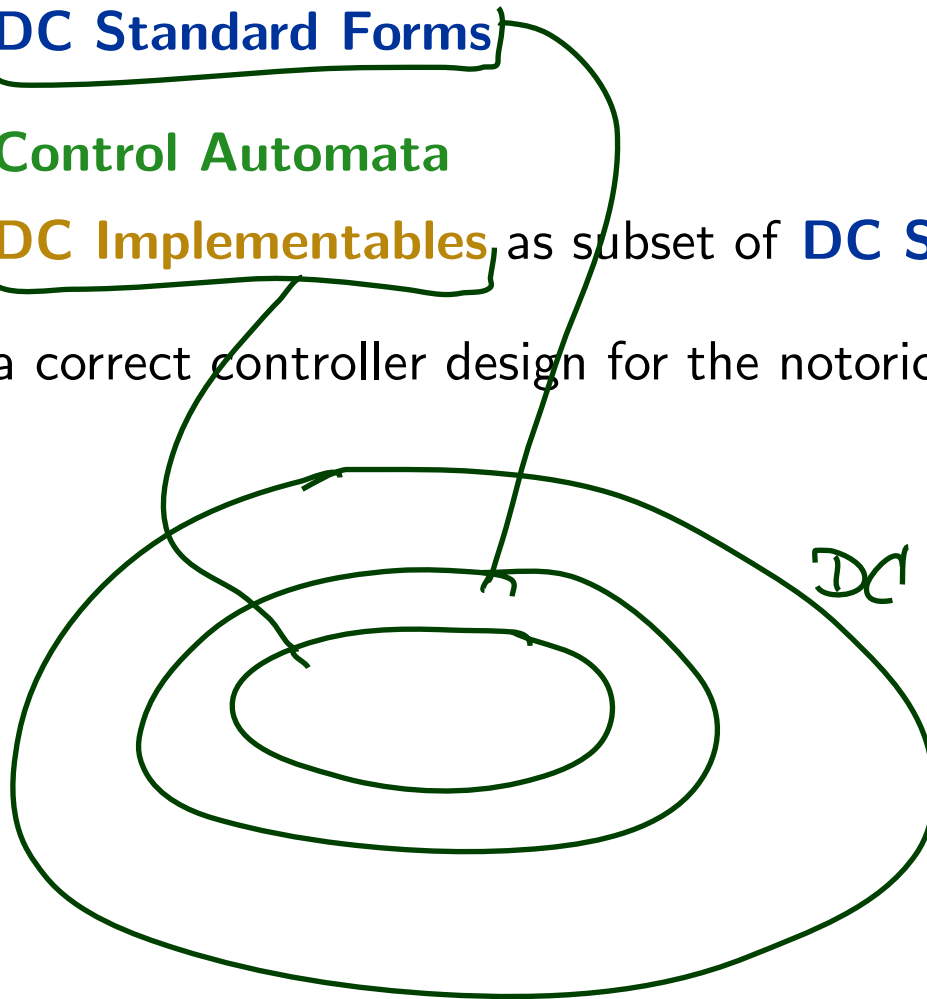
- proving correctness amounts to proving  $\models_0 \text{Impl} \implies \text{Req}$  (in DC)
- and we (more or less) know how to program (the correct) '**Impl**' in a PLC language, or in C on a real-time OS, or or or...

# 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  $\ell$ ,  
no  $\ell$

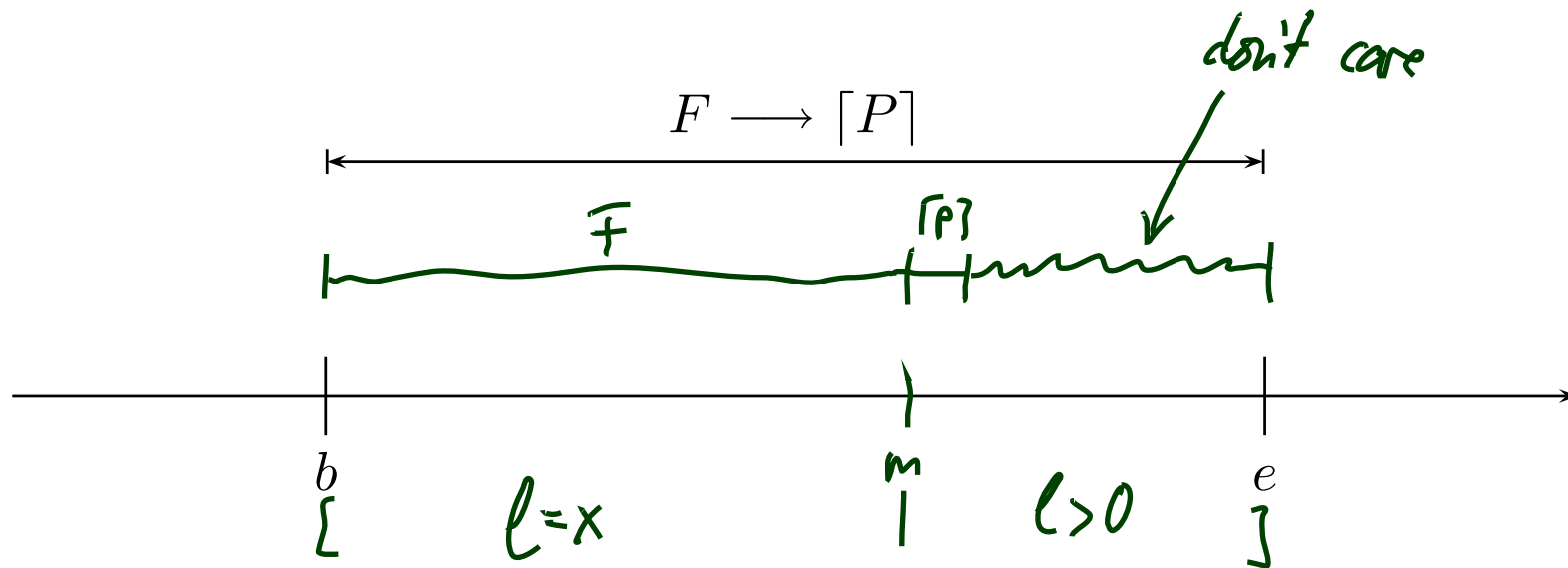
In the following:  $F$  is a DC **formula**,  $P$  a **state assertion**,  $\theta$  a **rigid term**.

- Followed-by:**

$$F \longrightarrow [P] :\iff \neg\Diamond(F ; [\neg P]) \iff \Box\neg(F ; [\neg P])$$

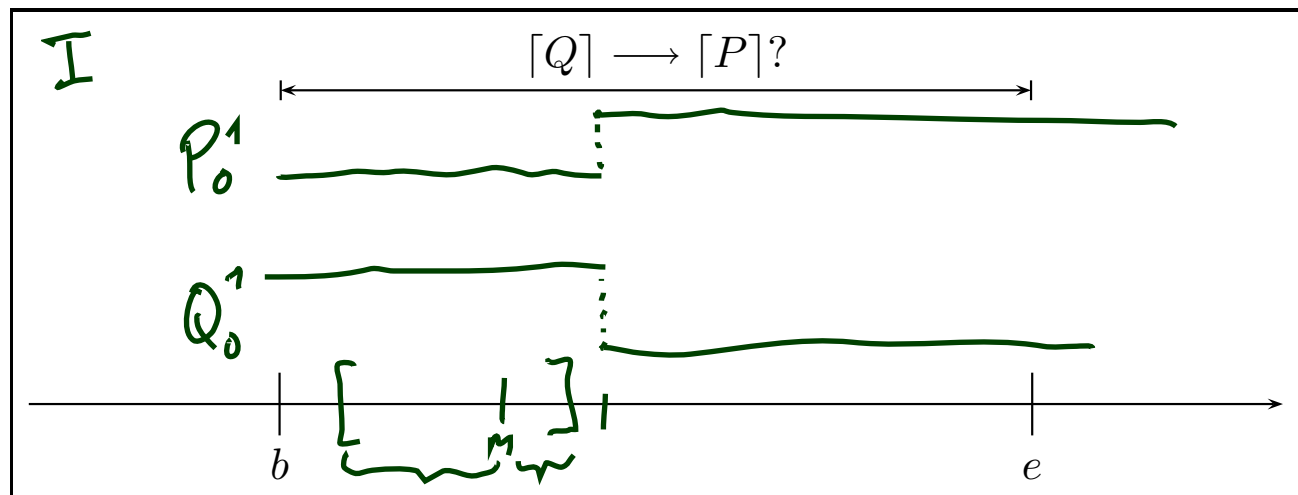
in other symbols

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

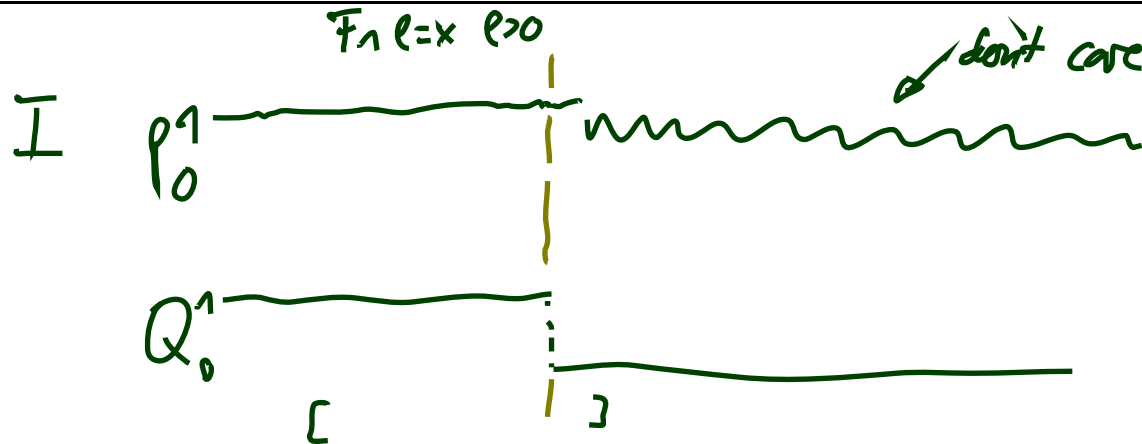


# DC Standard Forms: Followed-by Examples

$$\forall x \bullet \square((F \wedge l = x); l > 0 \implies (F \wedge l = x); [P]; true)$$

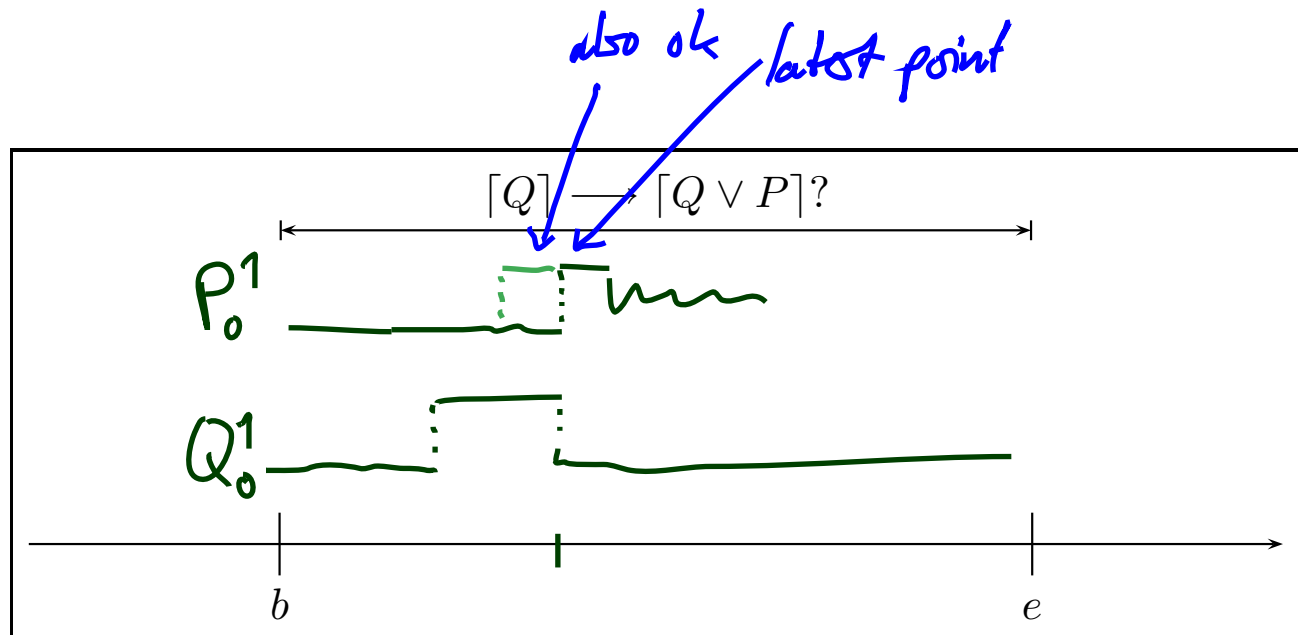


I on  $\{b, e\}$   
 does not  
 satisfy  
 $[Q] \rightarrow [P]$



# DC Standard Forms: Followed-by Examples

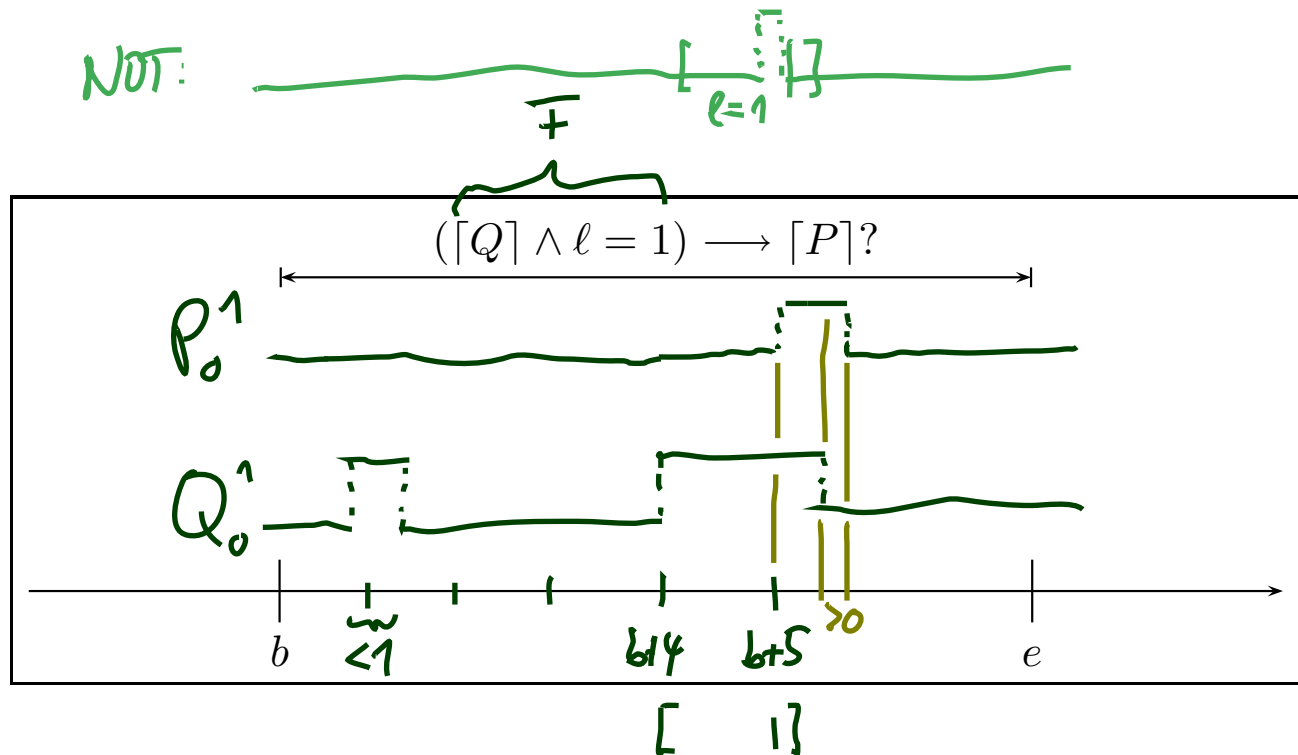
$$\forall x \bullet \square((F \wedge l = x); l > 0 \implies (F \wedge l = x); [P]; true)$$





# DC Standard Forms: Followed-by Examples

$$\forall x \bullet \square((F \wedge l = x); l > 0 \implies (F \wedge l = x); [P]; true)$$

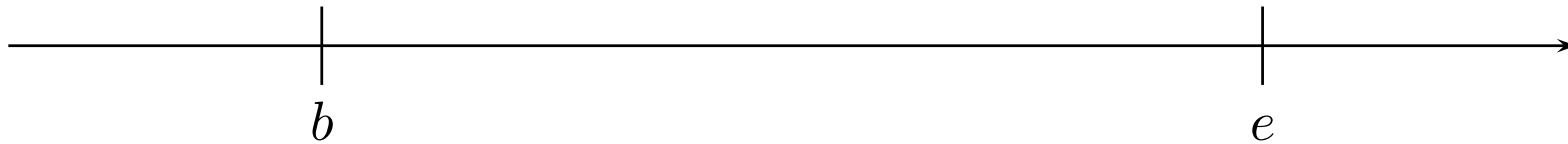
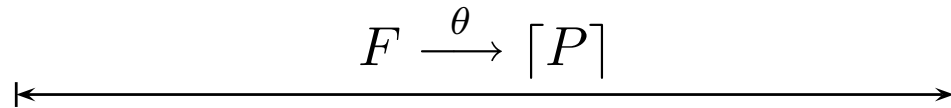


# DC Standard Forms: (Timed) leads-to

- (Timed) leads-to:

*rigid!*

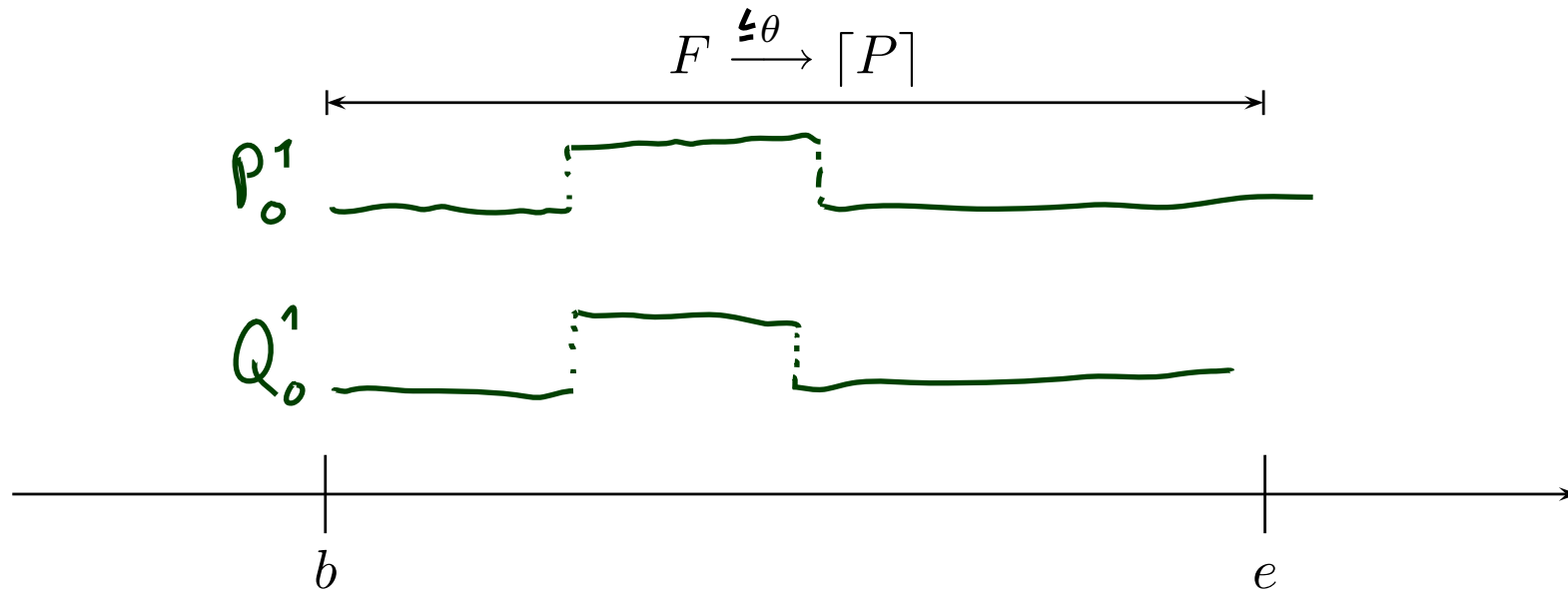
$$F \xrightarrow{\theta} [P] :\iff (F \wedge \ell = \theta) \longrightarrow [P]$$



# DC Standard Forms: (Timed) up-to

- (Timed) up-to:

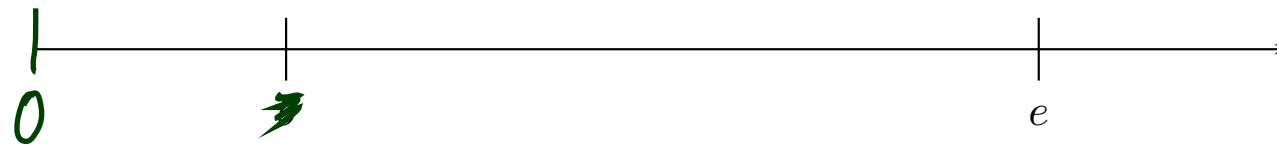
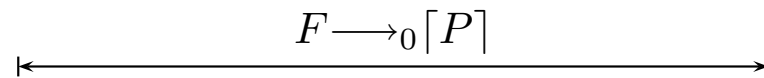
$$F \xrightarrow{\leq \theta} [P] \iff (F \wedge \ell \leq \theta) \longrightarrow [P]$$



# DC Standard Forms: Initialisation

- Followed-by-initially:

$$F \longrightarrow_0 [P] :\iff \neg(F ; [\neg P])$$



- (Timed) up-to-initially:

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

- Initialisation:

$$\square \vee ([P] ; true)$$

# 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

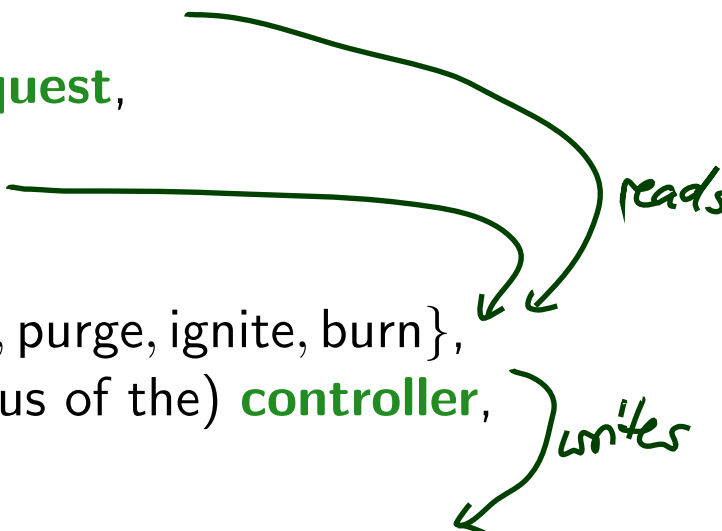
$$X_i = d_i, \quad d_i \in \mathcal{D}(X_i),$$

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

- **Basic phase** of  $C$ :

$C = \text{purge}$  (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  $X_i$ ,
  - $\varphi$  denotes a state assertion not depending on  $X_i$ .
- $\theta$  denotes a **rigid** term.

- **Initialisation:**

$$[\ ] \vee [\pi] ; true$$

- **Sequencing:**

$$[\pi] \longrightarrow [\pi \vee \pi_1 \vee \dots \vee \pi_n]$$

- **Progress:**

$$[\pi] \xrightarrow{\theta} [\neg\pi]$$

- **Synchronisation:**

$$[\pi \wedge \varphi] \xrightarrow{\theta} [\neg\pi]$$

# DC Implementables Cont'd

- **Bounded Stability:**

$$\underbrace{([\neg\pi] ; [\pi \wedge \varphi])}_{\bar{F}} \xrightarrow{\leq \theta} \underbrace{[\pi \vee \pi_1 \vee \dots \vee \pi_n]}_{\mathcal{P}}$$

- **Unbounded Stability:**

$$[\neg\pi] ; [\pi \wedge \varphi] \longrightarrow [\pi \vee \pi_1 \vee \dots \vee \pi_n]$$

- **Bounded initial stability:**

$$[\pi \wedge \varphi] \xrightarrow{\leq \theta}_0 [\pi \vee \pi_1 \vee \dots \vee \pi_n]$$

- **Unbounded initial stability:**

$$[\pi \wedge \varphi] \longrightarrow_0 [\pi \vee \pi_1 \vee \dots \vee \pi_n]$$



# *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  $\mathcal{D}(C) = \{\text{idle, purge, ignite, burn}\}$ ,  
representing the **controller**, (local)
- $G$  : Boolean,  
representing **gas valve**. (output)

# Gas Burner Controller Specification

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

input effect

timing

link output to ctrl. phases

# Gas Burner Controller Specification: Untimed

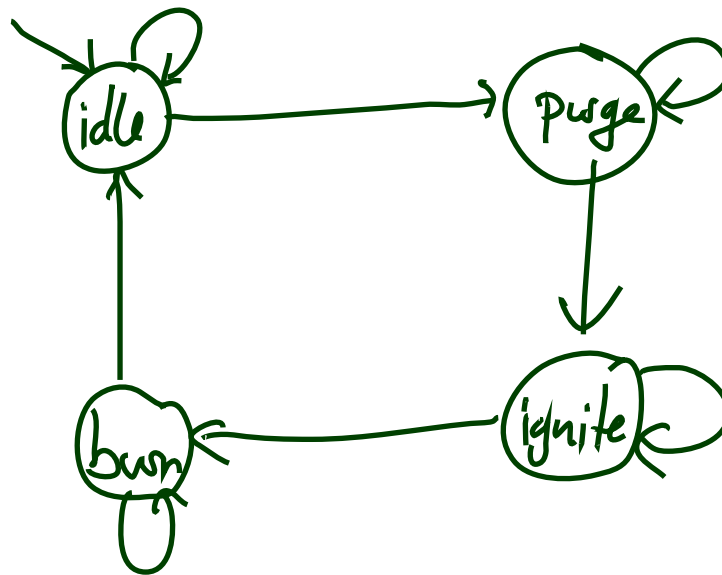
$\lceil \rceil \vee \lceil \text{idle} \rceil ; \text{true}$  (Init-1)

$\lceil \text{idle} \rceil \longrightarrow \lceil \text{idle} \vee \text{purge} \rceil$  (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)



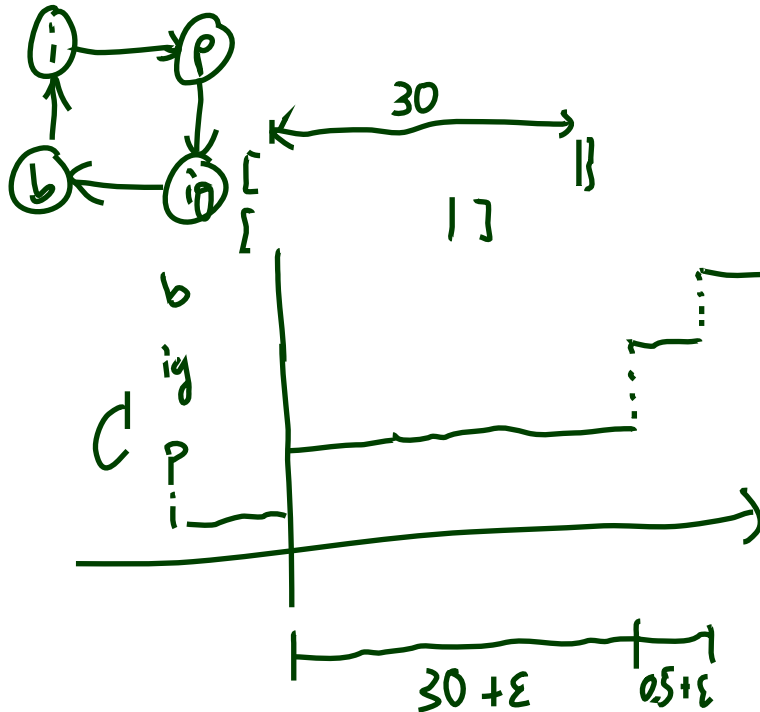
# Gas Burner Controller Specification: Timing

$$[\text{purge}] \xrightarrow{30+\epsilon} [\neg\text{purge}] \quad (\text{Prog-1})$$

$$[\text{ignite}] \xrightarrow{0.5+\epsilon} [\neg\text{ignite}] \quad (\text{Prog-2})$$

$$[\neg\text{purge}] ; [\text{purge}] \xrightarrow{\leq 30} [\text{purge}] \quad (\text{Stab-2})$$

$$[\neg\text{ignite}] ; [\text{ignite}] \xrightarrow{\leq 0.5} [\text{ignite}] \quad (\text{Stab-3})$$



# Gas Burner Controller Specification: Outputs

$$\overset{\pi}{\downarrow} \quad \underbrace{\varphi}_{\text{---}} \quad \overset{\uparrow}{\uparrow} \\ [G \wedge (\text{idle} \vee \text{purge})] \xrightarrow{\varepsilon} [\neg G] \quad (\text{Syn-3})$$

$$[\neg G \wedge (\text{ignite} \vee \text{burn})] \xrightarrow{\varepsilon} [G] \quad (\text{Syn-4})$$

$$[G] ; [\neg G \wedge (\text{idle} \vee \text{purge})] \longrightarrow [\neg G] \quad (\text{Stab-6})$$

$$[\neg G \wedge (\text{idle} \vee \text{purge})] \longrightarrow_0 [\neg G] \quad (\text{Stab-6-init})$$

$$[\neg G] ; [G \wedge (\text{ignite} \vee \text{burn})] \longrightarrow [\dot{G}] \quad (\text{Stab-7})$$

$$\uparrow \uparrow \vee \uparrow \neg G \uparrow ; \text{true} \quad (\text{Init-4})$$

value close/open  
after  $\varepsilon$  Tu the  
latest

value stays closed/open

# Gas Burner Controller Specification: Inputs

if  $H$  is "on" for less than  $\epsilon$ , we need not change phase

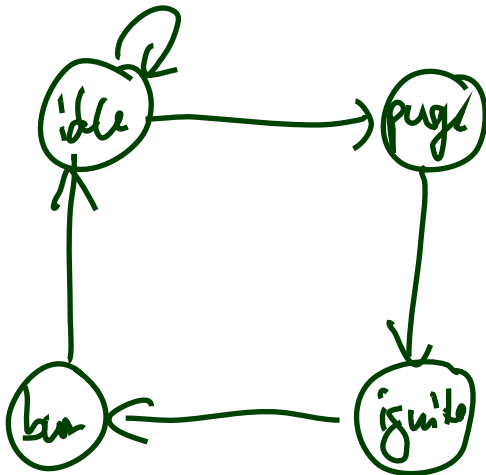
$$[\text{idle} \wedge H] \xrightarrow{\epsilon} [\neg \text{idle}] \quad (\text{Syn-1})$$

$$[\text{burn} \wedge (\neg H \vee \neg F)] \xrightarrow{\epsilon} [\neg \text{burn}] \quad (\text{Syn-2})$$

$$[\neg \text{idle}] ; [\text{idle} \wedge \neg H] \longrightarrow [\text{idle}] \quad (\text{Stab-1})$$

$$[\text{idle} \wedge \neg H] \longrightarrow_0 [\text{idle}] \quad (\text{Stab-1-init})$$

$$[\neg \text{burn}] ; [\text{burn} \wedge H \wedge F] \longrightarrow [\text{burn}] \quad (\text{Stab-4})$$





# Gas Burner Controller Specification: Assumptions

---

$\square \vee [\neg H] ; true$  (Init-2)

$\square \vee [\neg F] ; true$  (Init-3)

~~$\square \vee [\neg G] ; true$  (Init-4)~~

$[F] ; [\neg F \wedge \neg \text{ignite}] \longrightarrow [\neg F]$  (Stab-5)

$[\neg F \wedge \neg \text{ignite}] \longrightarrow_0 [\neg F]$  (Stab-5-init)

no spontaneous flame

# Gas Burner Controller Correctness Proof

---

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

## Recall:

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

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

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

for the **simplified**

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

Here we show

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

# Lemma 3.15

$$\models \underbrace{\text{GB-Ctrl}}_{[c,d]} \implies \square \left( \begin{array}{l} ([\text{idle}] \implies \int G \leq \varepsilon) \\ \wedge ([\text{purge}] \implies \int G \leq \varepsilon) \\ \wedge ([\text{ignite}] \implies \ell \leq 0.5 + \varepsilon) \\ \wedge ([\text{burn}] \implies \int \neg F \leq 2\varepsilon) \end{array} \right) \quad (*)$$

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

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

$$[G \wedge (\text{idle} \vee \text{purge})] \xrightarrow{\varepsilon} [\neg G] \quad (\text{Syn-3})$$

$$[G] ; [\neg G \wedge (\text{idle} \vee \text{purge})] \longrightarrow [\neg G] \quad (\text{Stab-6})$$

*conclude*

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

*(\*)*

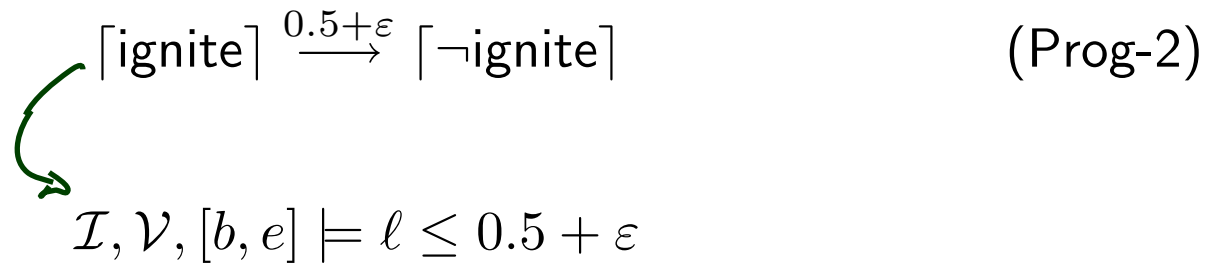
- Case 2:  $\mathcal{I}, \mathcal{V}, [b, e] \models [\text{purge}]$  Analogously to case 1.

# Lemma 3.15 Cont'd

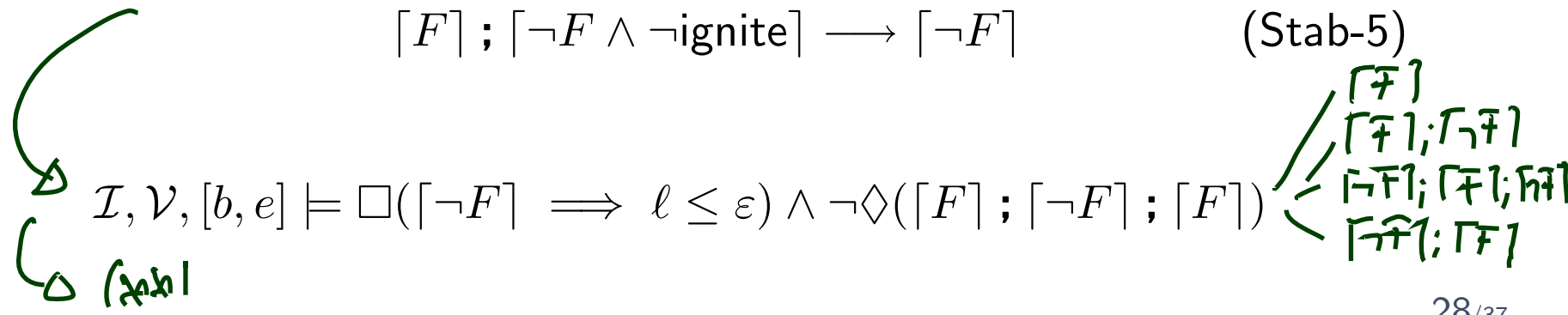
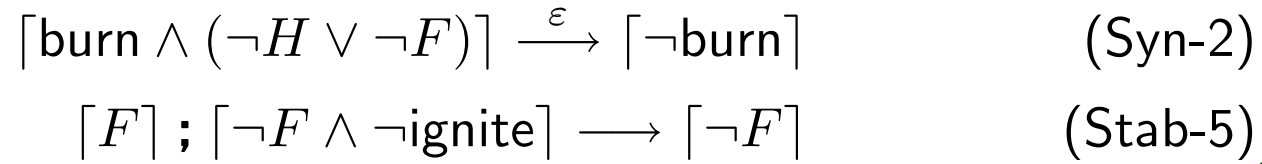
$(\llbracket \text{idle} \rrbracket \implies \int G \leq \varepsilon)$
$(\llbracket \text{purge} \rrbracket \implies \int G \leq \varepsilon)$
$(\llbracket \text{ignite} \rrbracket \implies \ell \leq 0.5 + \varepsilon)$
$(\llbracket \text{burn} \rrbracket \implies \int \neg F \leq 2\varepsilon)$

(\*)

- Case 3:  $\mathcal{I}, \mathcal{V}, [b, e] \models \llbracket \text{ignite} \rrbracket$



- Case 4:  $\mathcal{I}, \mathcal{V}, [b, e] \models \llbracket \text{burn} \rrbracket$



## Lemma 3.16

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

### Proof Sketch

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

Distinguish 5 cases:

$$\begin{array}{ll} I, V, [b, e] \models \top & (0) \\ \forall (\top \text{idle}; \text{true} \wedge \ell \leq 30) & (1) \\ \forall (\top \text{page}; \text{true} \wedge \ell \leq 30) & (2) \\ \forall (\top \text{quit}; \text{true} \wedge \ell \leq 30) & (3) \\ \forall (\top \text{down}; \text{true} \wedge \ell \leq 30) & (4) \end{array}$$

## Lemma 3.16 Cont'd

- Case 0:  $\mathcal{I}, \mathcal{V}, [b, e] \models \square$  ✓
- Case 1:  $\mathcal{I}, \mathcal{V}, [b, e] \models [\text{idle}] ; \text{true} \wedge \ell \leq 30$

$$[\text{idle}] \longrightarrow [\text{idle} \vee \text{purge}] \quad (\text{Seq-1})$$

$$[\neg \text{purge}] ; [\text{purge}] \xrightarrow{\leq 30} [\text{purge}] \quad (\text{Stab-2})$$

$$\hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models \neg [\text{idle}] \vee [\text{idle}] ; \neg [\text{purge}]$$

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

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

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

## Lemma 3.16 Cont'd

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

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

$$\begin{aligned} & \hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models (\lceil \text{burn} \rceil \vee \underbrace{\lceil \text{burn} \rceil; \lceil \text{idle} \rceil; \text{true}}) \wedge \ell \leq 30 \\ 3.15, (1) & \hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models (\sqrt{\ell} \leq 2\varepsilon \vee \sqrt{\ell} \leq 2\varepsilon; \sqrt{\ell} \leq 2\varepsilon) \wedge \ell \leq 30 \\ & \hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models \sqrt{\ell} \leq 4\varepsilon \end{aligned}$$

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

## Lemma 3.16 Cont'd

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

$$[\text{ignite}] \longrightarrow [\text{ignite} \vee \text{burn}] \quad (\text{Seq-3})$$

$$\begin{aligned} & \hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models ([\text{ignite}] \vee [\text{ignite}]; [\text{burn}]; \text{true}) \wedge \ell \leq 30 \\ 3.5, (2) & \hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models \int L \leq 0.5 + \varepsilon \vee (\int L \leq 0.5 + \varepsilon; \int L \leq 4\varepsilon) \wedge \ell \leq 30 \\ & \hookrightarrow \mathcal{I}, \mathcal{V}, [b, e] \models \int L \leq 0.5 + 5\varepsilon \end{aligned}$$

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



## Lemma 3.16 Cont'd

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

$$[\text{purge}] \longrightarrow [\text{purge} \vee \text{ignite}] \quad (\text{Seq-2})$$

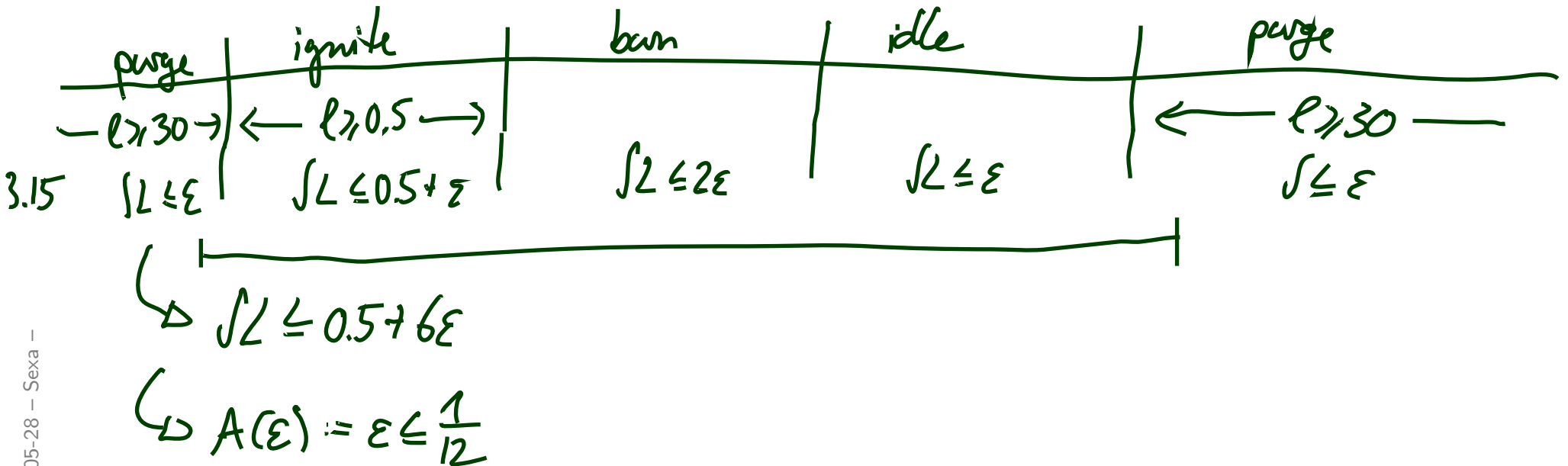
$$\begin{aligned} & \hookrightarrow \mathcal{I}, \mathcal{U}, [b, e] \models ([\text{purge}] \vee [\text{ignite}]; \text{true}) \wedge \ell \leq 30 \\ 3.15, (3) & \hookrightarrow \mathcal{I}, \mathcal{U}, [b, e] \models \sqrt{L} \leq \varepsilon \vee (\sqrt{L} \leq \varepsilon; \sqrt{L} \leq 0.5 + \varepsilon) \\ & \hookrightarrow \mathcal{I}, \mathcal{U}, [b, e] \models \sqrt{L} \leq 0.5 + 6\varepsilon \end{aligned}$$

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

# Correctness Result

## Theorem 3.17.

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



# 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

$$\text{Prog-1} = [\text{purge}] \xrightarrow{30+\varepsilon} [\neg\text{purge}]$$

for instance?

Niqa, 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.