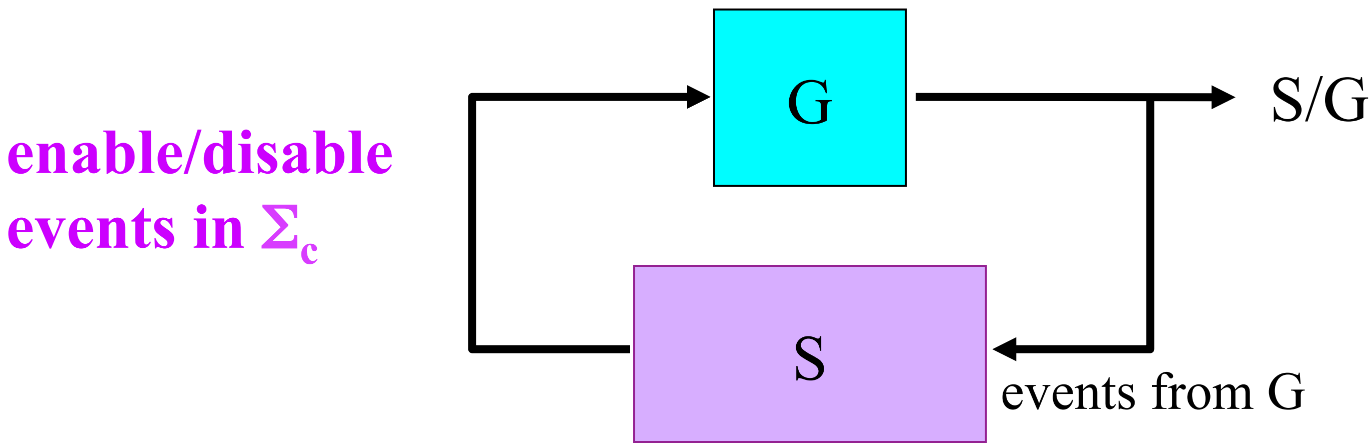
State-Based Supervisory Control

Dr Rong Su
S1-B1b-59, School of EEE
Nanyang Technological University
Tel: +65 6790-6042, Email: rsu@ntu.edu.sg

Outline

- Motivation
- Predicates and Predicate Transformers
- State Feedback Control
- Example
- Conclusions

The Main Challenge in Event-Based Feedback



$\Sigma = \Sigma_c \cup \Sigma_{uc}$

$\Sigma = \Sigma_o \cup \Sigma_{uo}$

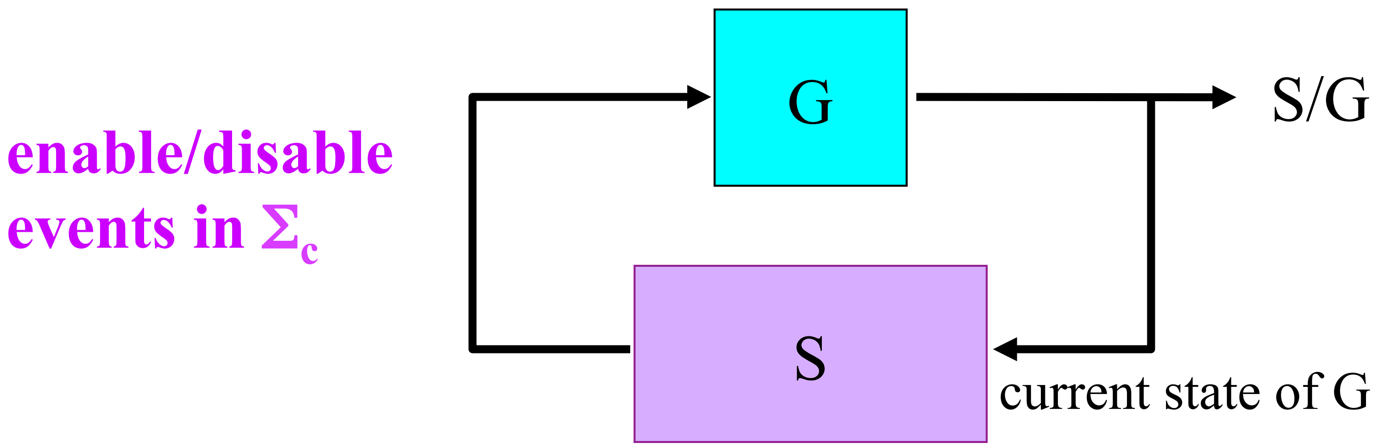
Σ_c:=controllable alphabet

Σ_o:=observable alphabet

- Control map $V : L(G) \rightarrow \Gamma = \{ \gamma \subseteq \Sigma \mid \Sigma_{uc} \subseteq \gamma \}$
- To encode V, the language L(S/G) needs to be stored.

As a consequence, a huge amount of memory is needed !

If Switch to State-Based Feedback



$\Sigma = \Sigma_c \cup \Sigma_{uc}$

$\Sigma = \Sigma_o \cup \Sigma_{uo}$

Σ_c:=controllable alphabet

Σ_o:=observable alphabet

- Control map $f : X \rightarrow \Gamma = \{ \gamma \subseteq \Sigma \mid \Sigma_{uc} \subseteq \gamma \}$
- To encode f, only relevant states are needed.

We probably can't save synthesis time, but we can save memory!

Outline

- Motivation
- Predicates and Predicate Transformers
- State Feedback Control
- Example
- Conclusions

The Concept of Predicates

- Let $G = (Q, \Sigma, \delta, q_0, Q_m)$
- A *predicate* P over Q is a function $P : Q \rightarrow \{0,1\}$
- Let $Q_P := \{ q \in Q \mid P(q) = 1 \}$
- $q \,?\, P \Leftrightarrow P(q) = 1 \Leftrightarrow q \in Q_P$
- Let Pred(Q) be the collection of all predicates on Q

Boolean Expressions over Predicates

- Given P, define $\neg P$, where $(\neg P)(q) = 1$ iff $P(q) = 0$
- Given P_1 and P_2, define $P_1 \wedge P_2$ and $P_1 \vee P_2$, where
  - $(P_1 \wedge P_2)(q) = 1$ iff $P_1(q) = 1 \wedge P_2(q) = 1$
  - $(P_1 \vee P_2)(q) = 1$ iff $P_1(q) = 1 \vee P_2(q) = 1$
- Recall the De Morgen rules
  - $\neg(P_1 \wedge P_2) = (\neg P_1) \vee (\neg P_2)$
  - $\neg(P_1 \vee P_2) = (\neg P_1) \wedge (\neg P_2)$

- For any $P_1, P_2 \in Pred(Q)$, P_1 ? P_2 iff $P_1 \wedge P_2 = P_1$ iff $\neg P_1 \vee P_2$

- (Pred(Q), ?) is a complete lattice

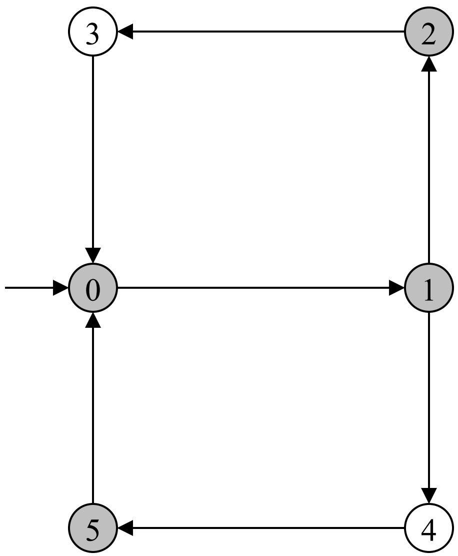
- The top element of (Pred(Q), ?) is ?, where $Q_{?} = Q$
  - ? can be interpreted as *true*

- The bottom element of (Pred(Q), ?) is $\perp$, where $Q_{\perp} = \emptyset$
  - $\perp$ can be interpreted as *false*

The Reachability Predicate R(G,P)

- Given P∈Pred(Q), we define a new predicate R(G,P) as follows
  - q_0 ? P $\Rightarrow$ q_0 ? R(G,P)
  - q ? R(G,P) $\wedge$ $\sigma \in \Sigma$ $\wedge$ $\delta(q,\sigma)!$ $\wedge$ $\delta(q,\sigma)$? P $\Rightarrow$ $\delta(q,\sigma)$? R(G,P)
  - No other states satisfy R(G,P)
- R(G,P) is the set of all states reachable from q_0 and satisfy P
- Clearly, R(G,P) ? P

Example

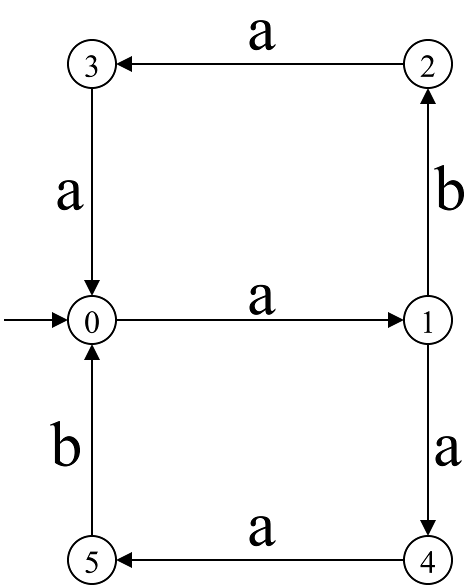


- $Q_P = \{0, 1, 2, 5\}$
- $Q_{R(G,P)} = \{0, 1, 2\}$
- $Q_{R(G,P)} \subseteq Q_P$ because $R(G,P)$? P

The Weakest Liberal Precondition $M_\sigma(P)$

- For any $\sigma \in \Sigma$ and $P \in Pred(Q)$, let $M_\sigma(P)$ be a predicate such that

$$q \,?\, M_\sigma(P) \text{ iff } \neg\delta(q,\sigma)! \vee \delta(q,\sigma) \,?\, P$$

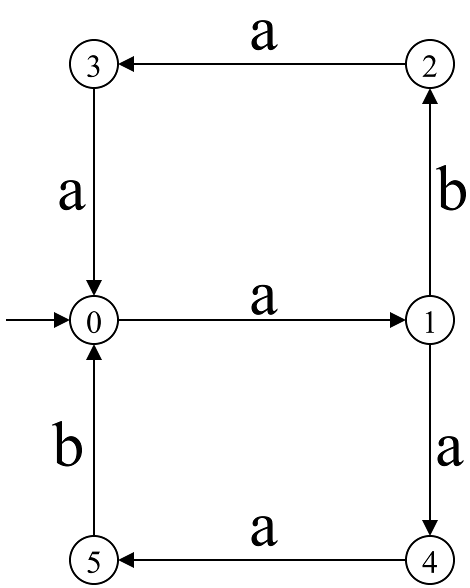


- $Q_P = \{2, 3, 5\}$
- $Q_{Ma(P)} = \{2, 4, 5\}$ (why?)

The Strongest Postcondition $N_\sigma(P)$

- For any $\sigma \in \Sigma$ and $P \in Pred(Q)$, let $N_\sigma(P)$ be a predicate such that

$$q \boxed{?} N_\sigma(P) \text{ iff } (\exists q' \in Q)\ \delta(q',\sigma)=q \wedge q' \boxed{?} P$$



- $Q_P = \{0, 1, 2, 4\}$
- $Q_{Na(P)} = \{1, 3, 4, 5\}$ (why?)

Outline

- Motivation
- Predicates and Predicate Transformers
- State Feedback Control
- Example
- Conclusions

State Feedback

- Let $f : Q \rightarrow \Gamma = \{\gamma \subseteq \Sigma | \Sigma_{uc} \subseteq \gamma\}$ be a state feedback control (SFBC)
  - We say $\sigma \in \Sigma$ is *enabled* at q, if $\sigma \in f(q)$, and is *disabled* otherwise
- For each $\sigma \in \Sigma$, introduce a predicate $f_\sigma \in Pred(Q)$ such that

$$f_\sigma(q) = 1 \text{ iff } \sigma \in f(q)$$

- The closed-loop transition map induced by f is defined as

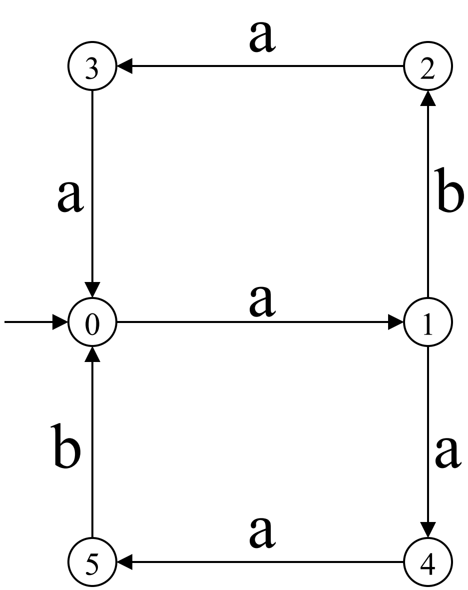
$$\delta^f(q,\sigma) = q' \text{ iff } \delta(q,\sigma) = q' \wedge f_\sigma(q) = 1$$

- Write $G^f = (Q, \Sigma, \delta^f, q_0, Q_m)$ for the closed-loop system by (G,f)
- Clearly, for any P∈Pred(Q), we have $R(G^f,P)$ ⍰ $R(G,P)$

The Concept of Controllability

- $P \in Pred(Q)$ is *controllable* with respect to G, if

$$P \boxed{?} R(G,P) \wedge (\forall \sigma \in \Sigma_{uc})\ P \boxed{?} M_{\sigma}(P)$$



- $\Sigma_c = \{a\}$
- $Q_P = \{0, 1, 4, 5\}$
- Is P controllable? Why?

- **Theorem 1**

Let $P \in Pred(Q)$ and $P \neq \mathit{false}$. Then P is controllable with respect to G if and only if there exists a SFBC f such that $R(G^f, \mathit{true}) = P$.

The Supremal Controllable Predicate

- For each P∈Pred(Q), let

$$\mathcal{CP}(P) := \{K \in Pred(Q) \mid K \boxed{?} P \wedge K \text{ is controllable}\}$$

  – Since *false*∈$\mathcal{CP}$(P), we know that $\mathcal{CP}(P) \neq \emptyset$.

- **Proposition 1**

  – $\mathcal{CP}$(P) is closed under arbitrary disjunctions.

  – In particular, the supremal controllable predicate sup$\mathcal{CP}$(P) exists in $\mathcal{CP}$(P).

Compute The Supremal Controllable Predicate

- Given $P \in Pred(Q)$, define a new predicate $<P>$ as follows:

$$q \boxed{?} <P> \text{ iff } (\forall w \in \Sigma_{uc}^{*})\ \delta(q,w)! \Rightarrow \delta(q,w) \boxed{?} P$$

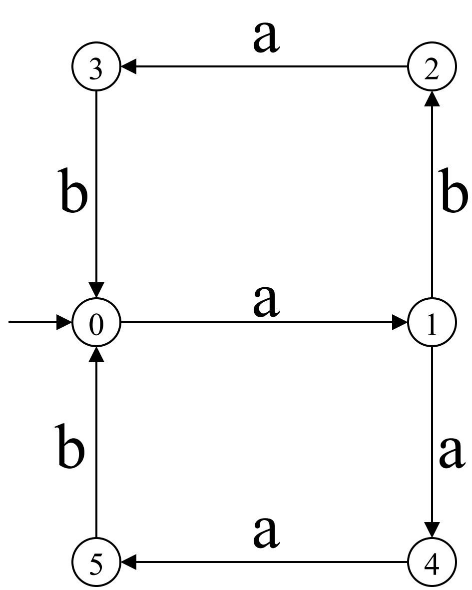
- **Proposition 2**

$$\sup \mathcal{CP}(P) = R(G,<P>)$$

- **Corollary 1**

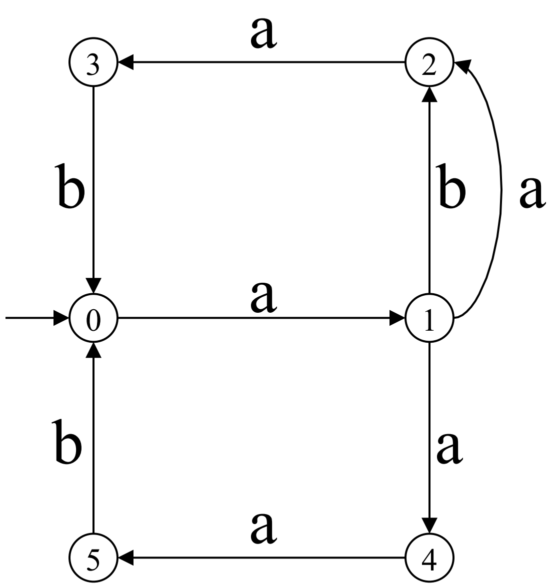
$$\sup \mathcal{CP}(P) \neq \mathit{false} \text{ iff } R(G,<P>) \neq \mathit{false} \text{ iff } q_0 \boxed{?} <P>$$

Example



- $\Sigma_c = \{a\}$
- $Q_P = \{0, 1, 4, 5\}$
- P is not controllable. (Why?)
- $Q_{<P>} = \{0, 4, 5\}$
- $Q_{R(G,<P>)} = \{0\}$

Improving Permissiveness in SFBC



- $\Sigma_c = \{a\}$
- $Q_P = \{0, 1, 2, 4\}$
- $Q_{R(G,<P>)} = \{0, 1, 2, 4\}$
- $f_1|_{Q-\{1\}} = f_2|_{Q-\{1\}}$
- $f_1(1) = \{b\}$ and $f_2(1) = \{a,b\}$
- $R(G^{f1},\mathit{true}) = R(G^{f2},\mathit{true}) = <P>$

We call that f_2 is *balanced*.

Formally Speaking

- An SFBC $f: Q \rightarrow \Gamma$ is *balanced*, if

$$(\forall q,q' \in Q)(\forall \sigma \in \Sigma)\ q,q' \in R(G^f, \mathit{true}) \wedge \delta(q,\sigma)=q' \Rightarrow f_\sigma(q)=1$$

Modular SFBC

- Suppose we have a collection of predicates $\{P_i | i \in I = \{1,\dots,n\}\}$
  - Each P_i can be interpreted as a local requirement
- Let $P := \wedge_{i \in I} P_i$
- For each $i \in I$, let $f_i : Q \to \Gamma$ be an optimal SFBC for P_i, namely

$$R(G^{f_i}, \mathrm{true}) = \sup \mathcal{CP}(P_i)$$

- Let $f : Q \to \Gamma$ such that $f(q) := \cap_{i \in I} f_i(q)$
  - Or symbolically, for each $\sigma \in \Sigma$, $f_\sigma := \wedge_{i \in I} f_{i,\sigma}$

- **Theorem 2**

Assume that, for each $i \in I$, f_i is balanced. Then f is balanced and

$$R(G^f, \mathit{true}) = \sup \mathcal{CP}(P)$$

Dynamic State Feedback Control

- Given $P \in Pred(Q)$, let

$$L(G,P) := \{s \in \Sigma^* \mid \delta(q_0,s)! \wedge (\forall s' \boxed{?} s)\ \delta(q_0,s') \boxed{?} P\}$$

- Let E be a requirement, and H a memory, whose state set is Y.
- Let $P \in Pred(Q \times Y)$. We call (E,P) is *compatible* with $G \times H$, if

$$L(G \times H, P) = L(E) \cap L(G) = L(G \times E)$$

Dynamic State Feedback Control (cont.)

- Let $\mathcal{CP}_{G\times H}(P)$ be the set of all controllable subpredicates (on $Q\times Y$) of P
- Let $\mathcal{C}_G(L(G\times E))$ be the set of all controllable sublanguages of $L(G\times E)$

- **Theorem 3**

If (E,P) is compatible with $G\times E$, then

$$L(G\times H, \sup\mathcal{CP}_{G\times H}(P)) = \sup\mathcal{C}_G(L(G\times E))$$

Dynamic State Feedback Control (cont.)

- Let $\{E_i \mid i \in I = \{1,\ldots,n\}\}$ be a set of requirements and $E = \cap_{i \in I} E_i$
- Let H_i $(i \in I)$ a memory for G, whose state set is Y_i and $H = \cap_{i \in I} H_i$
- Let $P_i \in Pred(Q \times Y_i)$ such that (E_i, P_i) is *compatible* with $G \times H_i$
- Define $P \in Pred(Q \times Y_1 \times \ldots \times Y_n)$ such that

$$(q, y_1, \ldots, y_n) \,?\, P \text{ iff } (q, y_i) \,?\, P_i \text{ for each } i \in I$$

Dynamic State Feedback Cobtrol (cont.)

- Let f_i be a balanced SFBC for $G\times H_i$ such that

$$L(G^{fi}, \mathit{true}) = \sup\mathcal{C}_G(L(G\times E_i))$$

- Define $f : Q\times Y_1\times\ldots\times Y_n \to \Gamma$ such that

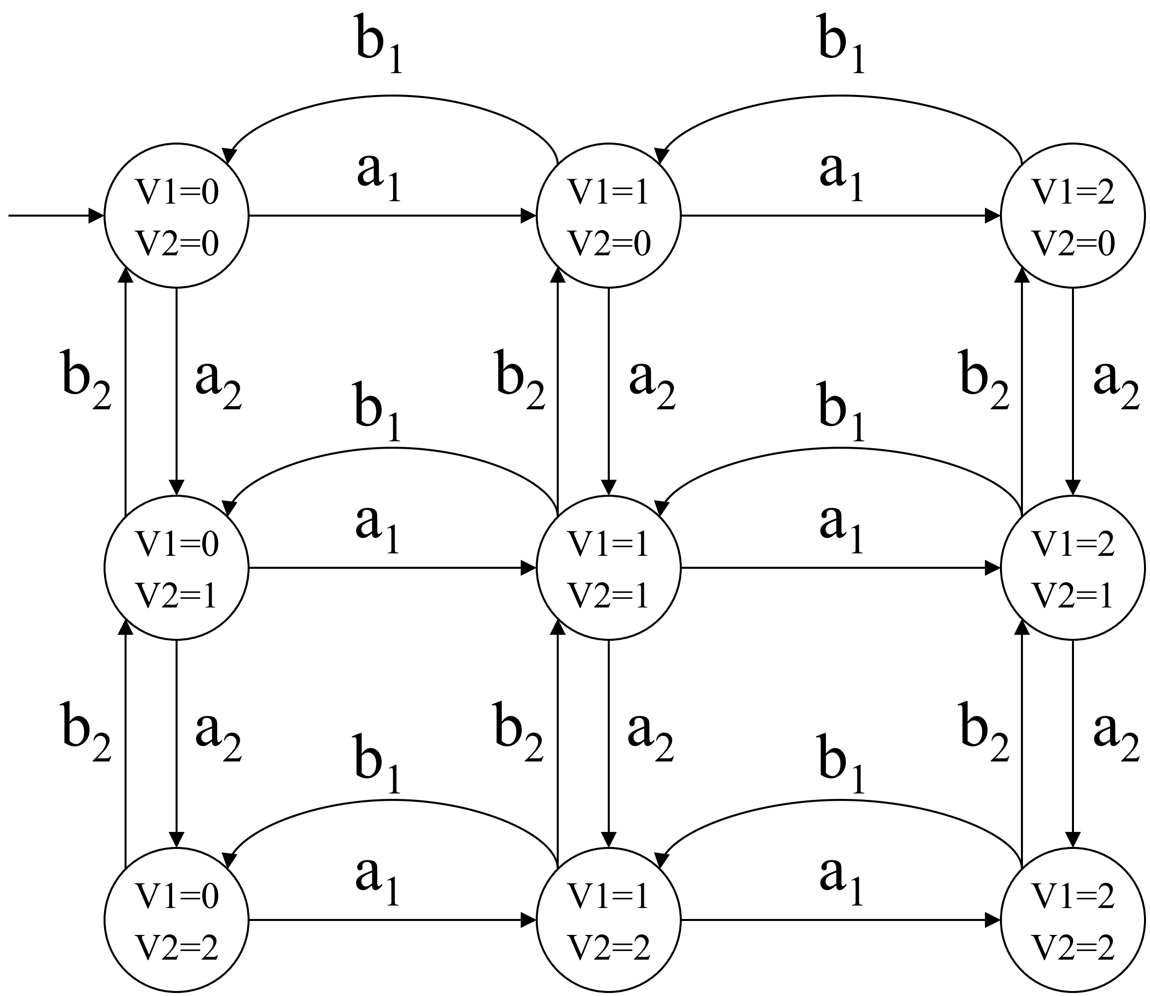
$$\sigma\in f(q,y_1,\ldots,y_n) \text{ iff } \sigma\in\cap_{i\in I} f_i(q,y_i)$$

- **Theorem 4**
  - (E,P) is compatible with $G\times H$
  - f is a balanced SFBC for $G\times H$
  - $L(G\times H, \sup\mathcal{CP}_{G\times H}(P)) = \sup\mathcal{C}_G(L(G\times E))$

Outline

- Motivation
- Predicates and Predicate Transformers
- State Feedback Control
- Example
- Conclusions

A Plant Model



- $\Sigma = \{a_1, a_2, b_1, b_2\}$
- $\Sigma_c = \{a_1, a_2\}$

G

Requirements

- Requirement 1

P_1 := the value of V1 should be no more than 1

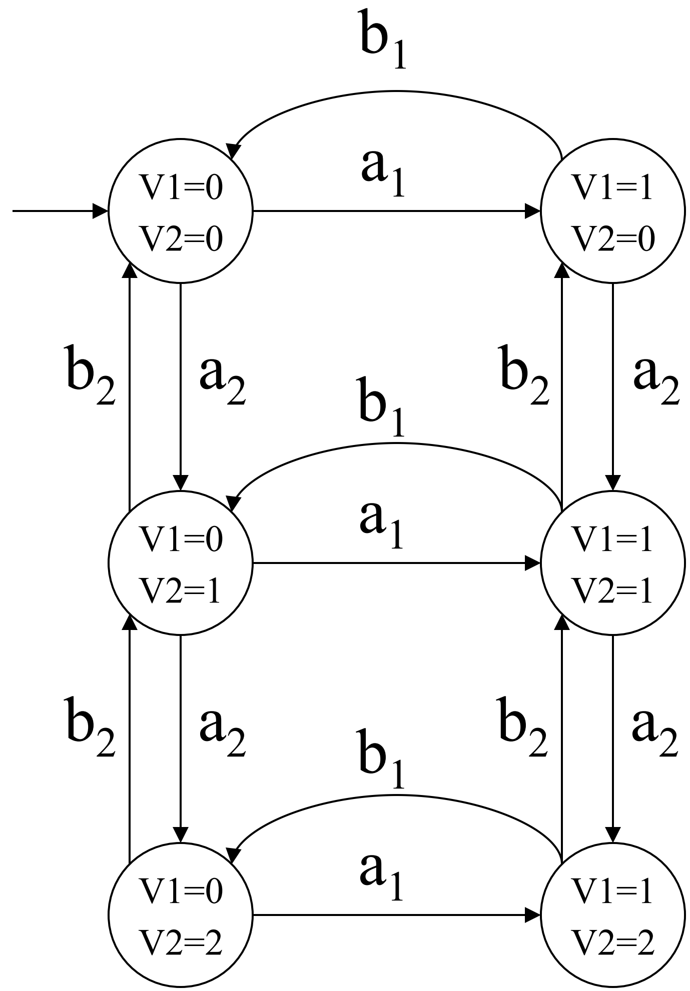
- Requirement 2

P_2 := the value of V2 should be no more than 1

State Feedback Control for Requirement 1

- We can check that P_1 is controllable with respect to G (why?)
- The corresponding state feedback control $f_1 : Q \rightarrow \Gamma$ is:
  - $f_1(V1=1,V2=0) = f_1(V1=1,V2=1) = f_1(V1=1,V2=2) = \{a_2, b_1, b_2\}$
  - For the rest of states q, set $f_1(q) = \Sigma$

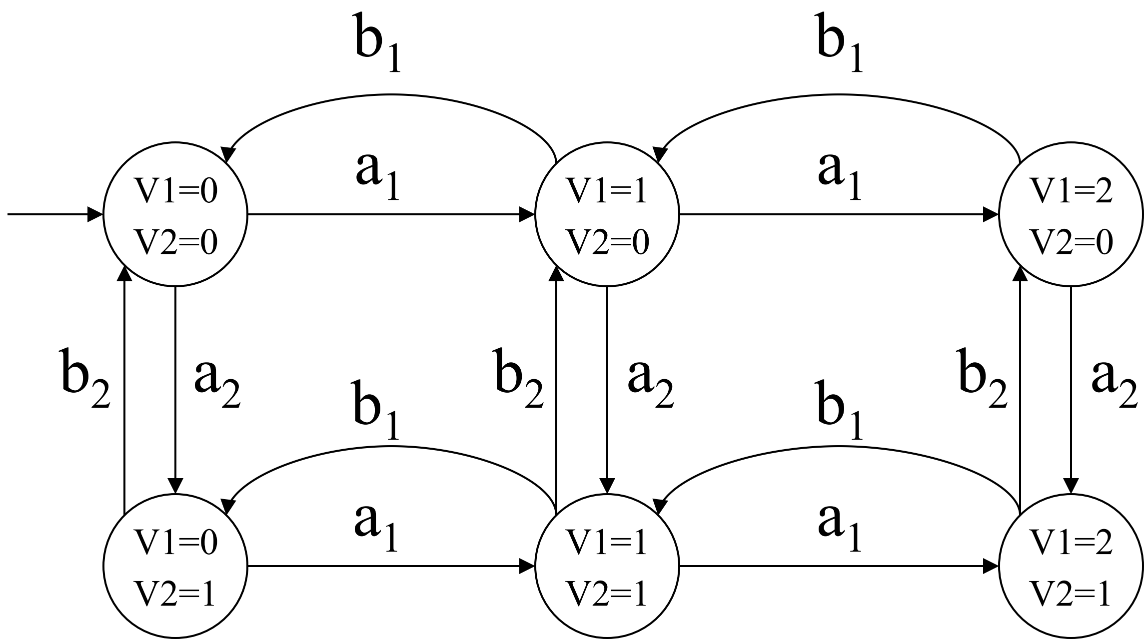
The Closed-loop System G^{f1}



State Feedback Control for Requirement 2

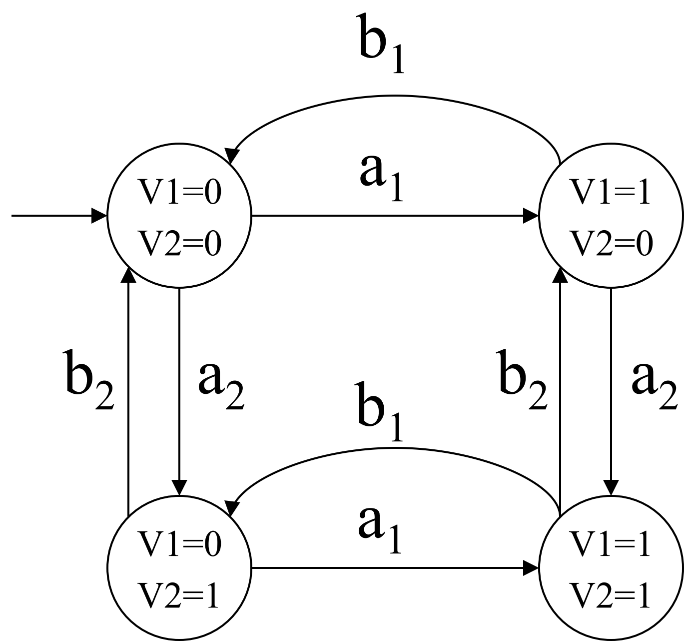
- We can check that P_2 is controllable with respect to G (why?)
- The corresponding state feedback control $f_2 : Q \rightarrow \Gamma$ is:
  - $f_2(V1=0,V2=1) = f_1(V1=1,V2=1) = f_1(V1=2,V2=1) = \{a_1, b_1, b_2\}$
  - For the rest of states q, set $f_2(q) = \Sigma$

The Closed-loop System G^{f2}



The Closed-Loop System $G^{f1\wedge f2}$

- Let $f := f_1 \wedge f_2$ such that $\sigma \in f(q)$ iff $\sigma \in f_1(q) \cap f_2(q)$
  - $f(V1=1,V2=0) = \{a_2, b_1, b_2\}$
  - $f(V1=0,V2=1) = \{a_1, b_1, b_2\}$
  - $f(V1=1,V2=1) = \{b_1, b_2\}$



Encode State Feedback Control Maps

- $f_{1,a1}(q) = 0$ if V1=1; otherwise, $f_{1,a1}(q) = 1$
- $f_{1,a2} = f_{1,b1}(q) = f_{1,b2}(q) = 1$
- $f_{2,a2}(q) = 0$ if V2=1; otherwise, $f_{2,a2}(q) = 1$
- $f_{2,a1} = f_{2,b1}(q) = f_{2,b2}(q) = 1$

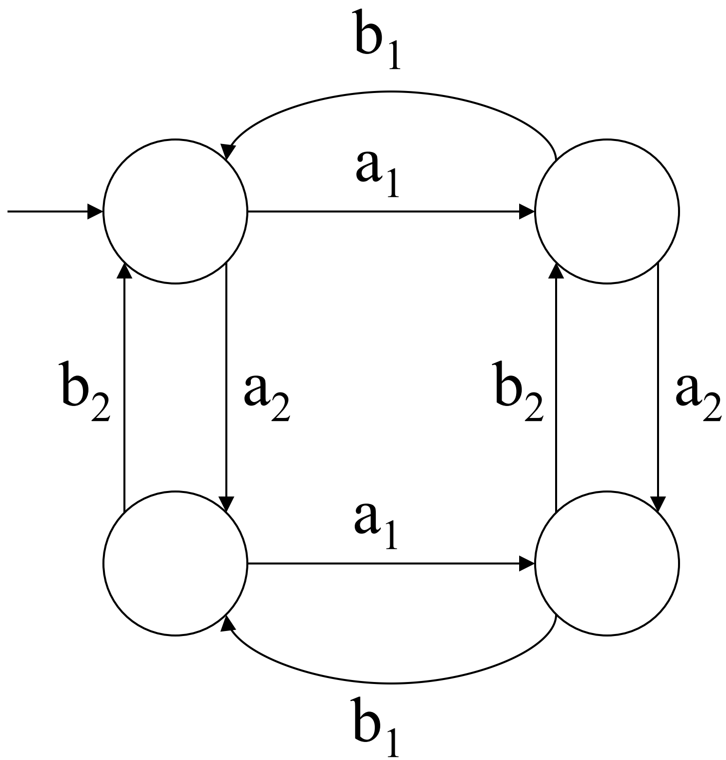
If in terms of event disablement, we have the following rules

$$V1 = 1 \Rightarrow a_1 \text{ is disabled}$$

$$V2 = 1 \Rightarrow a_2 \text{ is disabled}$$

Compare with Event-Based Feedback Control

- To encode the event-based feedback control map $f : L(G) \rightarrow \Gamma$



EBFC needs more memory than SFBC does

Conclusions

- Advantages of state feedback control
  - No memory for previous executions is required
  - The control map can be effectively encoded
- Disadvantage
  - It is applicable only when states are observable