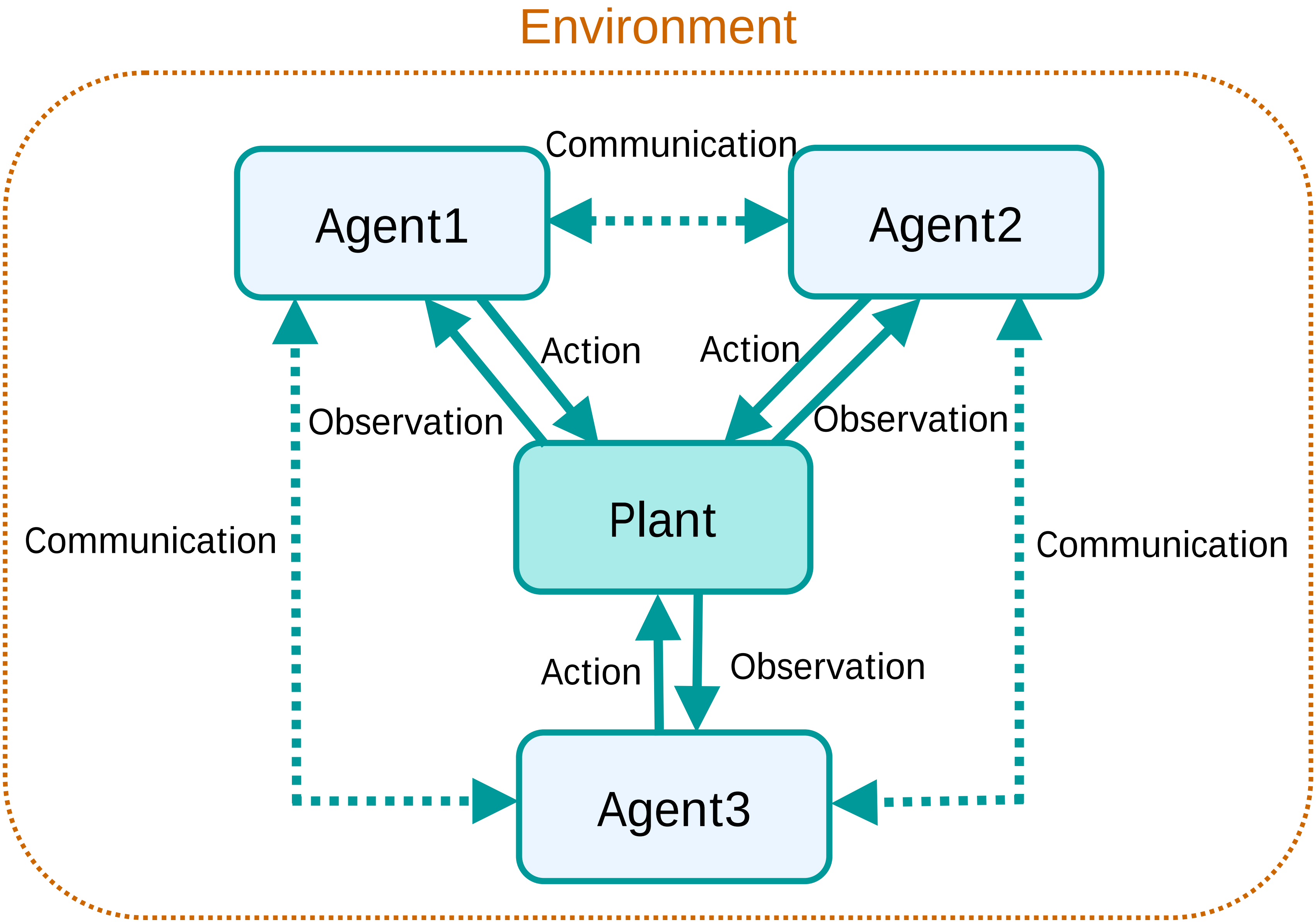
On Solvability of an Agent-Based Control Problem under Dynamic Environment

Kunihiko Hiraishi
School of Information Science
Japan Advanced Institute of Science and Technology

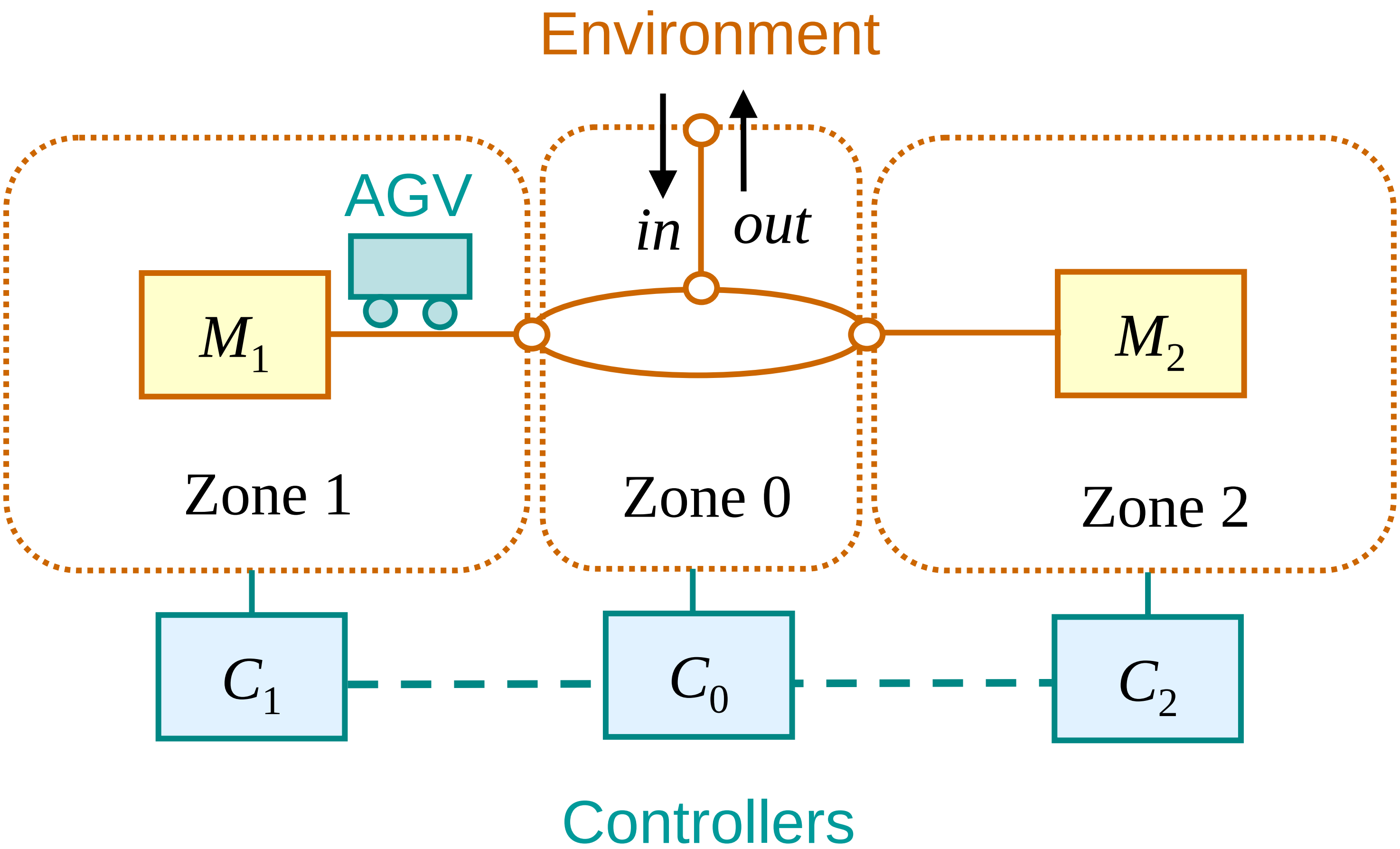
Contents

1. Decentralized control under dynamic environment.
2. Problem formulation: agent-based control under dynamic environment .
3. Solvability of the problem.

Agent-based control of DES



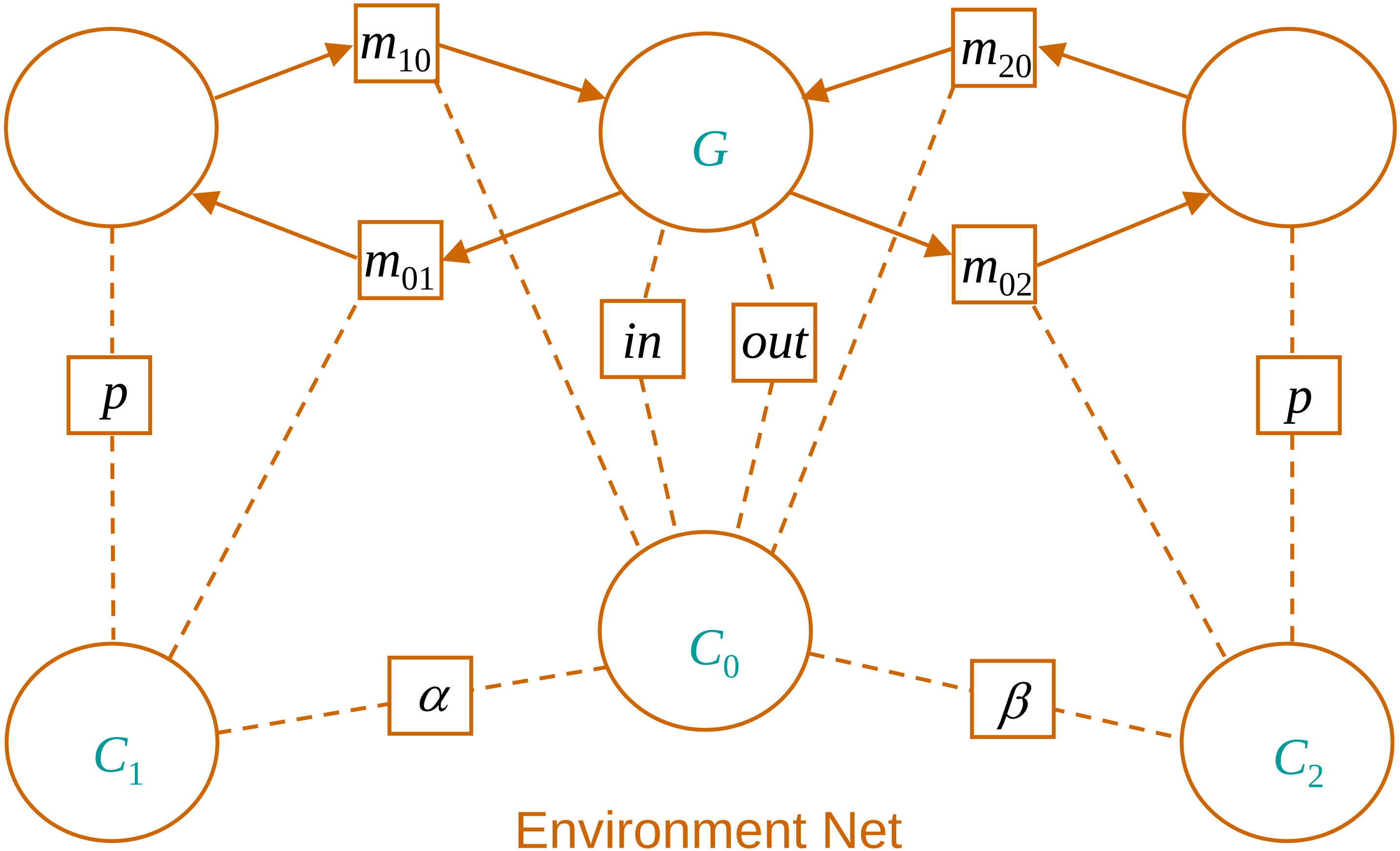
A simple AGV system



Dynamic environment

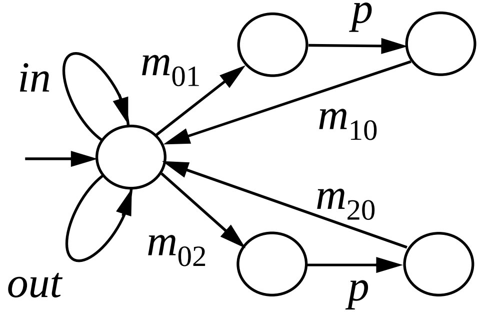
- The sets of controllable/observable events may change dynamically in the environment → Each controller C_i can control/observe the behavior of AGV if it is in zone Z_i.
- Communication link may change, e.g., mobile agent systems.

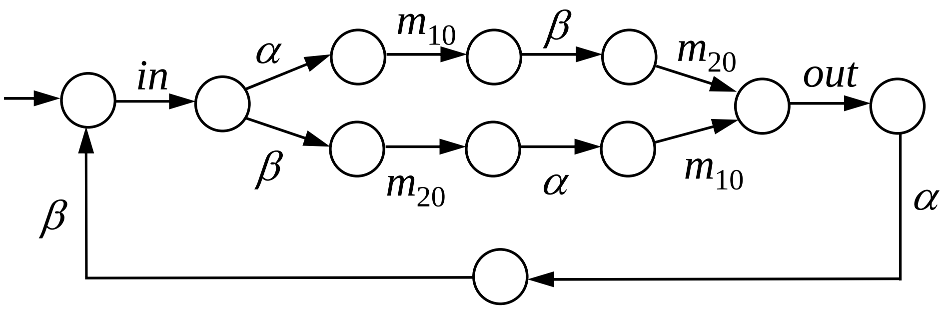
Modeling



*PN*²: Petri Nets in a Petri Net

Modeling

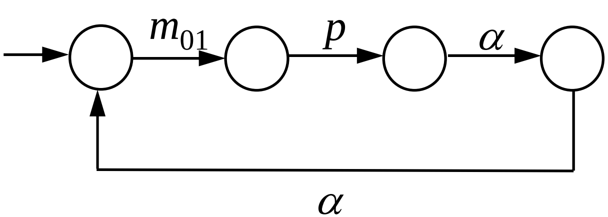


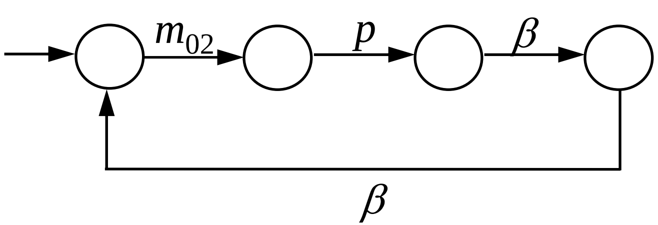


G

C_0

Agent Nets





C_1

C_2

Agent-Based Control under Dynamic Environment (ABCDE)

Given:

- A finite automaton G defined over Σ that represents the plant.
- The environment under which G and controllers $Z = \{ C_1, ..., C_n \}$ act.
- A finite automaton S that represents the desired behavior.

Find:

- Controllers $Z = \{ C_1, ..., C_n \}$ such that G/Z and S are Σ–bisimilar.

Why bisimulation?

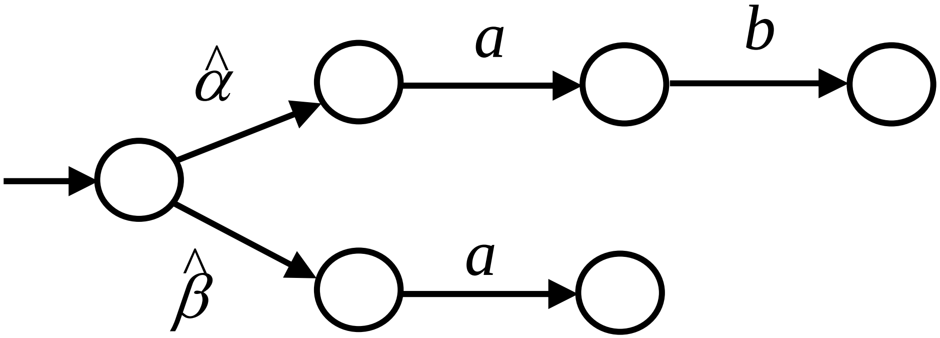
The controlled system has two kinds of events:

Σ: the set of *plant events*

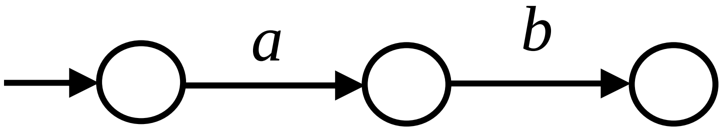
$\hat{\Sigma}$: the set of *synchronization events* (communication, movement of agents, ...)

However, the plant G and the spec. S are defined over Σ.

Why bisimulation?



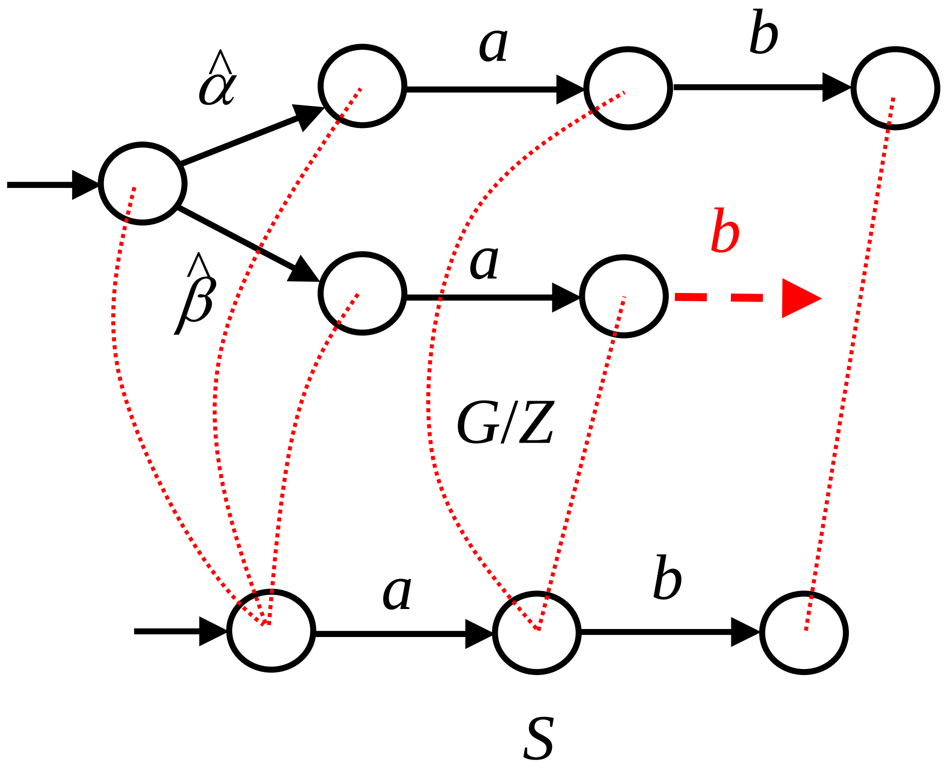
G/*Z*



S

$L(G/Z)|_{\Sigma} = L(S)$, but *b* does not always occur after *a*.

Why bisimulation?



G/Z and S are not Σ-bisimilar.

How to describe dynamic environment

- $\Sigma_{c,i}$ (controllable events), $\Sigma_{o,i}$ (observable events), $\hat{\Sigma}_i$ (sync. events) depends on *the configuration* π of the system, i.e, $\Sigma_{c,i}(\pi)$, $\Sigma_{o,i}(\pi)$, $\Sigma_i(\pi)$.
- Let Π be the finite set of all configurations.
- The configuration changes according to a transition function δ_Π: $\Pi \times (\Sigma \cup \hat{\Sigma}) \to \Pi$.
- PN^2 gives an uniform way to define the problem including dynamic environment.
- Since Π is finite, existence of dynamic environment has no effect on the decidability of the problem.

Decidability/undecidability results on decentralized control problem

Let A, E be regular languages and Let G be a finite automaton.

Without communication

$A \subseteq L(G/Z) \subseteq E$	decidable [Rudie and Wonham 92]
$A \subseteq L(G/Z) \subseteq E$, ω-*language, deadlock-free* $\equiv$ *-*language, nonblocking*	undecidable [Lamouchi and Thistle 00]
$L_m(G/Z) = E$, *nonblocking*	decidable [Rudie and Wonham 92]
$L_m(G/Z) \subseteq E$, *nonblocking*	undecidable [Tripakis 01]

Decidability/undecidability results on decentralized control problem

Let A, E be regular languages and Let G be a finite automaton.

With communication

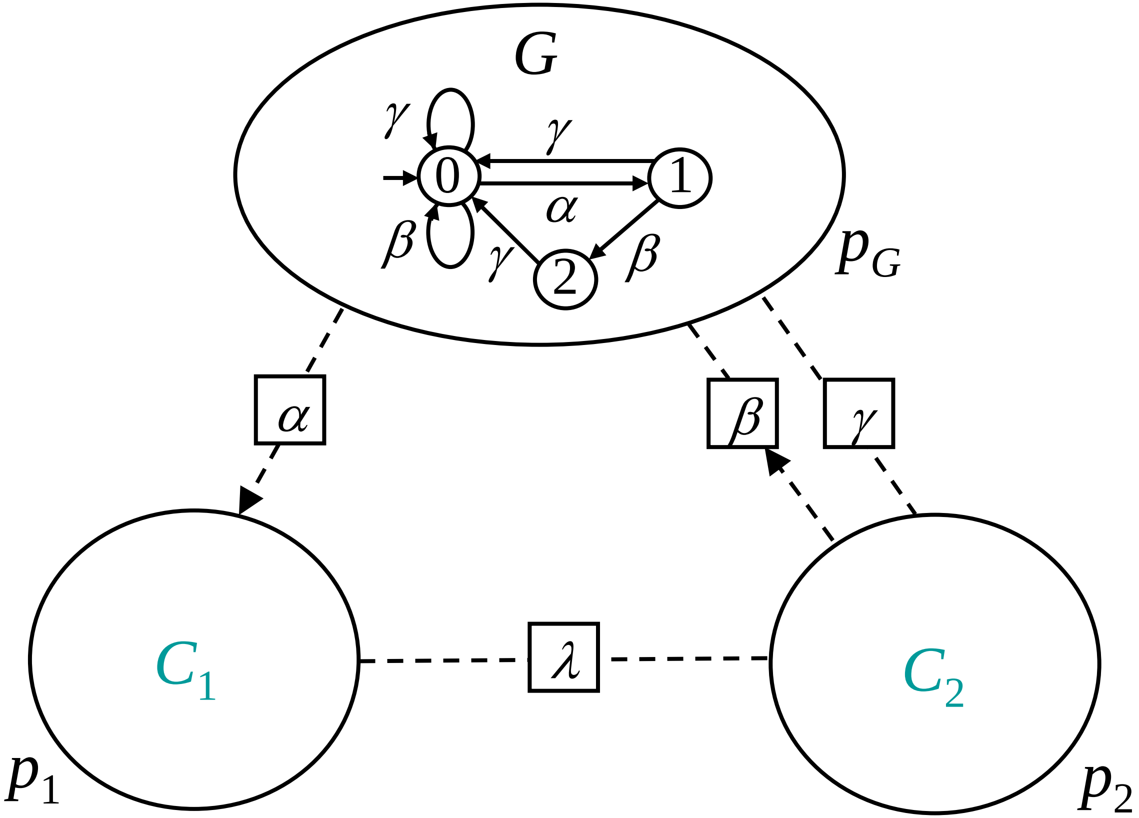
$L_m(G/Z)\|_{\Sigma} = E$, *nonblocking, no-delay communication*	decidable [Barett and Lafortune 98]
$L_m(G/Z) \models \phi$, ϕ : *responsible property*, *unbounded delay communication*	undecidable [Tripakis 00]
$L_m(G/Z) \models \phi$, ϕ : *responsible property*, *k-bounded delay communication*	(maybe) decidable [Tripakis 00]

Responsible property: $a \to b$ (b occurs after every a).

Undecidability

- Problem ABCDE is undecidable in general. This is proved by simulating Tripakis's architecture of decentralized control with unbounded delay communication.
- How can we find finite-state controllers?

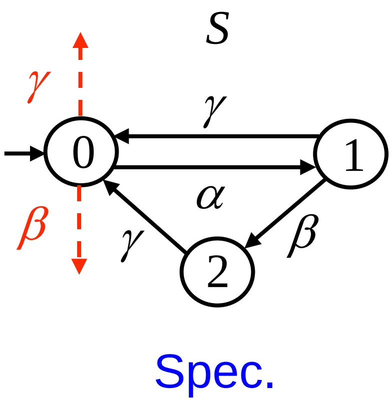
An instance of ABCDE

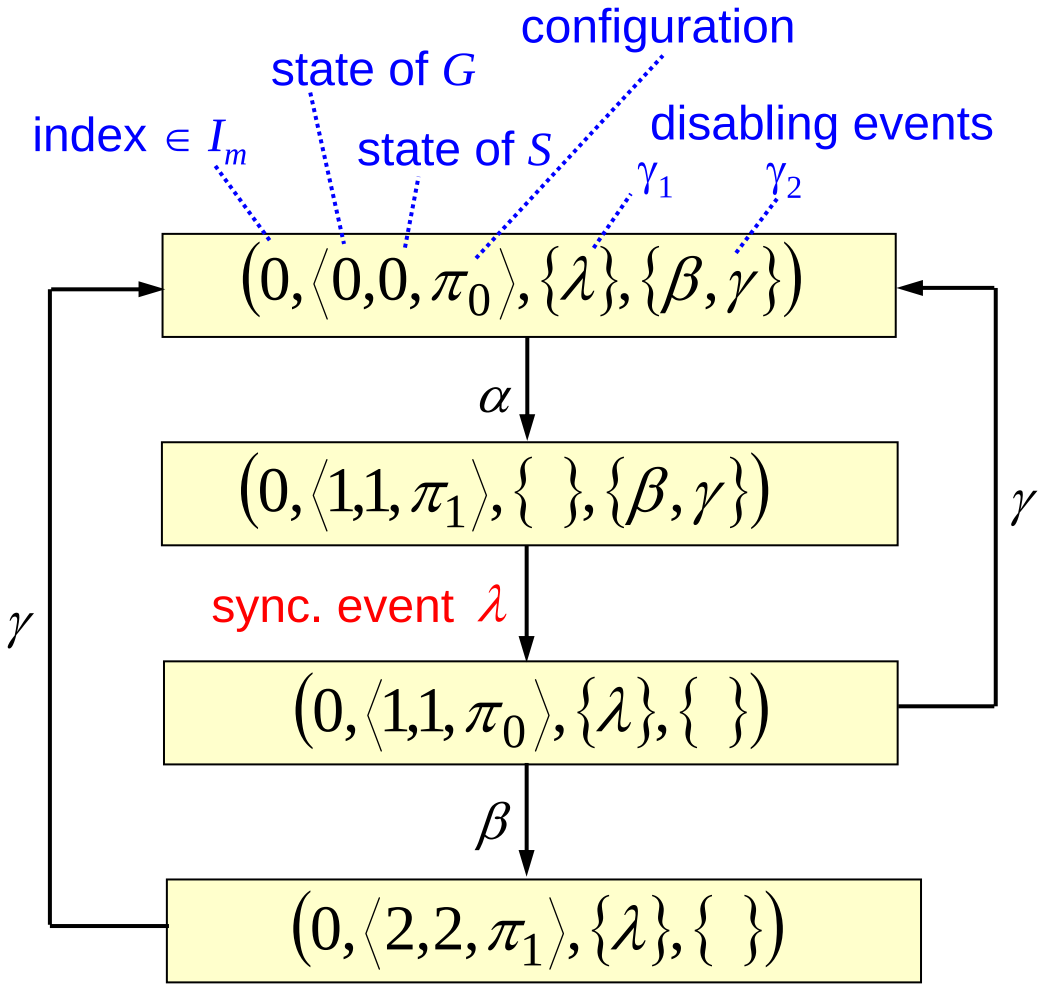


$L(G) = ((\beta^* + \gamma^*)\alpha(\gamma + \beta\gamma))^*$.
$Z = \{ C_1, C_2 \}$. $\Sigma_{o,1} = \{ \alpha \}$, $\Sigma_{c,1} = \varnothing$; $\Sigma_{o,2} = \{ \gamma \}$, $\Sigma_{c,2} = \{ \beta, \gamma \}$.
$E = (\alpha(\gamma + \beta\gamma))^*$.

Occurrence graph N

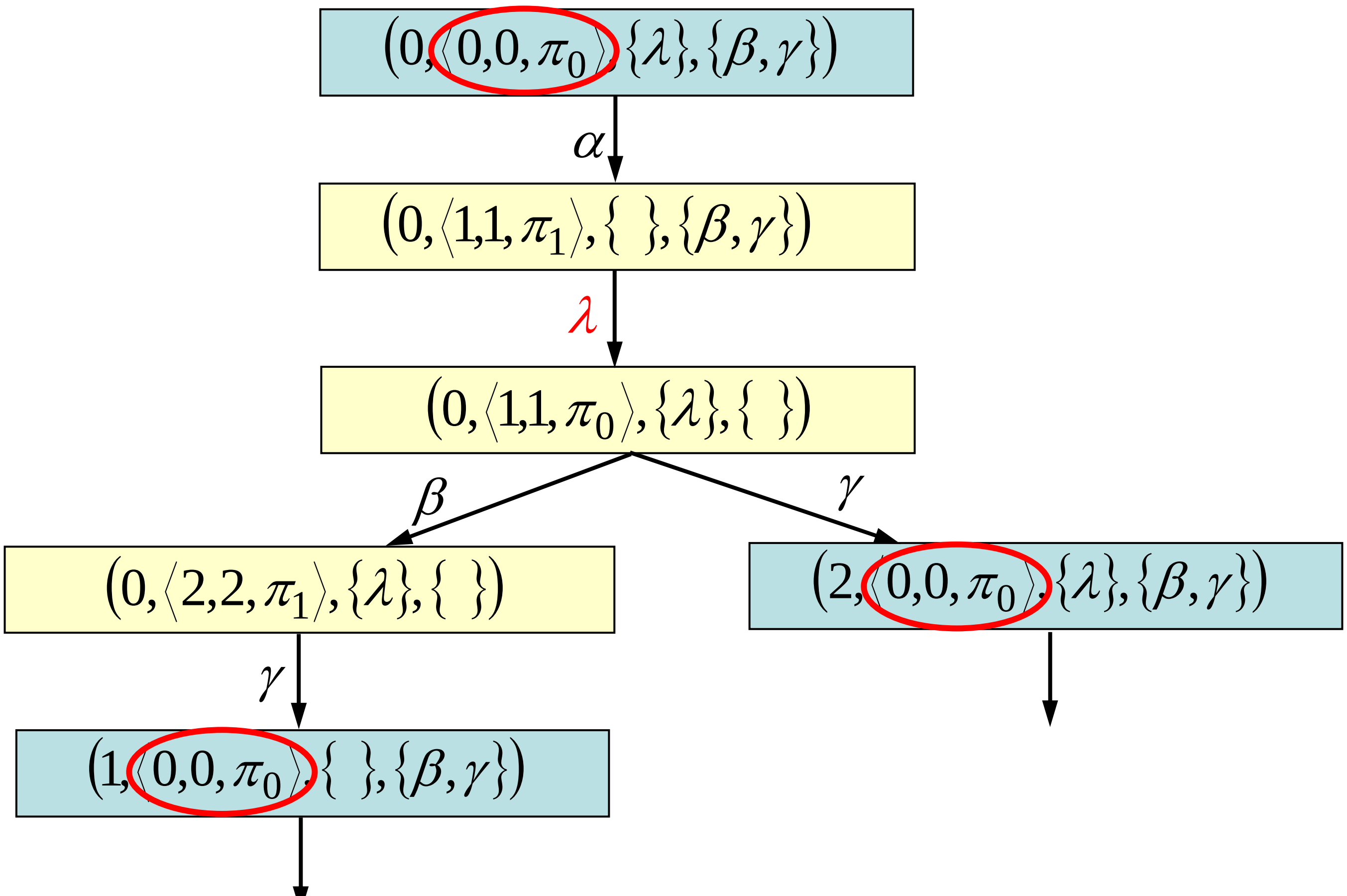
I_m := { 1, ..., m }: index set





Occurrence graph N

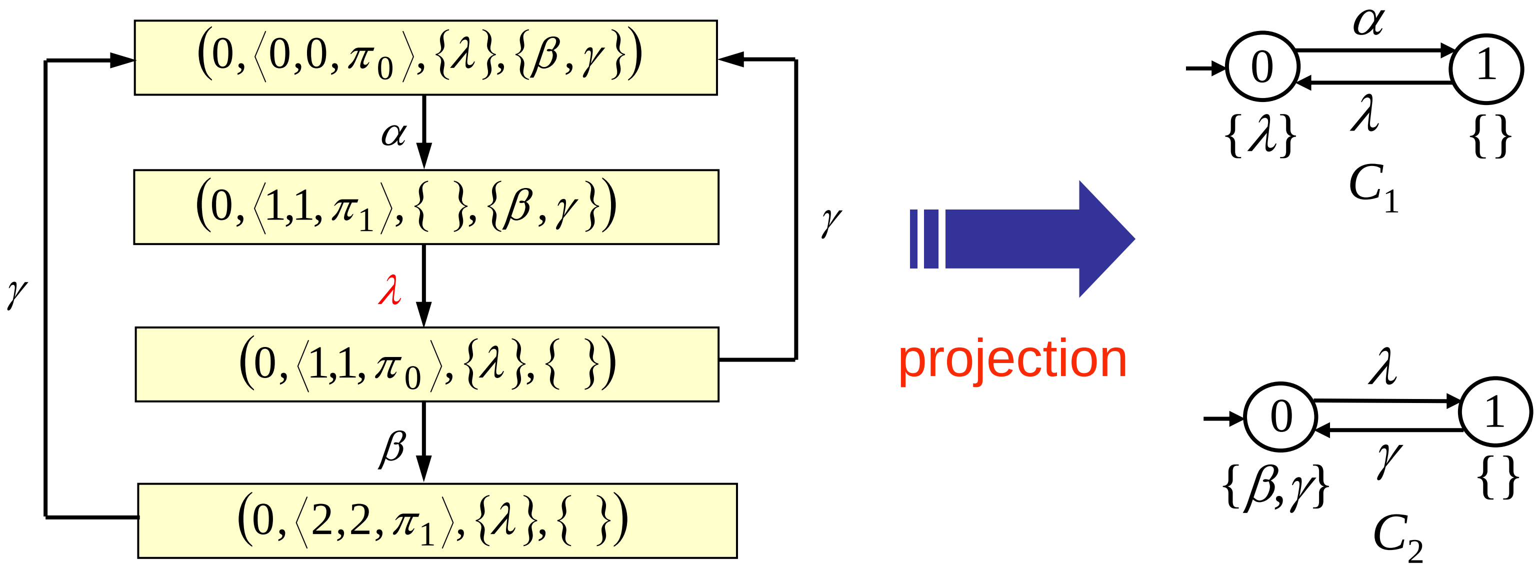
I_m is used for representing internal states.



Occurrence graph

- An occurrence graph N is called *consistent* if for any $s, s' \in L(N)$: $P_i(s) = P_i(s') \Rightarrow \gamma_i(s) = \gamma_i(s')$.
- An occurrence graph N is called *legal* if for any state $y = (k, < x, z, \pi >, \gamma_1, \gamma_2)$:
  - (i) If $\delta_G(x, \sigma)! \wedge \neg\delta_S(z, \sigma)!$, then $\neg\delta_N(y, \sigma)!$.
  - (ii) If $\delta_G(x, \sigma)! \wedge \delta_S(z, \sigma)!$, then there exists a finite sequence of sync. events u such that $\delta_N(y, u\sigma)!$.
- Lemma. N is legal if and only if N and S are Σ-bisimilar.
- Given a legal and consistent occurrence graph N, we can have finite-state controllers by projecting it.

Occurrence graph



Occurrence graph

- For a given index set I_m, the number of possible occurrence graphs is finite.
- Increasing m from 0, we can enumerate all occurrence graphs, and can check legality and consistency of them. That is, the set of all occurrence graphs is *a recursively enumerable set*.
- This implies that we can have a procedure to find finite-state controllers if they exist.

Communication behavior

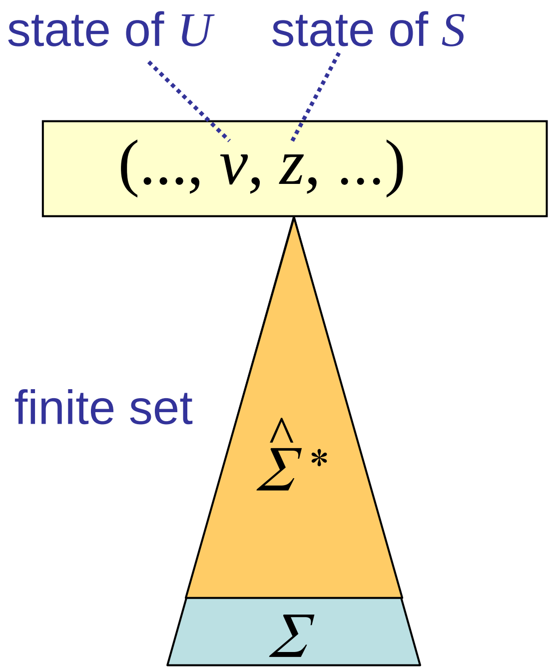
- On of the reasons why the undecidability arises is that the behavior on communication is unspecified in the problem.
  - Even if the languages of the plant and the specification are regular, the controlled behavior including communication may not.
- *A communication behavior* is a function that maps each trajectory observed so far to a set of synchronization events to be enabled.
- We assume the communication behavior of each C_i by a transition system W_i. Using W_i's and G, we construct a transition system U that represents the uncontrolled behavior including communication.

Communication behavior

- Then we solve a decentralized control problem without communication to find controllers such that S and U/Z are Σ-bisimilar.
- This problem is decidable if
  - the communication behavior is *rational*: it is given by a finite transition system that does not allow occurrence of infinite strings consisting only of synchronization events.
  - observation by each controller C_i does not *diverges*: $P^{-1}_i(P_i(s)) \cap L(G)$ is infinite for some $s \in L(G)$ (this condition may be dropped).

Idea of the proof

Similar to the language equivalence $L(G/Z) = E$, Σ- bisimulation can be checked locally and it does not require the system to be *nonblocking*.



These are sufficient for determining control actions.

Instances of rational communication

- State-estimation-based controllers: each controller tries to send the current state estimate to all other controllers after every observation of plant events.
- k-bounded-delay communication.

Further work

- Methods to compute finite controllers that are optimal in a sense that
  - reduction of communication,
  - reduction of the sizes.
- It is easy to expect that finding optimal solution is NP-hard.