SUPERVISORY CONTROL OF DISCRETE EVENT SYSTEMS: AN INTRODUCTION

W.M. Wonham
Systems Control Group
Dept. of Electrical & Computer Engineering
University of Toronto

June 1999

- $\star$ THEORY $\|$ DESIGN $\|$ SOFTWARE
- $\star$ COMPUTATIONAL EXAMPLE: DOSETANK
- $\star$ CONCLUSIONS

DISCRETE-EVENT SYSTEMS

- ⋆ Manufacturing workcells
- ⋆ Traffic systems
- ⋆ Database management
- ⋆ Communication protocols
- ⋆ Logistic (service) systems
- ⋆ Industrial (chemical) processes

Keywords:

- ⋆ EVENTS, CONDITIONS, SIGNALS

DES determine processes that are

- ⋆ discrete (time, state)
- ⋆ asynchronous (event-driven)
- ⋆ nondeterministic (generative)

TCT PROCEDURES

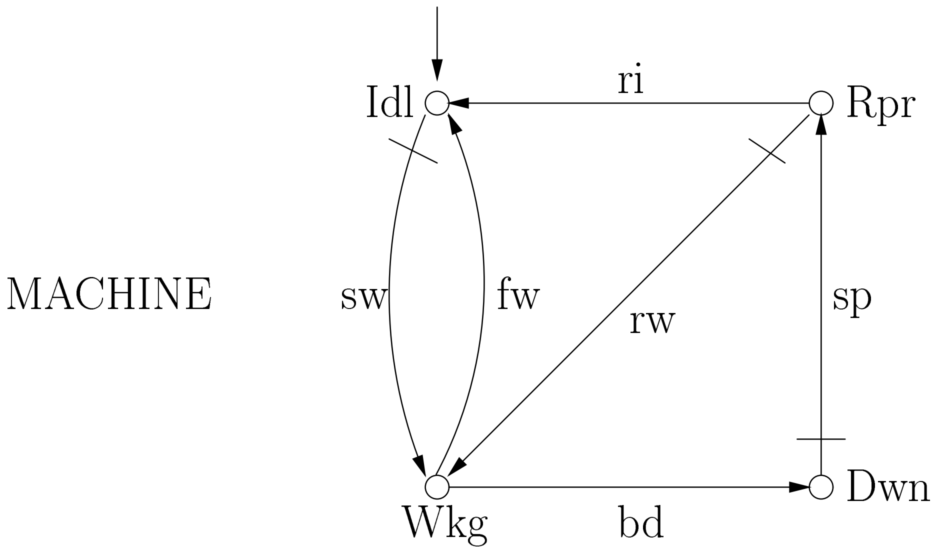
0: Create	P0: Project	N: Nonconflict
1: Selfloop	P1: Convert	I: Isomorph
2: Trim		
3: Sync	H0: Vocalize	E: Edit
4: Meet	H1: Outconsis	D: User file directory
5: Supcon	H2: Hiconsis	SE/SA/SX: Show DES/DAT/TXT
6: Mutex	H3: Higen	FE/FA: File DES/DAT
7: Condat	H4: Allevents	
8: Minstate		
9: Complement	R1: Supnorm	X: Exit to main menu

Procedure desired:

User directory: ICIT2000
Number files:

BASE MODEL

Finite Transition Structure



TCT "Create"

name: MACHINE # states: 4 state set: {Idl,Wkg,Dwn,Rpr}

init state: Idl marker state(s): Idl

controllable events: {sw,sp,rw} uncontrollable events: {fw,bd,ri}

transitions: 6

transitions:
[Idle, sw, Wkg] [Wkg, fw, Idl] ...

TCT FORMAT

TCT "Show"

MACHINE # states: 4 state set: 0 ... 3 initial state: 0

marker states: 0

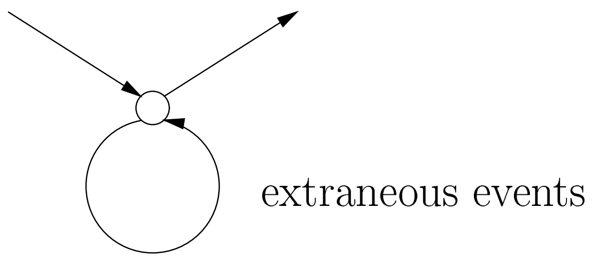
vocal states: none

transitions: 6

transitions:
[0, 1, 1] [1, 0, 0] [1, 2, 2] [2, 3, 3]
[3, 4, 0] [3, 5, 1]

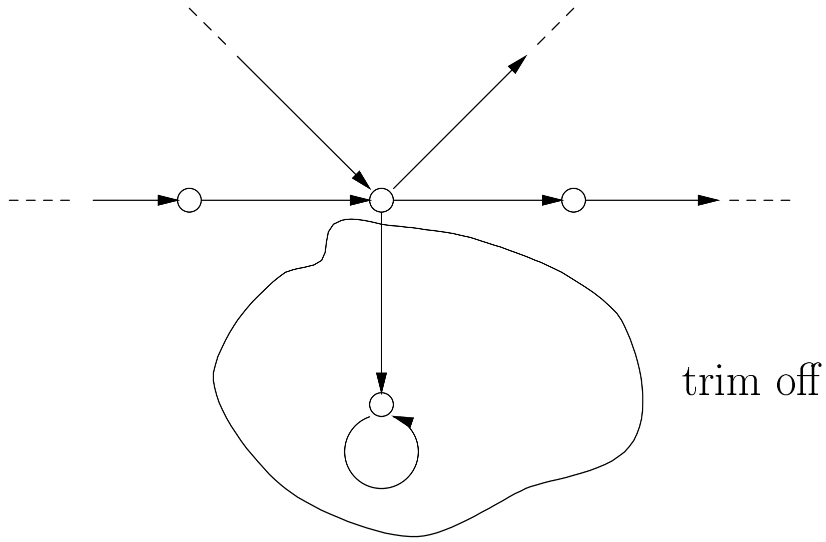
MODEL MODIFICATION

TCT "Selfloop"



"Selfloop" attaches extraneous events for compatibility with other DES.

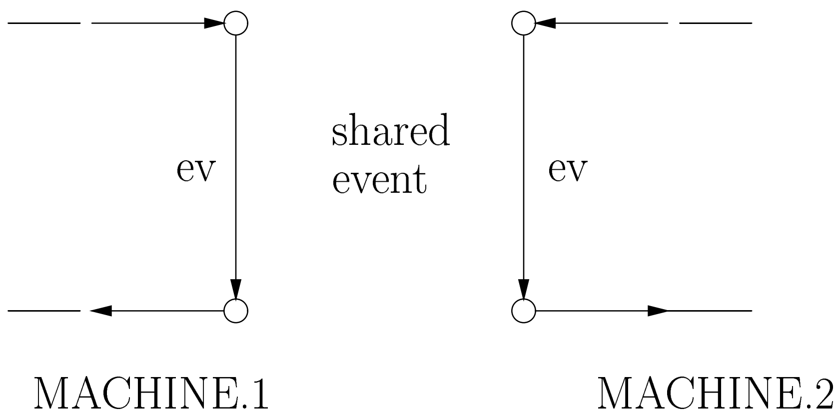
TCT "Trim"



"Trim" ensures marker states always remain reachable, by eliminating any non-coreachable states.

MODEL COMPOSITION

TCT "Sync"



Synchronous product couples several machines together by synchronizing any shared events:

MACHINE.3 = **sync**(MACHINE.1, MACHINE.2)

You can build up complex entities from simple ones:

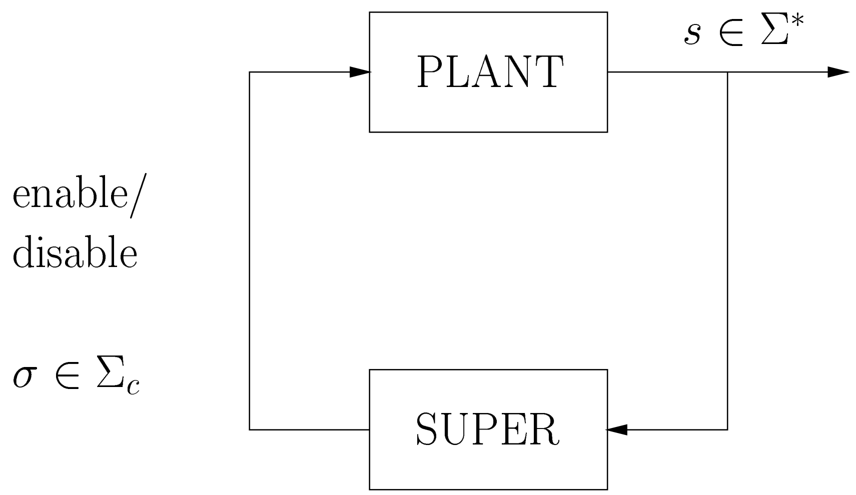
PLANT = **sync**(MACHINE.1, MACHINE.2, ..., MACHINE.N)

BASE CONTROL PROBLEM: FRAMEWORK

Alphabet of event labels: $\Sigma \;=\; \Sigma_c \dot{\cup} \Sigma_u$

Language environment: $\Sigma^* =$ all finite strings of symbols from Σ

Plant "behavior" is a sublanguage of Σ^*



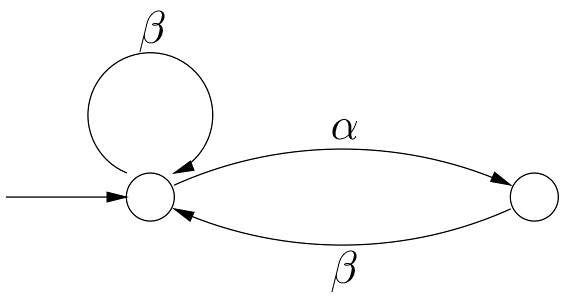
Supervisor SUPER enforces legal behavior of PLANT by enabling/disabling controllable events

BASE CONTROL PROBLEM: SPECIFICATION

Legal behavior   L(SPEC) $\subseteq \Sigma^*$

e.g. "If event α occurs, then event β must occur at least once before α may occur again."

SPEC.1



SPEC.1 = **selfloop**(SPEC.1,[list of extraneous events])

You can build up complex specifications from simpler ones by language intersection (= logical conjunction):

SPEC = **meet**(SPEC.1, SPEC.2, ..., SPEC.K)

BASE CONTROL PROBLEM: SOLUTION

Given: plant PLANT
specification SPEC

Find: supervisor SUPER such that

$$L_m(\text{SUPER}/\text{PLANT}) \subseteq L_m(\text{SPEC})$$

This problem is "optimally" solvable:

$$\text{SUPER} = \textbf{supcon}(\text{PLANT}, \text{SPEC})$$

TCT "Supcon" finds the supervisor which enforces specification on plant, while otherwise permitting plant maximal freedom of behavior.

MODULAR CONTROL PROBLEM: FRAMEWORK

Given: PLANT, SPEC.1, SPEC.2

Obtain SUP.1, SUP.2 to enforce SPEC.1, SPEC.2 respectively:

e.g. by intuition,
or by computing SUP.I = **supcon**(PLANT, SPEC.I) I = 1,2

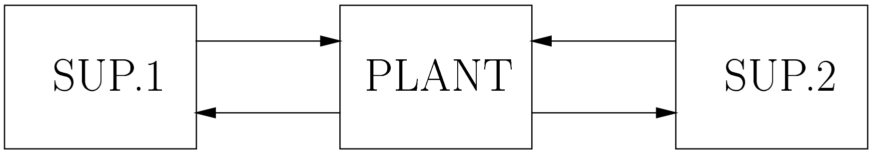
Use TCT "Condat" to check that SUP.I is controllable

Use TCT "Nonconflict" to check that SUP.1, SUP.2 are nonconflicting

If SUP.1 and SUP.2 pass these tests then you can take

MODSUP = **meet**(SUP.1, SUP.2)

as your solution.



The effect is that SUP.1 and SUP.2 are on line together.

Behavior of PLANT under control of MODSUP is

MODPLANT = **meet**(PLANT, MODSUP)

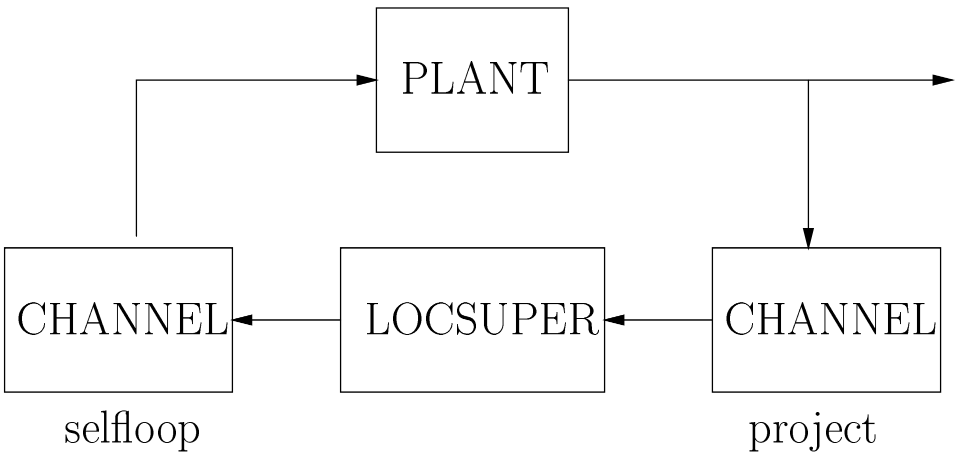
To check if this solution is optimal, first compute

SPEC = **meet**(SPEC.1, SPEC.2)
SUPER = **supcon**(PLANT, SPEC)

Then check whether

isomorph(MODPLANT, SUPER) = true

"LOCAL" CONTROL WITH PARTIAL OBSERVATIONS



NULL = list of events that local controller can't see.

TCT "Project" captures effect of Channel that erases NULL.

Observer's local model of PLANT is

$$\text{LOCPLANT} = \textbf{project}(\text{PLANT, NULL})$$

Given "global" specification SPEC.

Observer's local specification must be carefully computed:

$$\text{LOCSPEC} = \text{"}\textbf{observ}\text{"}(\text{SPEC, PLANT, NULL})$$

One way is to take "**observ**" to be TCT "Supnorm". In any case

$$\text{LOCSUPER} = \textbf{supcon}(\text{LOCPLANT, LOCSPEC})$$

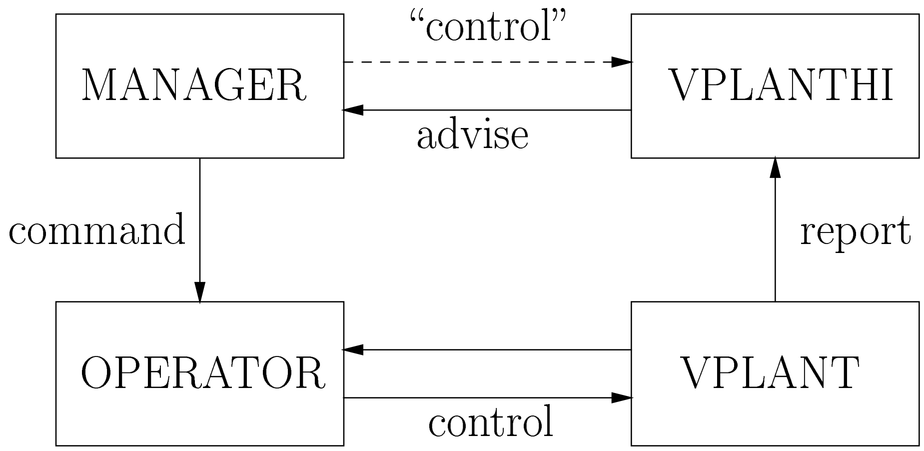
This is observer's local supervisor. Its global action is

$$\text{GLOBSUPR} = \textbf{selfloop}(\text{LOCSUPER, NULL})$$

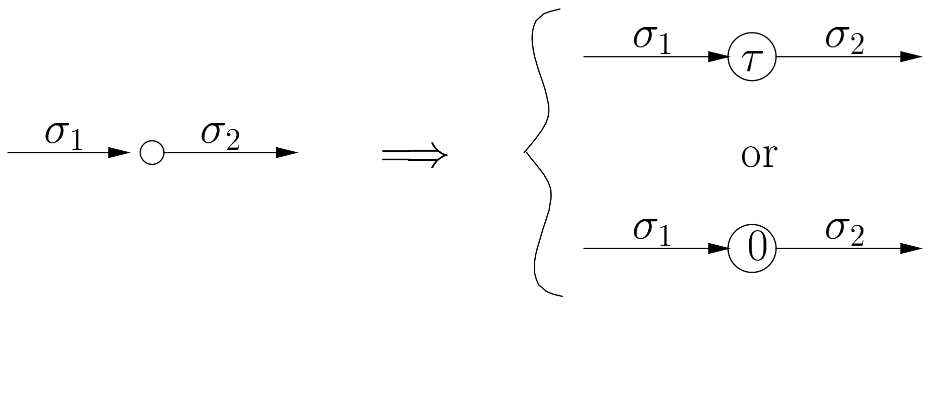
Check to see if it's nontrivial.

Check to see if it's optimal.

HIERARCHICAL CONTROL: FRAMEWORK



TCT "Vocalize" or "Edit"
PLANT $\Longrightarrow$ VPLANT



HIERARCHICAL CONTROL: DESIGN

Start with PLANT, operator's low-level model.

Identify events "significant" to manager.

Use TCT "Vocalize".

$$\text{VPLANT} = \textbf{vocalize}(\text{PLANT})$$

VPLANT need not provide a meaningful high-level model for control.

If necessary, refine it further:

$$\text{HCVPLANT} = \textbf{hiconsis}(\text{VPLANT})$$

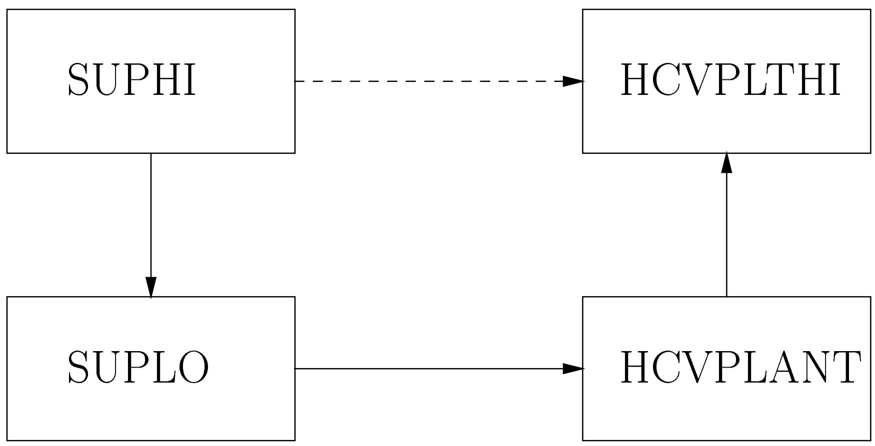
Then compute the manager's high-level model.

$$\text{HCVPLTHI} = \textbf{higen}(\text{HCVPLANT})$$

Proceed with control design as before,

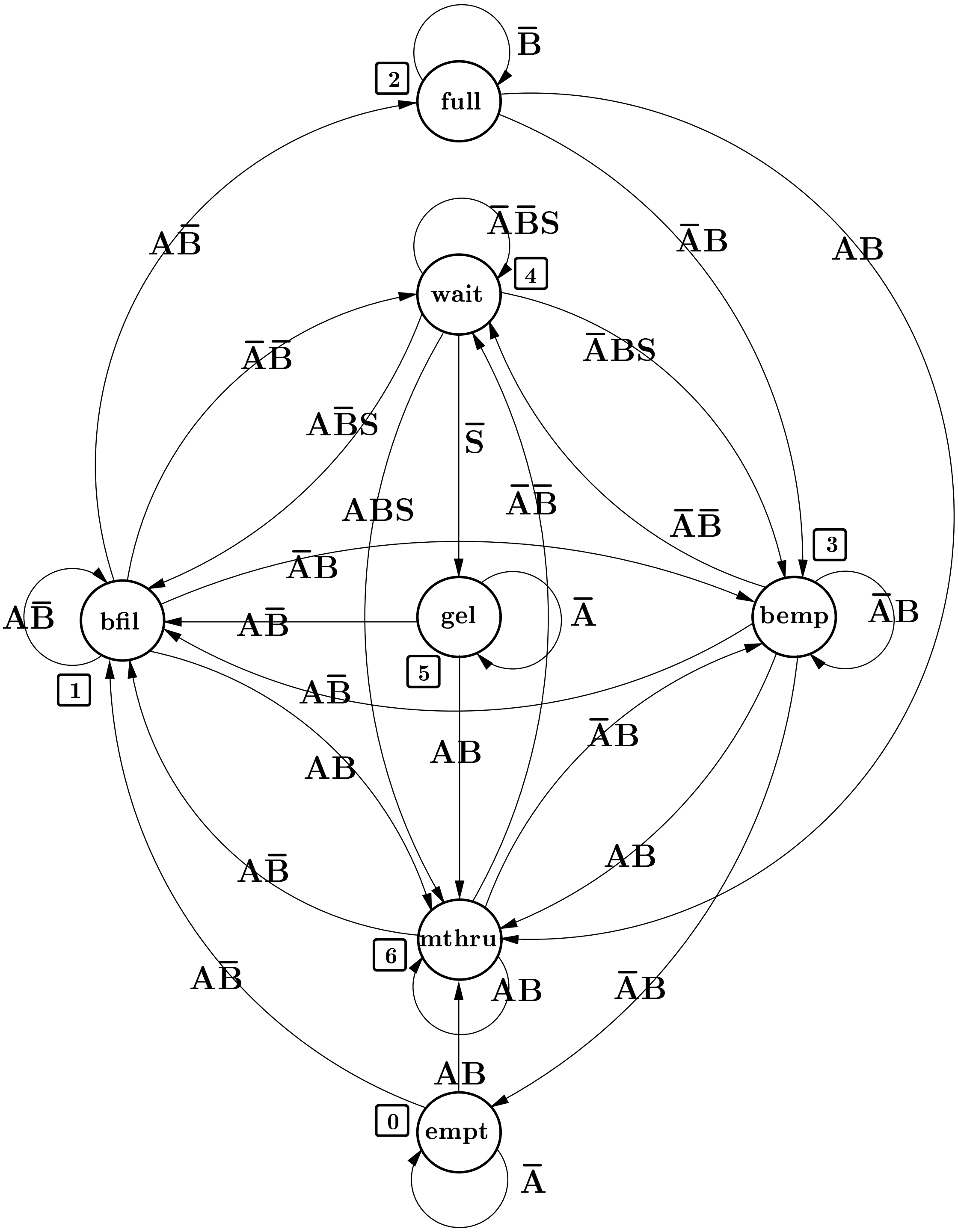
but now at the manager's level.

We hope design will be simpler to conceive and execute!



COMPUTATIONAL EXAMPLE: DOSETANK

Ref. Hanisch & Kowalewski, 1994



DOSETANK: STATES & EVENTS

Tank is equipped with VALVEA, VALVEB, STIRRER, which can be turned on/off by OPERATOR.

In each of OPERATOR's 8 states, exactly 1 event (label) can occur in DOSETANK.

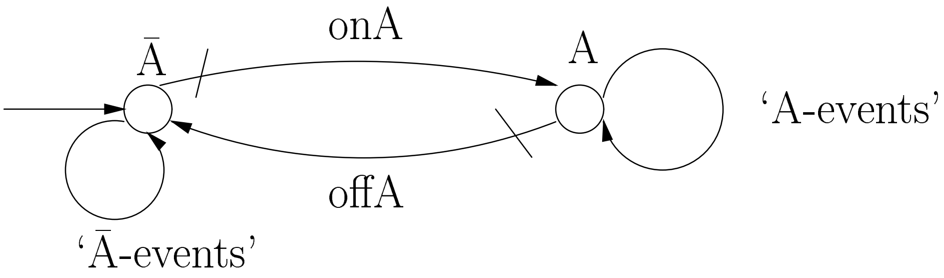
DOSETANK events labelled according to state of (VALVEA, VALVEB, STIRRER).

DOSETANK state		OPERATOR state	DOSETANK event
H&K	TCT	H&K	TCT
empt	0	$\bar{A}\bar{B}\bar{S}$	0
bfil	1	$\bar{A}\bar{B}S$	2
full	2	$\bar{A}B\bar{S}$	4
bemp	3	$\bar{A}BS$	6
wait	4	$A\bar{B}\bar{S}$	8
gel	5	$A\bar{B}S$	10
mthru	6	$AB\bar{S}$	12
		ABS	14

DOSETANK = **create**(DOSETANK)

DOSETANK: CONTROLS

VALVEA



TCT Coding: onA: 101; offA: 103; 'A-events': 0,2,4,6; '$\bar{A}$-events': 8,10,12,14

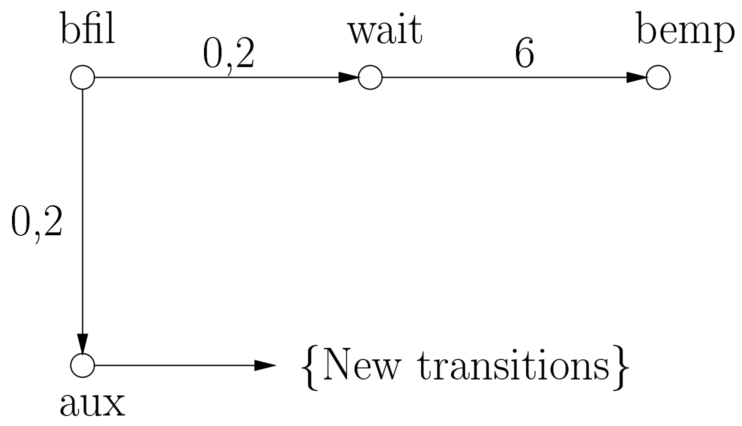
VALVEB, STIRRER: similar

OPERATOR = **sync**(VALVEA, VALVEB, STIRRER) (8,8)

PLANT = **sync**(DOSETANK, OPERATOR) (56,209)

DOSETANK: SPECIFICATIONS (H&K)

Specification 1: "Prohibit state sequence bfil - wait - bemp"



SPEC1 = **edit**(DOSETANK)

{create a modified copy of DOSETANK under name SPEC1}

In Editor, delete transitions: [1,0,4], [1,2,4]
adjoin transitions: [1,0,7], [1,2,7]
[7,0,5], [7,4,5], [7,8,5]
[7,10,1], [7,12,5], [7,14,6]

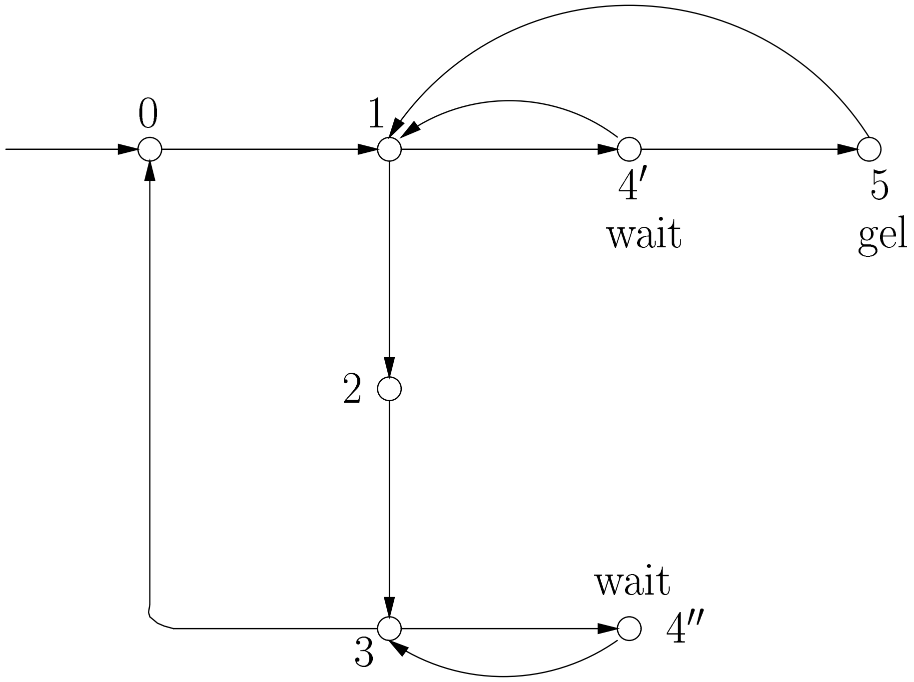
Specification	Prohibit
SPEC1	bfil-wait-bemp
SPEC2	bemp-wait-bfil
SPEC3	bemp-wait-gel
SPEC4	mthru
SPEC5	bfil-bemp
SPEC6	bemp-bfil

SPEC = **meet**(SPEC1, SPEC2, ..., SPEC6)
SPEC = **selfloop**(SPEC, [control events onA,offA,onB,offB,onS,offS])

DOSETANK: SOLUTION

SUPER = **supcon**(PLANT, SPEC) (32,84)

PSUPER = **project**(SUPER, [onA,offA,onB,offB,onS,offS]) (7,18)



PSUPER satisfies SPEC.

But DOSETANK may still enter gel!

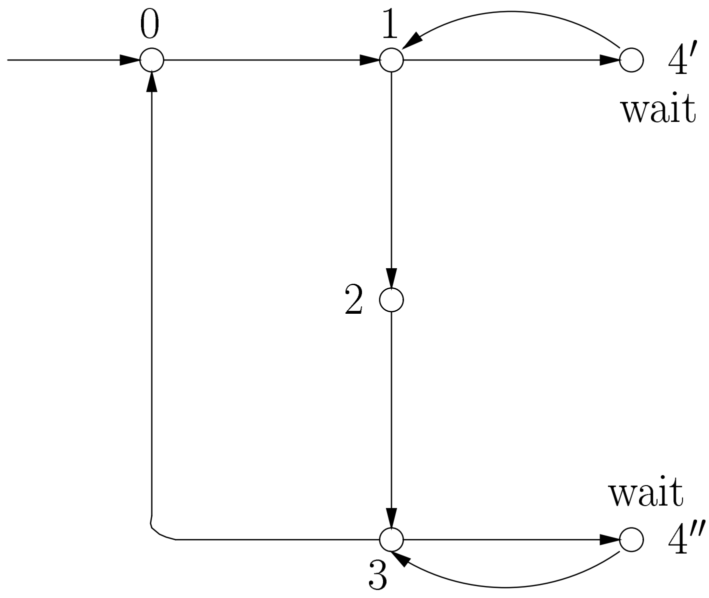
DOSETANK: NEWSPEC

To prohibit entrance of DOSETANK to state gel, adjoin SPEC7.

NEWSPEC = **meet**(SPEC, SPEC7) (6,50)

NEWSUP = **supcon**(PLANT, NEWSPEC) (22,52)

PNEWSUP = **project**(NEWSUP, [onA,...,offS]) (6,12)



Never enters gel

Events corresponding to turning off stirrer in 'wait' now disabled.

DOSETANK: CYCSPEC

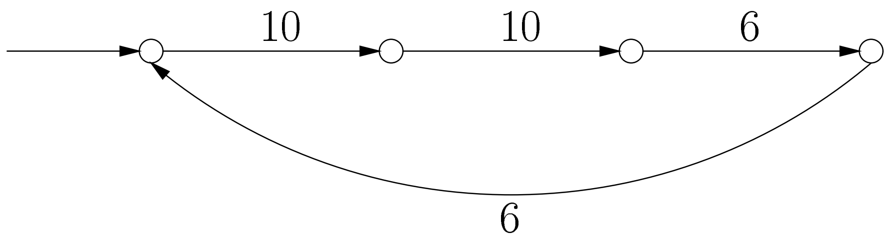
Specify a "standard repetitive work cycle":

state sequence: empt - bfil - full - bemp - empt

via event sequence: 10, 10, 6, 6

$$\text{CYCSPEC} = \textbf{create}(\text{CYCSPEC})$$

CYCSPEC

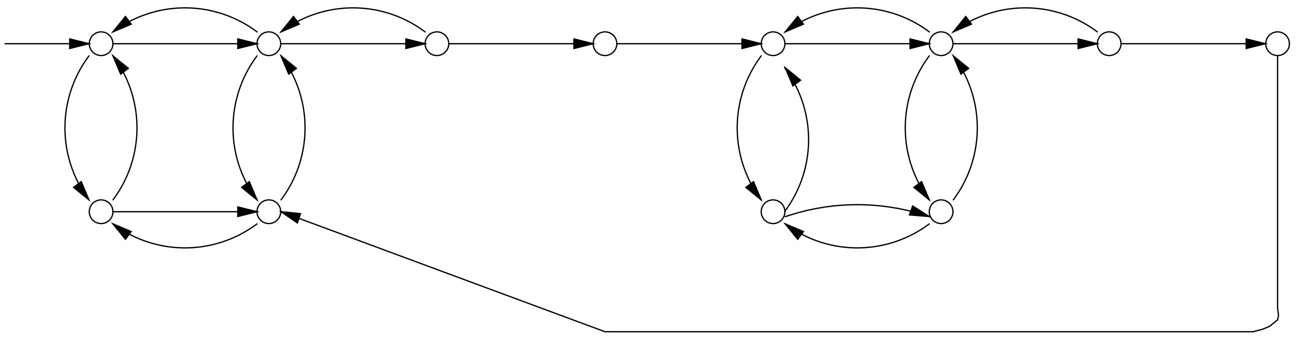


$$\text{CYCSUP} = \textbf{supcon}(\text{PLANT, CYCSPEC})\ (12{,}24)$$

Relatively complicated structure, because we haven't prevented the operator from flipping switches on and off to no purpose, generating futile combinatorics.

Need "intelligence" to prune!

CYCSUP



DOSETANK: MODULAR SUPERVISION, PARTIAL OBSERVATIONS

Attempt modular control, to satisfy SPEC4 alone,

i.e. exclude DOSETANK state mthru.

Solution:

$$\text{SUP4} = \textbf{supcon}(\text{PLANT, SPEC4}) \quad (36,107)$$

Simplify?

Notice (from SUP4.DAT) that SUP4 never controls STIRRER!

So, project out control events onS, offS,
using procedures for control under partial observations.

In this case, it suffices to project onS, offS from SUP4:

$$\text{OBSUP4} = \textbf{project}(\text{SUP4, [onS, offS]}) \quad (18,47)$$

By local control with partial observations, we've cut the complexity of SUP4 in half.

DOSETANK: HIERARCHICAL CONTROL

Manager's significant events:

Enter DOSETANK states empt, full, gel

Associate high-level events:

hiempt, hifull, higel (90, 92, 94)

Modify PLANT for information flow to Manager:

VPLANT = **edit**(PLANT, Vocalize appropriate PLANT states)

Manager's high-level model:

VPLANTHI = **higen**(VPLANT) (7,15)

VPLANT can't be used for control as is, since not hierarchically consistent.

To achieve consistency, compute Operator's refined plant model:

HCVPLANT = **hiconsis**(VPLANT) (74,274)

HCVPLANT is about 30% more complex than VPLANT.

TCT "Hiconsis" has introduced auxiliary states and events.

Advantage: HCVPLANT can be used for any high-level control specification!

DOSETANK: HIERARCHICAL CONTROL (concl.)

Manager's plant model for control:

HCVPLTHI = **higen**(HCVPLANT) (39,300)

Manager's specification: "Exclude events corresponding to 'enter state gel'" (i.e. events 940, 941)

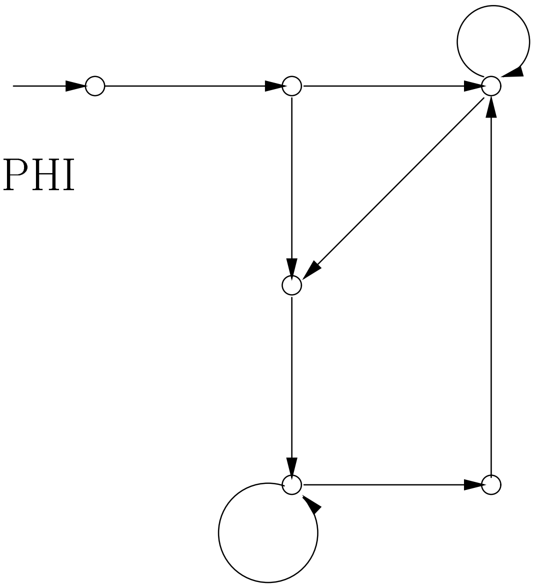
SPECHI = **create**(SPECHI) (1,30)

Manager's supervisor:

SUPHI = **supcon**(HCVPLTHI, SPECHI) (27,187)

To see what's going on, use TCT "Project" and "Convert" to project out the auxiliary events and fuse the controllable and uncontrollable siblings of hiempt and hifull (900,901 fuse back to 90; 920,921 fuse back to 92):

CPSUPHI



CSUPHI is just the basic work cycle

empt - full - empt

as we might expect.

CONCLUSIONS

Advantages of the approach:

SYNTHETIC

RESULTS CORRECT BY CONSTRUCTION

MODULAR & HIERARCHICAL ARCHITECTURE FOR CONTROL OF COMPLEXITY

EASY TO USE

Disadvantages:

EXPONENTIAL STATE EXPLOSION WITH INCREASING NUMBERS OF PLANT COMPONENTS

RESULTS MAY INCLUDE UNNECESSARY COMBINATORIAL DETAIL

RESULTS MAY REQUIRE "INSIGHT" TO INTERPRET

ABSTRACTION BY PROJECTION IS EXPONENTIALLY COMPLEX

FURTHER DEVELOPMENTS

Timing and forcing (Brandin 1994 ...)

Top-down hierarchical supervision (Wang 1995, Gohari 1998, Ma 1999 ...)

Implementation via PLCs (Leduc 1995 ...)

Mixed types (automata, integer, boolean,...) (Chen 1996 ...)

Model reduction approaches (Lawford 1997 ...)

Nonblocking heterarchical control (Wong 1997)