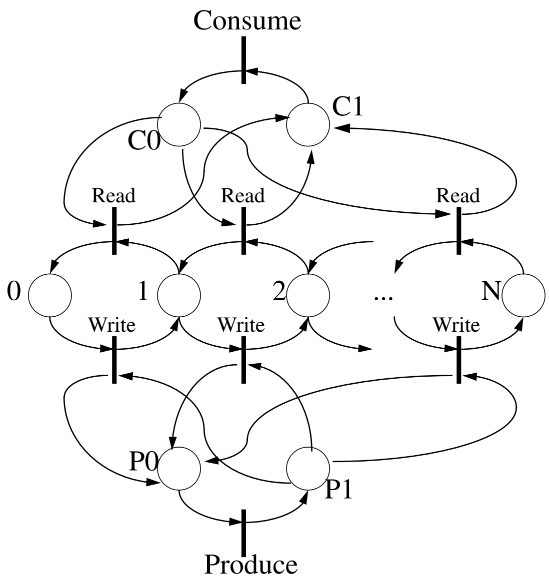
Petri-net for Buffered Consumer-Producer



Buffered Producer-Consumer model:
As long as the buffer is not empty, the consumer can read an input and consume it. As long as the buffer is not full, the producer can produce an output and write it.

Formal Specifications

Descriptive Specifications - State desired properties of system in a purely declarative way.

Operational Specification - Describe desired behaviour, usually by providing a model of the system. Model is an abstract device that in some way can simulate its behaviour.

Examples:
Input-output assertions P{Q}R
Algebraic specifications

Algebraic Specifications

Uses: (1) Set of objects
(2) Set of operations
(3) Axioms specifying behaviour of operations

Two parts to specification
Syntax
Axioms

Example: STACK

intuitive definitions of stack operations
NEW creates a new stack
PUSH adds a new item to top of stack
TOP returns a copy of top item
POP removes top item
EMPTY tests for an empty stack

Syntax

Operation	Domain		Range
NEW	()	→	STACK
PUSH	(STACK,ITEM)	→	STACK
POP	(STACK)	→	STACK
TOP	(STACK)	→	ITEM
EMPTY	(STACK)	→	BOOLEAN

Axioms:

(stk is of type STACK, itm is of type ITEM)

(1) EMPTY(NEW) = true
(2) EMPTY(PUSH(stk, itm)) = false
(3) POP(NEW) = error
(4) TOP(NEW) = error
(5) POP(PUSH(stk, itm)) = stk
(6) TOP(PUSH(stk, itm)) = itm

Axioms stated in english:

1 A new stack is empty
2 A stack is not empty immediately after pushing an item onto it
3 Attempting to pop a new stack results in an error
4 There is no top item on a new stack
5 Pushing an item onto a stack and immediately popping it off leaves the stack unchanged
6 Pushing an item onto a stack and immediately requesting the top item returns the item just pushed onto the stack

Canonical Form

A particular stack written in terms of the operations used to form it:

d
c
b
a

stk = PUSH(PUSH(PUSH(PUSH(NEW,a), b), c), d)

TOP(POP(stk)) =
TOP(POP(PUSH(PUSH(PUSH(PUSH(NEW,a), b), c), d))) =
TOP(PUSH(PUSH(PUSH(NEW,a), b), c)) = c

Extending the specifications

Suppose we want to add an operator REPLACE which replaces the top item of a stack with its *itm* argument.

Syntax:
REPLACE: (stk,itm) → stk

Axioms:
REPLACE(stk, itm) =
if EMPTY(stk) then
    error
else
    PUSH(POP(stk, itm))

or

REPLACE(NEW, itm) = error
REPLACE(PUSH(stk, old-itm), itm) = PUSH(stk, itm)

Axiomatic specification of a queue

Syntax:

Operation	Domain		Range
NEW	()	→	QUEUE
ADD	(QUEUE, ITEM)	→	QUEUE
REMOVE	(QUEUE)	→	QUEUE
FRONT	(QUEUE)	→	ITEM
EMPTY	(QUEUE)	→	BOOLEAN

Axioms:
(1) EMPTY(NEW) = true
(2) EMPTY(ADD(que, itm)) = false
(3) FRONT(NEW) = error
(4) REMOVE(NEW) = error
(5) FRONT(ADD(que, itm)) =
if EMPTY(que) then
    itm
else
    FRONT(que)
(6) REMOVE(ADD(que, itm)) =
if EMPTY(que) then
    NEW
else
    ADD(REMOVE(que), itm)