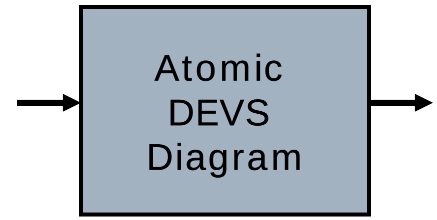
ECE575/ Appendix 1. Atomic DEVS Diagrams

Moon Ho Hwang
mhhwang@ece.arizona.edu

2006 Fall

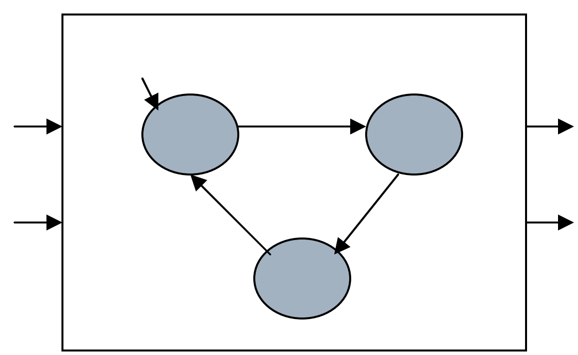
ECE Department,
University of Arizona

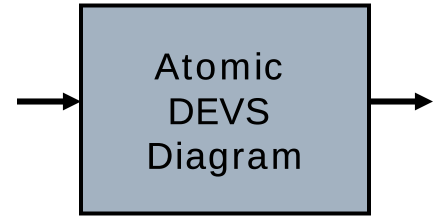
Motivation



- To show an atomic DEVS *A*, we can we usually define three sets and four functions of $A = \langle X, Y, S, ta, \delta_{ext}, \delta_{int}, \lambda \rangle$.
- However, if we can have a diagram which is equivalent to A, it can be more intuitive than the 7-tuple definition.

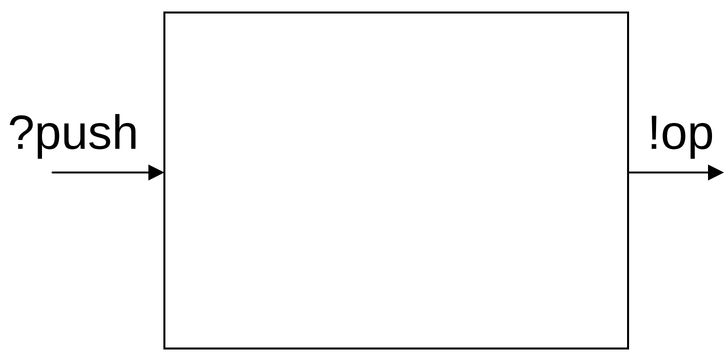
$A = \langle X, Y, S, ta, \delta_{ext}, \delta_{int}, \lambda \rangle$ $\Leftrightarrow$

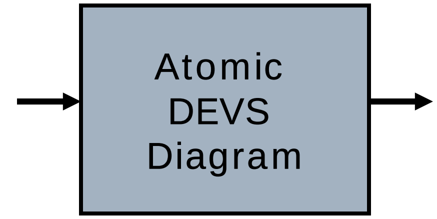




Events

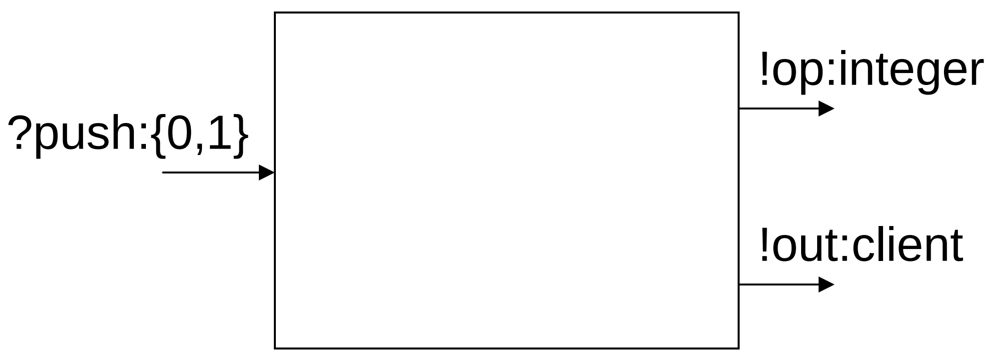
- Events can be represented as (Port, Value).
- If we **don't** use value, event is represented by **port only**. For example, ?push and !op



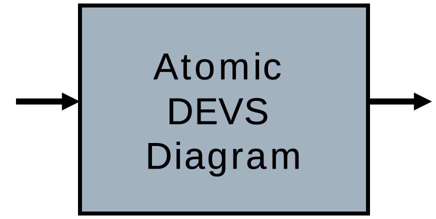


Events (continued.)

- If we use (event, value), event is represented by **port:value_type**. For example, ?push:{0,1} and !op:integer, !out:client where client is a value type.



State and Phase

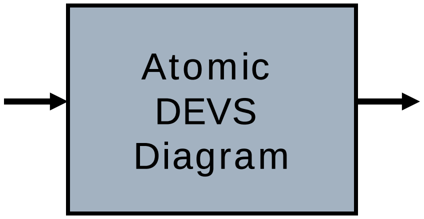


- The state set can be represented as cross product the *phase* set and *other variable*.
  - For example,

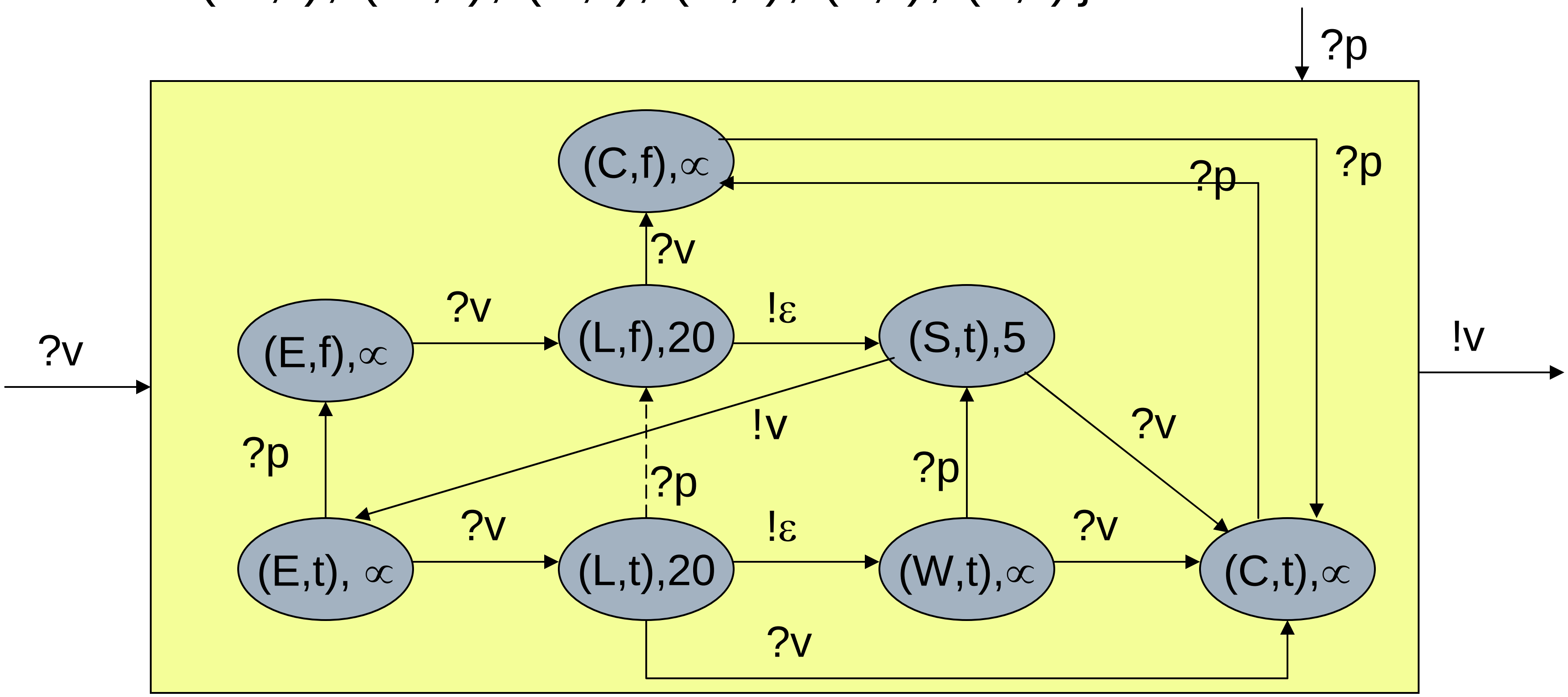
    Phase={Empty (E), Loading (L), Sending (S), Waiting (W), Collided (C), Fault (F)} and next_occupied ={false, (f), true (t)},

S=Phase × next_occupied;

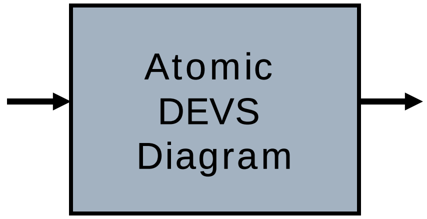
State Transition Diagram



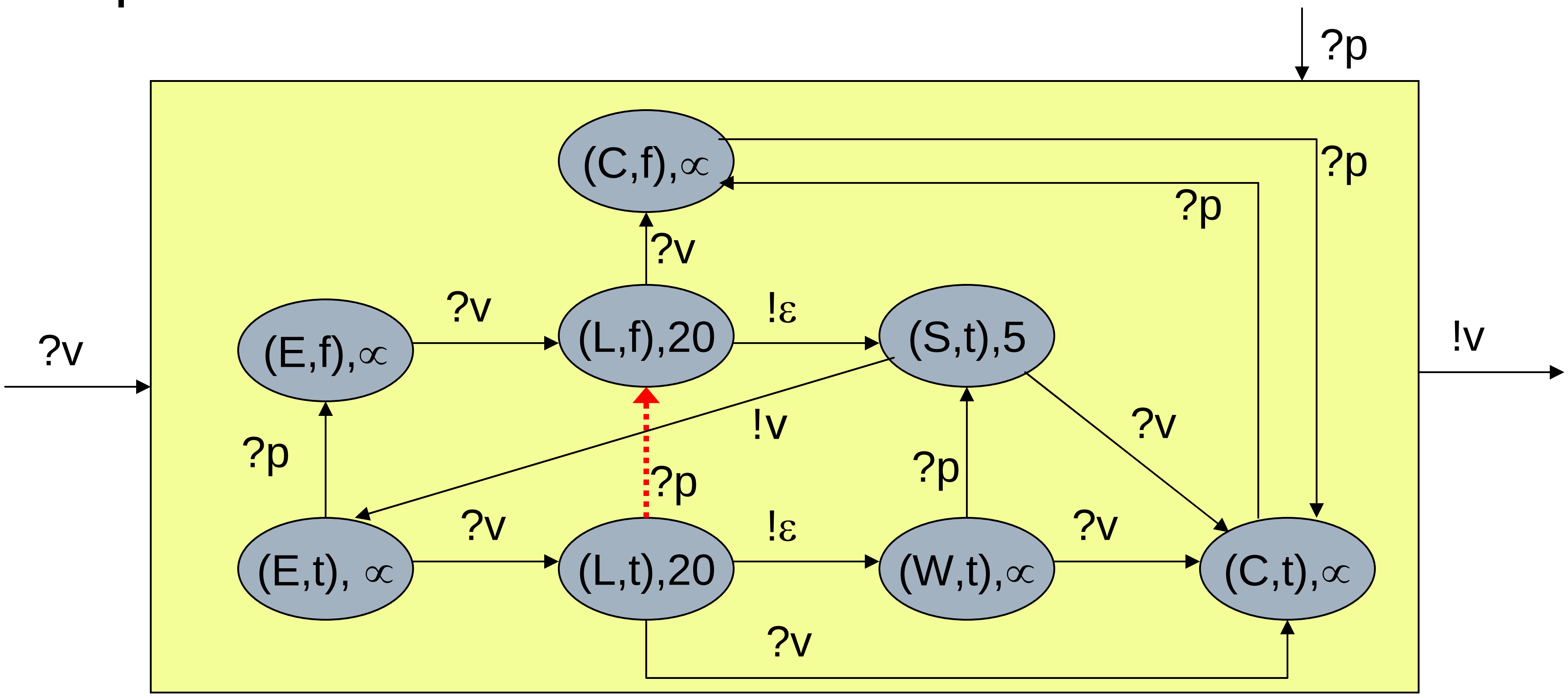
- A node indicates a *state s* with *ta(s)*, and an arc stands for either δ_{ext} or δ_{int} with λ.
  - For example, S={(E,f), (E,t), (L,f), (L,t), (S,f), (S,t), (W,f), (W,t), (C,f), (C,t), (F,f), (F,t)}



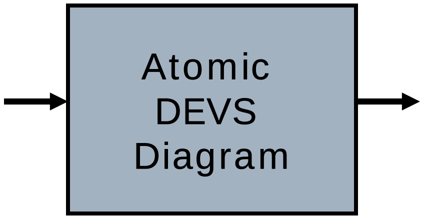
State Transition Diagram



- A dashed arc for δ_{ext} represents *continuing schedule* case such that $\delta_{ext}((L,b),te,?p)=(L,i)$ changes *b* to *i* but the next scheduled time is preserved.



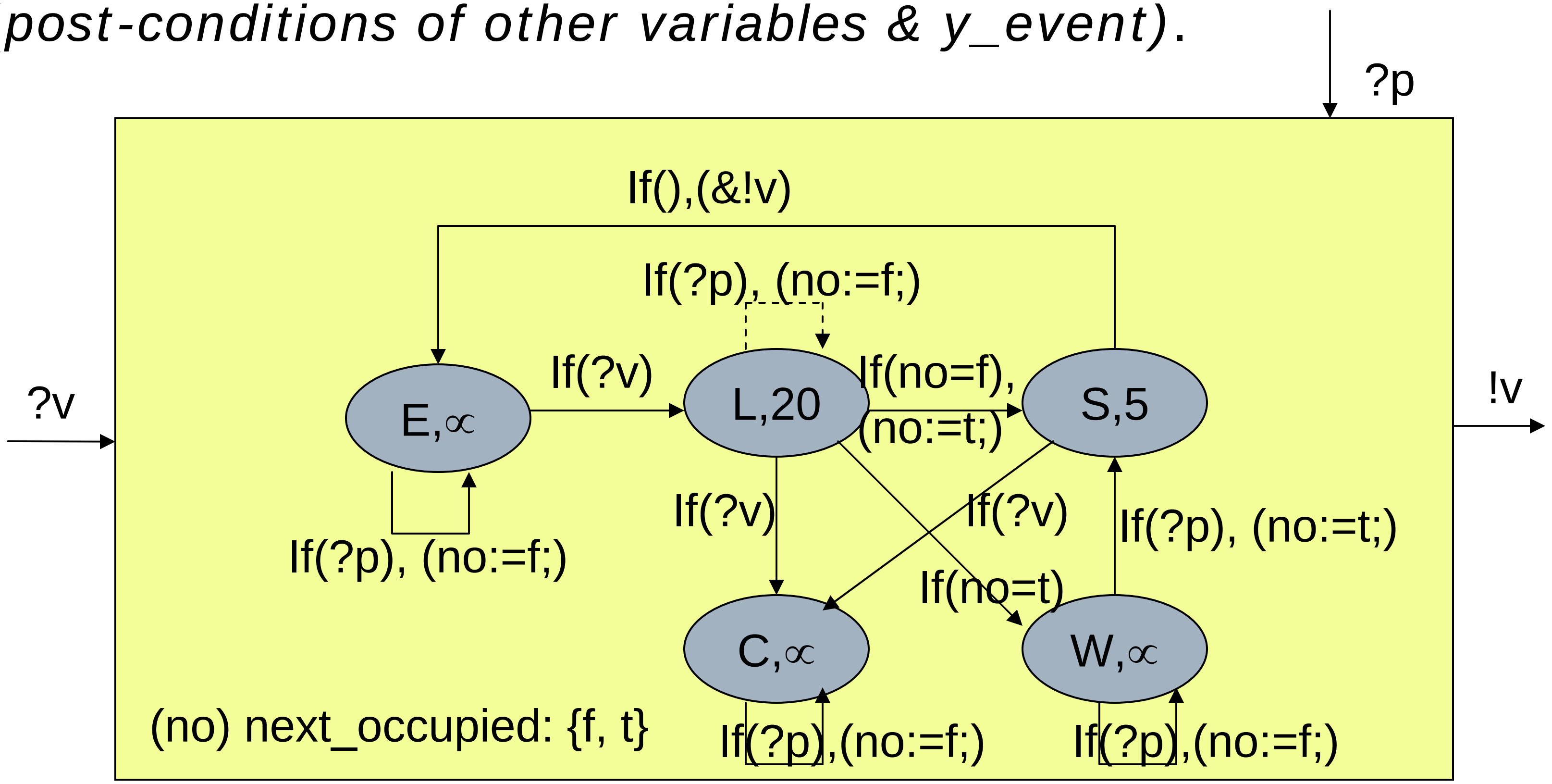
Phase Transition Diagram



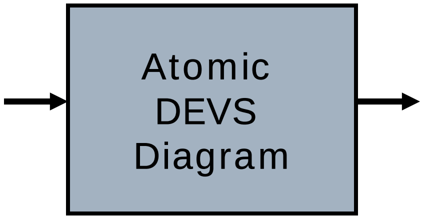
- A node indicates a *phase p* with *ta(s)*, and an arc stands for a *phase transition* with

If(x_event & *pre-conditions of other variables*),

(*post-conditions of other variables & y_event)*.



Discussion



- Generally, the state transition diagram is useful if the transition logic can be clear with only on state variable like phase. But it can be very complicated if the rule is based on more than two state variables.
- In the meanwhile, the phase transition diagram reduces the number of states so it can be useful in case that the number of state variables are many. However, the user makes sure that the *pre-conditions of arcs* coming from the same phase should be *mutually exclusive*.
- Sometime it seems to be impossible to describe all logics of atomic DEVS using any diagrams.
- However, both diagrams of state transition and phase transition give us *insight* of its dynamics, even though it is not perfect.