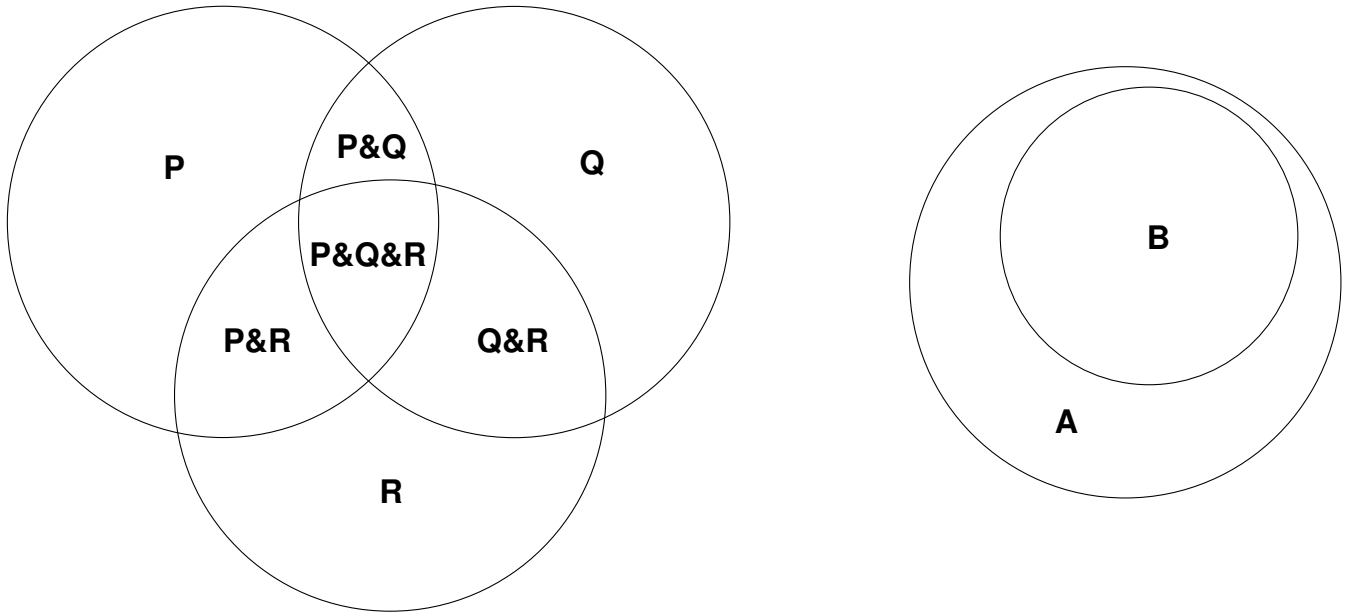
Statecharts aka Harel Charts

- visual formalism
- application of higraphs
- diverse applications;
  in particular: concurrent, reactive systems behaviour modelling

Visualising Information

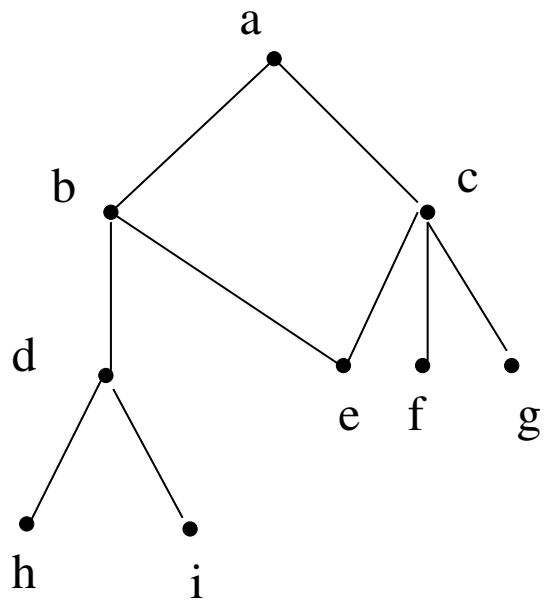
- complex
- non-quantitative, structural
- topological, not geometrical
- Euler
  - Venn diagrams (Jordan curve: inside/outside): enclosure, intersection
  - graphs (nodes, edges: binary *relation*); hypergraphs

Venn diagrams, Euler circles

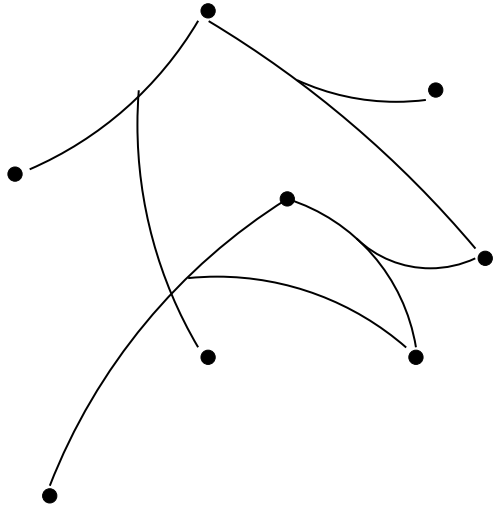


- *topological* notions:
  enclosure, intersection, exclusion
- Used to represent *mathematical set operations*:
  union, intersection, difference

Hypergraph



a graph



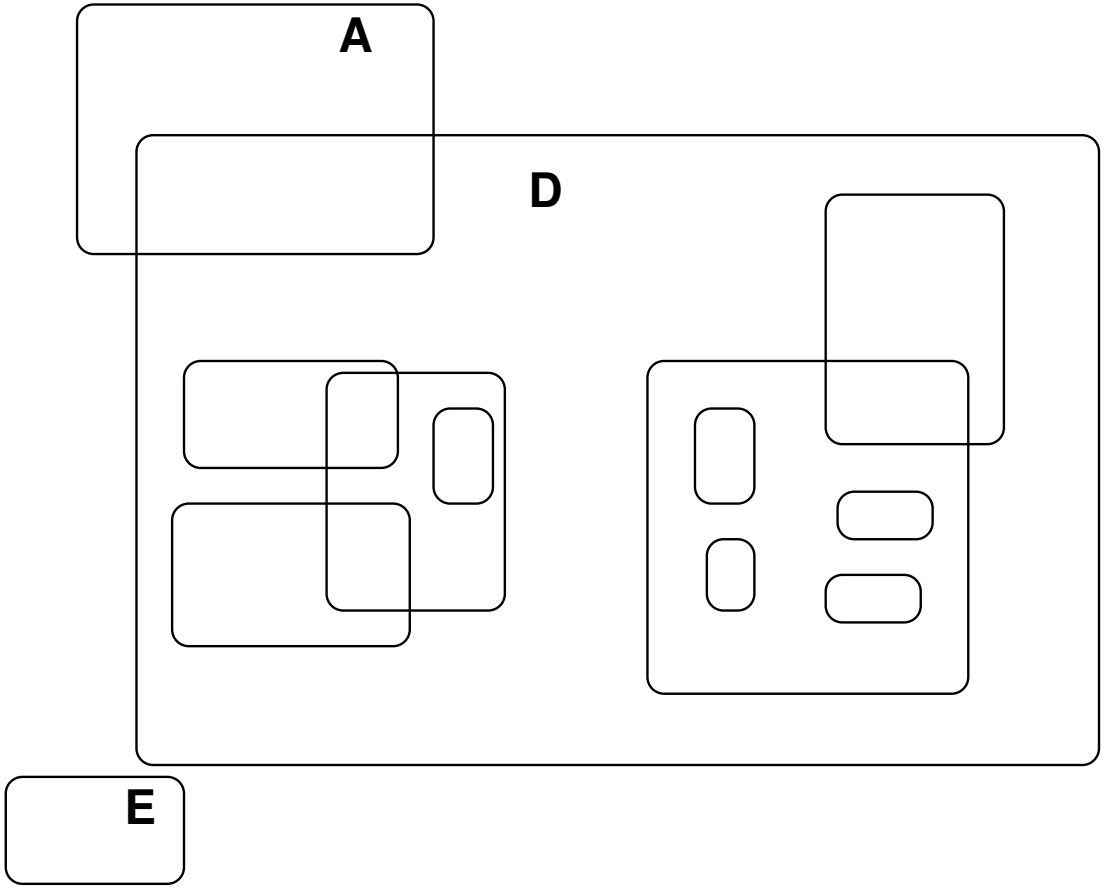
a hypergraph

- *topological* notion: connectedness
- Used to represent *relations* between sets.
- Hyperedges: non longer binary relation ($\subseteq X \times X$): $\subseteq 2^X$ (undirected), $\subseteq 2^X \times 2^X$ (directed).

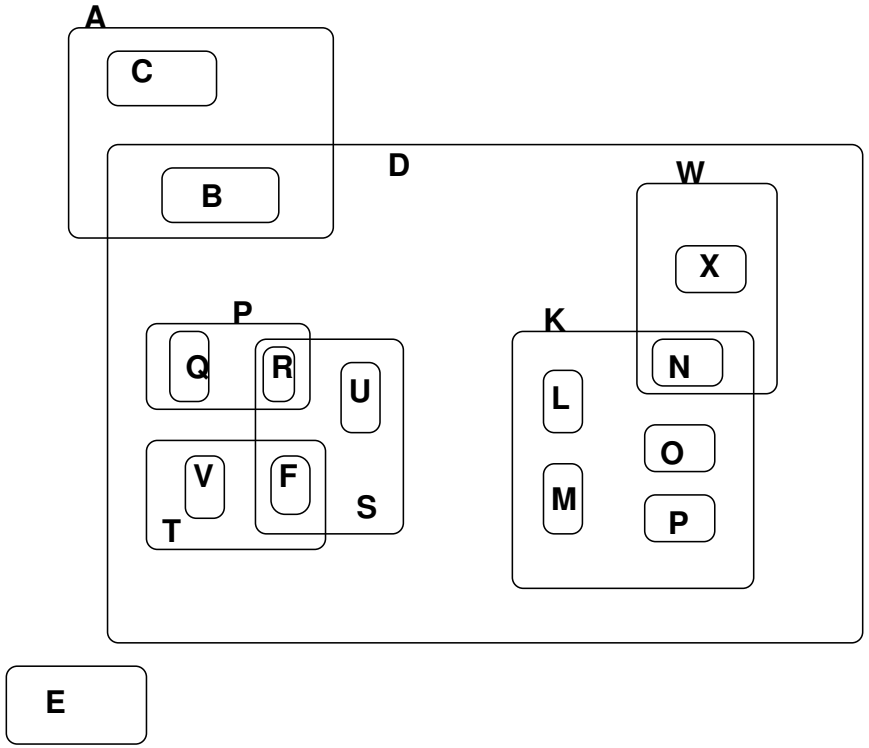
Higraphs: combining hypergraphs and Venn diagrams

- sets + cartesian product
- hypergraphs

Blobs: set *inclusion, not membership*

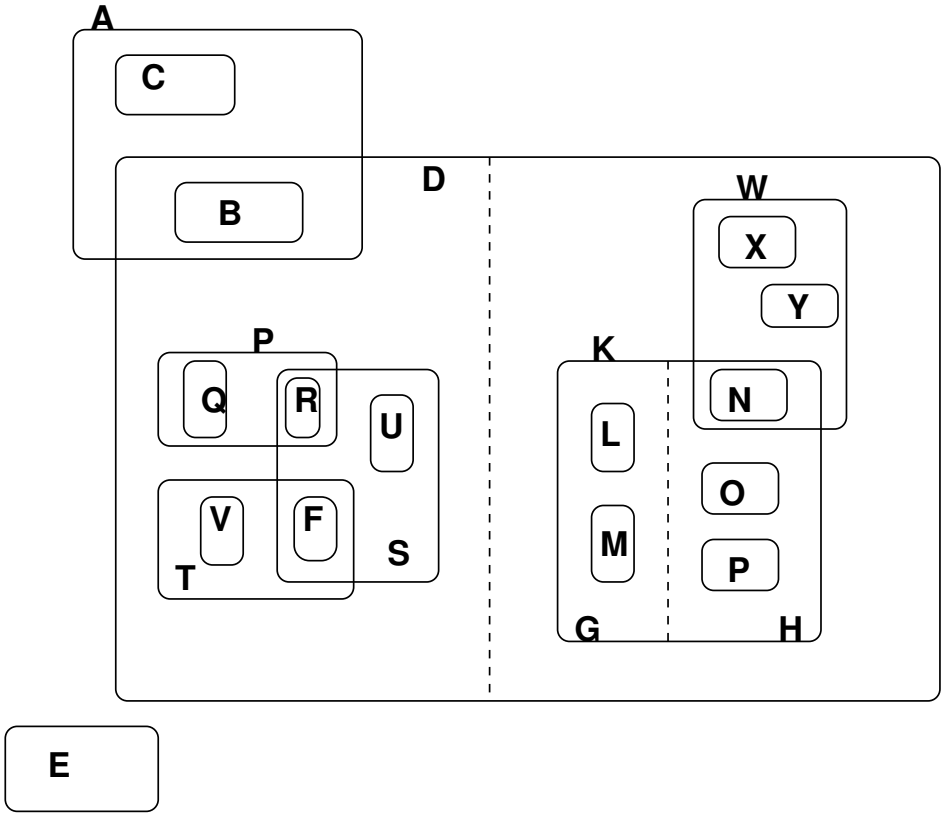


Unique Blobs (atomic sets, no intersection)



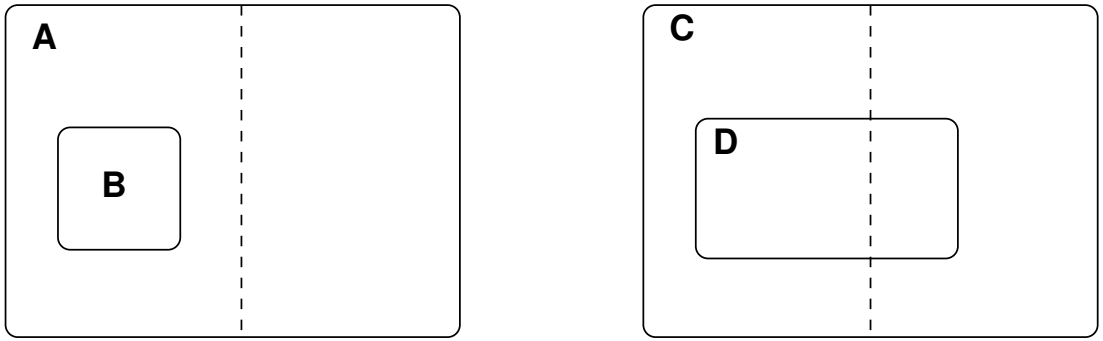
- empty space has no meaning, intersection must be identified
- atomic blobs are identifiable sets
- other blobs are union of enclosed sets

Unordered Cartesian Product: *Orthogonal* Components

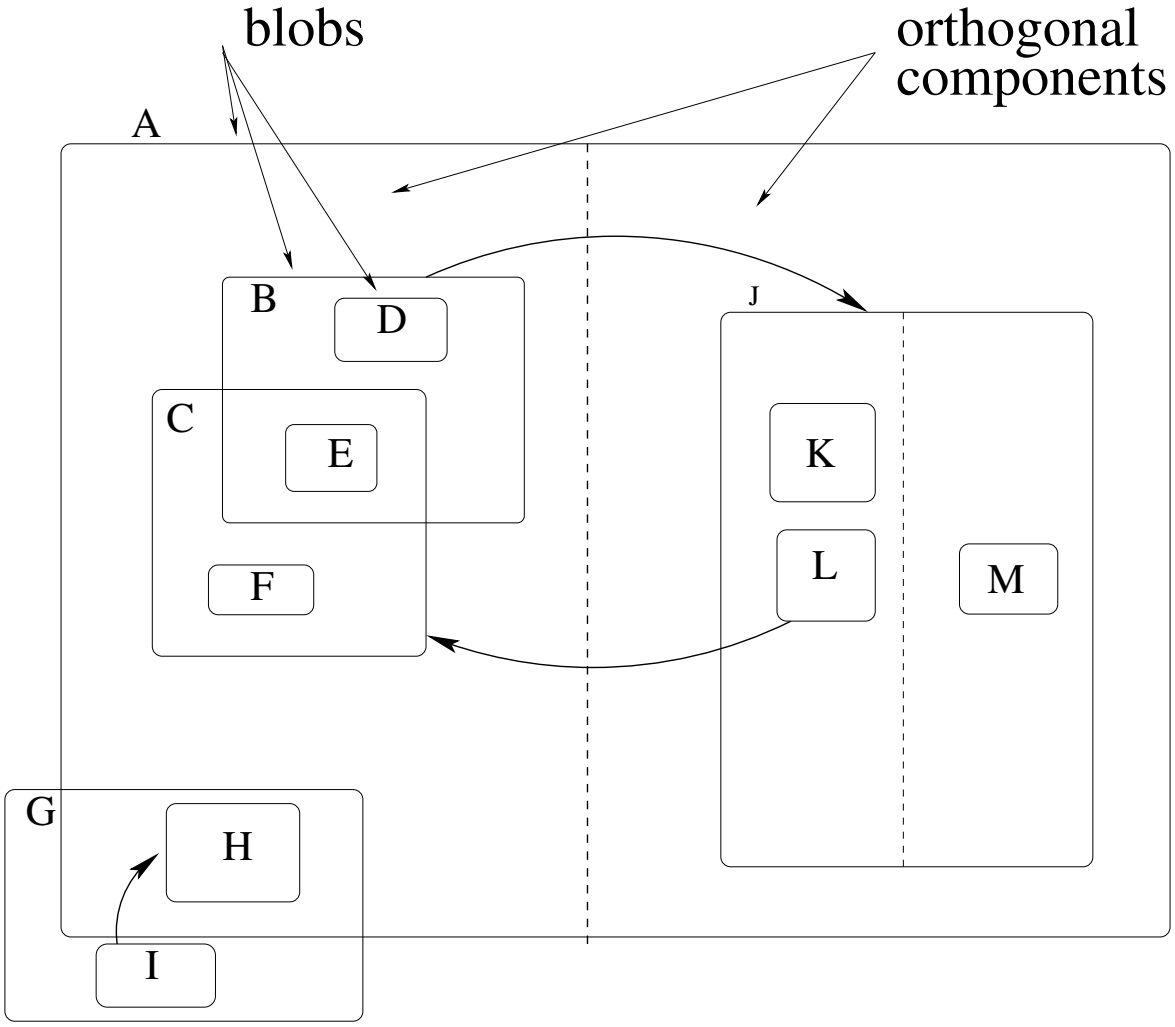


$$K = G \times H = (L \cup M) \times (N \cup O \cup P)$$

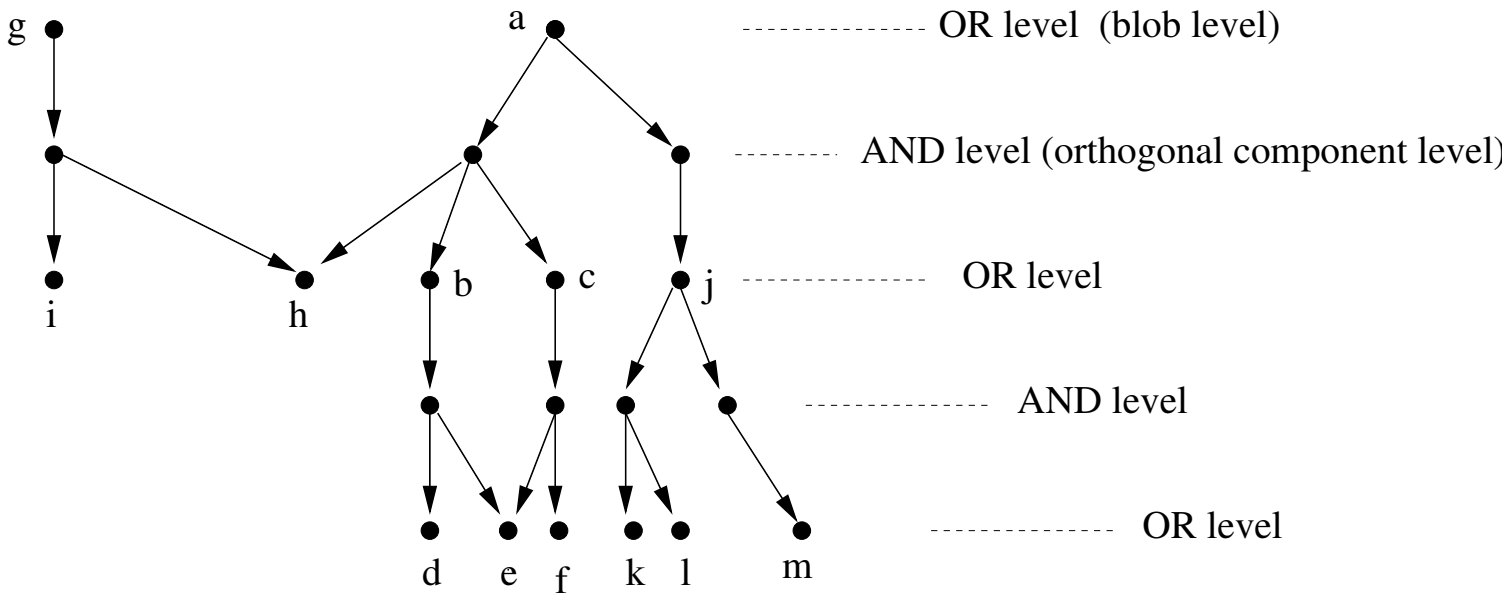
Meaningless constructs



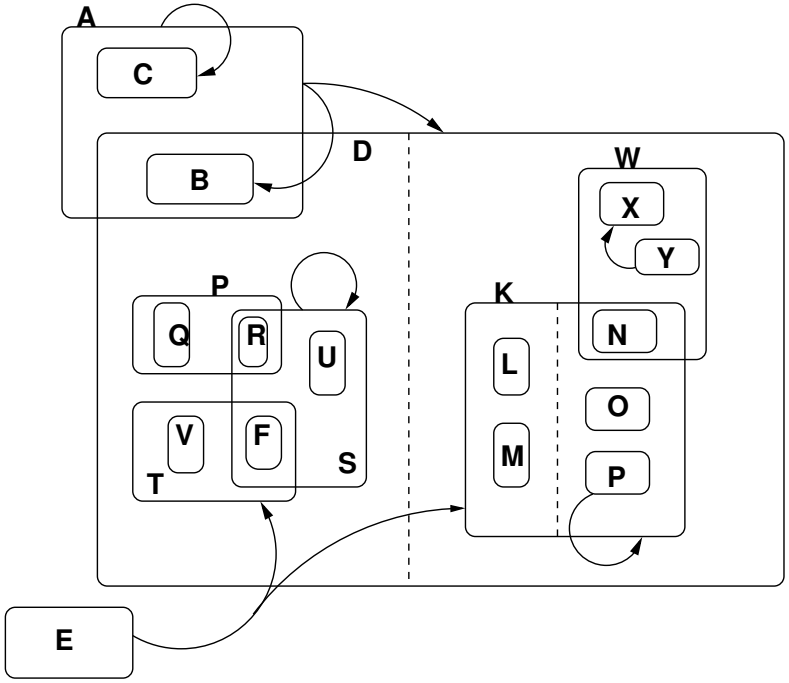
Simple Higraph



Induced Acyclic Graph (blob/orth comp alternation)

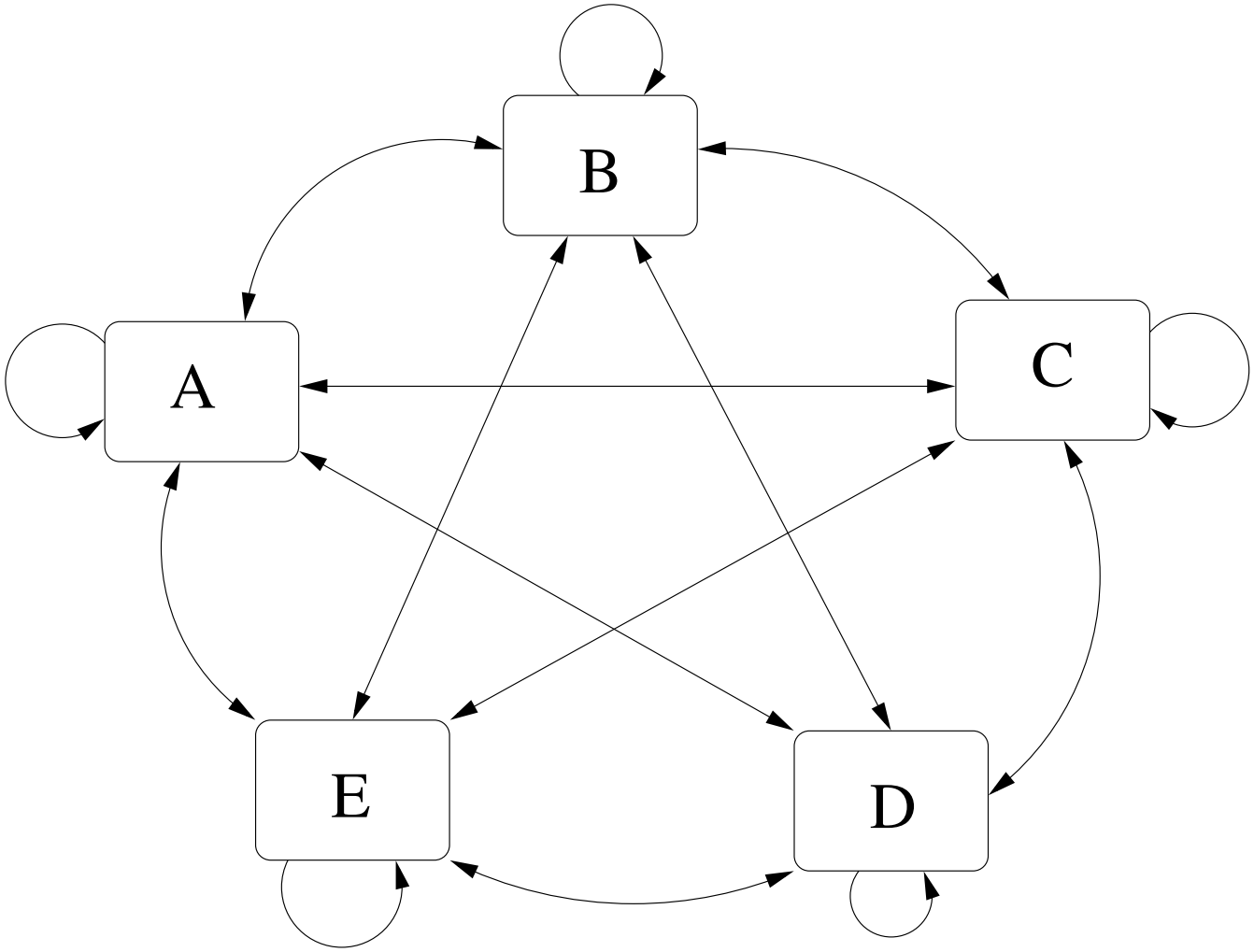


Adding (hyper) edges

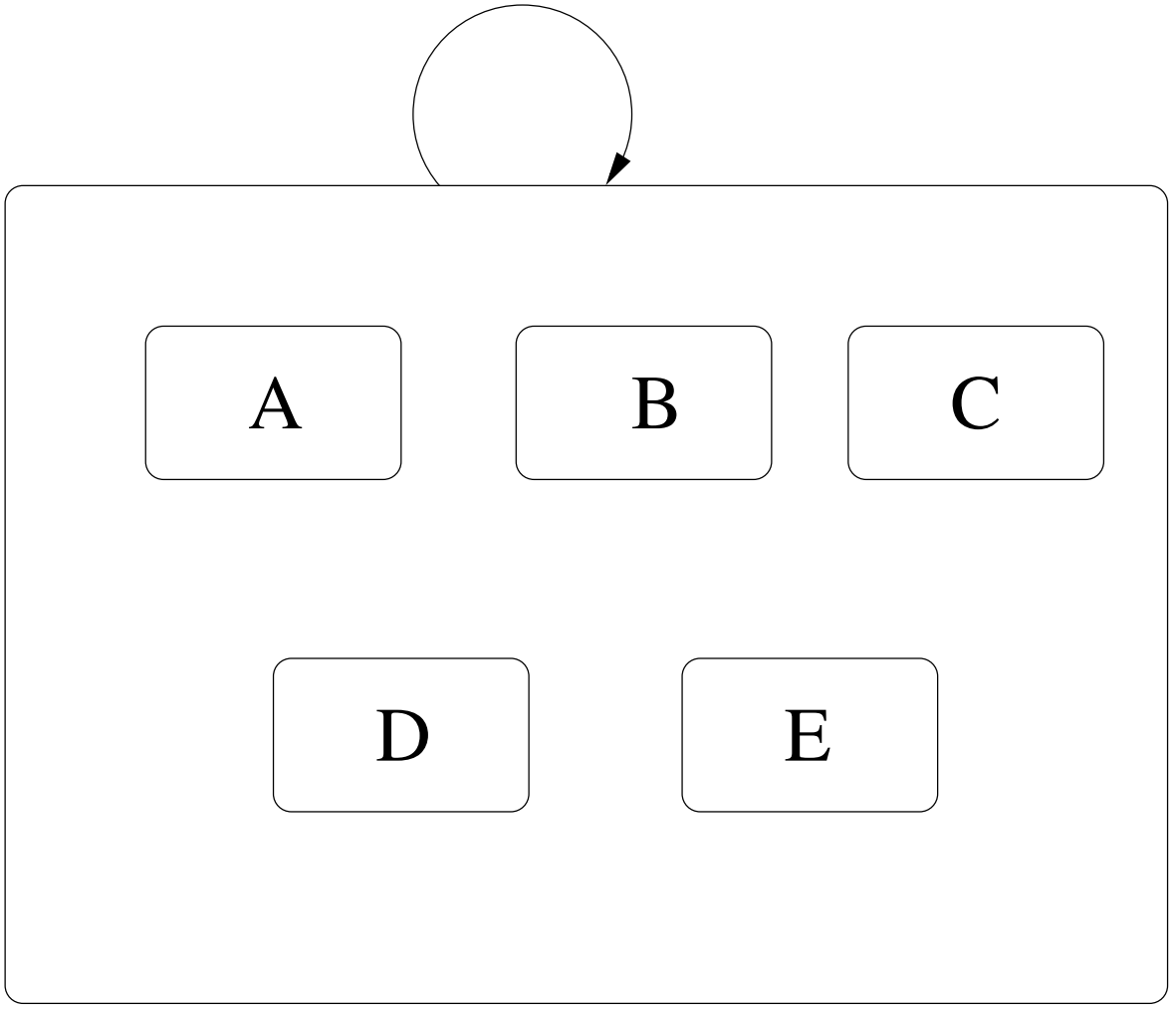


- hyperedges
- attach to contour of any blob
- inter-level possible

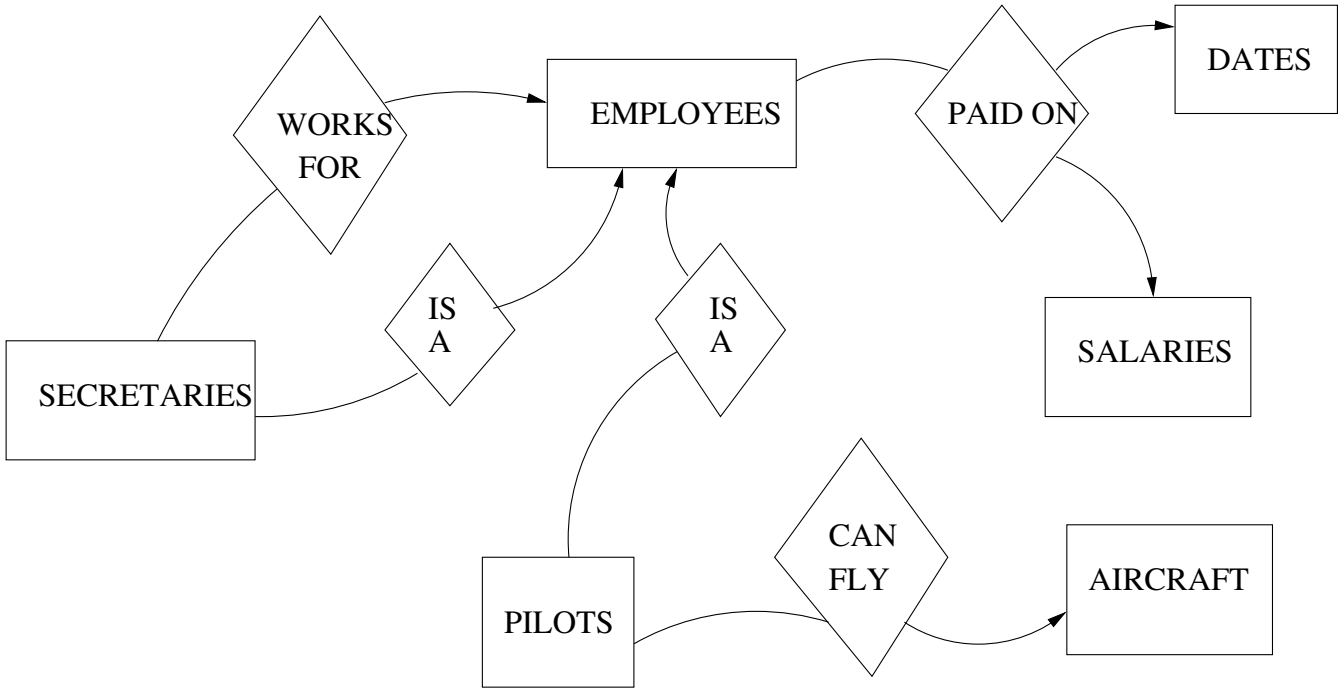
Clique Example



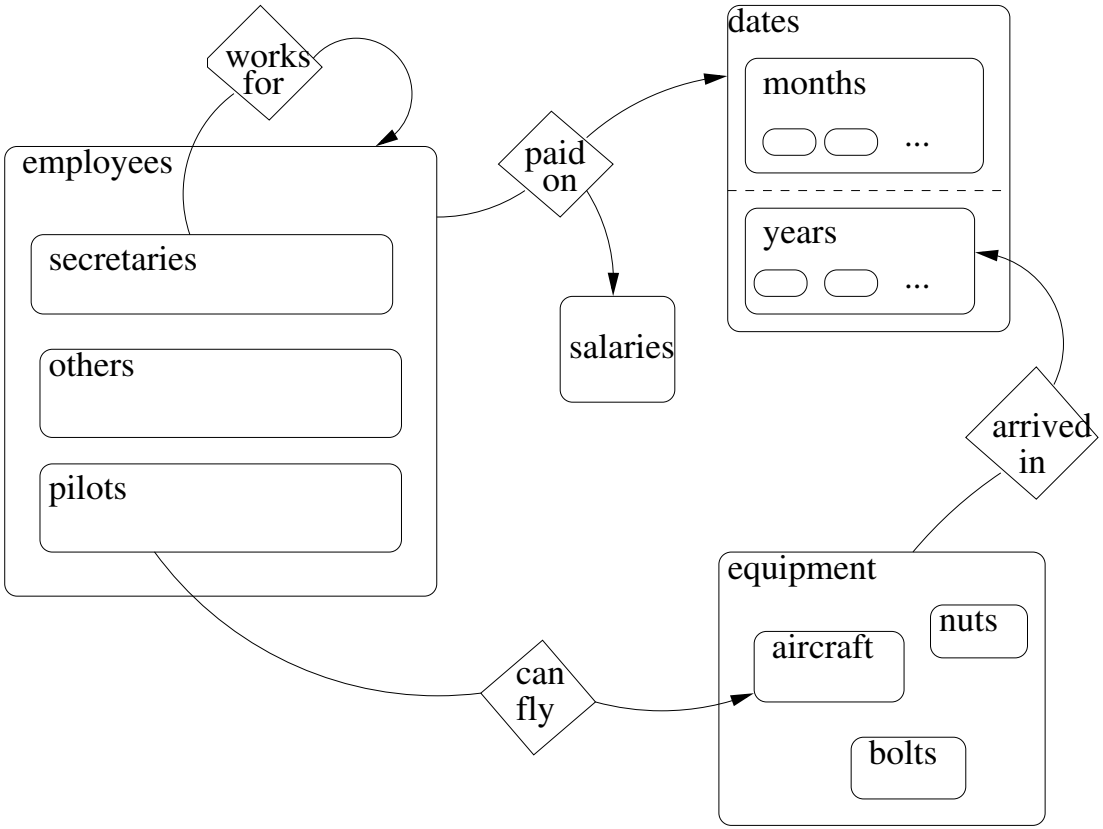
Clique: higraph with fully connected semantics



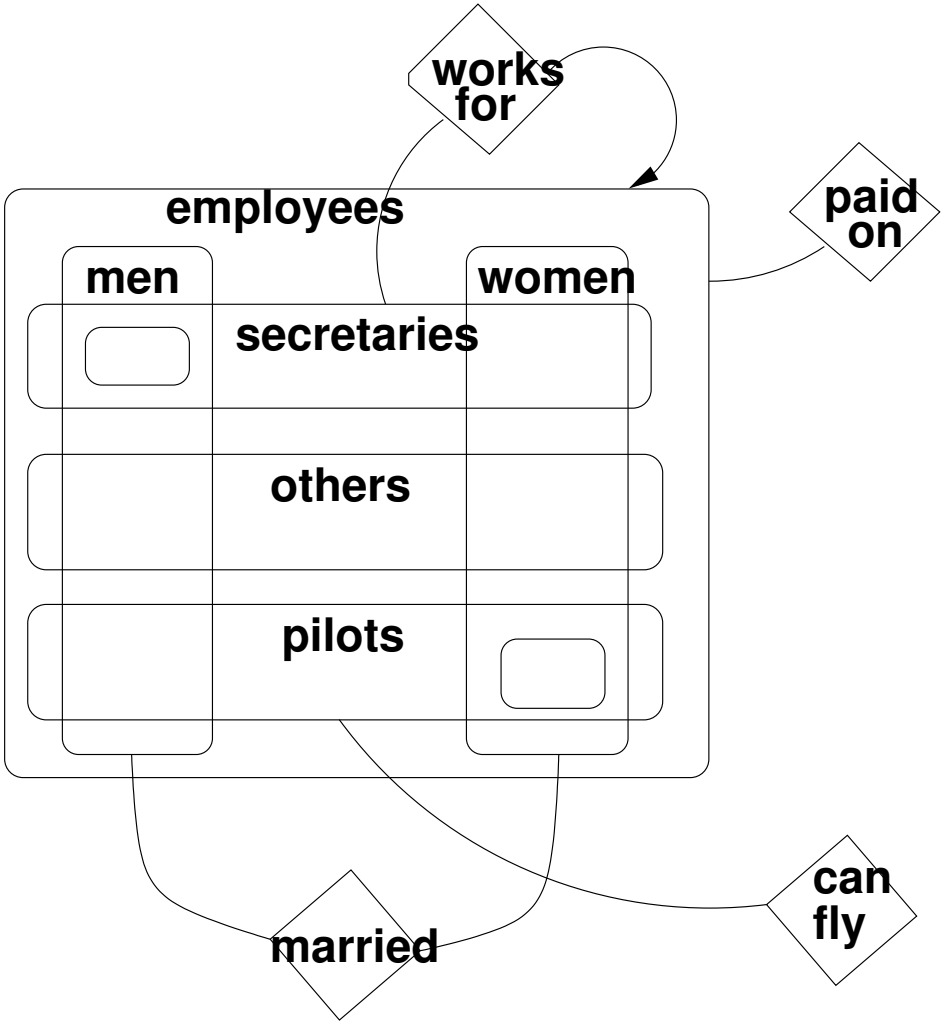
Entity Relationship Diagram (`is-a`)



Higraph version of E-R diagram



Extending the E-R diagram



Formally (syntax)

A higraph H is a quadruple

$$H = (B, E, \sigma, \pi)$$

B: finite set of all unique *blobs*

E: set of hyperedges

$$\subseteq X \times X, \quad \subseteq 2^X, \quad \subseteq 2^X \times 2^X$$

The subblob (direct descendants) function σ

$$\sigma : B \to 2^B$$

$$\sigma^0(x) = \{x\},\ \sigma^{i+1} = \bigcup_{y \in \sigma^i(x)} \sigma(y),\ \sigma^+(x) = \bigcup_{i=1}^{+\infty} \sigma^i(x)$$

Subblobs$^+$ cycle free

$$x \notin \sigma^+(x)$$

The partitioning function π associates *equivalence relationship* with x

$$\pi : B \rightarrow 2^{B \times B}$$

Equivalence classes π_i are *orthogonal components* of x

$$\pi_1(x), \pi_2(x), \ldots, \pi_{k_x}(x)$$

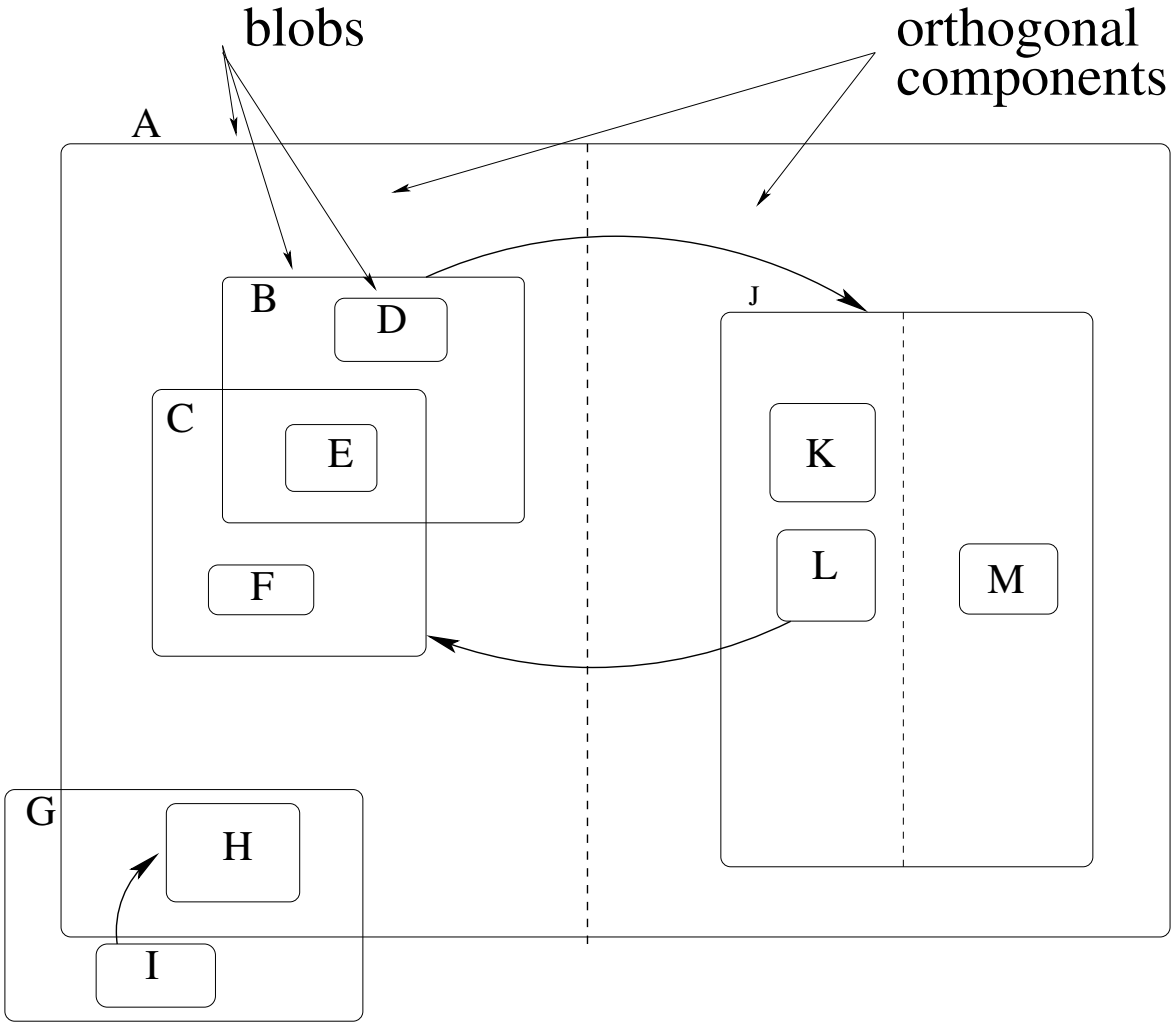
$k_x = 1$ means a single orthogonal component (no partitioning)

Blobs in different orthogonal components of x are *disjoint*

$$\forall y, z \in \sigma(x) : \sigma^+(y) \cap \sigma^+(z) = \emptyset$$

unless in the same equivalence class

Simple Higraph



Induced Orthogonal Components

$$B = \{A, B, C, D, E, F, C, G, H, I, J, K, L, M\}$$

$$E = \{(I, H), (B, J), (L, C)\}$$

$$\rho(A) = \{B, C, H, J\}, \rho(G) = \{H, I\}, \rho(B) = \{D, E\}, \rho(C) = \{E, F\},$$

$$\rho(J) = \{K, L, M\}$$

$$\rho(D) = \rho(E) = \rho(F) = \rho(H) = \rho(I) = \rho(K) = \rho(L) = \rho(M) = \emptyset$$

$$\pi(J) = \{(K, K), (K, L), (L, L), (L, K), (M, M)\}$$

Induces *equivalence classes* $\pi_1(J) = \{K, L\}$ and $\pi_2(J) = \{M\}, \ldots$

These are the *orthogonal components*

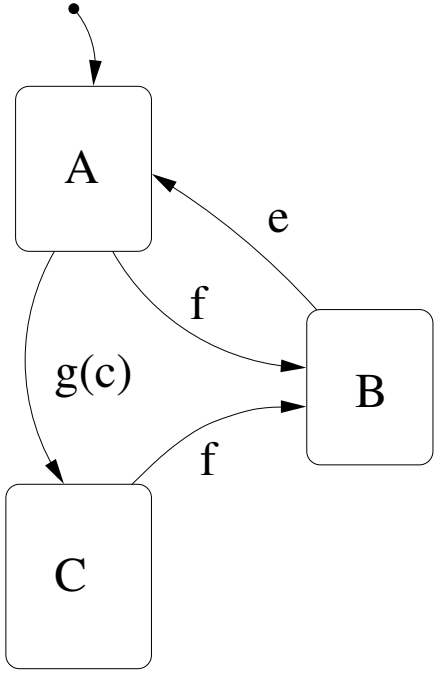
Higraph applications

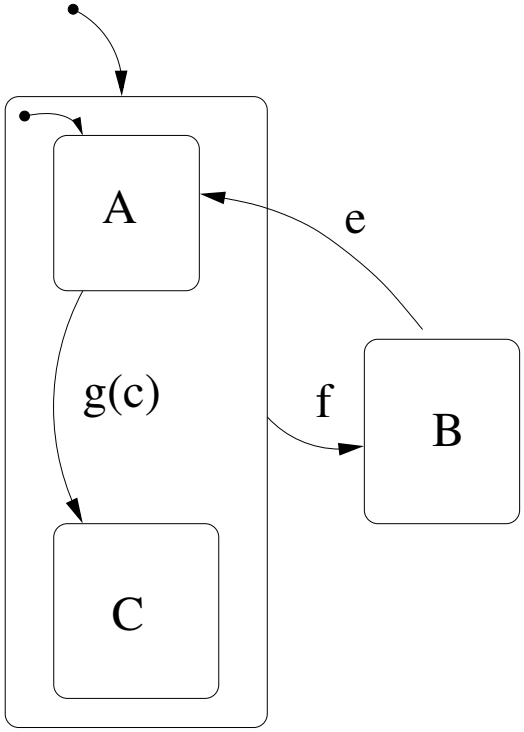
- E-R diagrams
- data-flow diagrams (activity diagrams)
  edges represent (flow of) data
- inheritance
- Statecharts

StateCharts = state diagrams + depth + orthogonality + broadcast

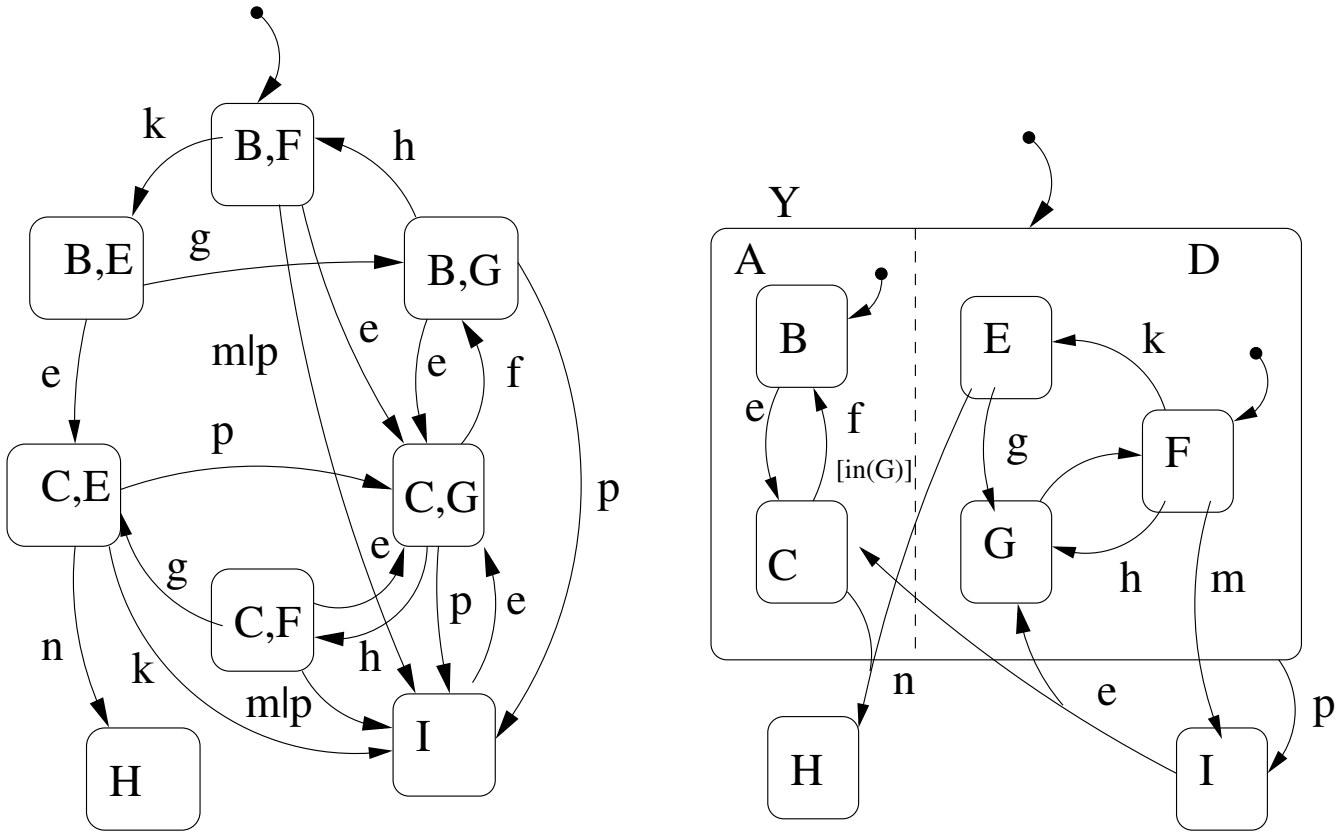
- Reactive Systems (event driven, react to internal and external stimuli)
- like Petri Nets, CSP, CCS, sequence diagrams, ...
- graphical but formal and rigourous for
  - analysis
  - code generation
- solve FSA problems:
  - flat $\Rightarrow$ hierarchy $\Rightarrow$ re-use
  - represent large number of transitions concisely
  - represent large number of states concisely
  - sequential $\Rightarrow$ concurrent

Depth (XOR)

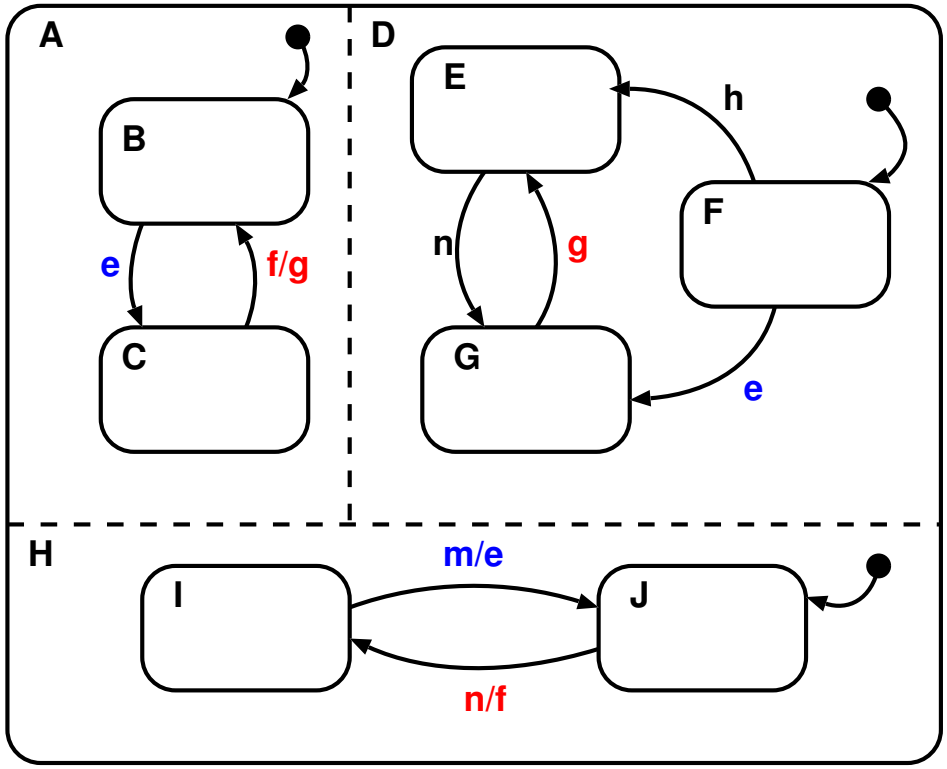




Orthogonality (AND), flattening $\Rightarrow$ semantics

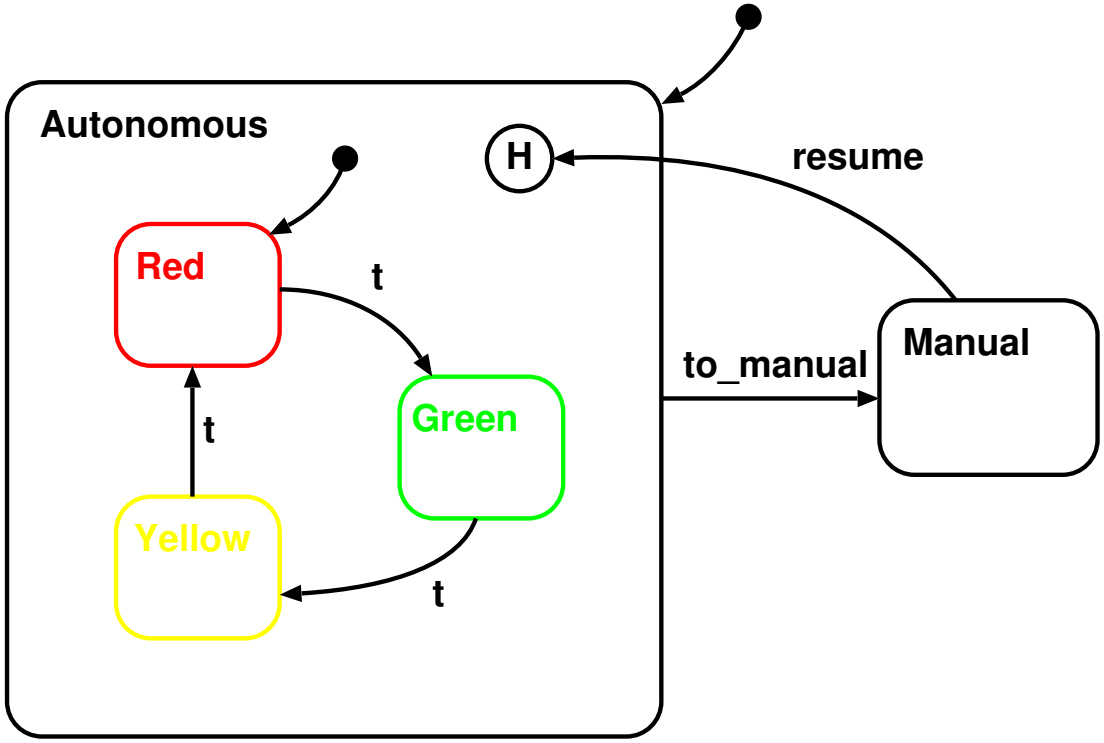


Broadcasting (output events)

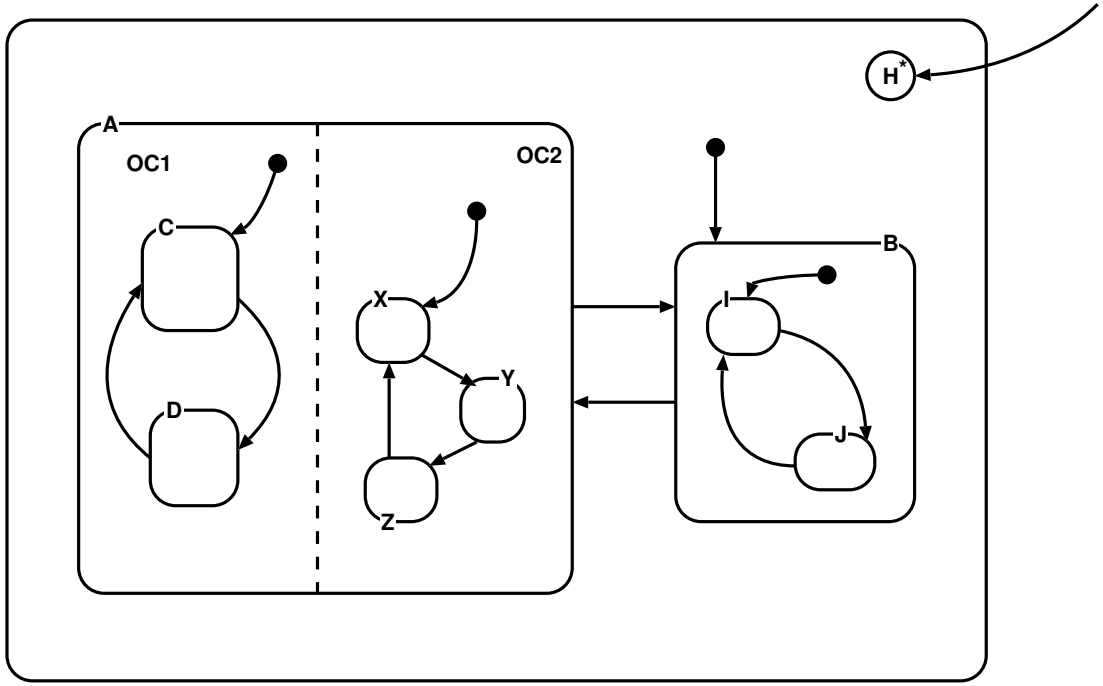


Input Segment: **nmnn**

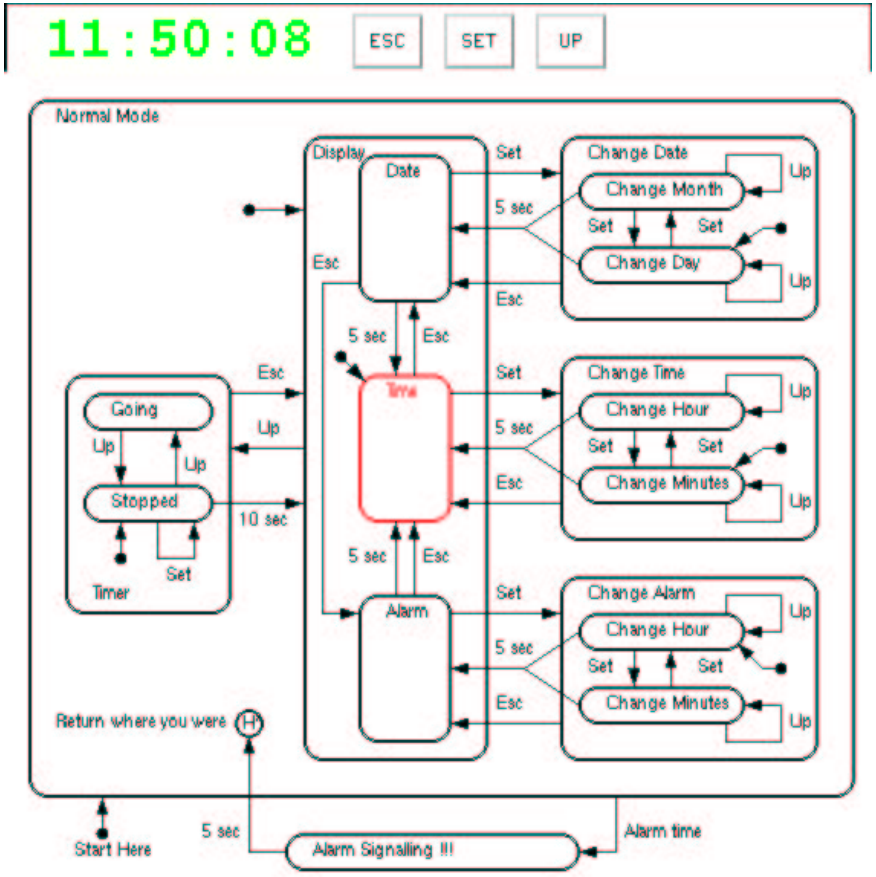
History States



Deep History

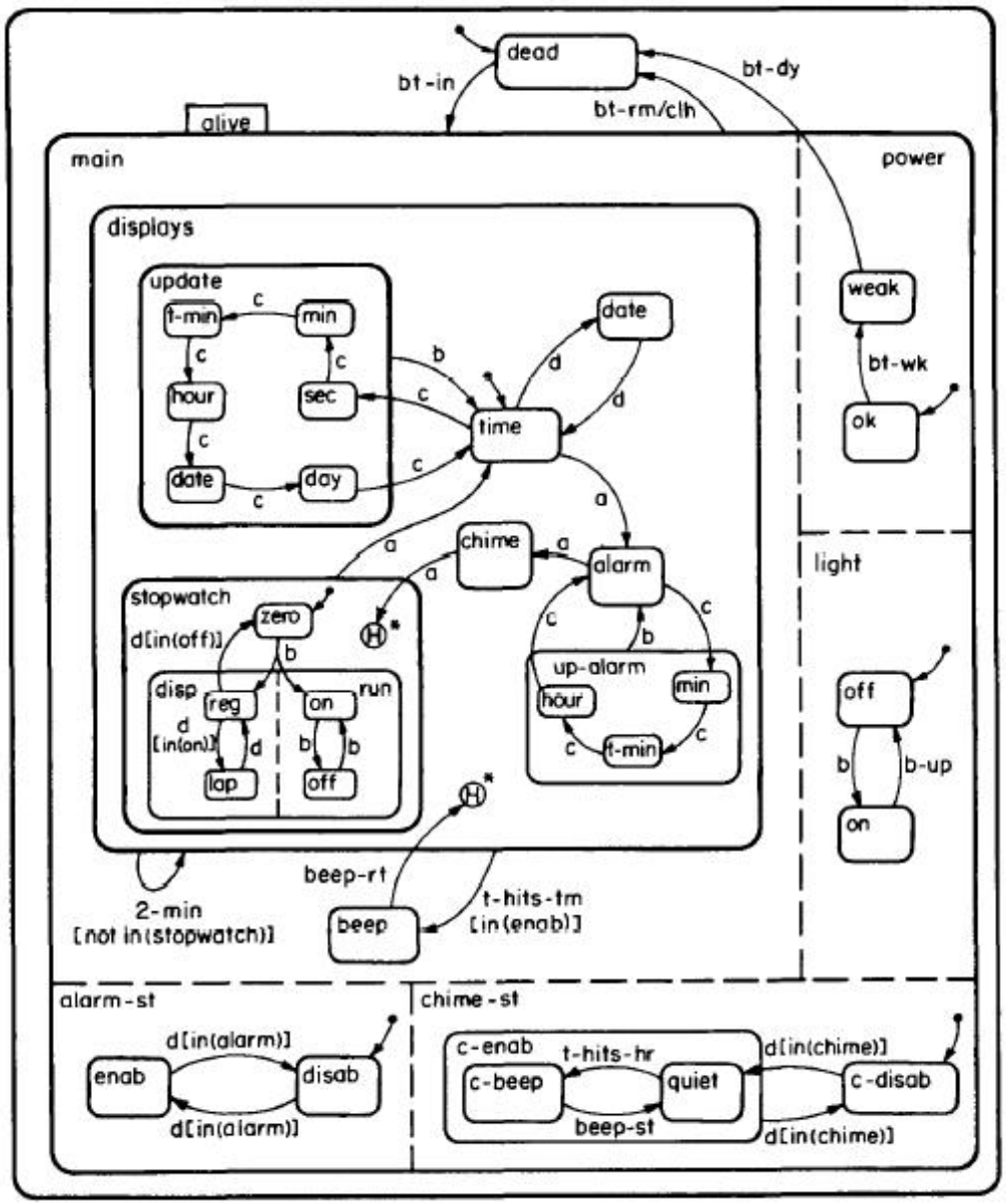


Simple Stopwatch Example



http://www.xjtek.com/products/xjcharts/demo/

Stopwatch Example


dead
bt-in
bt-dy
bt-rm/clh
alive
main
power
displays
update
t-min
min
hour
sec
date
day
time
date
weak
bt-wk
ok
chime
alarm
light
stopwatch
zero
d[in(off)]
disp
reg
lap
run
on
off
up-alarm
hour
min
t-min
off
b-up
on
beep-rt
2-min
[not in(stopwatch)]
beep
t-hits-tm
[in(enab)]
alarm-st
enab
disab
d[in(alarm)]
chime-st
c-enab
t-hits-hr
c-beep
quiet
beep-st
c-disab
d[in(chime)]

Messages

- `object.method(...)` synchronous method invocation
- `object->method(...)` queued, a-synchronous