



Lecția 9: Statechart-uri

v1.0

Gheorghe Stefanescu — Universitatea București

Metode de Dezvoltare Software, Sem.2

Februarie 2007— Iunie 2007



Statechart-uri

Cuprins:

- *Generalități*
- Grupări și rafinări
- Independență și concurență
- Alte facilități
- Concluzii, diverse, etc.



Statechart-uri:

- Introduse de Harel în 1987 pentru specificarea și proiectarea sistemelor complexe, bazate pe componente și interacția lor
- Exemple: sisteme multi-computer, protocoale de comunicare, unități de control, etc.
- Curent au fost incorporate în UML și sunt intensiv folosite pentru sisteme software complexe
- Folosim articolul original: *D. Harel, Statecharts: A visual formalism for complex systems, 1987.*



..Generalități

Caracteristici de baza:

- *statechart-uri* = *diagrame de stări*
 - + *adâncime*
 - + *ortogonalitate*
 - + *comunicare prin broadcast*
- Pe scurt stările,
 - se grupează în structuri hierarhice,
 - se descompun în substări concurente,
 - definesc procese care comunică prin sincronizare, broadcast

Cuprins:

- Generalități
- *Grupări și rafinări*
- Independență și concurență
- Alte facilități
- Concluzii, diverse, etc.

Grupari si rafinari

Arii, săgeți comune

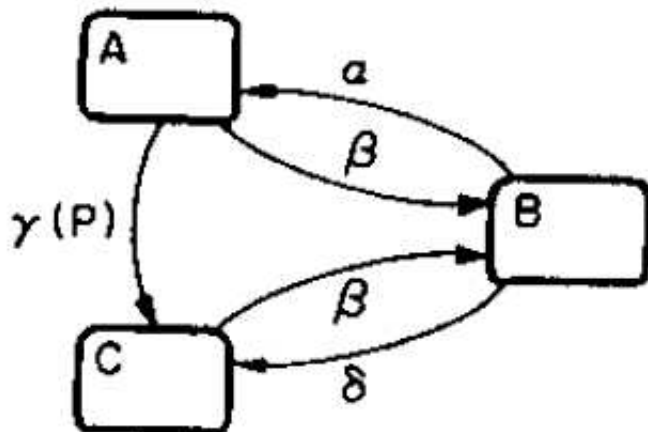


Fig.1

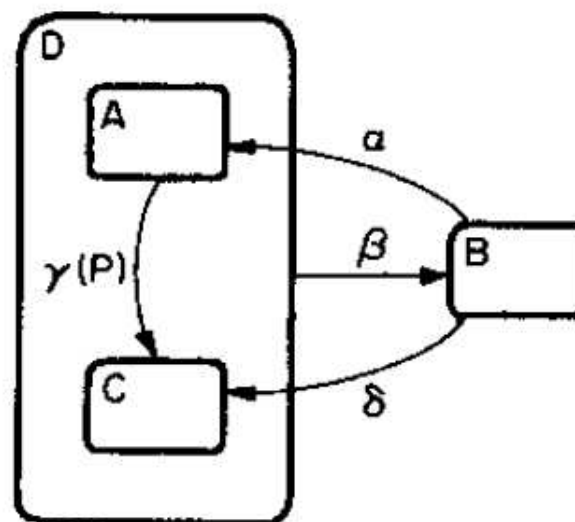


Fig.2

- în Fig.1 avem o diagramă uzuală, plată
- în Fig.2 avem o reprezentare echivalentă: stările A,C sunt grupate într-o superstare D, delimitată cu un contur rectangular
- β din Fig.2 notează *o mulțime de arce β* , câte *unul pentru fiecare stare* din D



..Grupari si rafinari

Grupare, abstracție

- trecerea de la Fig.1 la Fig.2 se face prin *grupare*
- noile super-stări introduse sunt de tip *exclusiv-or (XOR)*:
 - a fi într-o stare din D înseamnă a fi *în A* ori *în C*, dar *nu în amundouă*
- stările sunt reprezentare prin contururi rectangulare care nu se intersecrează
- săgețile sunt etichetate cu litere reprezentând *evenimente* (având, opțional, atașate *condiții* ca în $\gamma(P)$)

..Grupari si rafinari

Rafinare

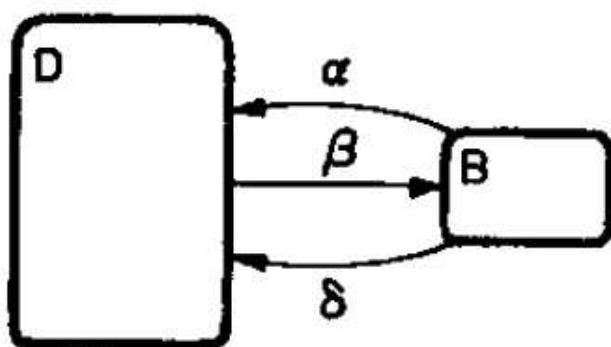


Fig.3

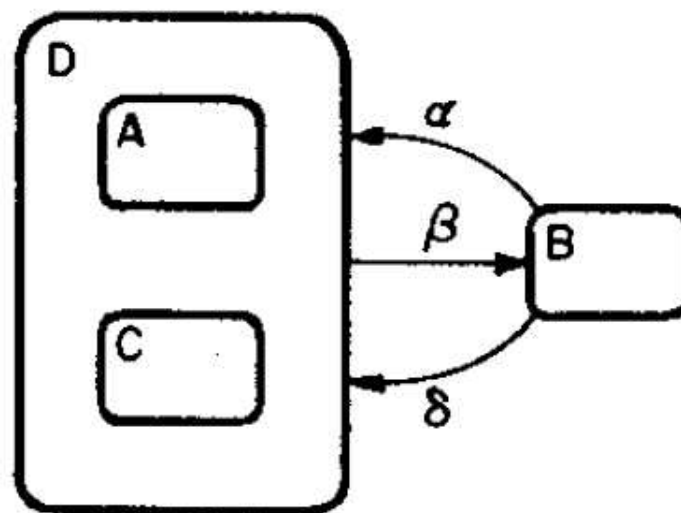


Fig.4

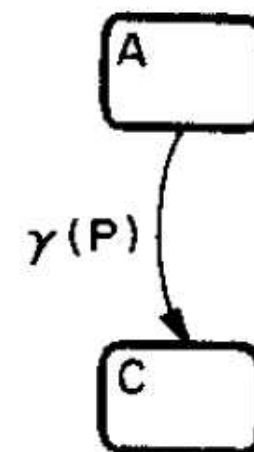


Fig.5

Pe dos, diagrama din Fig.2 poate fi gândită ca rezultatul unei *rafinări*:

- se pleacă cu Fig.3, apoi se rafinează D în A și C (Fig.4)
- se include comportamentul intern din D (Fig.5)
- în final, se clarifică ținta săgeților incomplet definite α, δ



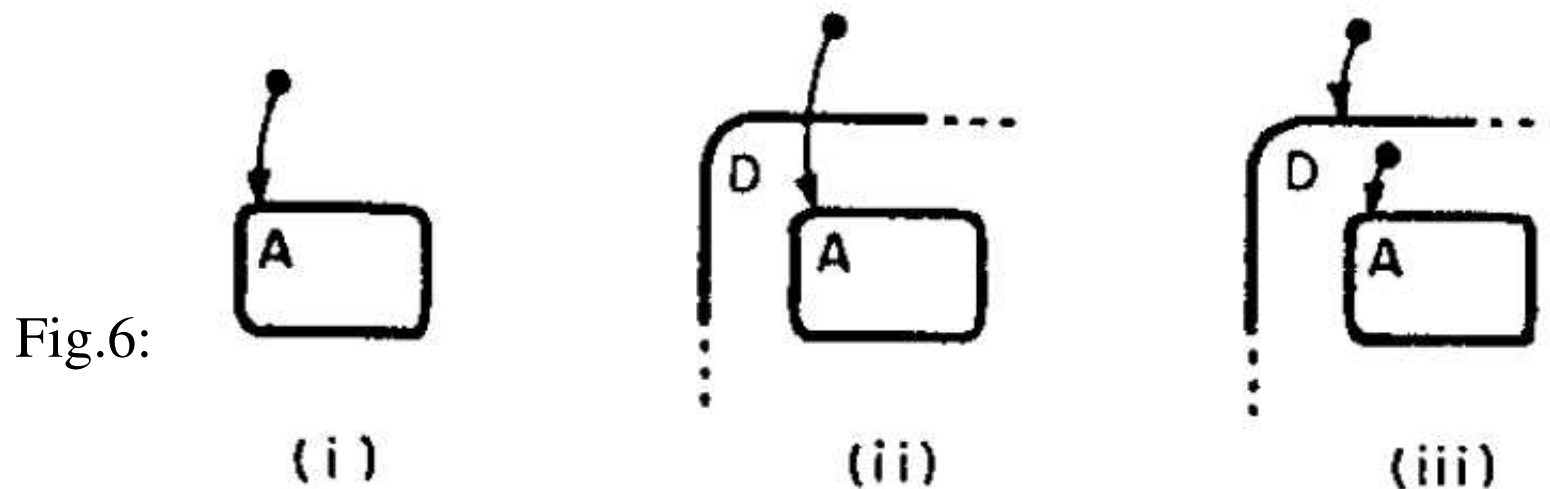
..Grupari si rafinari

Zooming-in și zooming-out

- *abstracția* și *rafinarea* sunt activități *complementare* de zoom-out și zoom-in
- reprezintă un mecanism eficient de a face inteligibile specificații mari

..Grupari si rafinari

Intrarea într-o superstare



- în Fig.6(i) avem specificarea unei stări de start (o mică săgeată)
- intrarea într-o *superstare* se poate face
 - direct într-o stare specificată - exemplu Fig.6(ii)
 - ori în starea de început a superstării - exemplu Fig.6(iii)

Exemplu: Ceas

Exemplu (ceas)

- *Citizen Quartz Multi-Alarm III*
- *Format din:*
 - o zonă principală de display
 - 4 zone mici de display
 - beeper cu 2 tonalități
 - 4 butoane de comandă a,b,c,d



- *Funcționalități:* afișează timpul ori data, are 2 alarme independente, cronometru, sursă de iluminat, baterie, test de beeper
- *Notății:* *a* - butonul *a* este apăsat; *â* - butonul *a* este eliberat

..Exemplu: CeasFig

Tranzițiile dintre *modul display* și *stările de beep*

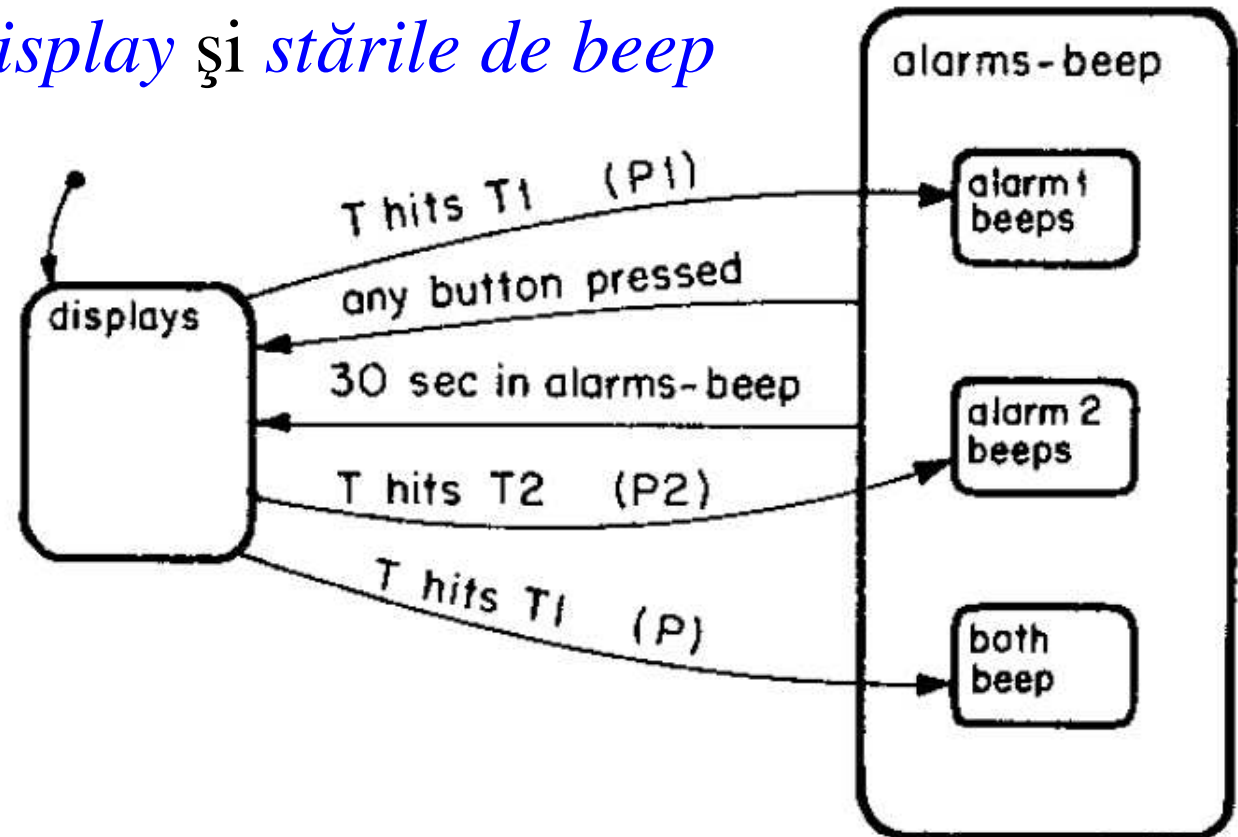


Fig.7:

- $T1, T2$ - timpii la care au fost setate alarmele
- condițiile $P1, P2, P$ sunt:
 - $P1 = alarm1 \text{ enabled} \wedge (\neg(alarm2 \text{ enabled}) \vee T1 \neq T2)$
 - $P2$ - similar
 - $P = alarm1 \text{ enabled} \wedge alarm2 \text{ enabled} \wedge T1 = T2$

..Exemplu: CeasFig

O *rafinare* a stării *display*

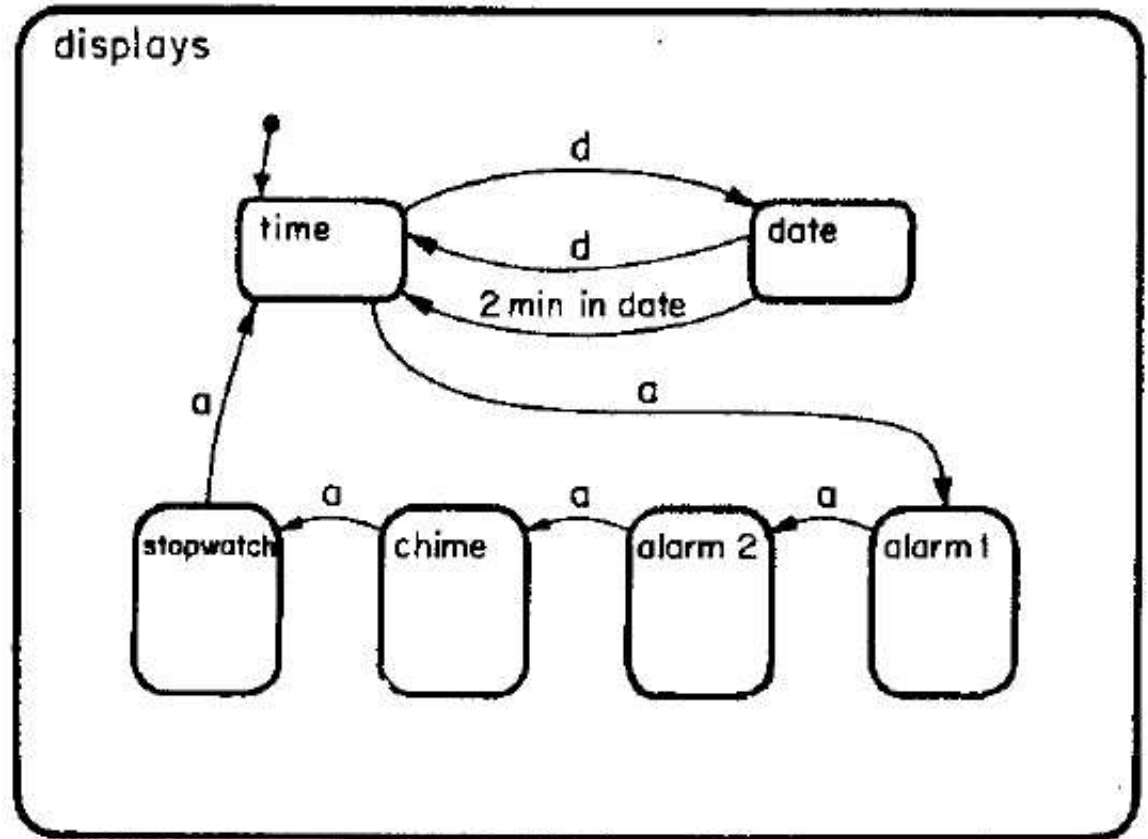


Fig.8:

- *d* - switch între afişare timp şi dată
- *a* - produce o trecere ciclică printr-o serie de substări display
- modul “default” la intrarea în starea displays este time

Reintrarea într-o stare

- se poate face în substarea “*default*”,
- în *ultima sub stare vizitată*,
- ori în diverse *combinații* (dacă starea este una ierarhică, complexă)

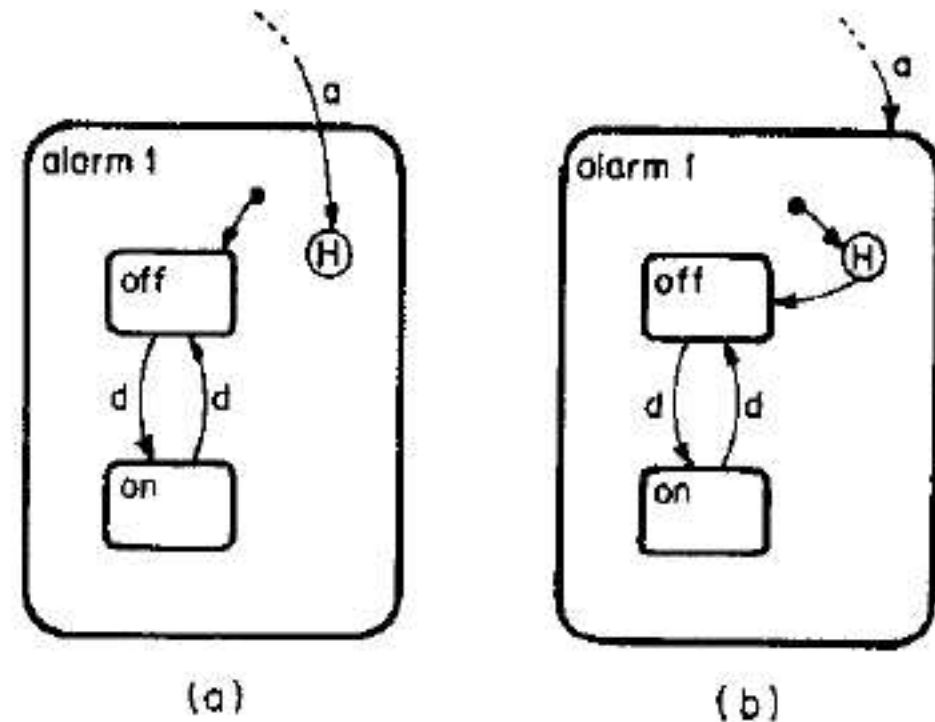


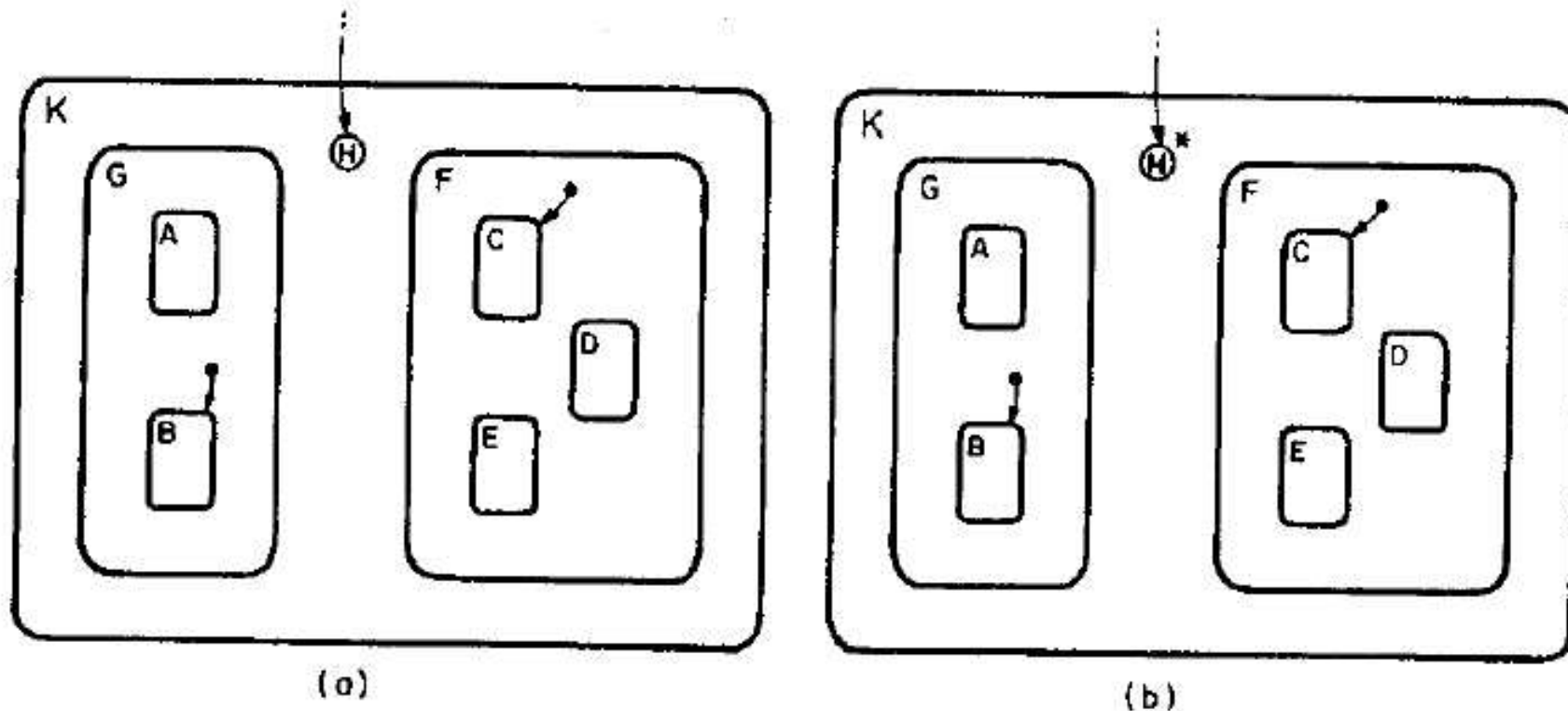
Fig.9

Notății:

- *H* indică reintrarea în *ultima stare vizitată*
- se poate combina cu starea “default” (folosită automat la prima intrare) ca în Fig.9(a),(b)

..Istorie

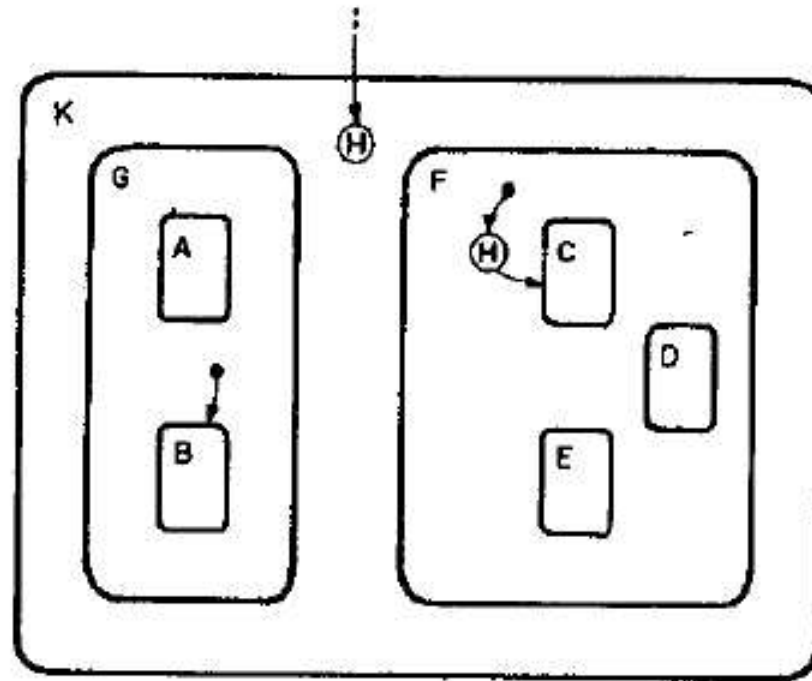
Combinatii de intrări într-o stare



- în (a), opțiunea de istorie indică alegerea ultimei stări vizitate dintre G,F, intrându-se în starea lor “default”
- în (b), se folosește notația H^* , pentru a indica faptul că opțiunea de *istorie* se aplică *recursiv* tuturor substărilor

..Istorie

..Combinații de intrări într-o stare (cont.)



- în figura de mai sus, avem o situație de *istorie mixtă* (nici primul nivel, nici peste tot)
 - se intră în ultima stare vizitată dintre G,F, în G intrându-se în starea “default”, iar în F în ultima stare vizitată

..Exemplu: Ceas

Rafinare a stării
displays

Nota: Acum de foloseste H^ , modificand optiunea din Fig.8 de a reveni in modul time.*

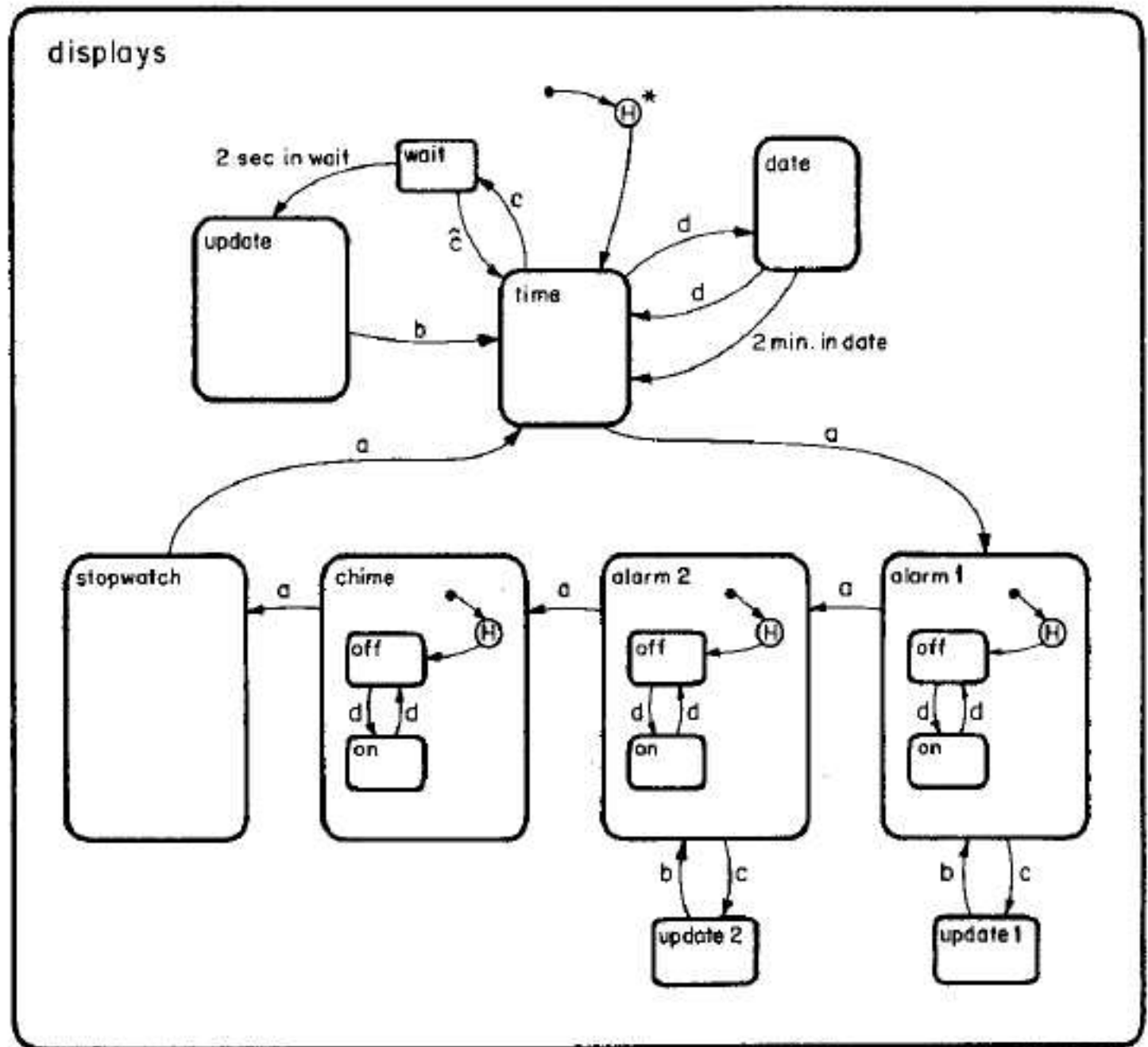


Fig.10

..Exemplu: Ceas

Rafinare a stărilor *update*

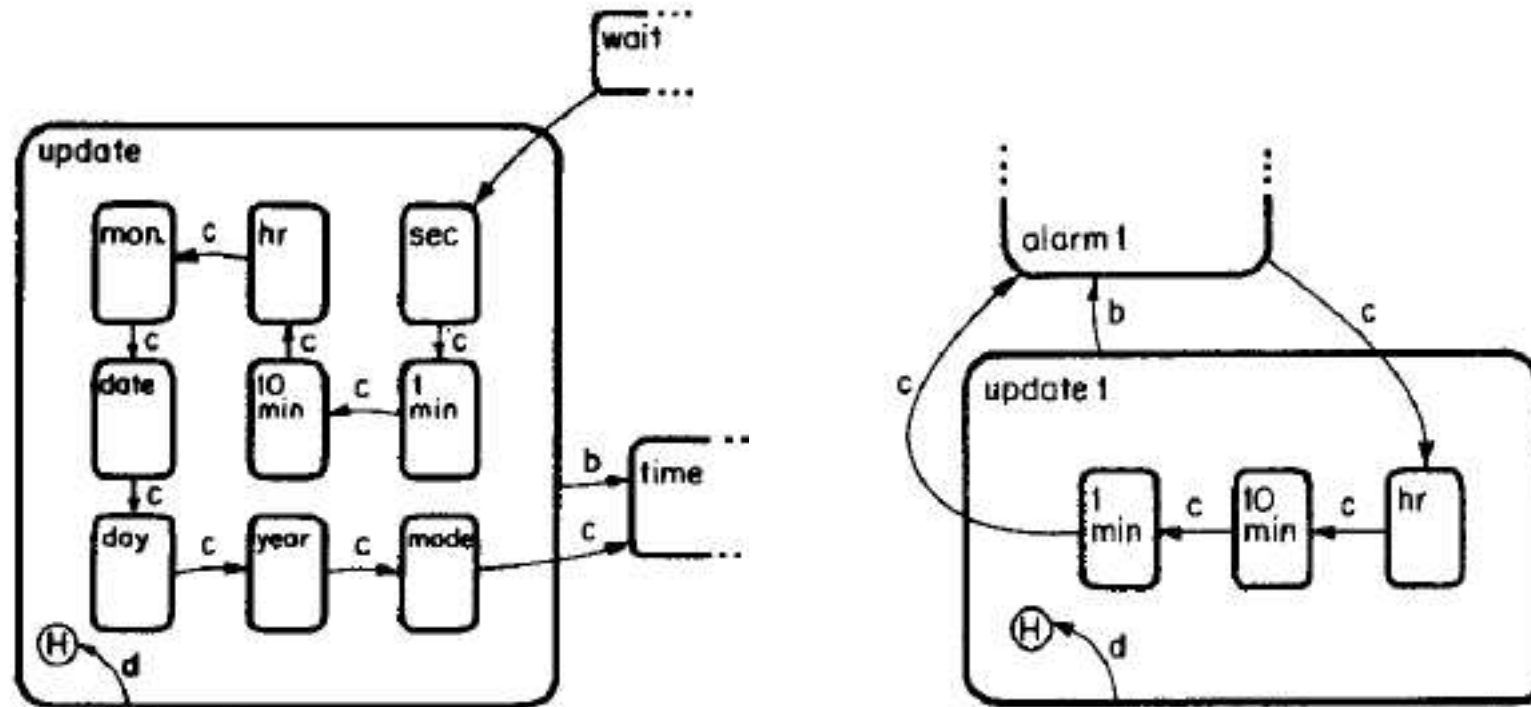
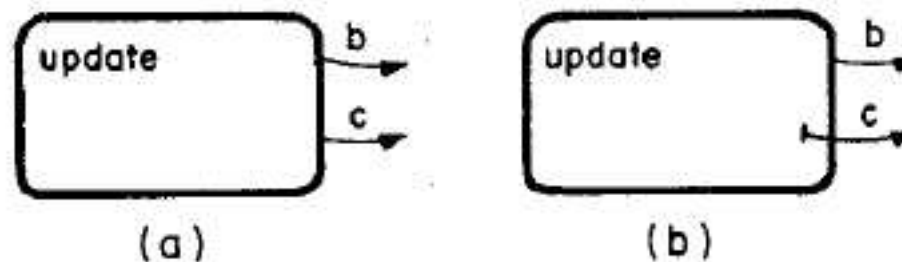


Fig.11

Observatii: de notat ca arcul c de iesire nu se aplica tuturor starilor, deci trebuie o notatie suplimentara - cea din (b)





..Exemplu: Ceas

Observații:

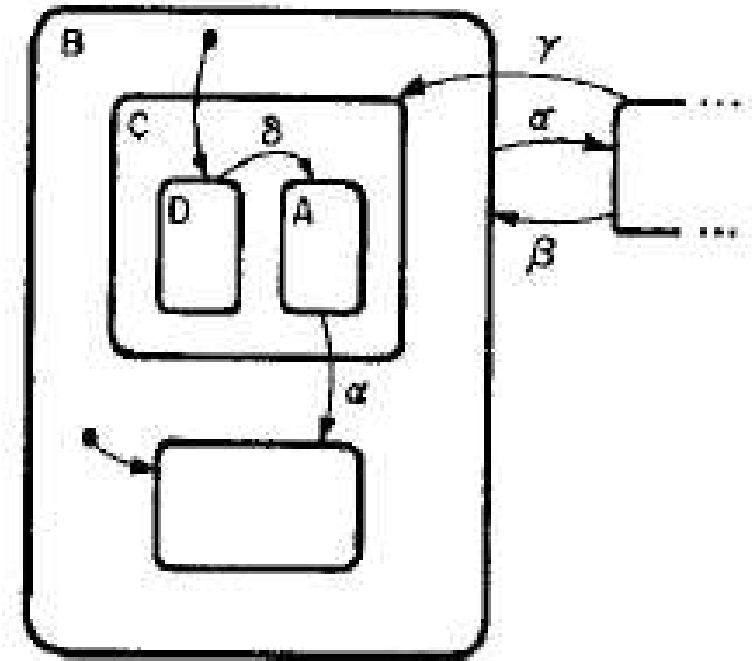
- arcul cu *d* și *H* din Fig.11 vrea să captureze următorul comportament (observat pe ceas)
 - apăsând *d* în orice stare din update, *se iese* imediat, iar apoi (după update, nespecificat aici) *se revine* în starea pe care tocmai am *părăsit-o*
- de notat că o acțiune ca *d* are efect în multiple *stări concurente* (pe care le vom descrie ulterior), deci chiar dacă nu are efect vizibil aici, poate avea efect în altă componentă

Arce cu surse ori destinații comune

Arce cu surse ori destinații comune:

Pot apare diverse erori

- α - contradicție: după A, se iese din C ori B?
- β - contradicție: unde se intră?
- γ - nespecificat: care este starea “default” din C?



Cuprins:

- Generalități
- Grupări și rafinări
- *Independență și concurență*
- Alte facilități
- Concluzii, diverse, etc.

Ortogonalitate

Descompunerea in latime:

- este ortogonală pe structura ierarhica anterioară
- stările complexe pot avea și *descompuneri-și* (pot fi AND-stări):
 - sistemul este într-o AND-stare când este într-o *stare pentru fiecare componentă*

Exemplu:

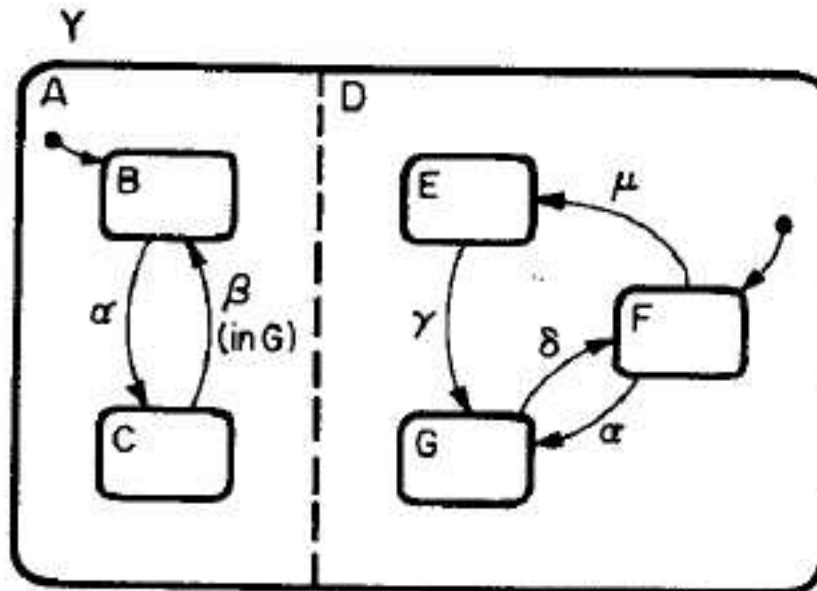


Fig.12



..Ortogonalitate

Comentarii:

- starea Y este AND-stare formată din componentele A și D
- reprezentarea grafică folosește *linii întrerupte* pentru a separa componentele unei AND-stări
- Y este un fel de *produs* al lui A și D, dar cu
 - *sincronizare*: o acțiune (ca α) poate apare în mai multe componente și se execută simultan în toate componentele
 - *independență*: o acțiune (ca μ) poate apare într-o singură componentă, executându-se independent
 - *dependență*: o acțiune dintr-o componentă (ca β) poate folosi condiții despre starea altei componente

..Ortogonalitate

- în fine, o reprezentare plată echivalentă cu cea din Fig.12:

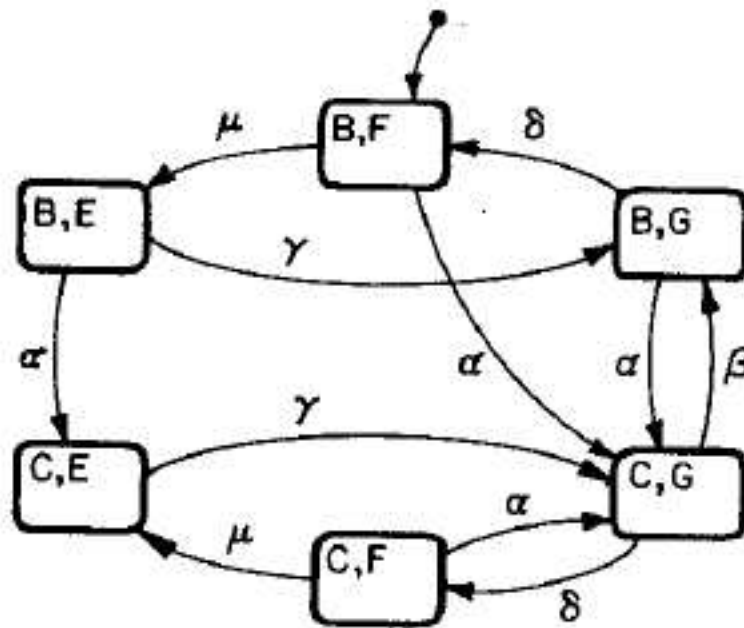


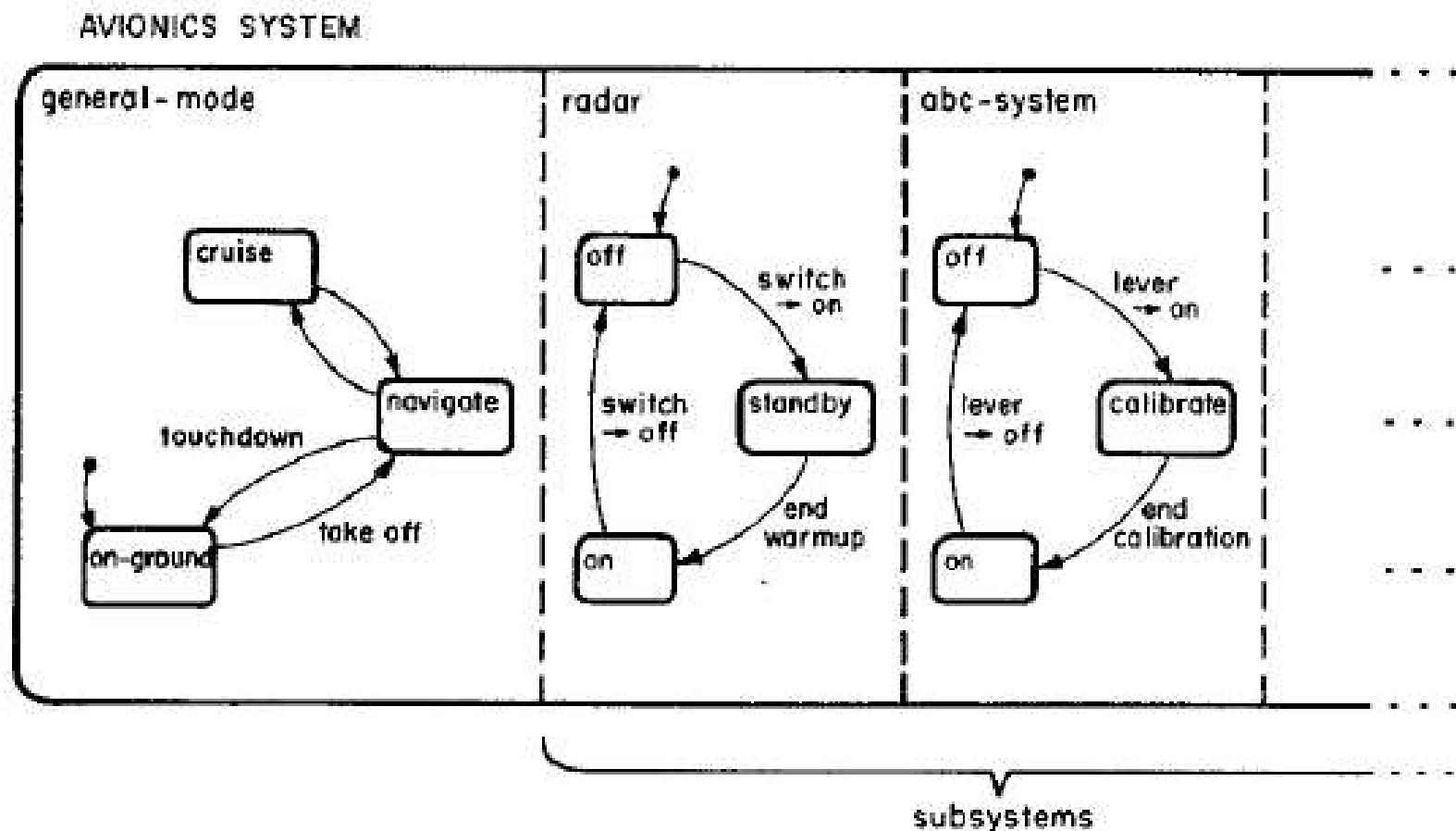
Fig.13

- economie:
 - în Fig.12 folosim 6 tranziții în loc de 11
 - în genere, trecerea de la stări AND la variantele platizate crează un *blow-up exponențial*

..Ortogonalitate

Criterii de descompunere

- deseori este utilă descompunerea unei stări AND în componente pe principii *structurale*
- exemplu (aviație)

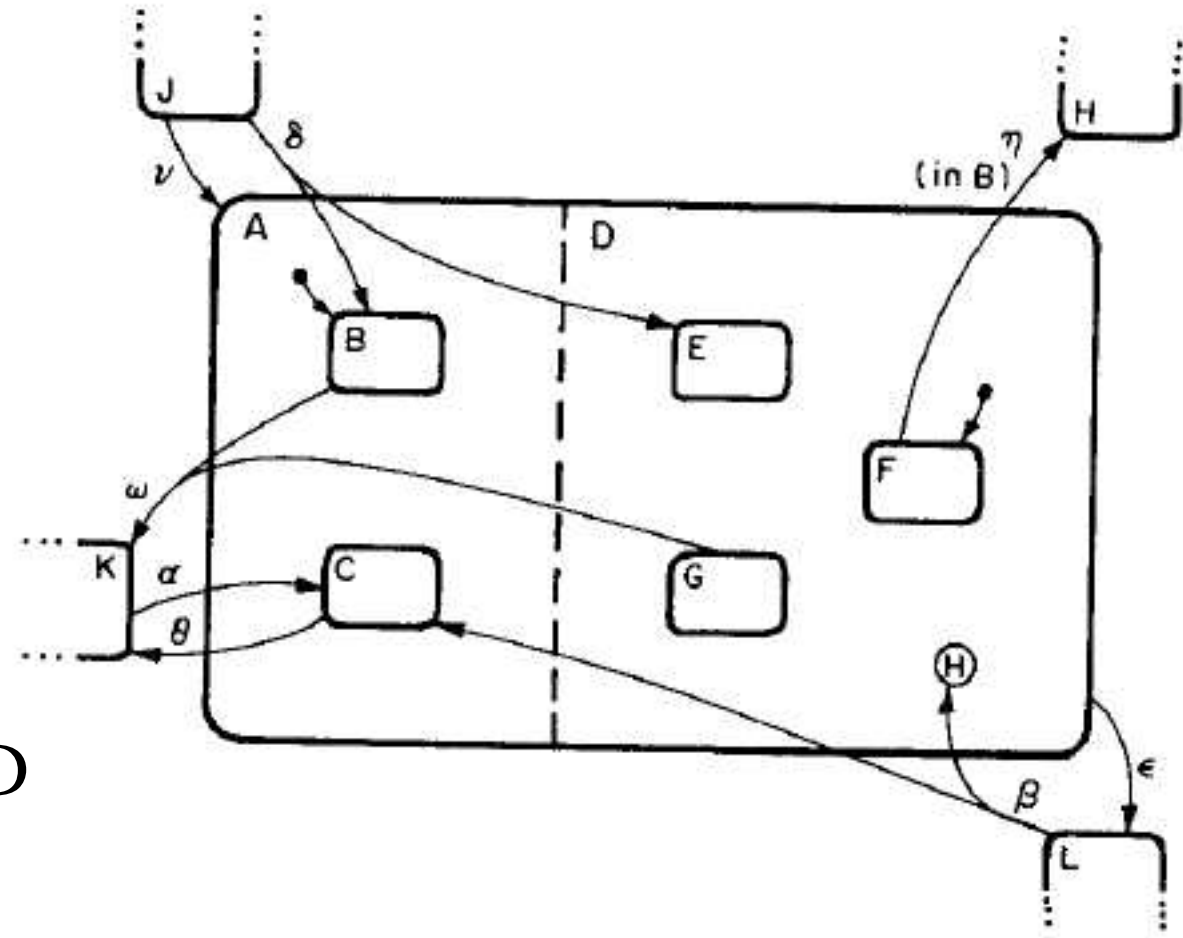


..Ortogonalitate

Intrări/ieșiri în/din AND-stări

In:

- δ - (B,E)
- α - (C,F)
- ν - (B,F)
- β - (C,X), unde X este starea anterior vizitată în D



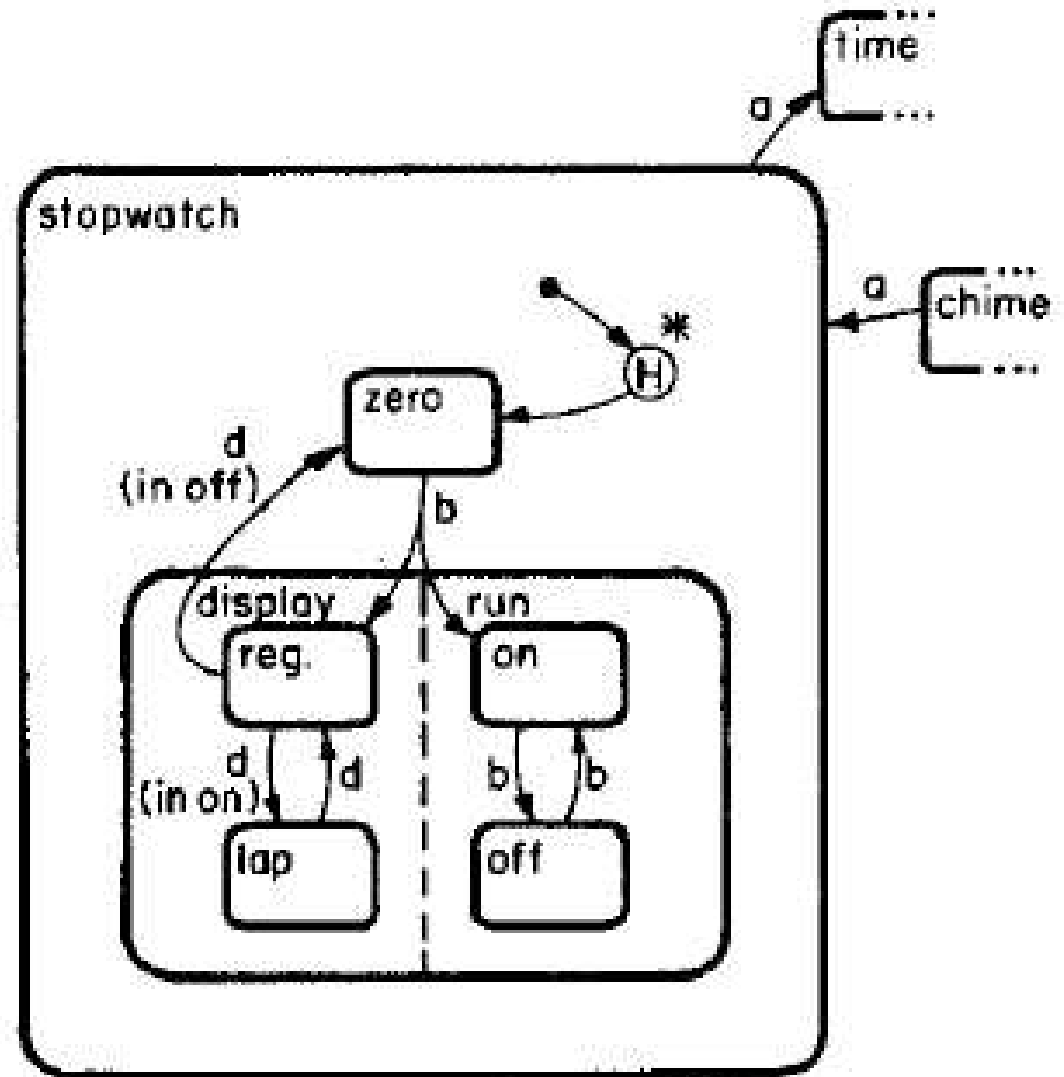
Out:

- ω - din (B,G) iese în K
- η - din (B,F) iese în H
- θ - din (C,*) iese în K
- ϵ - din (*,*), i.e., de oriunde, iese în L

Exemplu: Ceas

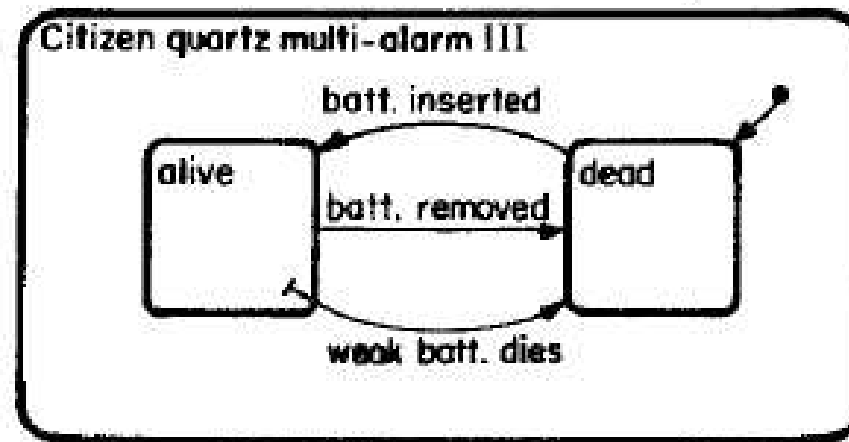
Stopwatch:

- zero - o componentă xor inițială cu cronometru off
- cealaltă componentă xor este o AND-stare, cu 2 componente display, run
- display are 2 afișări: normală și “lap”
- run are 2 stări: on-off



..Exemplu: Ceas

Baterie:

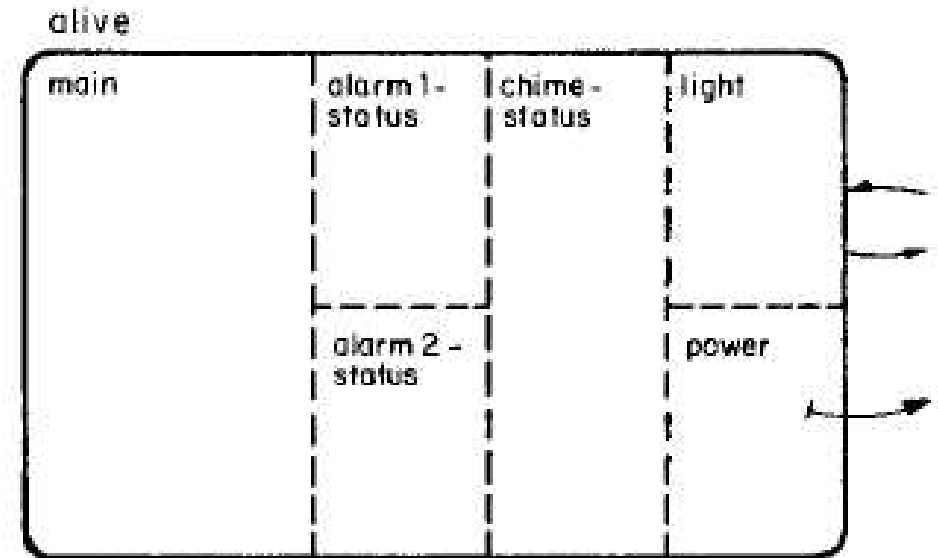


- 2 stări de bază `alive` și `dead`
- acțiuni/activități complexe au loc al trecerea dintr-o stare în alta

..Exemplu: Ceas

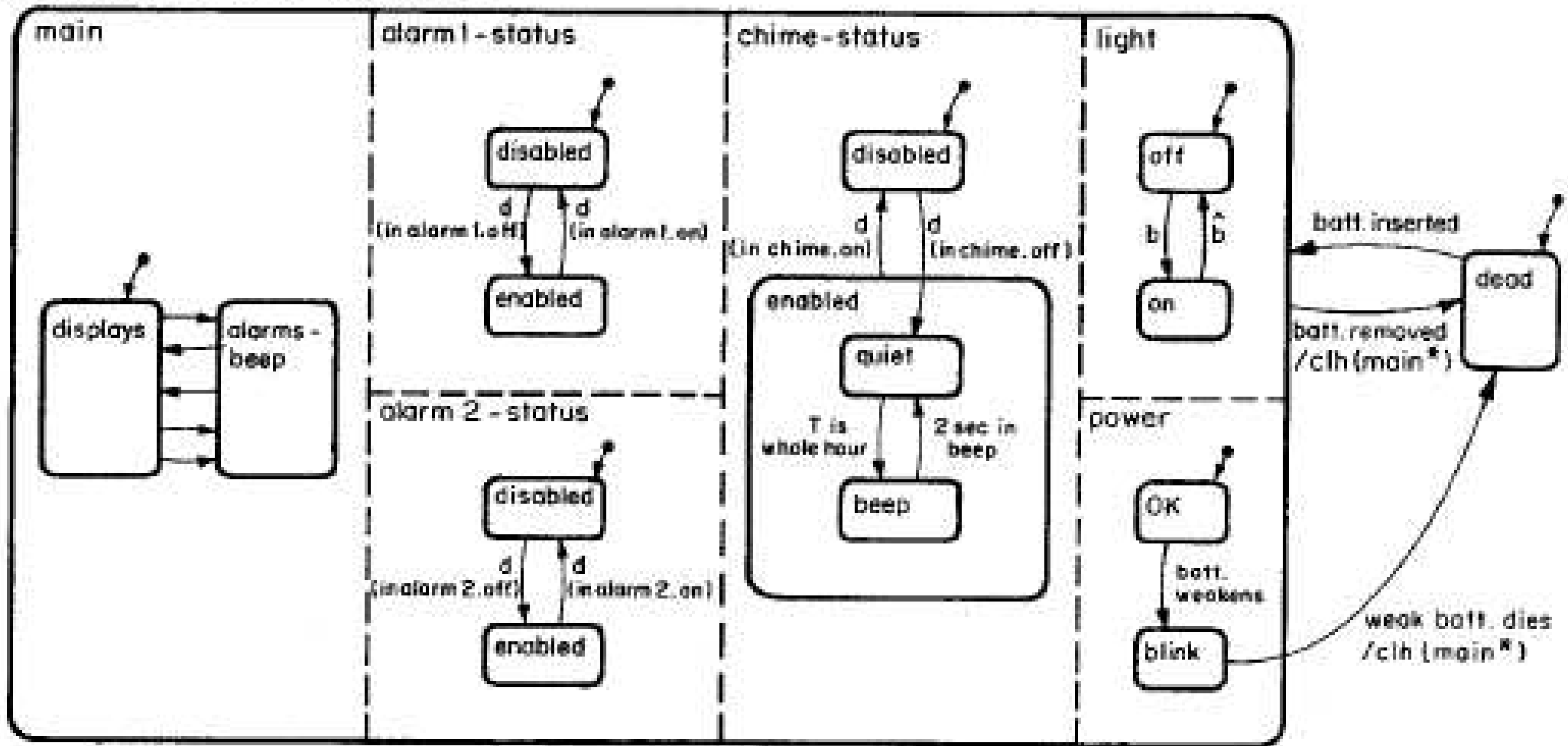
Structura generală, starea *alive*

- 6 componente ortogonale: main, 2 alarme, clinchet (chime), lumină, energie
- *disable/enable* din alarme și clinchet sunt legate de stările *on/off* din display (via condiții P1,P2,P)
- *H*-ul anterior trebuie nuanțat: dacă bateria este scoasă, nu se poate reveni la ultima stare vizitată - folosim acțiunea *clh* (clear-history) pentru modelare



..Exemplu: Ceas

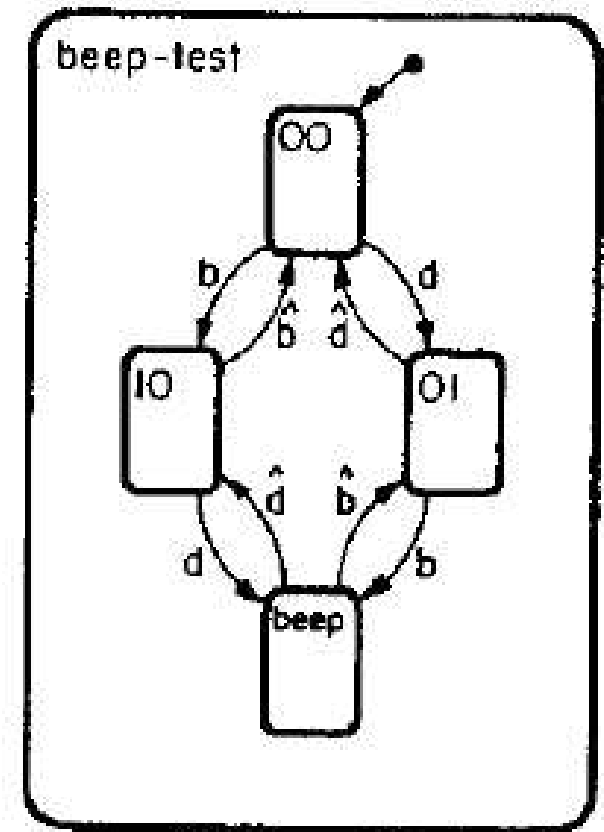
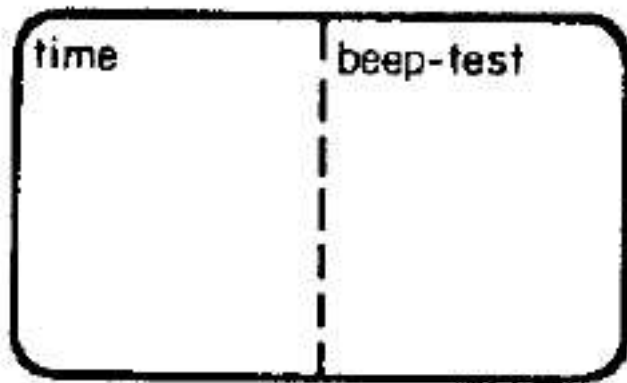
Citizen quartz multi - alarm III



..Exemplu: Ceas

Extensii - Test beeper:

- beeper-ul începe să ticăie dacă se apasă ambele b și d
- se aplică în paralel cu multe stări, dar nu cu toate (e.g., nu cu dead)
- exemplu de modelare a extensiei, adăugată la starea `time`:



- arată flexibilitatea modificări ulterioare a unui design cu statechart-uri



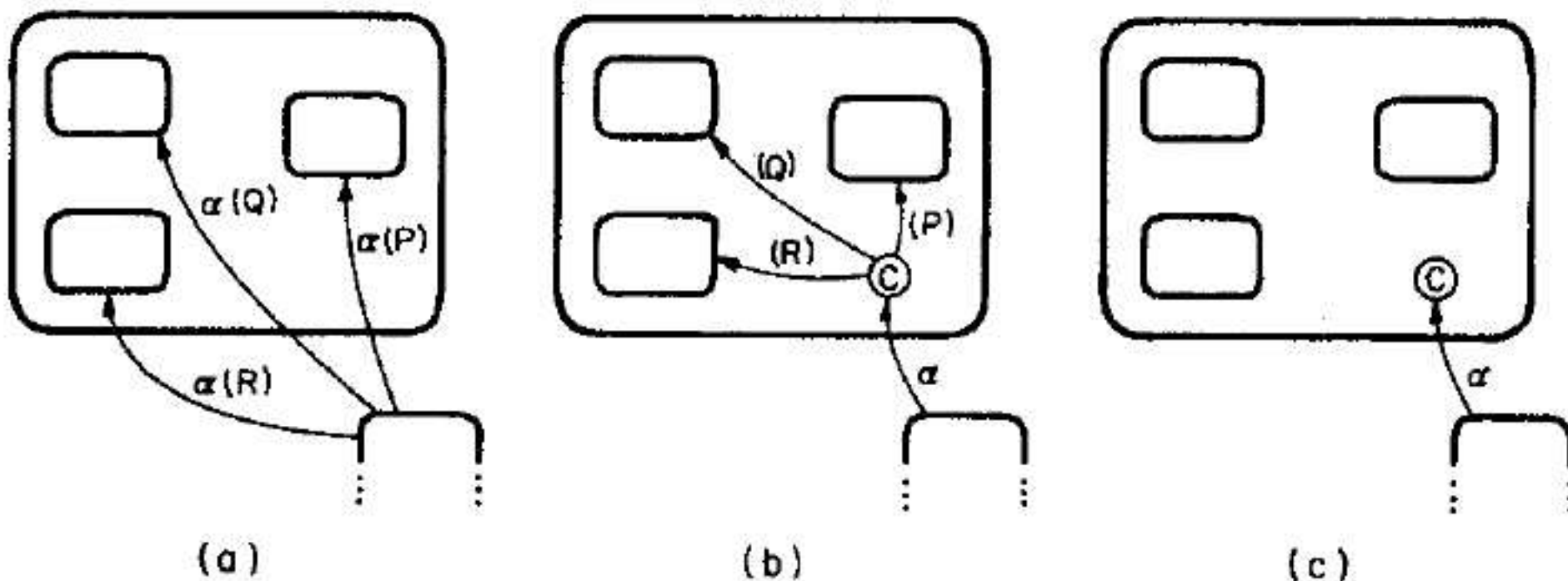
Cuprins:

- Generalități
- Grupări și rafinări
- Independență și concurență
- *Alte facilități*
- Concluzii, diverse, etc.

Conditii, intrari selectate

Conditii

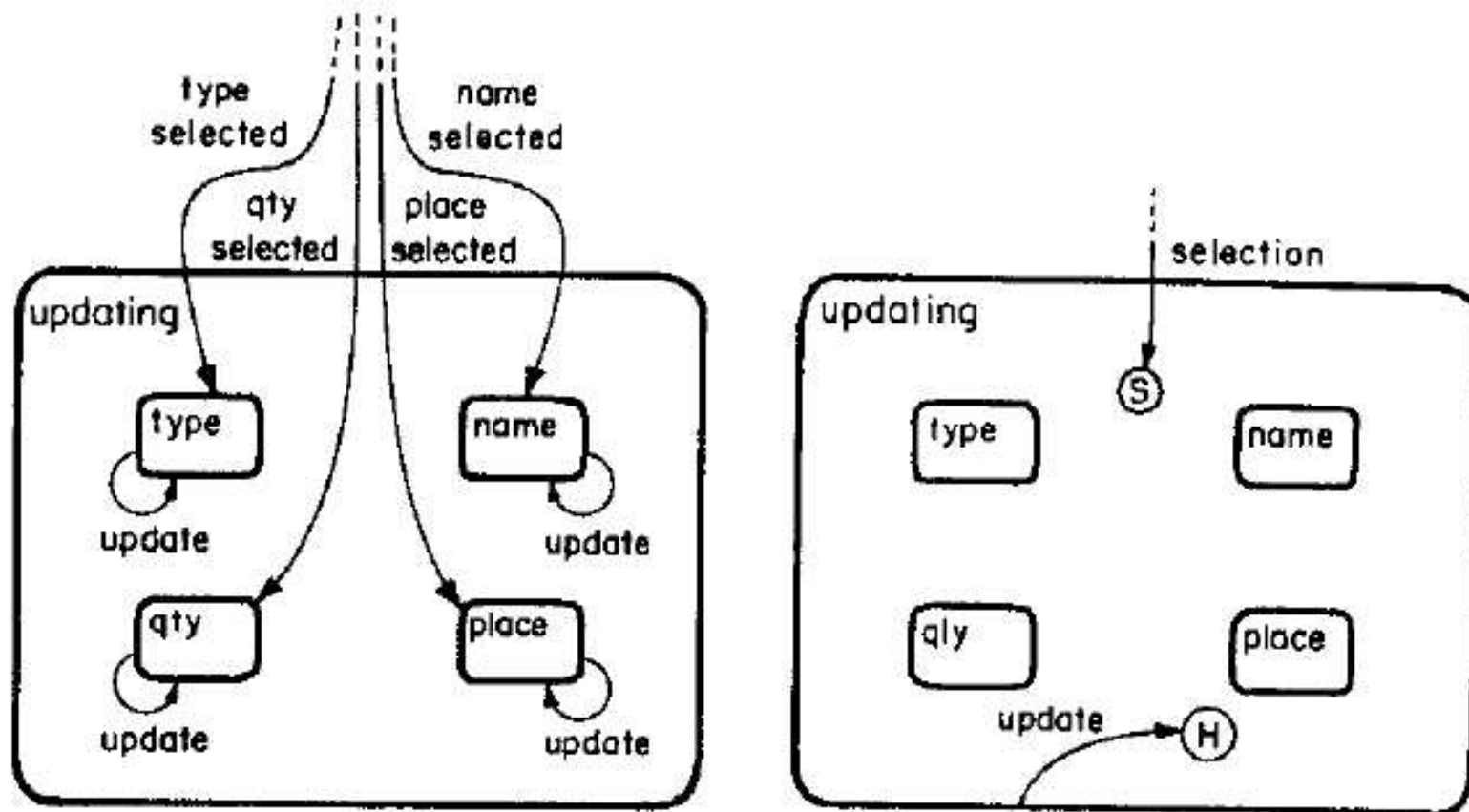
- tranzițiile/arcele pot fi și mai mult unificate dacă se folosesc *condiții C*
- exemplu:



..Conditii, intrari selectate

Selecții

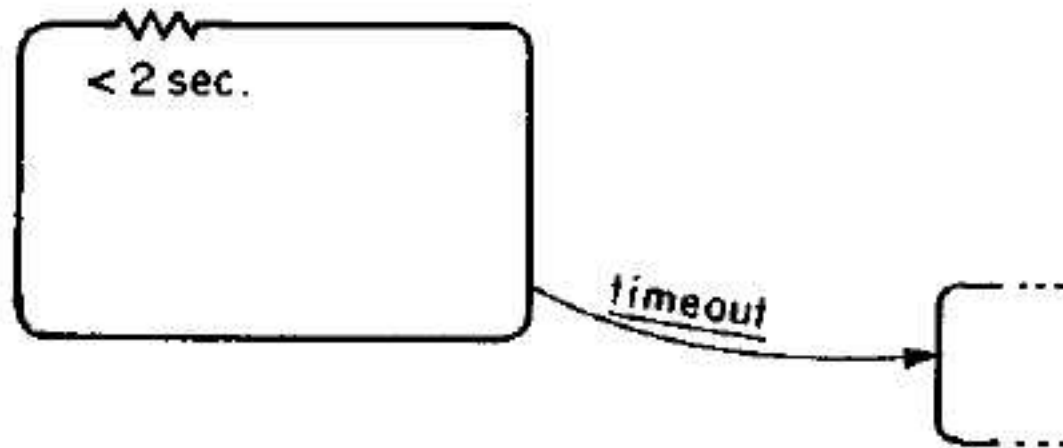
- un *selector* S este similar, dar alegerea se face dintr-o mulțime specifică de opțiuni, folosind un eveniment generic și valoarea sa concretă
- exemplu:



Delay, timeout

Timp

- se folosesc expresii *timeout(event,number)* - notează evenimentul care marchează trecerea a *number* unități de timp de la apariția evenimentului *event*
- se poate folosi o *notație grafică*

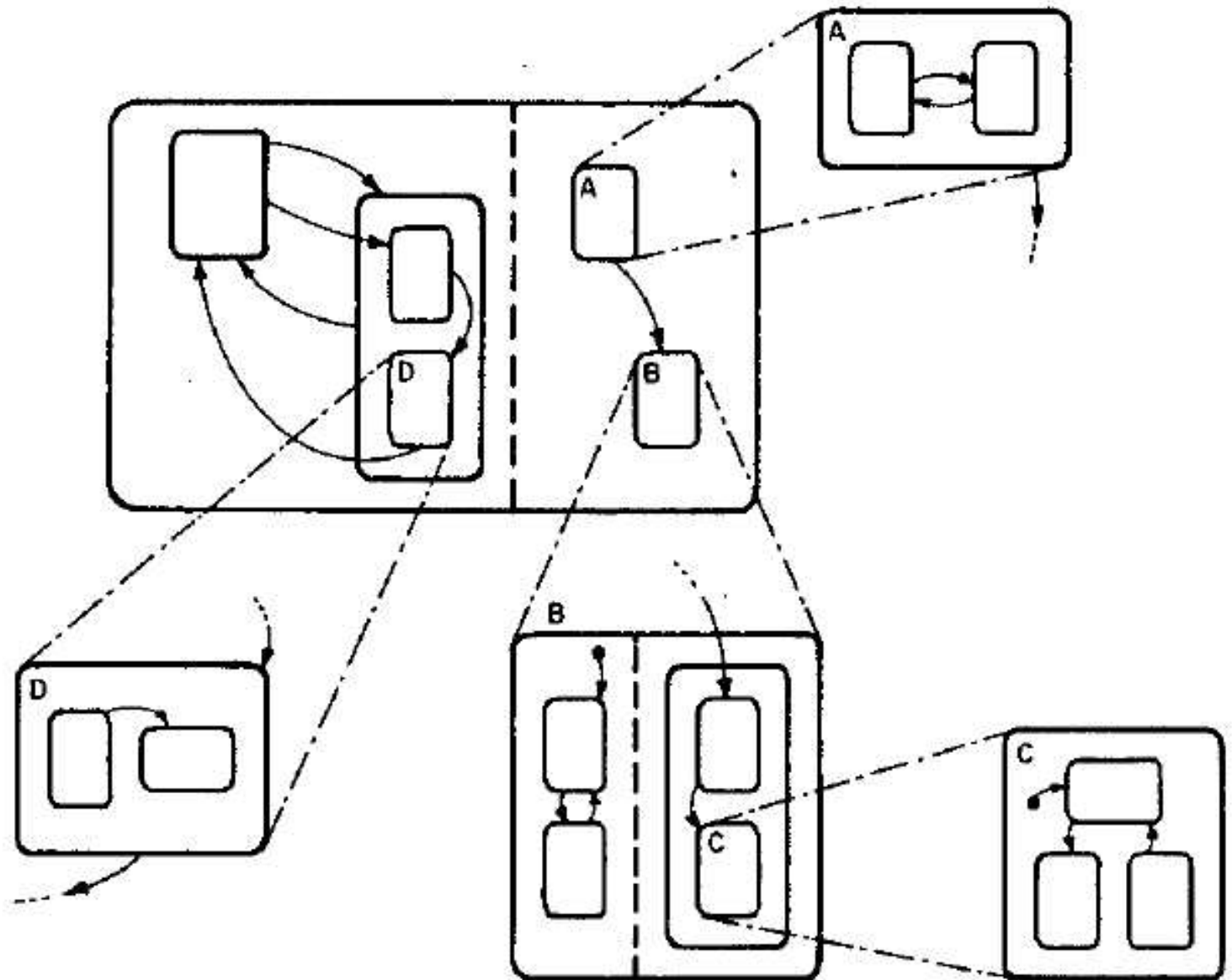


- se pot folosi și *expresii* mai complicate - intervale temporale, comparații, etc.

De-grupare

De-grupare

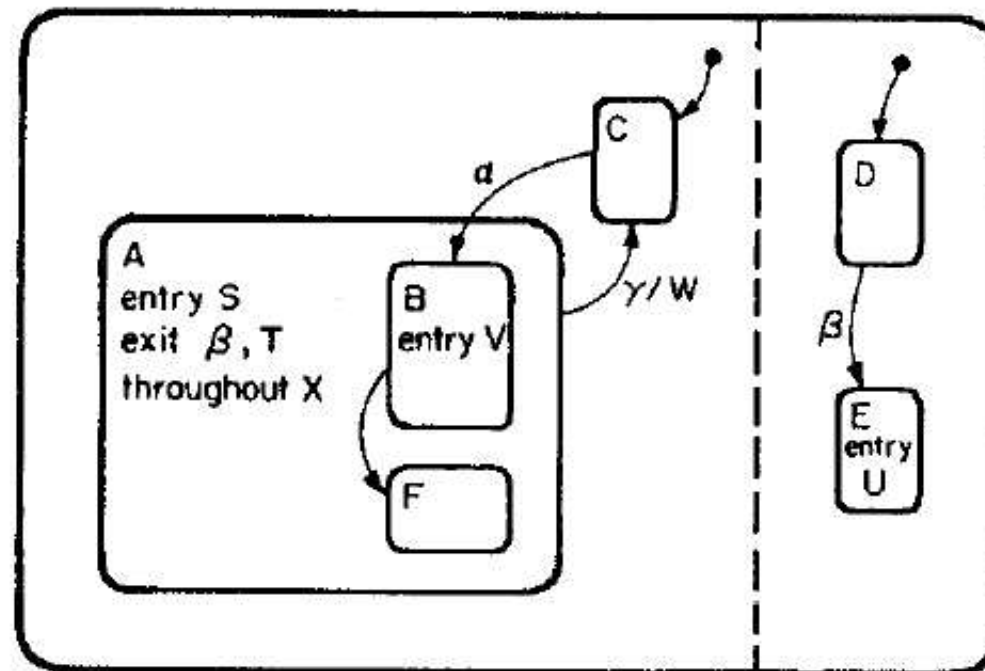
- pentru o mai mare flexibilitate, se pot *ataşa* descrieri de rafinări pentru stări din statechart-uri mai abstracte
- ... de folosit cu grijă că poate *de-structura* specificaţia



Actiuni vs. activitati

Actiuni

- se distinge între un *eveniment*, *acțiune* (care nu consumă timp) și o *activitate* (care durează)
- în statechart-uri (ca ulterior în UML) intrarea/ieșirea în/din stare poate fi acompaniată cu acțiuni specifice





..Actiuni vs. activitati

Activități

- activitățile pot fi modelate cu *start, throughout, stop* (cea din mijloc executându-se cât timp suntem în starea respectivă)

Exemplu (desenul anterior):

- fiind în (C,D), dacă apare α
 - se trece în (B,D)
 - se execută acțiunile S,V
- apoi, dacă apare γ
 - se iese din A
 - se execută acțiunile T,W; se execută și cauzând trecându-se, în a 2-a componentă, în E și executând U
- ...*semantica* acestor acțiuni poate fi destul de *delicată* (confuză)!

Cuprins:

- Generalități
- Grupări și rafinări
- Independență și concurență
- Alte facilități
- *Concluzii, diverse, etc.*



Concluzii, diverse, etc.

a se insera...