textual notations

grammars
production rules
CSP and event algebras

Textual - Grammars

- Regular expressions

```
sel-line click click* dble-click
```

- compare with JSD
  - same computational model
  - different notation
- BNF

```
expr ::= empty
       | atom expr
       | '(' expr ')' expr
```

- more powerful than regular exp. or STNs
- Still NO concurrent dialogue

Production rules

- Unordered list of rules:

  if *condition* then *action*

  - condition based on state or pending events
  - every rule always potentially active
- Good for concurrency
- Bad for sequence

Event based production rules

```
Sel-line → first
C-point first → rest
C-point rest → rest
D-point rest → < draw line >
```

- Note:
  - events added to list of pending events
  - '`first`' and '`rest`' are internally generated events
- Bad at state!

Prepositional Production System

- State based
- Attributes:
  - Mouse: { mouse-off, select-line, click-point, double-click }
  - Line-state: { menu, first, rest }
- Rules (feedback not shown):
  - select-line → mouse-off first
  - click-point first → mouse-off rest
  - click-point rest → mouse-off
  - double-click rest → mouse-off menu
- Bad at events!

CSP and process algebras

- used in Alexander's SPI, and Agent notation
- good for sequential dialogues

```
Bold-tog = select-bold? → bold-on → select-bold? →
                                                bold-off → Bold-tog

Italic-tog = . . .
Under-tog = . . .
```

- and concurrent dialogue

```
Dialogue-box = Bold-tog || Italic-tog || Under-tog
```

- but causality unclear

Semantics Alexander SPI (i)

- Two part specication:
  - EventCSP - pure dialogue order
  - EventISL - target dependent semantics
- dialogue description - centralised
- syntactic/semantic trade-off - tollerable

Semantics Alexander SPI (ii)

- EventCSP

```
Login  = login-mess -> get-name -> Passwd
Passwd = passwd-mess -> (invalid -> Login [] valid -> Session)
```

- EventISL

```
event: login-mess
     prompt: true
     out: "Login:"
event: get-name
     uses: input
     set: user-id = input
event: valid
     uses: input, user-id, passwd-db
     when:  passwd-id = passwd-db(user-id)
```

Semantics - raw code

- event loop for word processor
- dialogue description
  - very distributed
- syntactic/semantic trade-off
  - terrible!

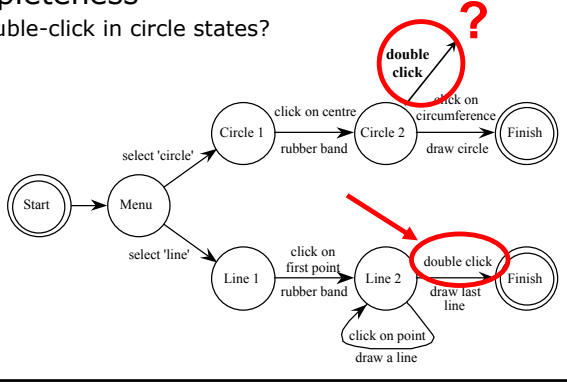
```
switch ( ev.type ) {
  case button_down:
    if ( in_text ( ev.pos ) ) {
        mode = selecting;
        mark_selection_start(ev.pos);
    }
    ...
  case button_up:
    if ( in_text ( ev.pos )
            && mode == selecting ) {
        mode = normal;
        mark_selection_end(ev.pos);
    }
    ...
  case mouse_move:
    if (mode == selecting ) {
        extend_selection(ev.pos);
    }
    ...
} /* end of switch */
```

Action properties

- completeness
  - missed arcs
  - unforeseen circumstances
- determinism
  - several arcs for one action
  - deliberate: application decision
  - accident: production rules
- nested escapes
- consistency
  - same action, same effect?
  - modes and visibility

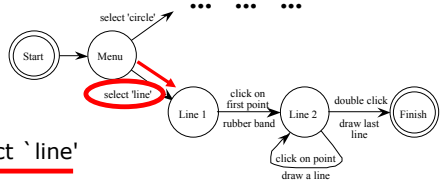
Checking properties (i)

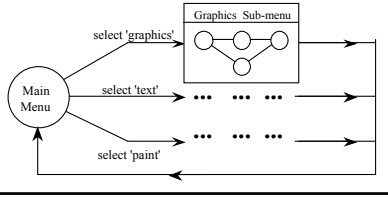
- completeness
  - double-click in circle states?



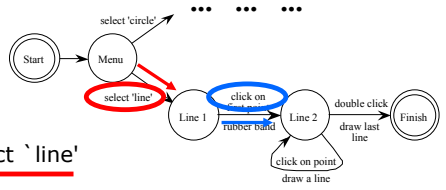
Checking properties (ii)

- Reversibility:
  - to reverse select `line'

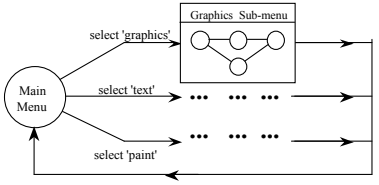




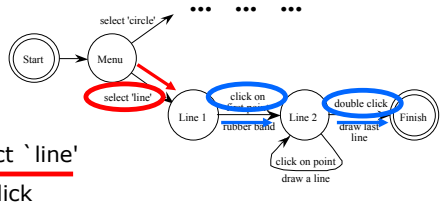
Checking properties (ii)



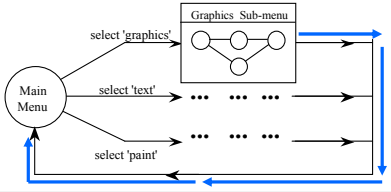
- Reversibility:
  - to reverse select `line'
  - click



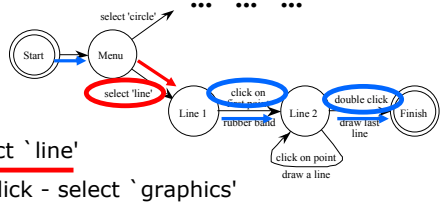
Checking properties (ii)



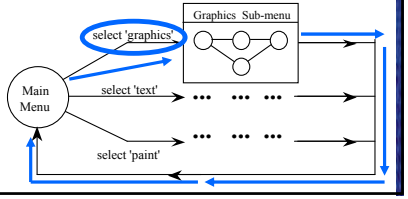
- Reversibility:
  - to reverse select `line'
  - click - double click



Checking properties (ii)



- Reversibility:
  - to reverse select `line'
  - click - double click - select `graphics'
  - (3 actions)
- N.B. not undo

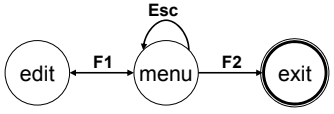


State properties

- reachability
  - can you get anywhere from anywhere?
  - and how easily
- reversibility
  - can you get to the previous state?
  - but NOT undo
- dangerous states
  - some states you don't want to get to

Dangerous States

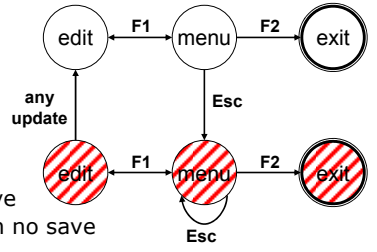
- word processor: two modes and exit
  - F1 - changes mode
  - F2 - exit (and save)
  - Esc - no mode change



but ... Esc resets autosave

Dangerous States (ii)

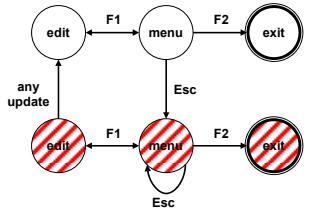
- exit with/without save ⇒ dangerous states
- duplicate states - semantic distinction



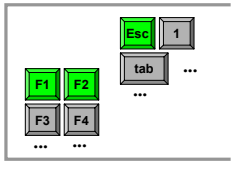
F1-F2 - exit with save
F1-Esc-F2 - exit with no save

layout matters

- word processor - dangerous states

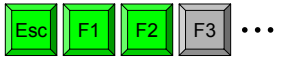


- old keyboard - OK



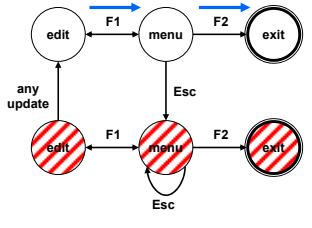
layout matters

- new keyboard layout



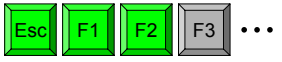
intend F1-F2 (save)

finger catches Esc



layout matters

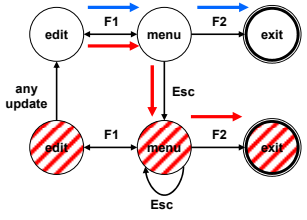
- new keyboard layout



intend F1-F2 (save)

finger catches Esc

F1-Esc-F2 - disaster!



HUMAN-COMPUTER INTERACTION

Dialogue Analysis - Summary

- Semantics and dialogue
  - attaching semantics
  - distributed/centralised dialogue description
  - maximising syntactic description
- Properties of dialogue
  - action properties: completeness, determinism, consistency
  - state properties: reachability, reversibility, dangerous states
- Presentation and lexical issues
  - visibility, style, layout
  - N.B. not independent of dialogue