Slide 1

textual notations

grammars
production rules
CSP and event algebras

Slide 2

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

Slide 3

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

Slide 4

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!

5

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

6

Semantics Alexander SPI (i)

- Two part specification:
  - EventCSP - pure dialogue order
  - EventISL - target dependent semantics

- dialogue description - centralized

- syntactic/semantic trade-off - tolerable

7

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)
  ```

8

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 */          9
```

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?
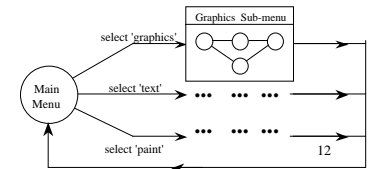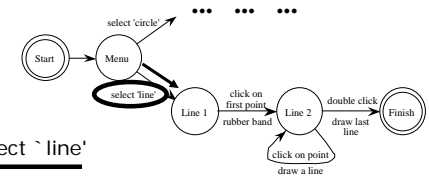 - modes and visibility

10

Checking properties (i)

- completeness
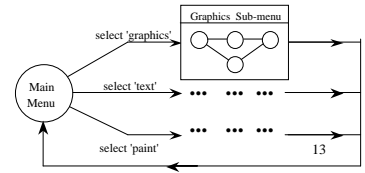 - double-click in circle states?
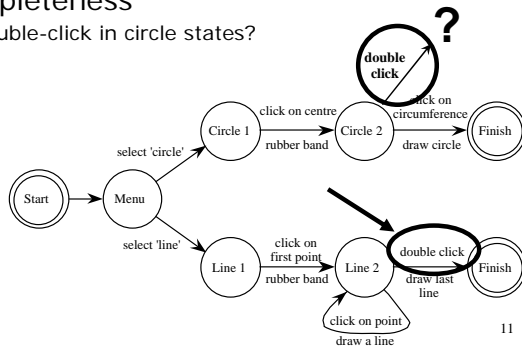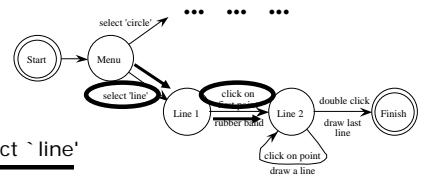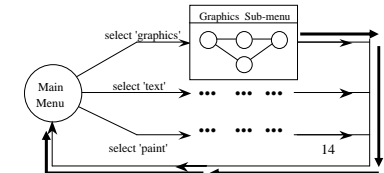

11

Checking properties (ii)
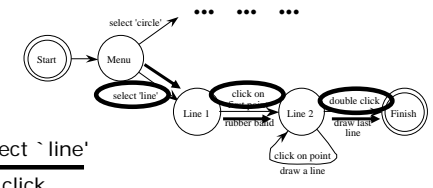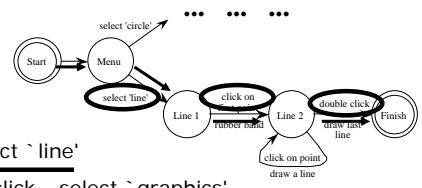
- Reversibility:
  - to reverse select `line'

12

Slide 13

Checking properties (ii)

select 'circle'
Start Menu
select 'line'
Line 1   click on   Line 2   double click   Finish
rubber band   draw last line
click on point
draw a line

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

Graphics Sub-menu
select 'graphics'
Main Menu   select 'text'
select 'paint'

13

Slide 14

Checking properties (ii)

select 'circle'
Start Menu
select 'line'
Line 1   click on   Line 2   double click   Finish
rubber band   draw last line
click on point
draw a line

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

Graphics Sub-menu
select 'graphics'
Main Menu   select 'text'
select 'paint'

14

Slide 15

Checking properties (ii)

select 'circle'
Start Menu
select 'line'
Line 1   click on   Line 2   double click   Finish
rubber band   draw last line
click on point
draw a line

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

Graphics Sub-menu
select 'graphics'
Main Menu   select 'text'
select 'paint'
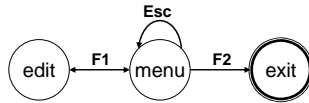
15

Slide 16
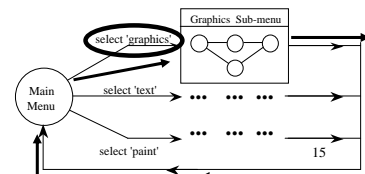
State properties

- 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

16

Dangerous States

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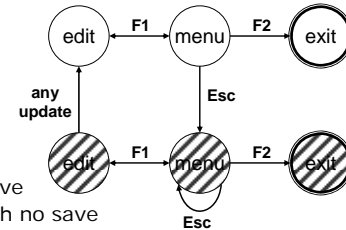 - F2 - exit (and save)
  - Esc - no mode change

edit → F1 → menu (Esc loop) → F2 → exit

but … Esc resets autosave

17

Dangerous States (ii)

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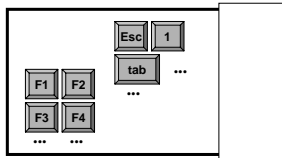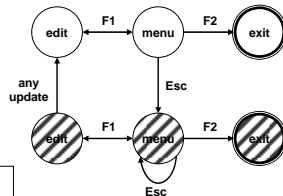- duplicate states - semantic distinction

edit → F1 → menu → F2 → exit
any update / Esc
edit → F1 → menu → F2 → exit (Esc loop)

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

18

---

layout matters

- word processor - dangerous states

edit → F1 → menu → F2 → exit
any update / Esc
edit → F1 → menu → F2 → exit (Esc loop)

- old keyboard - OK

Esc | 1
tab | ···
F1 | F2
F3 | F4
··· ···

19

---

layout matters

- new keyboard layout

Esc | F1 | F2 | F3 ···

intend F1-F2 (save)
→
finger catches Esc

edit → F1 → menu → F2 → exit
any update / Esc
edit → F1 → menu → F2 → exit (Esc loop)
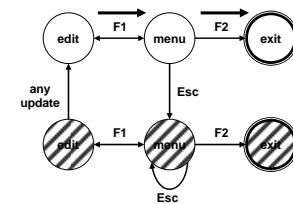
20

layout matters

- new keyboard layout

| Esc | F1 | F2 | F3 | ⋯ |
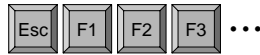
intend F1-F2 (save)

finger catches Esc

F1-Esc-F2 - disaster!


```
edit  --F1-->  menu  --F2-->  exit

any update

Esc

edit  --F1-->  menu  --F2-->  exit

Esc
```
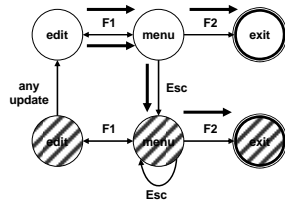

Dialogue Analysis - Summary

- Semantics and dialogue
  - attaching semantics
  - distributed/centralized dialogue description
  - maximizing 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
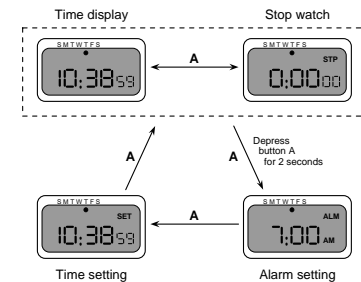

Digital watch – User Instructions

- two main modes

- limited interface - 3 buttons

- button A changes mode

Time display Stop watch

Depress button A for 2 seconds

Time setting Alarm setting
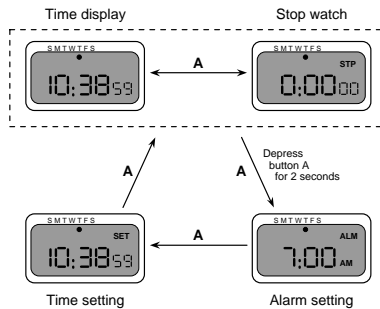
Digital watch – User Instructions

- dangerous states
  - *guarded*
    ... by two second hold

- completeness
  - distinguish depress A and release A
  - what do they do in all modes?

Time display Stop watch

Depress button A for 2 seconds

Time setting Alarm setting

Digital watch – Designers instructions

and …

that's just
one button

Time display   Stop watch

Depress A

Release A   Release A

Depress A   2 seconds   2 seconds

Release A   Depress A   Release A

Time setting   Alarm setting   25