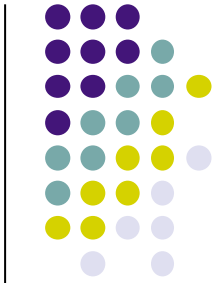
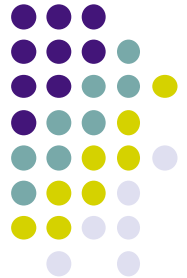
How to execute a program using the semantics



- We execute the program using the semantics by following two steps
- First, we translate the program into kernel language
  - The kernel language is a simple language that has all essential concepts
  - All programs in the practical language can be translated into kernel language
- Second, we execute the translated program on the abstract machine
  - The abstract machine is a simplified computer with a precise mathematical definition

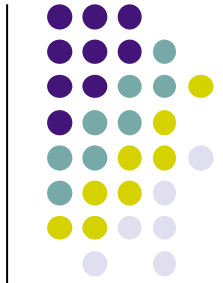
→ Let's take a closer look at the abstract machine

Kernel language of the functional paradigm



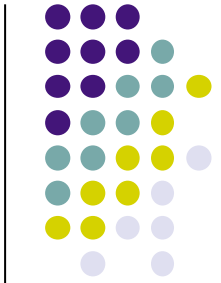
- <s> ::= **skip**
  | <s>$_1$ <s>$_2$
  | **local** <x> **in** <s> **end**
  | <x>$_1$=<x>$_2$
  | <x>=<v>
  | **if** <x> **then** <s>$_1$ **else** <s>$_2$ **end**
  | {<x> <y>$_1$ … <y>$_n$}
  | **case** <x> **of** <p> **then** <s>$_1$ **else** <s>$_2$ **end**
- <v> ::= <number> | <procedure> | <record>
- <number> ::= <int> | <float>
- <procedure> ::= **proc** {$ <x>$_1$ … <x>$_n$} <s> **end**
- <record>, <p> ::= <lit> | <lit>(<f>$_1$:<x>$_1$ … <f>$_n$:<x>$_n$)

Abstract machine concepts



- Single-assignment memory $\sigma = \{x_1=10, x_2, x_3=20\}$
  - Variables and the values they are bound to
- Environment $E = \{X \rightarrow x, Y \rightarrow y\}$
  - Link between identifiers and variables in memory
- Semantic instruction $(\langle s \rangle, E)$
  - An instruction with its environment
- Semantic stack $ST = [(\langle s \rangle_1, E_1), \ldots, (\langle s \rangle_n, E_n)]$
  - A stack of semantic instructions
- Execution state (ST, σ)
  - A pair of a semantic stack and the memory
- Execution $(ST_1, \sigma_1) \rightarrow (ST_2, \sigma_2) \rightarrow (ST_3, \sigma_3) \rightarrow \ldots$
  - A sequence of execution states

Abstract machine execution algorithm



- **procedure** execute(<s>)
  begin
  ST:=[(<s>,{})]; /* Initial semantic stack: empty environment */
  σ:={}; /* Initial memory: empty (no variables) */
  while (ST≠{}) **do**
  pop(ST, SI); /* Pop semantic instruction into SI */
  (ST,σ):=rule(SI, (ST,σ)); /* Execute SI */
  end
  end

- While the semantic stack is nonempty, pop the instruction at the top of the semantic stack, and execute it according to its semantic rule
- Each instruction of the kernel language has a rule that defines its execution in the abstract machine
- (Note: When we introduce concurrency, we will extend this algorithm to run with more than one semantic stack)