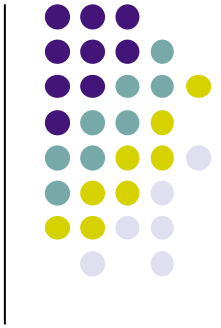
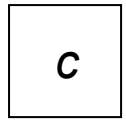
Adding explicit state to the language

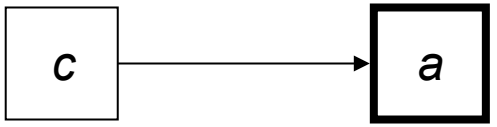


- We can make the state explicit by extending the language
- With this extension a program can directly observe the sequence of values in time
  - This was not possible in the functional paradigm
- We call our extension a cell
  - The word “cell” is chosen to avoid confusion with related terms, such as the overused word “variable”
- A cell is a box with a content
  - The content can be changed but the box remains the same
  - The same cell can have different contents: we can observe change
  - The sequence of contents is a state



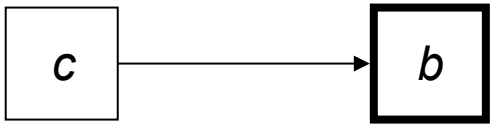
An unbound variable

Creating a cell with initial content *a* (=5)



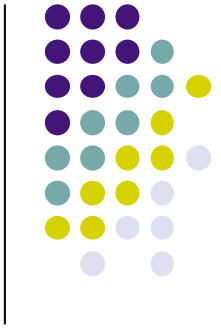
cell

Replace the content by another variable *b* (=6)



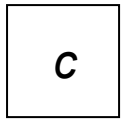
cell

A cell



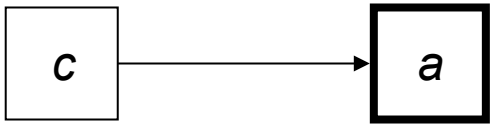
- A cell is a box with an identity and a content
  - The identity is a constant
    (the “name” or “address” of the cell)
  - The content is a variable
    (in the single-assignment store)
- The content can be replaced by another variable

```
A=5
B=6
C={NewCell A}  % Create a cell
{Browse @C}    % Display content
C:=B           % Change content
{Browse @C}    % Display content
```



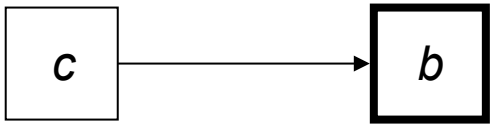
An unbound variable

Creating a cell with initial content *a* (=5)



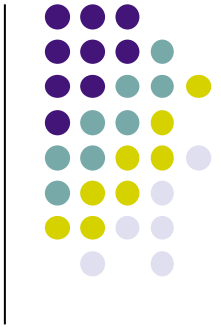
cell

Replace the content by another variable *b* (=6)



cell

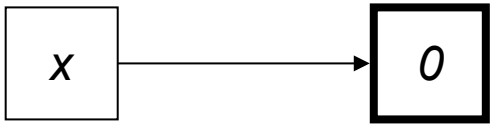
Adding cells to the kernel language



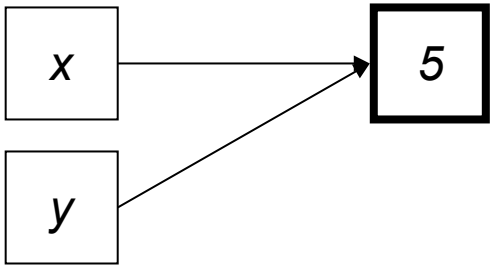
- We add cells and their operations
  - Cells have three operations
- C={NewCell A}
  - Create a new cell with initial content A
  - Bind C to the cell's identity
- C:=B
  - Check that C is bound to a cell's identity
  - Replace the cell's content by B
- Z=@C
  - Check that C is bound to a cell's identity
  - Bind Z to the cell's content

Some examples (1)

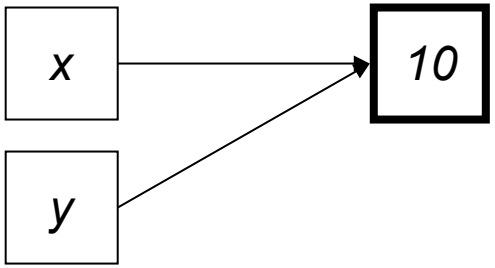
- X={NewCell 0}



- X:=5
- Y=X

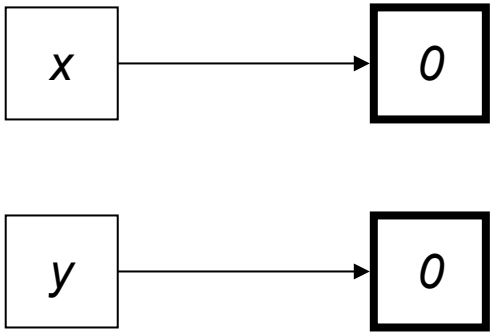


- Y:=10
- @X==10 % true
- X==Y % true



Some examples (2)

- X={NewCell 0}
- Y={NewCell 0}



- X==Y % false
- Because X and Y refer to different cells, with different identities

- @X==@Y % true
- Because the contents of X and Y are the same value