Separation Logic

Herbert Kruitbosch

April 17, 2013

Propositional dynamic logic

$$P \rightarrow \boxed{C} Q$$

Propositional dynamic logic

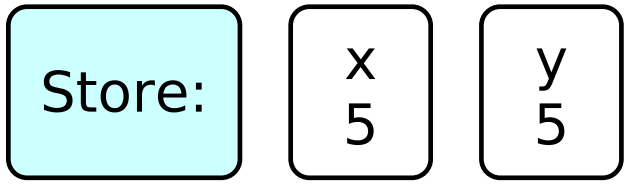
$$P \rightarrow \boxed{C} Q$$

Different notation:

$$\{P\} \quad \mathsf{C}; \quad \{Q\}$$

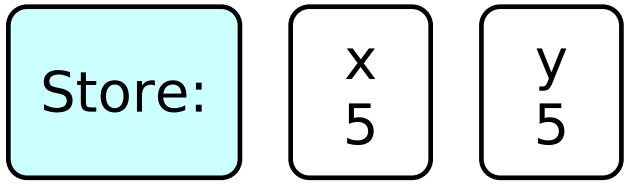
Store

```
int x=5;
int y=5;
```



Store

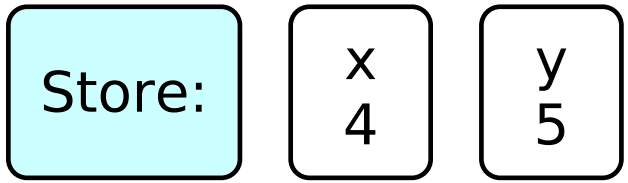
```
int x=5;
int y=5;
```



$\{x = 5, y = 5\}$

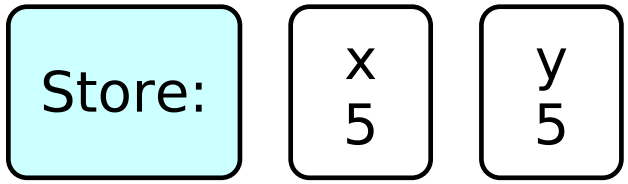
```
x=4;
```

$\{x = 4, y = 5\}$



Store

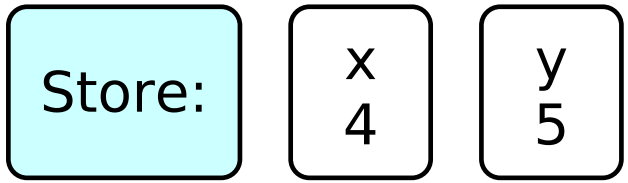
```
int x=5;
int y=5;
```



$\{x = 5, y = 5\}$

```
x=4;
```

$\{x = 4, y = 5\}$



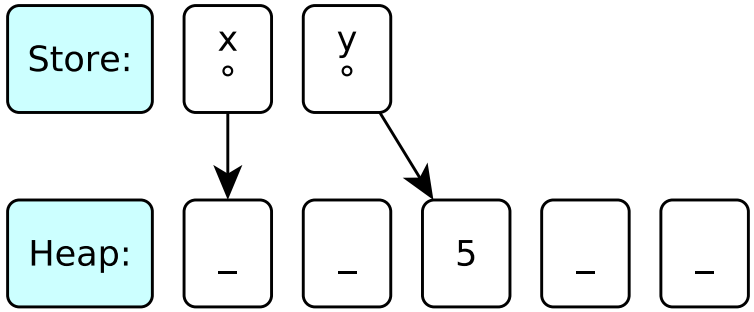
Formally:

$\{x = \_, y = M\}$ x:=N; $\{x = N, y = M\}$

$(x = \_ \text{ and } y = M) \rightarrow \boxed{\text{x:=N}}(x = N \text{ and } y = M)$

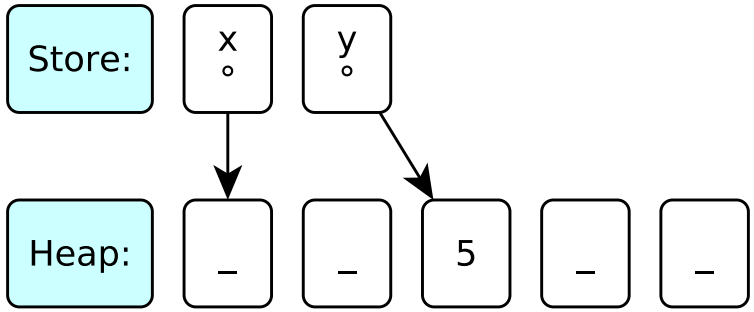
Heap

```
int *x=malloc(5*sizeof(int));
int *y=x+2;
x[2]=5;
y[0]=5;
```



Heap

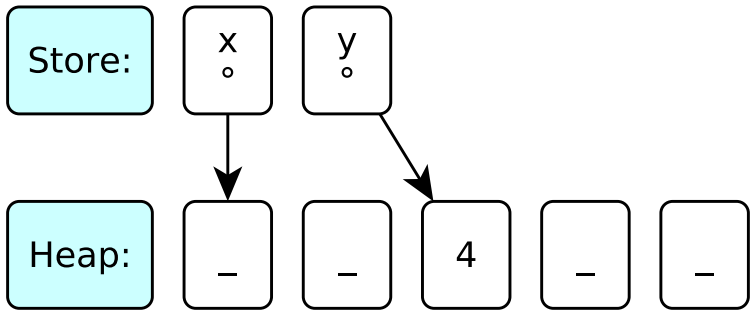
```
int *x=malloc(5*sizeof(int));
int *y=x+2;
x[2]=5;
y[0]=5;
```



$\{x[2] = 5, y[0] = 5\}$

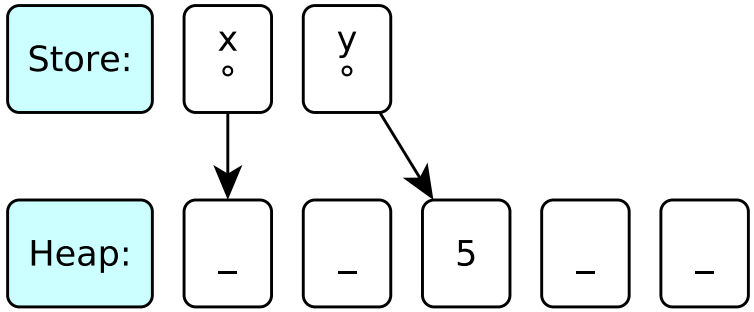
```
x[2]=4;
```

$\{x[2] = 4, y[0] = 5\}$ $\leftarrow$ **FAIL**



Heap

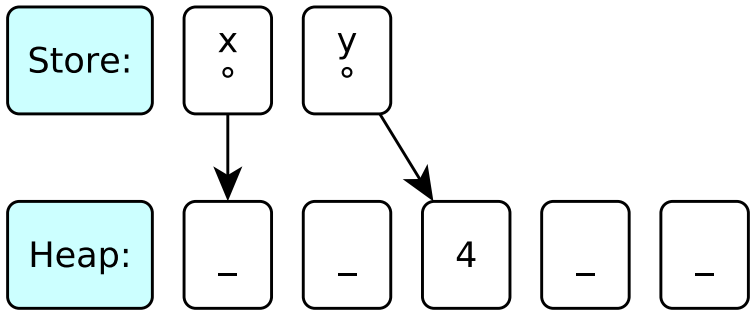
```
int *x=malloc(5*sizeof(int));
int *y=x+2;
x[2]=5;
y[0]=5;
```



$\{x[2] = 5, y[0] = 5\}$

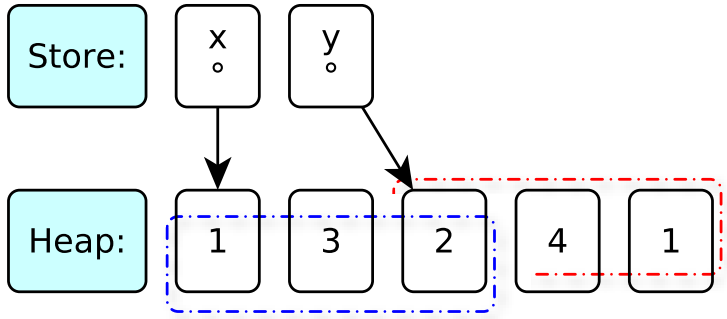
```
x[2]=4;
```

$\{x[2] = 4, y[0] = 5\}$ $\leftarrow$ **FAIL**



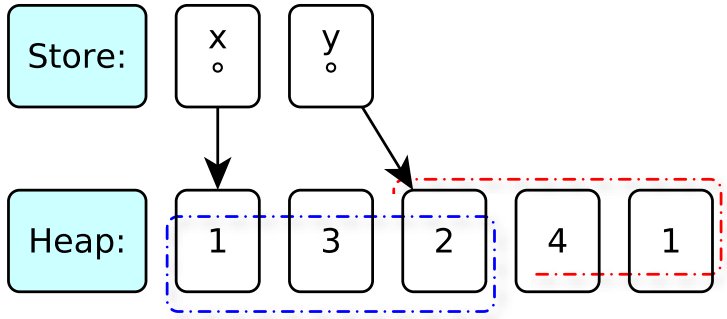
Merge Sort

```
int *x=malloc(5*sizeof(int));
int *y=x+2;
```

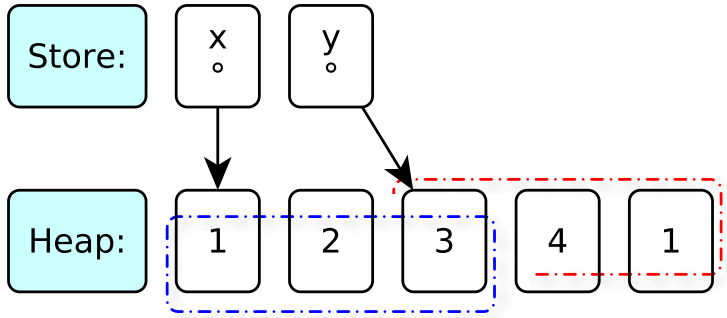


Merge Sort

```
int *x=malloc(5*sizeof(int));
int *y=x+2;
```

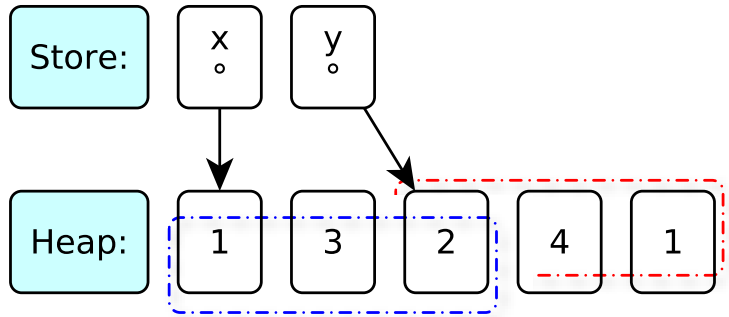


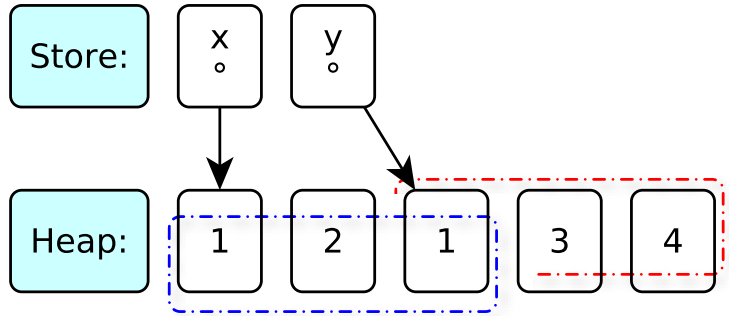
{true}
sort x[0], x[1] and x[2]
$\{x[0] < x[1] < x[2]\}$



Merge Sort

```
int *x=malloc(5*sizeof(int));
int *y=x+2;
```



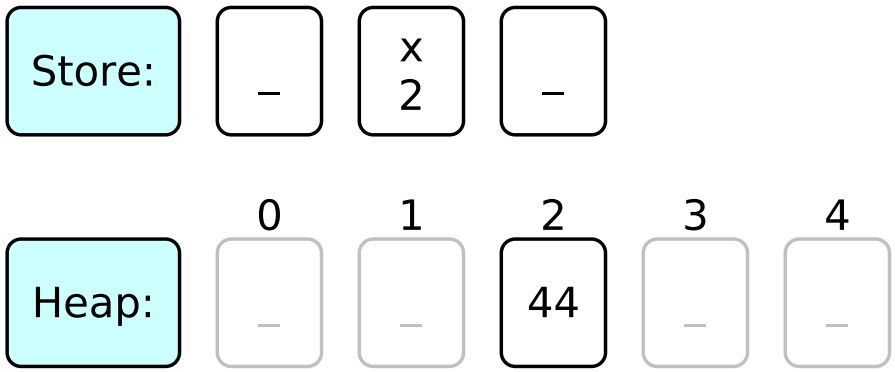


{true}
sort x[0], x[1] and x[2]
$\{x[0] < x[1] < x[2]\}$
sort y[0], y[1] and y[2]
$\{x[0] < x[1] < x[2] \text{ and } y[0] < y[1] < y[2]\}$ ← **FAIL**

Seperation Logic

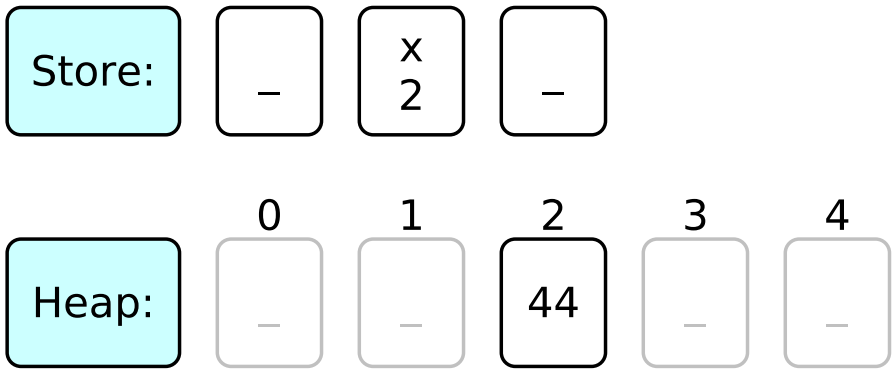
Model

1. Variables: x, y, etc
2. Locations: 0, 1, 2, 3, etc
3. Store: Variable $\rightarrow$ Integer
4. Heap: Location $\rightarrow$ Integer
5. States: Store $\times$ Heap



Model

1. Variables: x, y, etc
2. Locations: 0, 1, 2, 3, etc
3. Store: Variable $\rightarrow$ Integer
4. Heap: Location $\rightarrow$ Integer
5. States: Store $\times$ Heap



Assume a store s and a heap h then for example $s, h \models P$

Assertions

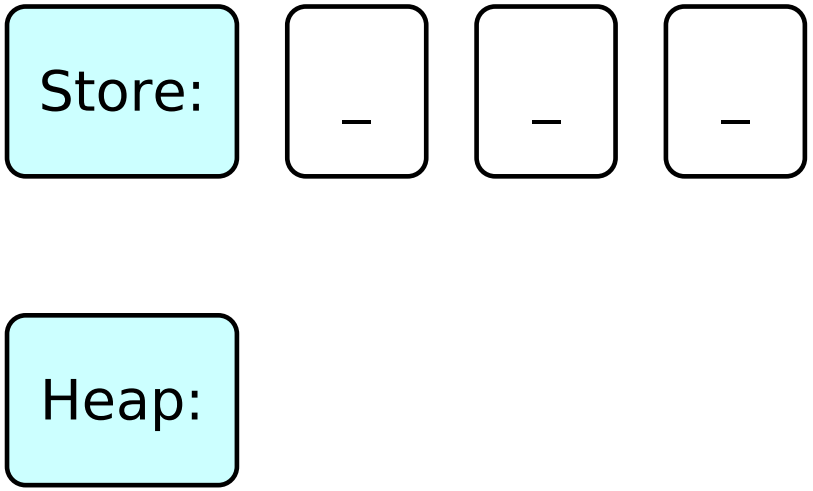
$P =$

1. Default: B, *false*, *true*, $p \Rightarrow Q$, $\forall x.P$, $x = 5$, etc

Assertions

$P =$

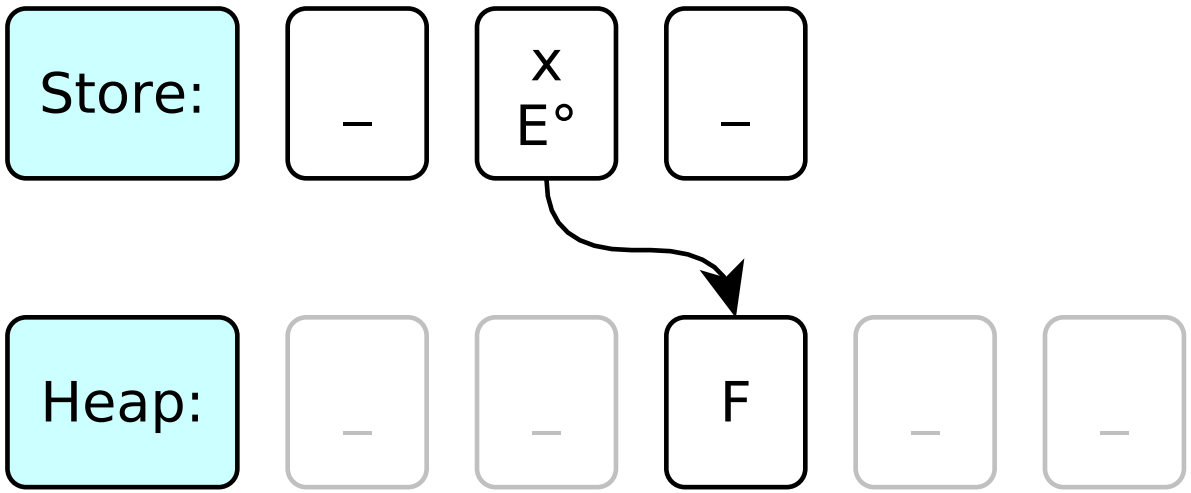
1. Default: B, *false*, *true*, $p \Rightarrow Q$, $\forall x.P$, $x = 5$, etc
2. $s, h \models \text{emp}$



Assertions

$P =$

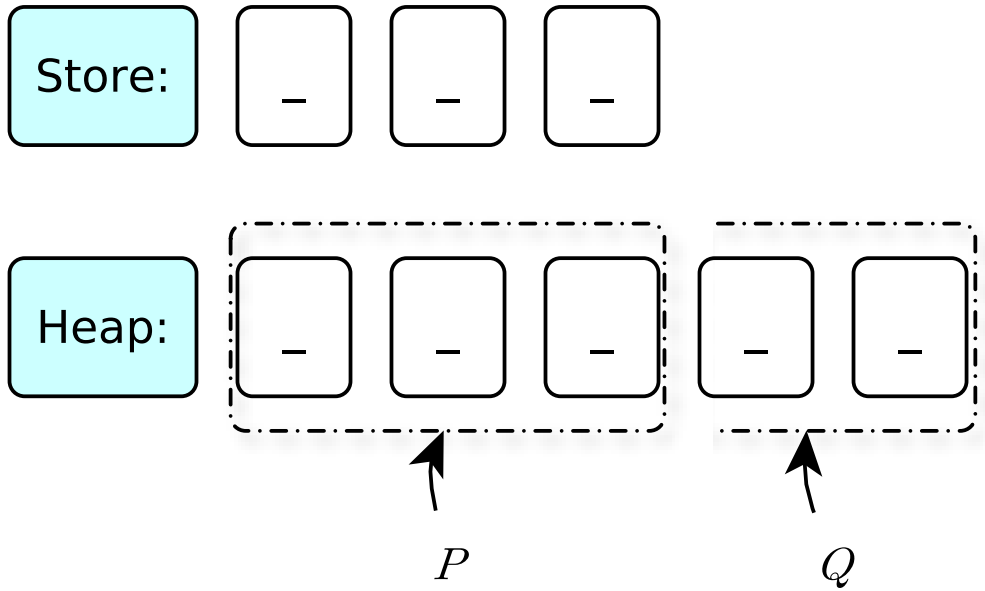
1. Default: B, *false*, *true*, $P \Rightarrow Q$, $\forall x.P$, $x = 5$, etc
2. $s, h \models \text{emp}$
3. $s, h \models E \mapsto F$



Assertions

$P =$

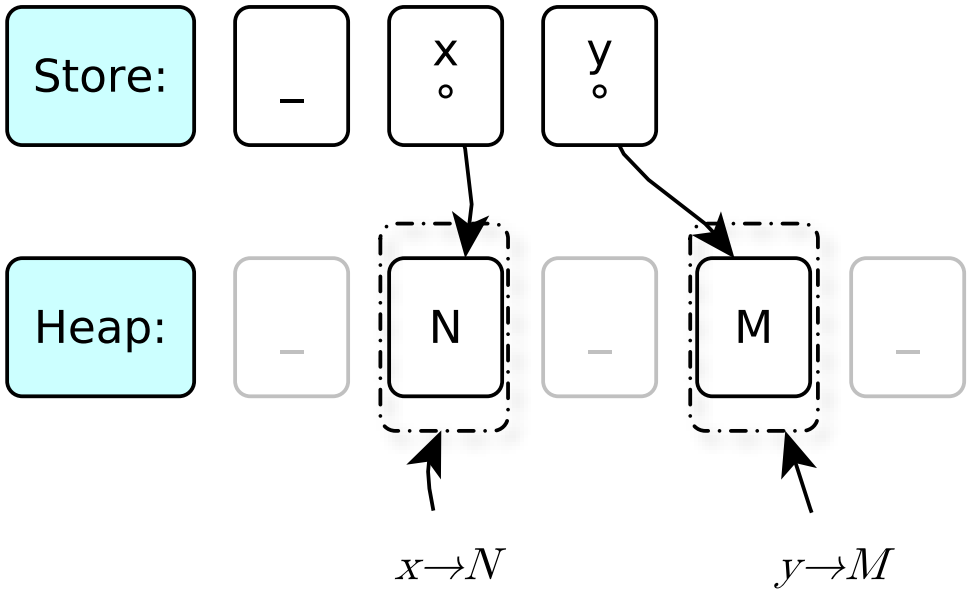
1. Default: B, *false*, *true*, $P \Rightarrow Q$, $\forall x.P$, $x = 5$, etc
2. $s, h \models \text{emp}$
3. $s, h \models E \mapsto F$
4. $s, h \models P * Q$



Assertions

$P =$

1. Default: B, *false*, *true*, $P \Rightarrow Q$, $\forall x.P$, $x = 5$, etc
2. $s, h \models \text{emp}$
3. $s, h \models E \mapsto F$
4. $s, h \models (x \mapsto N) * (y \mapsto M)$

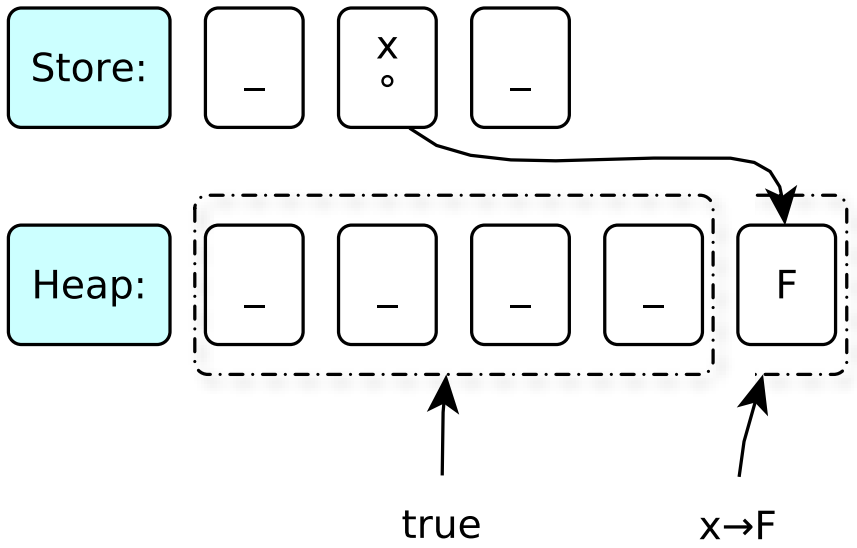


Assertions

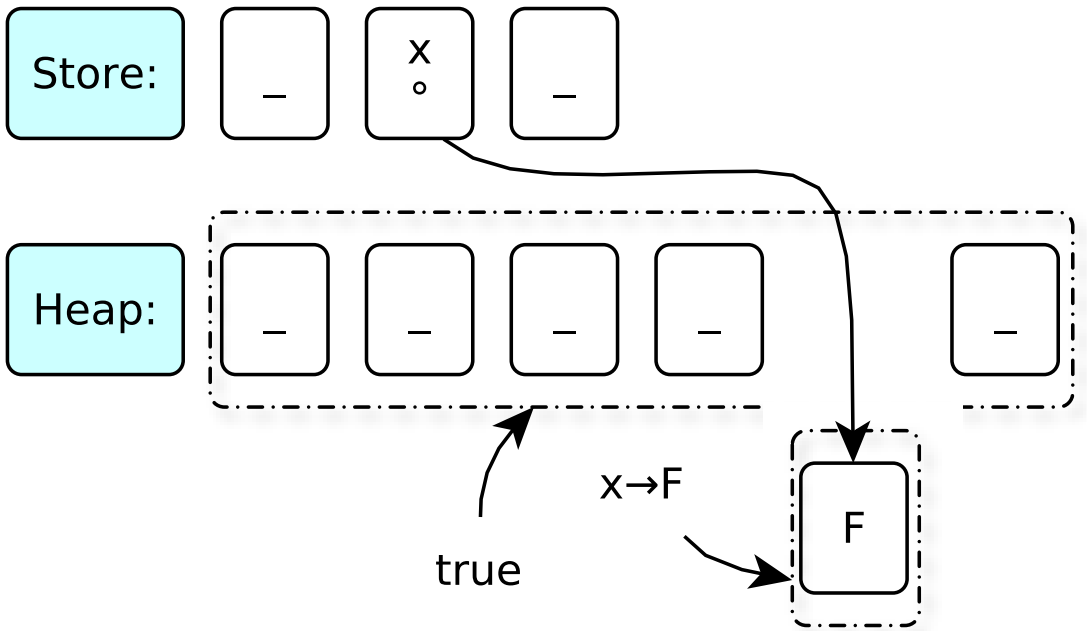
$P =$

1. Default: B, *false*, *true*, $P \Rightarrow Q$, $\forall x.P$, $x = 5$, etc
2. $s, h \models \text{emp}$
3. $s, h \models E \mapsto F$
4. $s, h \models P * Q$
5. $s, h \models P -\!\!* Q$

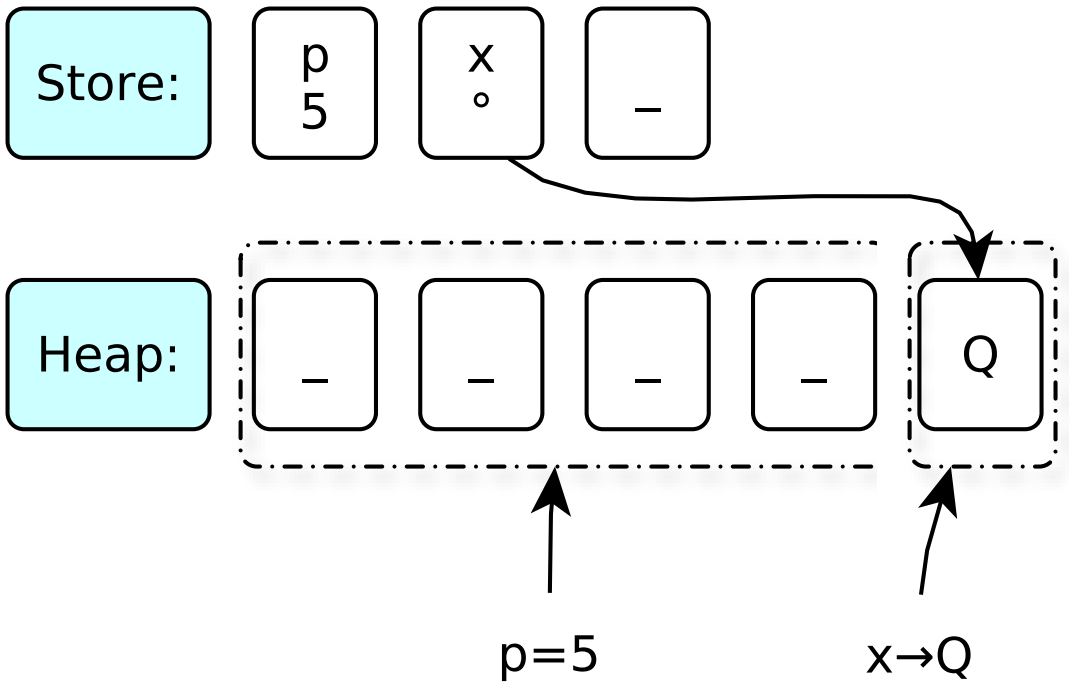
Assertion: $s, h \models \text{true} * (x \mapsto F)$



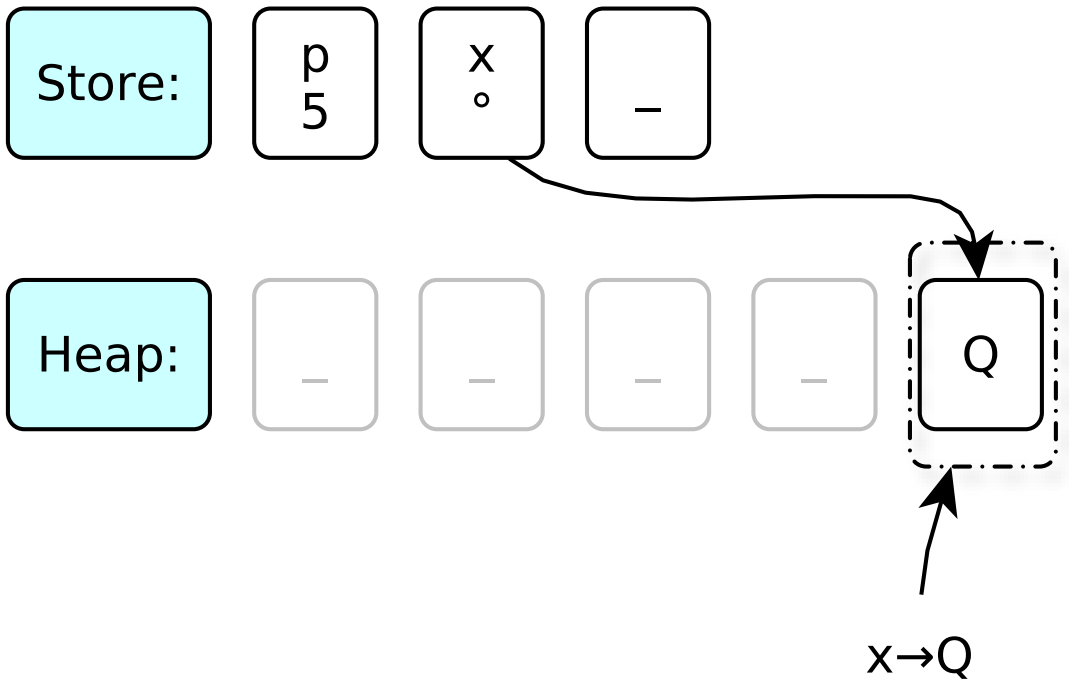
Assertion: $s, h \models \text{true} * (x \mapsto F)$



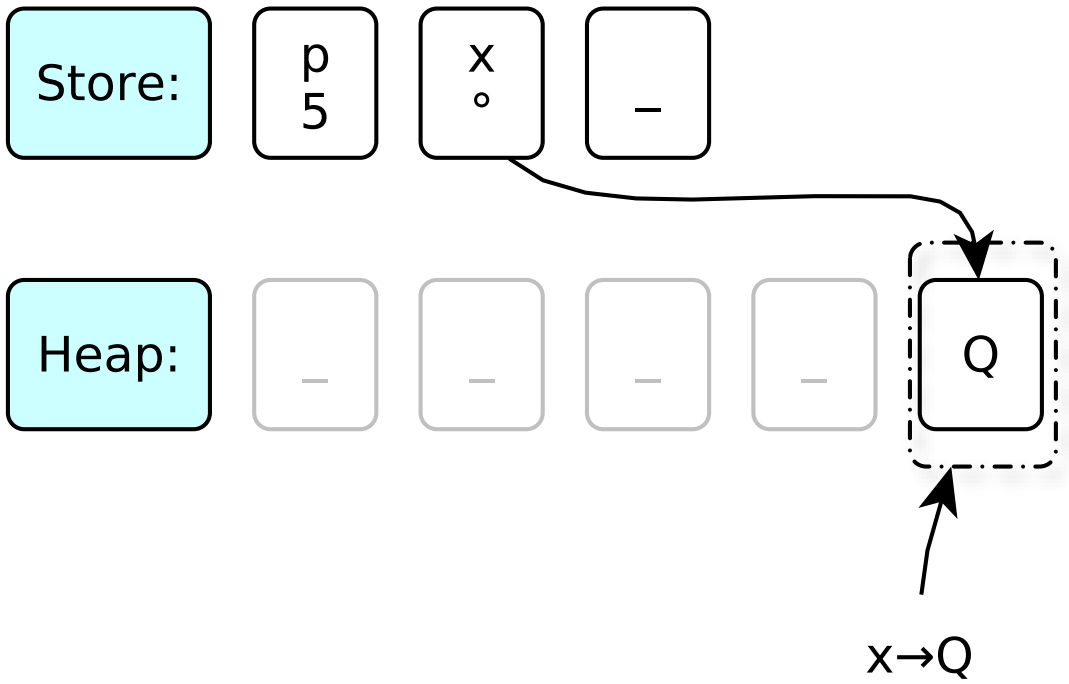
Assertion: $s, h \models (p = 5) * (x \mapsto A)$



Assertion: $s, h \models (p \doteq 5) * (x \mapsto A)$



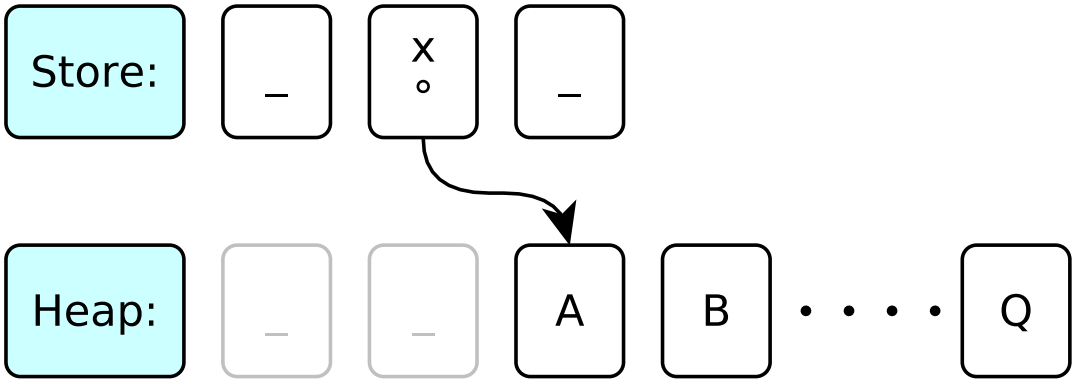
Assertion: $s, h \models (p \doteq 5) * (x \mapsto A)$



Formally:

$A \doteq B \quad \equiv \quad A = B$ and emp

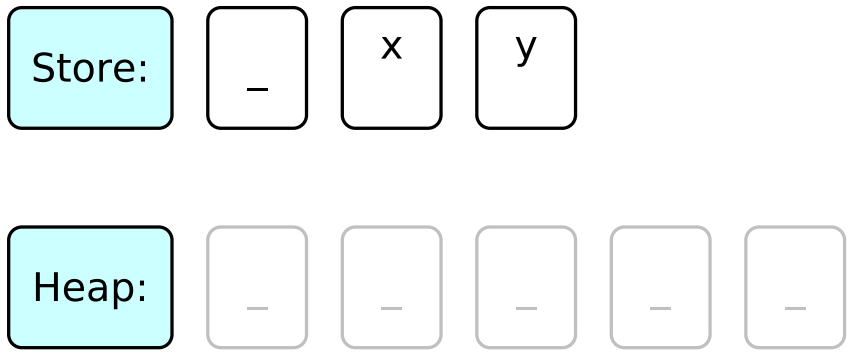
Assertion: $s, h \models x \mapsto A, B, \ldots, Q$



Linked List

Example

{emp}

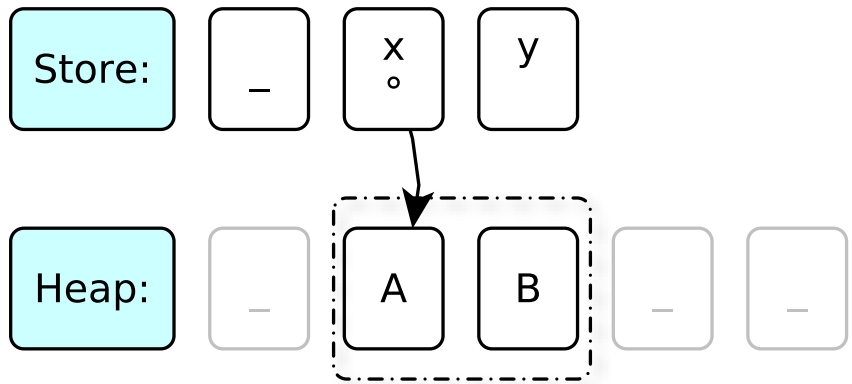


Example

$\{emp\}$

$x := cons(A, B);$

$\{x \mapsto A, B\}$



Example

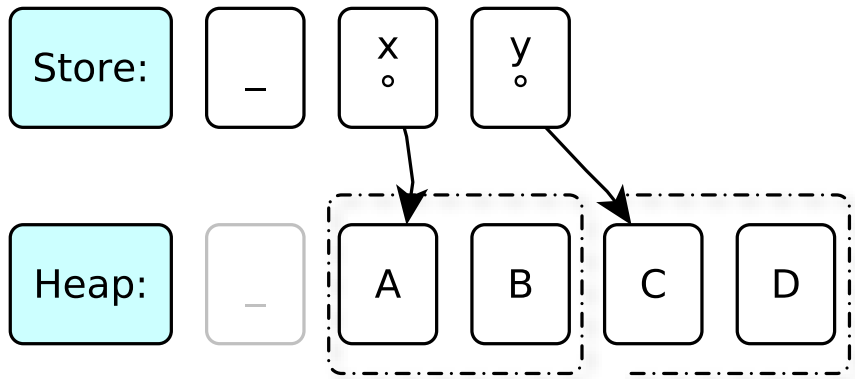
$\{emp\}$

$x := cons(A, B);$

$\{x \mapsto A, B\}$

$y := cons(C, D);$

$\{x \mapsto A, B * y \mapsto C, D\}$



Example

$\{emp\}$

$x := cons(A, B);$

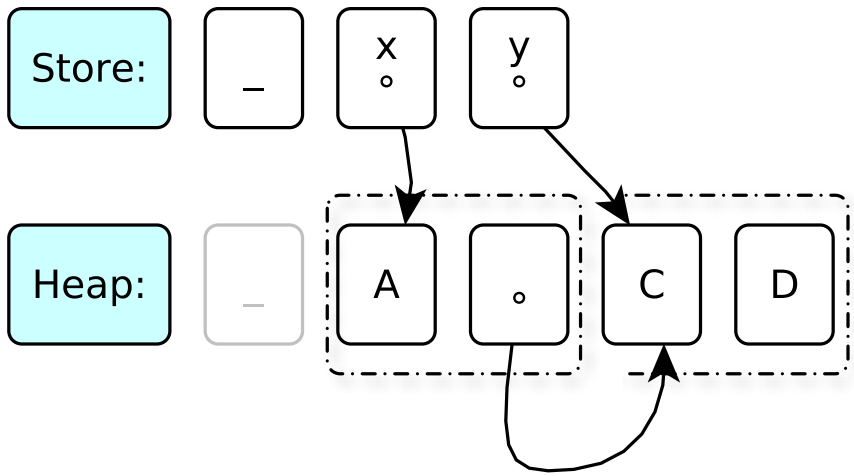
$\{x \mapsto A, B\}$

$y := cons(C, D);$

$\{x \mapsto A, B * y \mapsto C, D\}$

$x[1] := y$

$\{x \mapsto A, y * y \mapsto C, D\}$



Example

$\{emp\}$

x := cons(A, B);

$\{x \mapsto A, B\}$

y := cons(C, D);

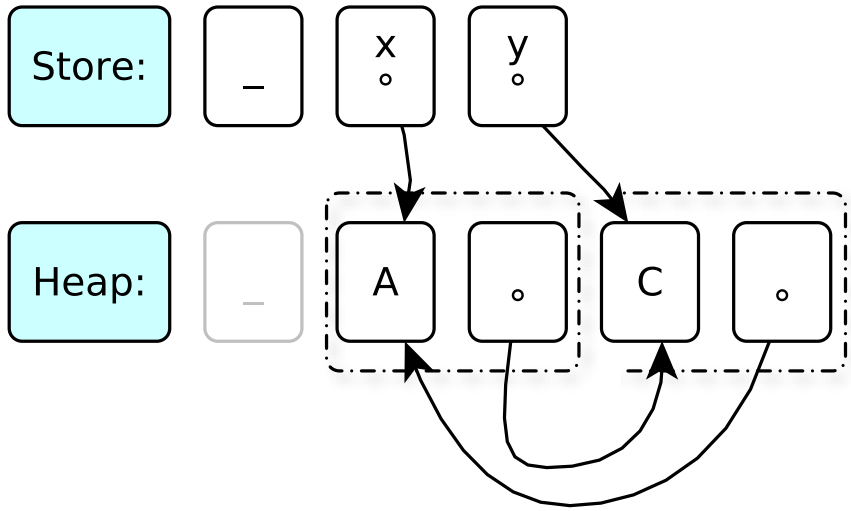
$\{x \mapsto A, B * y \mapsto C, D\}$

[x + 1] := y

$\{x \mapsto A, y * y \mapsto C, D\}$

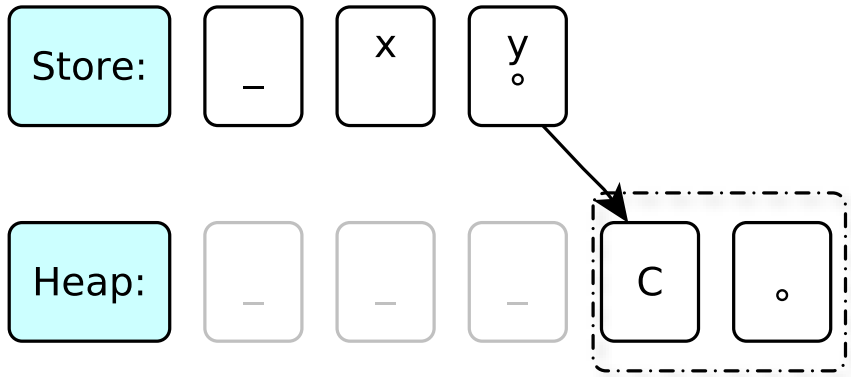
[y + 1] := x

$\{x \mapsto A, x * y \mapsto C, D\}$



Example

$\{emp\}$
x:=cons(A,B);
$\{x \mapsto A, B\}$
y:=cons(C,D);
$\{x \mapsto A, B * y \mapsto C, D\}$
[x+1]:=y
$\{x \mapsto A, y * y \mapsto C, D\}$
[y+1]:=x
$\{x \mapsto A, x * y \mapsto C, D\}$
⋮
[y+1]:=nil;
dispose(x+1); dispose(x);
$\{emp * y \mapsto C, D\} = \{y \mapsto C, D\}$



Small Axioms

1. $\{E \mapsto -\}[E] := F\{E \mapsto F\}$
2. $\{E \mapsto -\}\mathit{dispose}(E)\{\mathrm{emp}\}$
3. $\{x \doteq n\}x := E\{x \doteq E'\}$
4. $\{x \doteq n\}x = \mathit{cons}(E_1, E_2, \ldots, E_k)\{x \mapsto E'_1, E'_2, \ldots, E'_k\}$
5. $\{E \mapsto m \text{ and } x = n\}x = [E]\{x = m \text{ and } E' \mapsto m\}$

Where E' is E with each x is replaced by n.

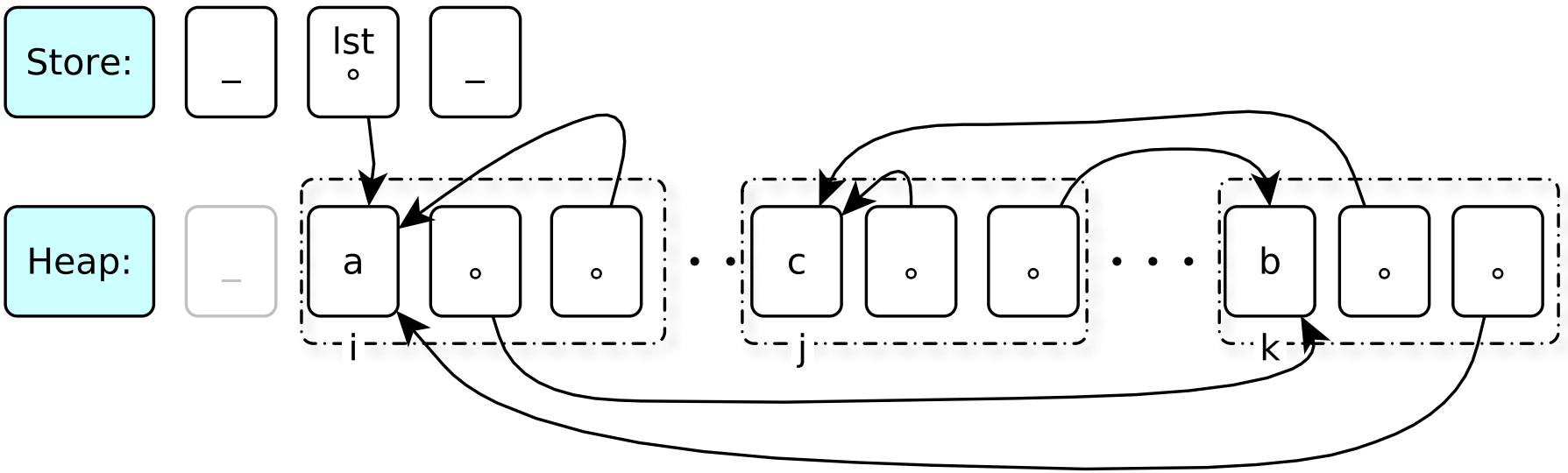
Structural Rules

$$\{P\}C\{Q\} \Rightarrow \{P * R\}C\{Q * R\}$$

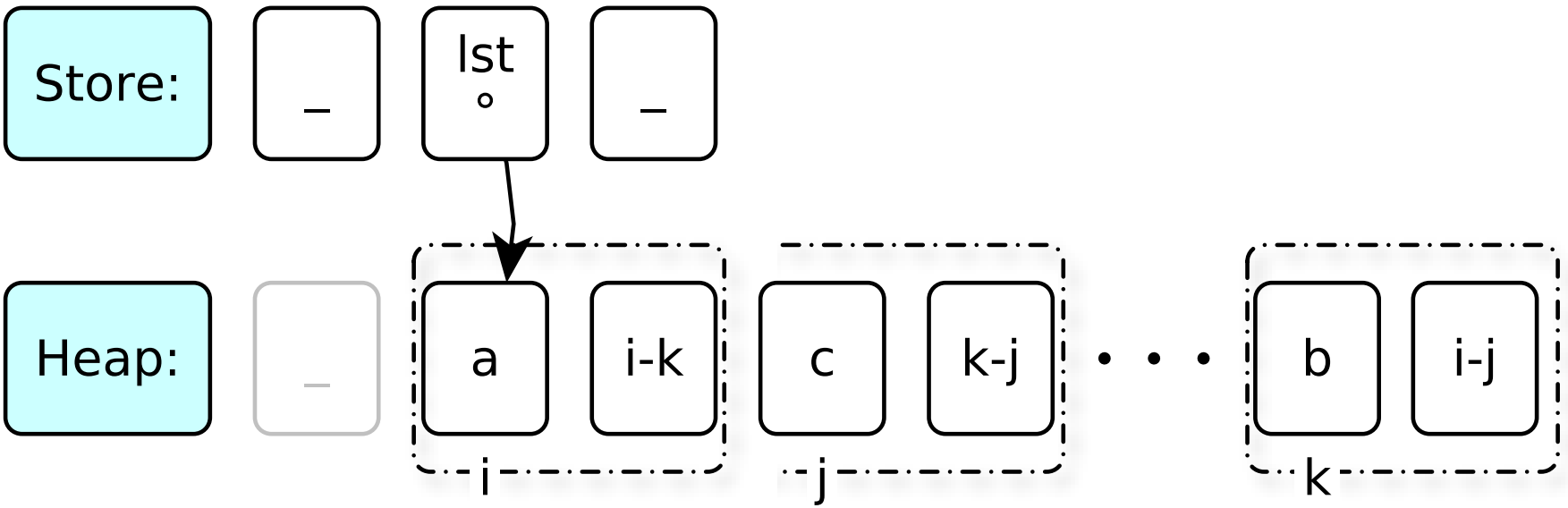
Only if C does not modify anything used in R

Difference list

Double linked list: [*a*, *b*, *c*]

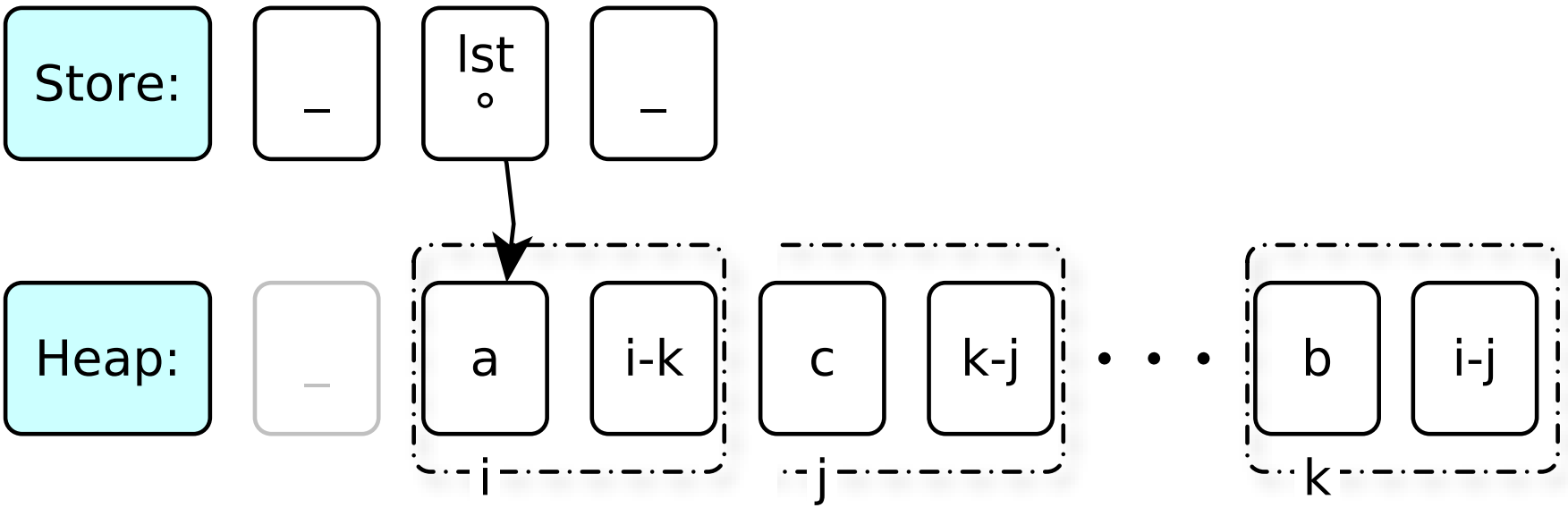


Difference list: $[a, b, c]$



Difference list: $[a, b, c]$

1. dl $[](i, i', j, j') \equiv$ emp and $i = j$ and $i' = j'$
2. dl $[a, \alpha](i, i', k, k') \equiv \exists j.(i \mapsto a, j - i') * \text{dl } \alpha(j, i, k, k')$



Thank you!