

# Separation Logic

Herbert Kruitbosch

April 17, 2013

Herbert Kruitbosch 1/41

# Propositional dynamic logic



Herbert Kruitbosch 2/41

# Propositional dynamic logic

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

Different notation:

#### Store

Store:

х 5 у 5

#### Store

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

Store:

#### Store

int 
$$x=5$$
; int  $y=5$ ;

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

Store:

#### Formally:

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

#### Heap

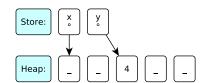
```
int *x=malloc(5*sizeof(int));
int *y=x+2;
x[2]=5;
y[0]=5;
Heap:
- - 5 - -
```

Herbert Kruitbosch 7/41

# Неар

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

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

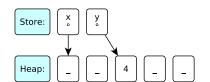


Herbert Kruitbosch 8/41

## Heap

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

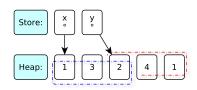
$${x[2] = 5, y[0] = 5}$$
  
x[2]=4;  
 ${x[2] = 4, y[0] = 5} \leftarrow FAIL$ 



Herbert Kruitbosch 9/41

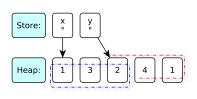
### Merge Sort

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

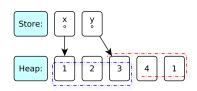


Herbert Kruitbosch 10/41

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



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



Herbert Kruitbosch

# Merge Sort

```
{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] and y[0] < y[1] < y[2]} \leftarrow FAIL
```

Herbert Kruitbosch 12/41



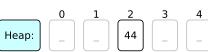
# Seperation Logic

Herbert Kruitbosch 13/41

#### Model

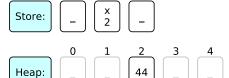
- Variables: x, y, etc
- Locations: 0, 1, 2, 3, etc
- Store: Variable → Integer
- Heap: Location → Integer
- States: Store × Heap





#### Model

- Variables: x, y, etc
- 2 Locations: 0, 1, 2, 3, etc
- Store: Variable → Integer
- Heap: Location → Integer
- States: Store × Heap



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

Herbert Kruitbosch 15/41

P =

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

Herbert Kruitbosch 16/41

$$P =$$

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

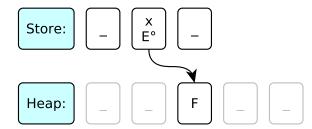


Heap:

Herbert Kruitbosch 17/41

$$P =$$

- **1** Default: B, false, true,  $P \Rightarrow Q$ ,  $\forall x.P$ , x = 5, etc
- $s, h \models emp$



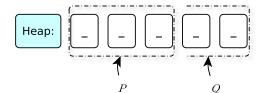
Herbert Kruitbosch 18/41

$$P =$$

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

- $\bullet$   $s, h \models P * Q$

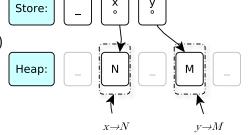




Herbert Kruitbosch 19/41

$$P =$$

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

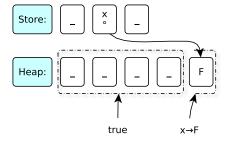


Herbert Kruitbosch 20/41

P =

- **1** Default: B, false, true,  $P \Rightarrow Q$ ,  $\forall x.P$ , x = 5, etc
- $s, h \models emp$
- $\bullet$   $s, h \models P * Q$
- $\bullet$   $s, h \models P *Q$

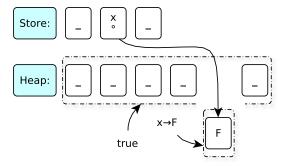
# Assertion: $s, h \models \text{true} * (x \mapsto F)$



Herbert Kruitbosch 22/41



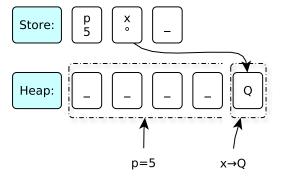
# Assertion: $s, h \models \text{true} * (x \mapsto F)$



Herbert Kruitbosch 23/41



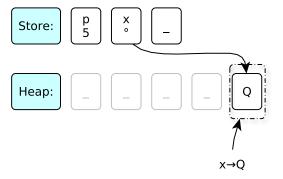
# Assertion: $s, h \models (p = 5) * (x \mapsto A)$



Herbert Kruitbosch 24/41



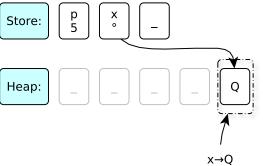
# Assertion: $s, h \models (p = 5) * (x \mapsto A)$



Herbert Kruitbosch 25/41



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



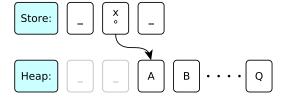
Formally:

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

Herbert Kruitbosch 26/41



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



Herbert Kruitbosch 27/41



# Linked List

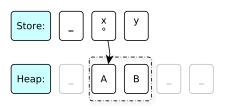
Herbert Kruitbosch 28/41

{emp}

Heap:

Herbert Kruitbosch 29/41

```
{emp}
x:=cons(A,B);
\{x \mapsto A, B\}
```



Herbert Kruitbosch 30/41

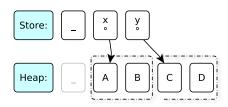
```
{emp}

x := cons(A,B);

\{x \mapsto A, B\}

y := cons(C,D);

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



Herbert Kruitbosch 31/41

```
{emp}

x := cons(A,B);

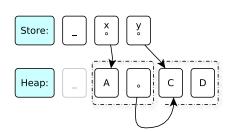
\{x \mapsto A, B\}

y := cons(C,D);

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

x[1] := y

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



Herbert Kruitbosch 32/41

```
{emp}

x := cons(A, B);

\{x \mapsto A, B\}

y := cons(C, D);

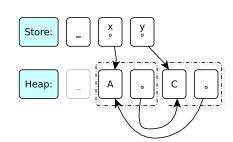
\{x \mapsto A, B * y \mapsto C, D\}

[x+1] := y

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

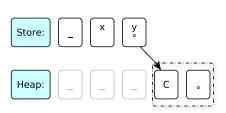
[y+1] := x

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



Herbert Kruitbosch 33/41

```
{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\}
[v+1]:=x
\{x \mapsto A, x * y \mapsto C, D\}
[y+1]:=nil;
dispose(x+1); dispose(x);
\{\mathsf{emp} * \mathsf{v} \mapsto \mathsf{C}, \mathsf{D}\} = \{\mathsf{v} \mapsto \mathsf{C}, \mathsf{D}\}
```



Herbert Kruitbosch 34/41

#### **Small Axioms**

- $\{E \mapsto -\}[E] := F\{E \mapsto F\}$
- $\{E \mapsto -\} dispose(E) \{emp\}$
- **3**  $\{x = n\}x := E\{x = E'\}$

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

Herbert Kruitbosch 35/41

#### Structural Rules

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

Only if C does not modify anything used in R

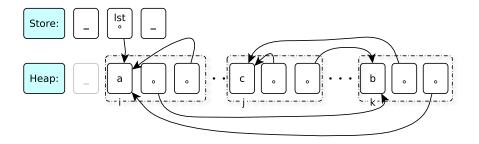
Herbert Kruitbosch 36/41



# Difference list

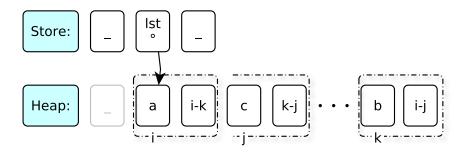
Herbert Kruitbosch 37/41

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



Herbert Kruitbosch 38/41

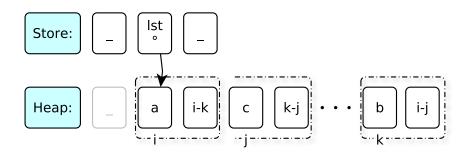
# Difference list: [a, b, c]



Herbert Kruitbosch 39/41

# Difference list: [a, b, c]

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



Herbert Kruitbosch 40/41



# Thank you!

Herbert Kruitbosch 41/41