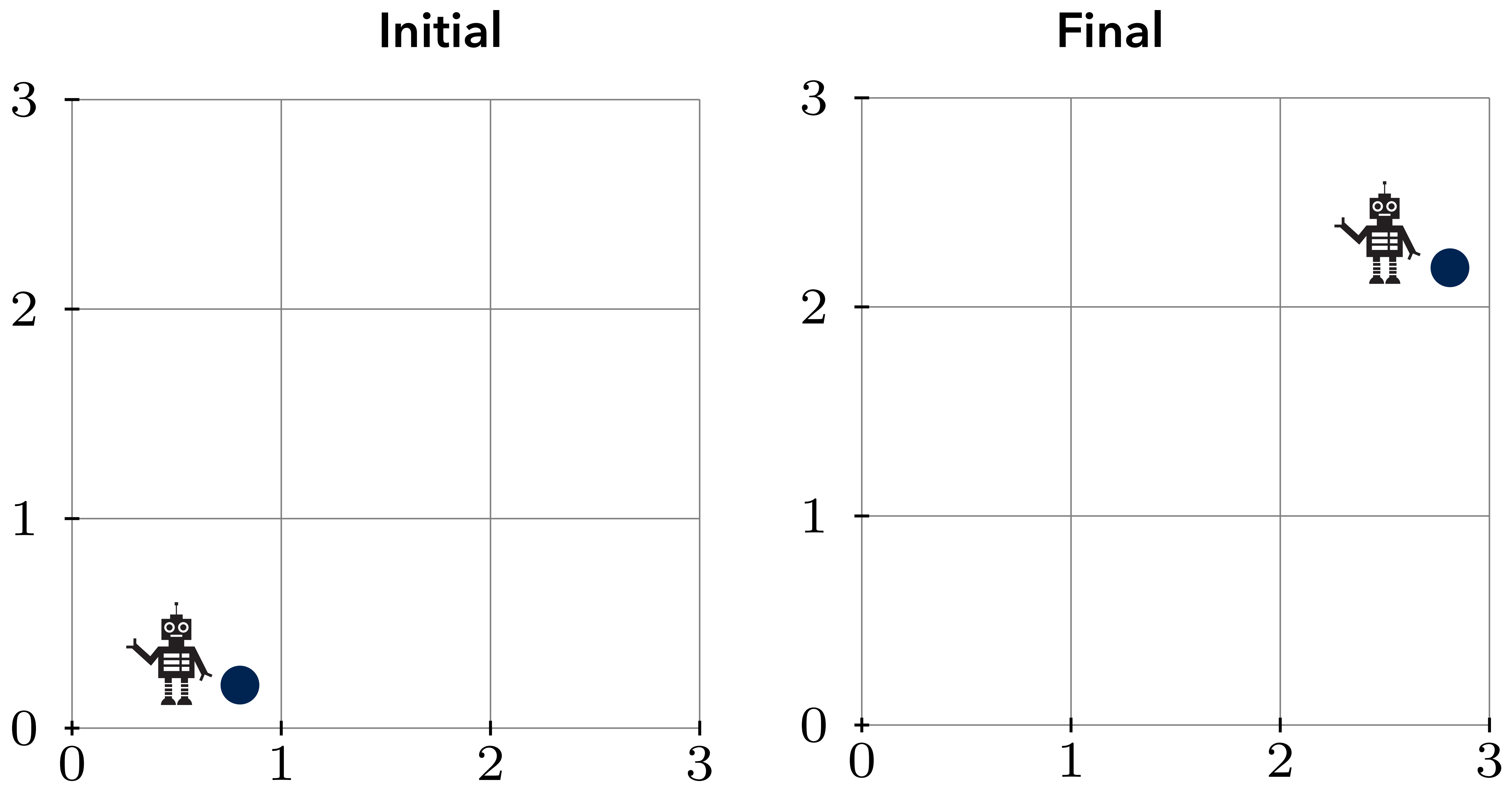
Learning efficient programs

Andrew Cropper and Stephen Muggleton

Imperial College London

Learning efficient robot strategies [IJCAI15]


Initial

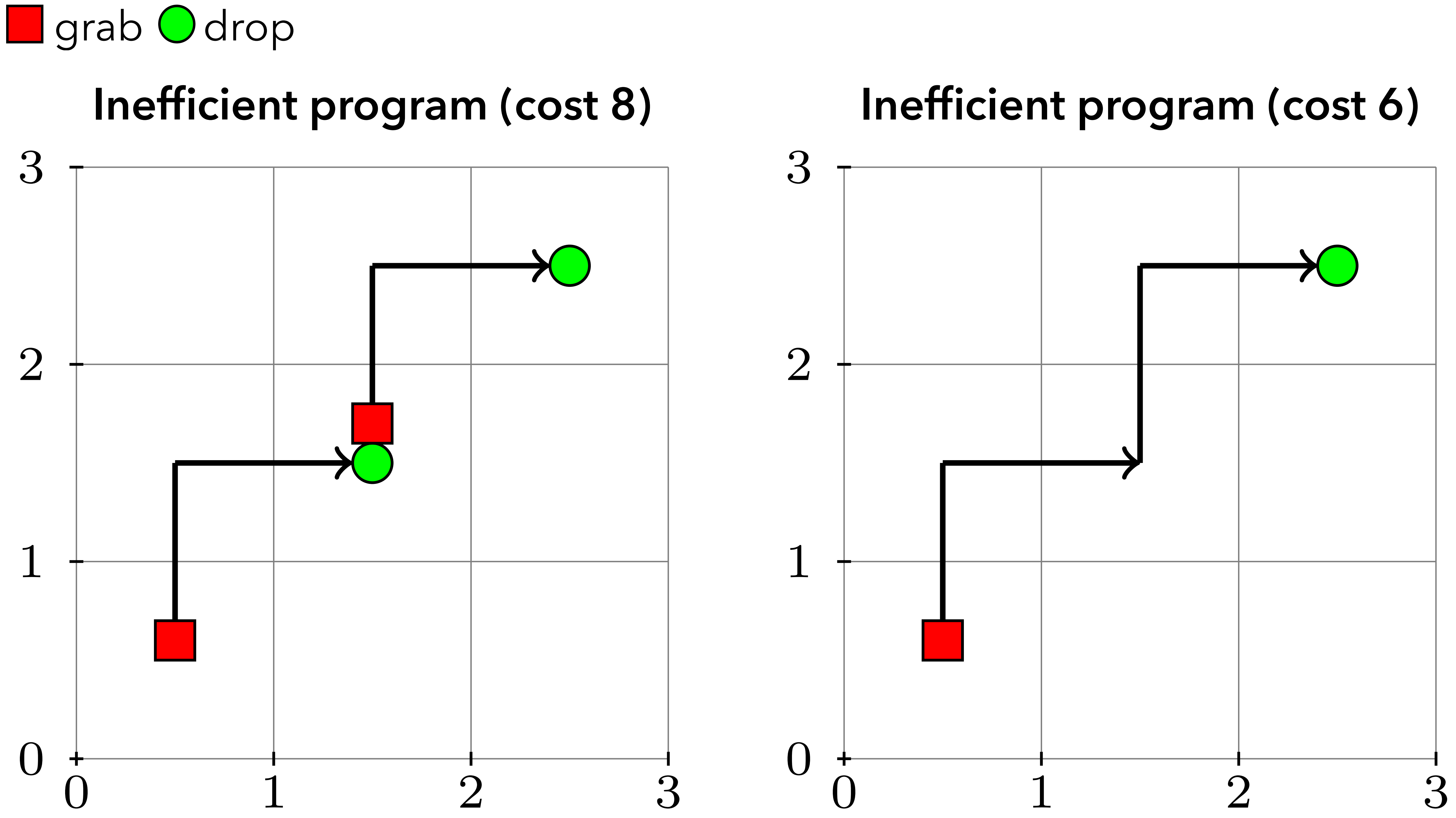
Final

```
% actions
north/2, south/2, east/2, west/2, pickup/2, drop/2

% learning task
A=[pos(robot,0/0),pos(ball,0/0),holding_ball(false)],
B=[pos(robot,2/2),pos(ball,2/2),holding_ball(false)],
learn([f(A,B)],Prog)
```

```
f(A,B):-f3(A,C),f3(C,B).
f3(A,B):-f2(A,C),drop(C,B).
f2(A,B):-grab(A,C),p1(C,B).
f1(A,B):-north(A,C),east(C,B).
```

```
f(A,B):-f3(A,C),drop(C,B).
f3(A,B):-grab(A,C),f2(C,B).
f2(A,B):-f1(A,C),f1(C,B).
f1(A,B):-north(A,C),east(C,B).
```



Inefficient program (cost 8)

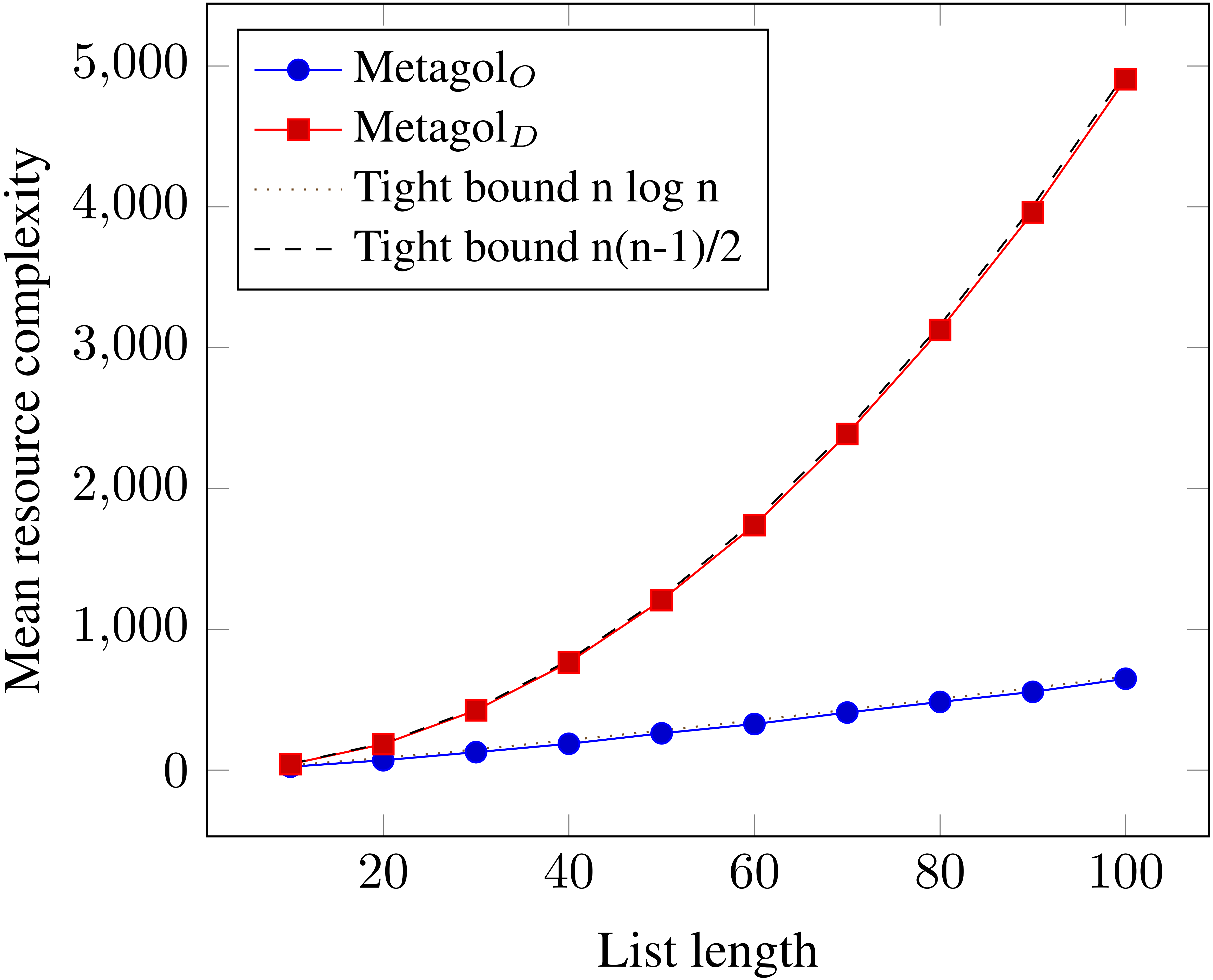
Inefficient program (cost 6)

```
f(A,B):-f3(A,C),f3(C,B).
f3(A,B):-f2(A,C),drop(C,B).
f2(A,B):-grab(A,C),p1(C,B).
f1(A,B):-north(A,C),east(C,B).
```

```
f(A,B):-f3(A,C),drop(C,B).
f3(A,B):-grab(A,C),f2(C,B).
f2(A,B):-f1(A,C),f1(C,B).
f1(A,B):-north(A,C),east(C,B).
```

Sorting experiment [IJCAI15]

Input	**Output**
[9,13,1,8,4]	[1,4,8,9,13]
[1,18,20,6,15,5]	[1,5,6,15,18,20]
[12,16,18,6,15,3,5]	???
[16,1,4,12,3,18,2,14]	???
[12,17,5,13,6,4,14,2,15]	???


Metagol$_O$
Metagol$_D$
Tight bound n log n
Tight bound n(n-1)/2
Mean resource complexity
5,000
4,000
3,000
2,000
1,000
0
20
40
60
80
100
List length

String transformations

Input	Output
My name is John.	John
My name is Bill.	Bill
My name is Josh.	Josh
My name is Albert.	Albert
My name is Richard.	Richard

```
% Inefficient program
f(A,B):-tail(A,C),f1(C,B).
f1(A,B):-dropLast(A,C),f2(C,B).
f2(A,B):-dropWhile(A,B,not_uppercase).

% Efficient program
f(A,B):-tail(A,C),f1(C,B).
f1(A,B):-f2(A,C),dropLast(C,B).
f2(A,B):-f3(A,C),f3(C,B).
f3(A,B):-tail(A,C),f4(C,B).
f4(A,B):-f5(A,C),f5(C,B).
f5(A,B):-tail(A,C),tail(C,B).
```