

# Learning Multi-Step Predictive State Representations - Appendix

## Abstract

This appendix contains pseudocode for the algorithms presented in the main paper, and more details on the experiments.

## Pseudocode of the encoding function

The pseudocode is given in Algorithm 1, using the following notation:

**bestEncoding:** A map from indices  $i$  of  $Q$  to the optimal encoding of  $Q[i]$ .

**minEncoding:** A map from indices  $i$  of  $Q$  to  $|bestE[i]|$

**opsEnding:** A map from indices  $i$  of  $Q$  to the set of strings in  $\Sigma'$ :  $\{x \in \Sigma'.tQ[i - |x| : i] == x\}$

---

### Algorithm 1 Encoding Algorithm

---

**INPUT:**  $x$

**OUTPUT:**  $\kappa(x)$

```
1: procedure DPENCODE
2:    $bestEncoding[] \leftarrow String[|x| + 1]$ 
3:    $minEncoding[] \leftarrow Int[|x| + 1]$ 
4:    $opsEnding[] \leftarrow String[|x| + 1][]$ 
5:    $bestEncoding[0] = x[0]$ 
6:    $minEncoding[0] = 0$ 
7:   for  $i$  in  $[1, |x|]$  do
8:      $opsEnding[i] \leftarrow \{s \in \Sigma', x[i - |s| : i] == s\}$ 
9:   end for
10:  for  $i$  in  $[1, |x|]$  do
11:     $bestOp \leftarrow null$ 
12:     $m \leftarrow null$ 
13:    for  $s \in opsEnding[i]$  do
14:       $t \leftarrow minE[i - |s|] + 1$ 
15:      if  $m == null$  or  $t < m$  then
16:         $m \leftarrow t$ 
17:         $bestOp \leftarrow s$ 
18:      end if
19:    end for
20:     $minEncoding[i + 1] \leftarrow m$ 
21:     $bestEncoding[i + 1] \leftarrow bestEncoding[i - |bestOp|] + bestOp$ 
22:  end for
23:  return  $bestEncoding[|x|]$ 
24: end procedure
```

---

## Experiments

The figures below depict the Double Loop and Pacman environments used in the experiments.

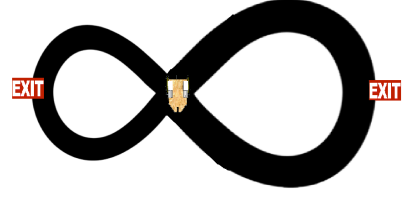


Figure 1: Double Loop Environment

---

### Algorithm 2 Base Selection Algorithm

---

**INPUT:**  $Train, Sub_M$

**OUTPUT:**  $\Sigma'$

```

1: procedure BASE SELECTION
2:    $\Sigma' \leftarrow \{s, s \in \Sigma\}$ 
3:    $bestEncoding \leftarrow null$ 
4:   for each obs in Train do
5:      $bestEncoding[obs] \leftarrow |obs|$ 
6:   end for
7:    $i \leftarrow 0$ 
8:   while  $i < numOps$  do
9:      $bestOp \leftarrow null$ 
10:     $m \leftarrow 0$ 
11:    for each  $s \in Sub_M$  do
12:       $c \leftarrow 0$ 
13:      for each obs in Train do
14:         $c \leftarrow c + DPEncode(obs) -$ 
 $prevBestE(obs)$ 
15:      end for
16:      if  $c > m$  then
17:         $bestOp \leftarrow obs$ 
18:         $m \leftarrow c$ 
19:      end if
20:    end for
21:     $\Sigma' \leftarrow \Sigma' \cup bestOp$ 
22:    for each obs in Train do
23:       $bestEncoding \leftarrow DPEncode(obs, \Sigma')$ 
24:    end for
25:     $i \leftarrow i + 1$ 
26:  end while
27:  return  $\Sigma'$ 
28: end procedure

```

---

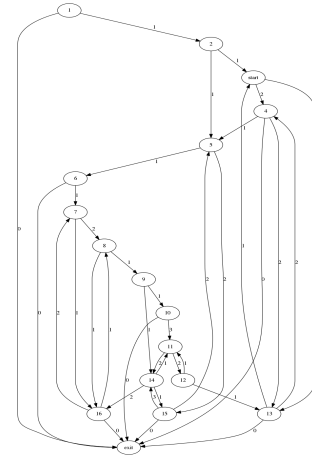


Figure 2: Graph of Pacman Labyrinth