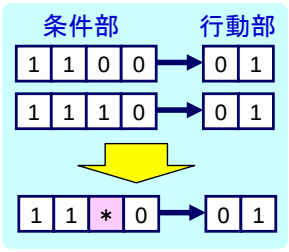
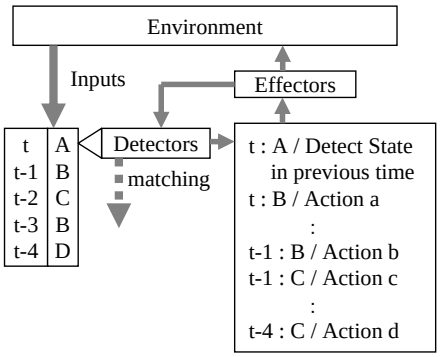
# 時系列依存型クラシファイアシステム TCS

- クラシファイアシステム（Classifier System；CS）は，条件部→行動部のルールに基づいてエージェントが行動する際，ルール数が爆発する問題を抑えるため，条件部にワイルドカード（＊）を導入して複数のルールをまとめて学習できるようにした方法である．
- 例えば，条件部が4ビット，行動部が2ビットの右のようなルールがあるとすると，0または1のいずれにも成り得る＊を導入すれば2つのルールを下の1つで表現することができる．一般に＊がn個含まれるルールは $2^n$ 通りのルールを表せる．
- ＊を無作為に含めたルール集合を強化学習によって強化することで，エージェントの行動を最適化することができる．



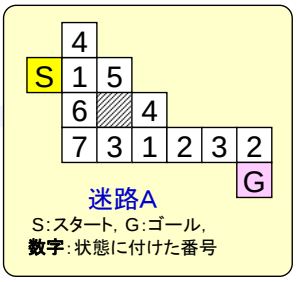

- 一方，従来のCSでは現在の環境入力だけを扱っており，同一の環境入力でも異なる行動が要求される問題には不向きであった．ここでは，過去の環境入力を考慮することでその問題を解決したTCS（Time-dependent CS）（福寄・原・長尾 '02）を紹介する．
- 右にTCSの原理を示す．現時刻 t の現在の入力だけでなく，過去に遡ってルールのマッチングを行うことで，時系列処理を実現している．








# 実験例と考察



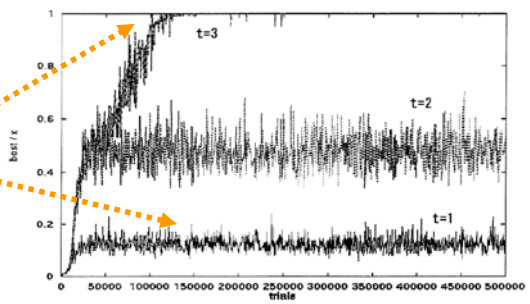

- ここでは右に例示するようなグリッド環境でのスタート地点（S）からゴール地点（G）までの最短経路を学習する問題を扱った．
- 問題の設定を次に示す．
  - エージェントは上下左右の壁の有無を知覚することができる．
  - エージェントの行動は上下左右の移動（Up, Down, Left, Right），あるいは前の状態の検出（Previous State）の5種類のいずれかとする．
  - ルールの記述は，（時刻）:（状態）／（行動） と表現する．
    例）0:S/R → 時刻 t=0 で 状態＝S のとき Right
  - ルールの強化は Profit Sharing で行い，強化関数としては公比0.8の等比減少関数を用いた．

```
0:S/R
0:1/PS, 1:S/D
0:6/D
0:7/R
0:3/R
0:1/PS, 1:3/PS, 2:7/R
0:2/PS, 1:1/R
0:3/R
0:2/PS, 1:3/PS, 2:2/D
```

R:Right, D:Down
PS:Previos State

上の迷路の正解

- 実験結果を右下のグラフに示す．次のことがわかる．
  - 迷路Aでは，入力が同じでも行動を適切に選択する時系列処理が必要であるため，従来のCS（t=1）では成功率は2割に満たないことがわかる．
  - 一方，提案手法のt=3では10割を達成した．
- 今後，“何ステップ前まで（tの値）考慮すべきか”を，問題に応じて学習中に適切に自動設定できるように改良することなどが考えられる．




試行回数と成功率