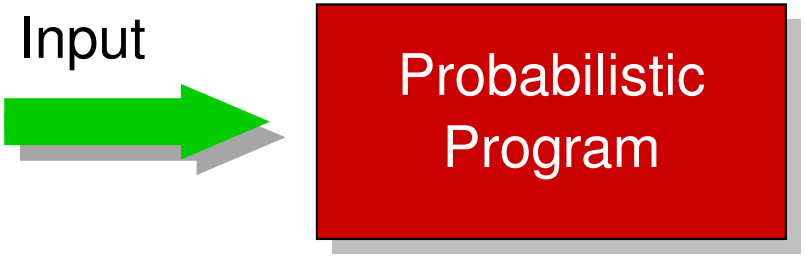
Analyzing recursive probabilistic Programs

Justin Winkens

30. June 2016

Table of contents

Probabilistic Programs



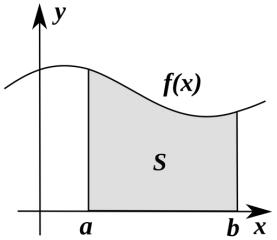
Probabilistic Programs

Probabilistic Programs

TrueSkill™

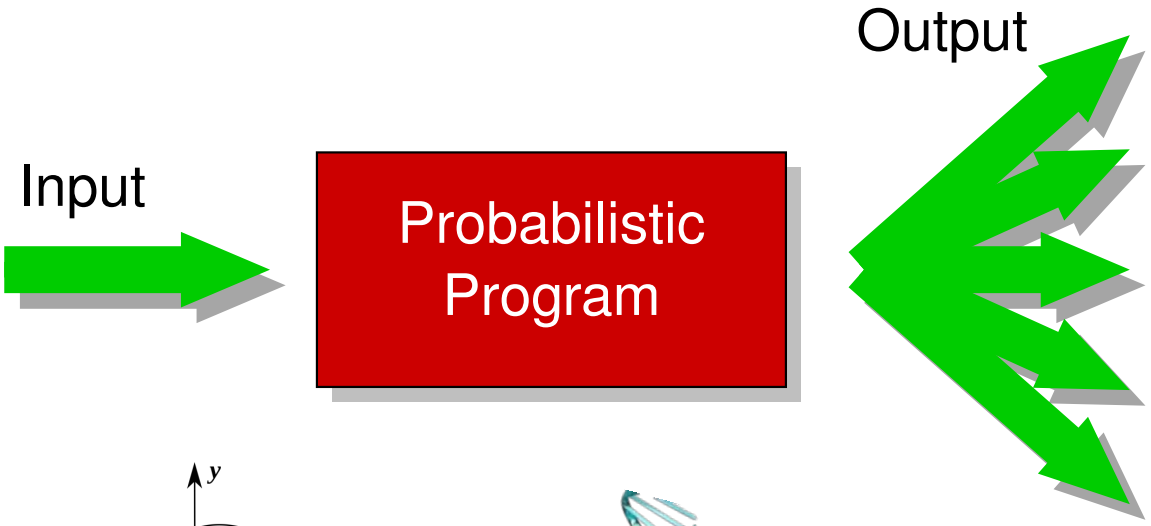
Probabilistic Programs

TrueSkill™

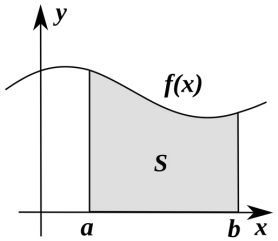


self-testing/ correcting

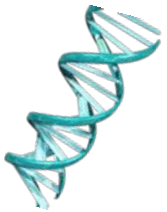
Probabilistic Programs



TrueSkill™

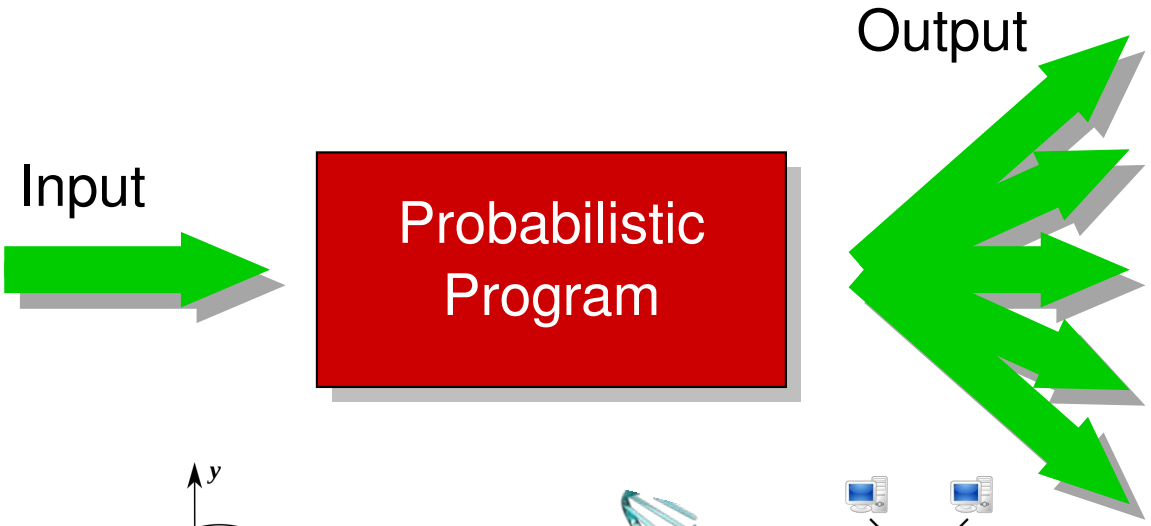


self-testing/ correcting

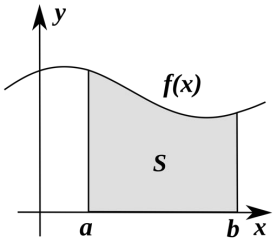


biology

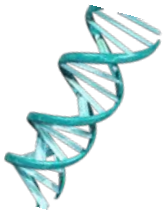
Probabilistic Programs



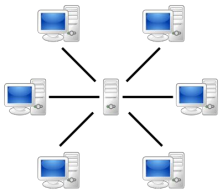
TrueSkill™



self-testing/ correcting

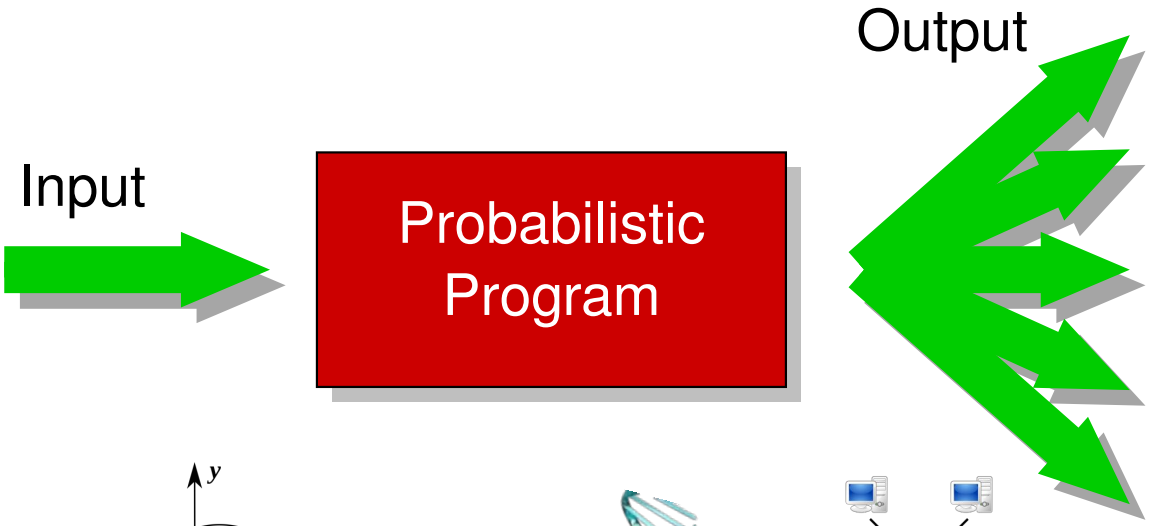


biology

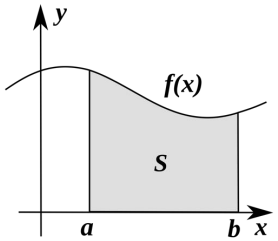


networks

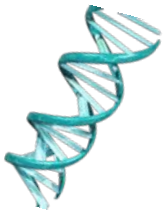
Probabilistic Programs



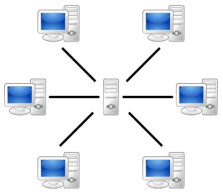
TrueSkill™



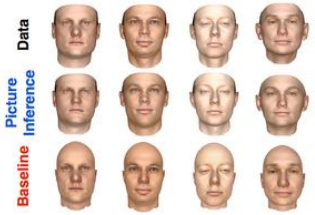
self-testing/ correcting



biology



networks



PICTURE

Recursive Probabilistic Programs

```
algorithm quicksort(A) is
  list less, equal, greater
  if length(A) <= 1
      return A
  pivot := A[length(A)]
  for x in A
    if x < pivot then append x to less
    if x = pivot then append x to equal
    if x > pivot then append x to greater
  return concat(quicksort(less), equal,quicksort(greater))
```

Recursive Probabilistic Programs

```
algorithm quicksort(A) is
  list less, equal, greater
  if length(A) <= 1
      return A
  pivot := A[length(A)]
  for x in A
    if x < pivot then append x to less
    if x = pivot then append x to equal
    if x > pivot then append x to greater
  return concat(quicksort(less), equal,quicksort(greater))
```

$\Rightarrow$ Average case complexity $O(n \log n)$

Recursive Probabilistic Programs

```
algorithm quicksort(A) is
  list less, equal, greater
  if length(A) <= 1
      return A
  pivot := A[length(A)]
  for x in A
    if x < pivot then append x to less
    if x = pivot then append x to equal
    if x > pivot then append x to greater
  return concat(quicksort(less), equal,quicksort(greater))
```

$\Rightarrow$ Average case complexity $O(n \log n)$

$\Rightarrow$ Worst case complexity $O(n^2)$

Recursive Probabilistic Programs

```
algorithm quicksort(A) is
  list less, equal, greater
  if length(A) <= 1
      return A
  pivot := uniformSel(A)
  for x in A
    if x < pivot then append x to less
    if x = pivot then append x to equal
    if x > pivot then append x to greater
  return concat(quicksort(less), equal,quicksort(greater))
```

Recursive Probabilistic Programs

```
algorithm quicksort(A) is
  list less, equal, greater
  if length(A) <= 1
      return A
  pivot := uniformSel(A)
  for x in A
    if x < pivot then append x to less
    if x = pivot then append x to equal
    if x > pivot then append x to greater
  return concat(quicksort(less), equal,quicksort(greater))
```

⇒ Worst case far less likely!

Recursive Probabilistic Programs

- We wish to...

Recursive Probabilistic Programs

- We wish to...
  - ...reason about a programs behavior

Recursive Probabilistic Programs

- We wish to...
  - ...reason about a programs behavior
  - ...estimate expected outcomes and runtimes

Recursive Probabilistic Programs

- We wish to...
  - ...reason about a programs behavior
  - ...estimate expected outcomes and runtimes
  - ...quantify probabilities

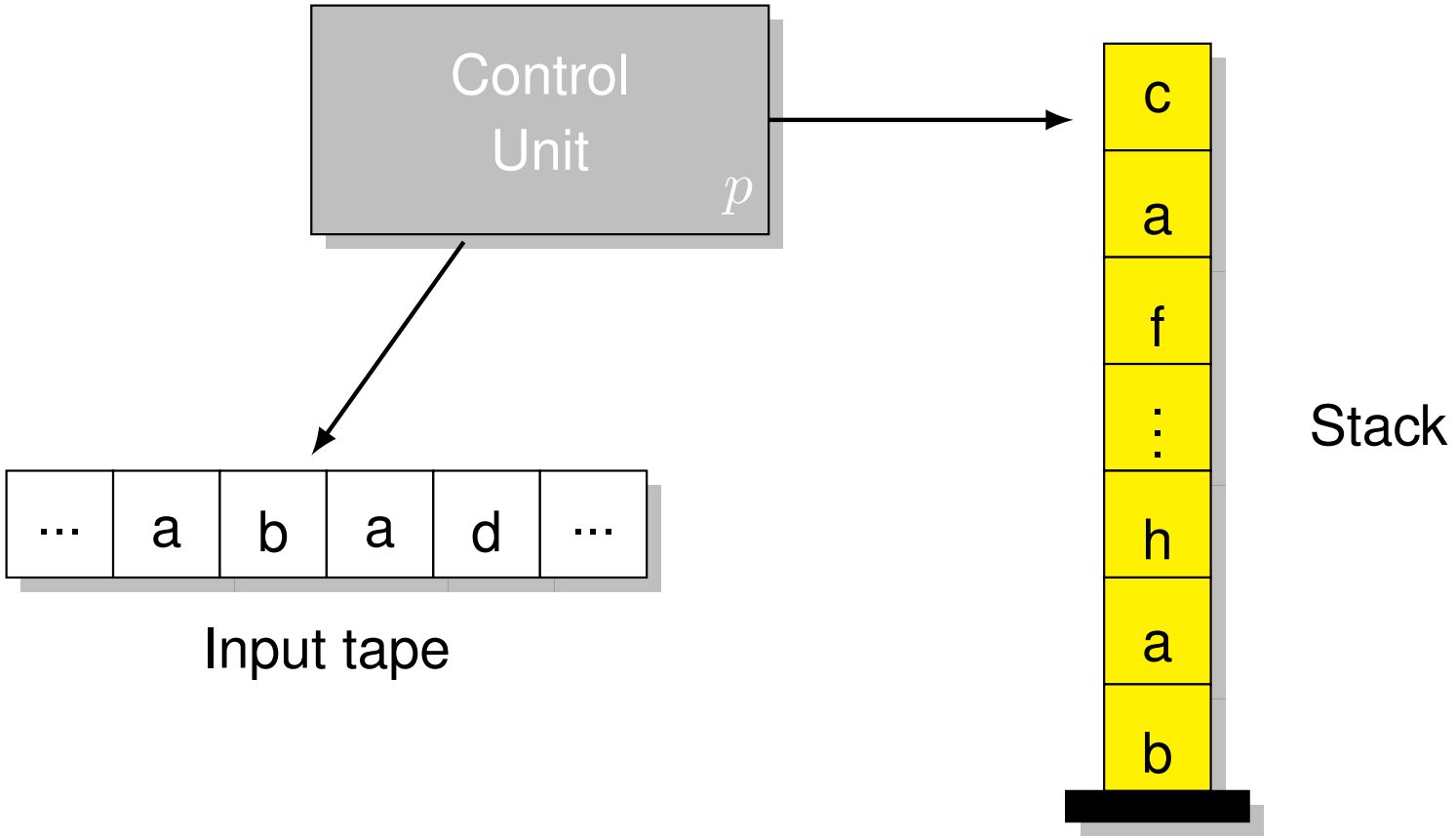
Recursive Probabilistic Programs

- We wish to...
  - ...reason about a programs behavior
  - ...estimate expected outcomes and runtimes
  - ...quantify probabilities
- Recursive programs written in the probabilistic Guarded Command Language

Recursive Probabilistic Programs

- We wish to...
  - ...reason about a programs behavior
  - ...estimate expected outcomes and runtimes
  - ...quantify probabilities
- Recursive programs written in the probabilistic Guarded Command Language
- Analysis by using probabilistic pushdown automata

Pushdown Automata



Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.
- Γ is a finite stack alphabet.

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.
- Γ is a finite stack alphabet.
- $\hookrightarrow \subseteq (Q \times \Gamma) \times (Q \times \Gamma^*)$ is a set of transitions such that

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.
- Γ is a finite stack alphabet.
- $\hookrightarrow \subseteq (Q \times \Gamma) \times (Q \times \Gamma^*)$ is a set of transitions such that
  - for every $pX \in Q \times \Gamma$ there is at least one transition of the form $pX \hookrightarrow q\alpha$

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.
- Γ is a finite stack alphabet.
- $\hookrightarrow \subseteq (Q \times \Gamma) \times (Q \times \Gamma^*)$ is a set of transitions such that
  - for every $pX \in Q \times \Gamma$ there is at least one transition of the form $pX \hookrightarrow q\alpha$
  - for every transition $pX \hookrightarrow q\alpha$ we have that $|\alpha| \leq 2$.

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- $\mathfrak{P}$ is a function which assigns probabilities to transitions

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- $\mathfrak{P}$ is a function which assigns probabilities to transitions
  - $\mathfrak{P}(pX \hookrightarrow q\alpha) \in [0, 1]$

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- $\mathfrak{P}$ is a function which assigns probabilities to transitions
  - $\mathfrak{P}(pX \hookrightarrow q\alpha) \in [0, 1]$
  - $\sum_{pX \hookrightarrow q\alpha} \mathfrak{P}(pX \hookrightarrow q\alpha) = 1$

Probabilistic Pushdown Automata

Definition: Probabilistic Pushdown Automaton (pPDA)

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- $\mathfrak{P}$ is a function which assigns probabilities to transitions
  - $\mathfrak{P}(pX \hookrightarrow q\alpha) \in [0, 1]$
  - $\sum_{pX \hookrightarrow q\alpha} \mathfrak{P}(pX \hookrightarrow q\alpha) = 1$
  - We write $pX \overset{x}{\hookrightarrow} q\alpha$ instead of $\mathfrak{P}(pX \hookrightarrow q\alpha) = x$

Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

What does this pPDA represent?

Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

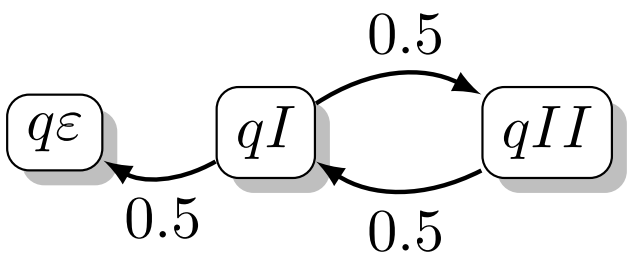
qI

Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

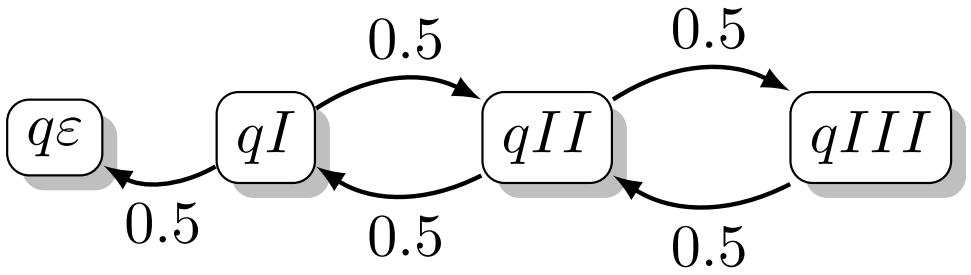


Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

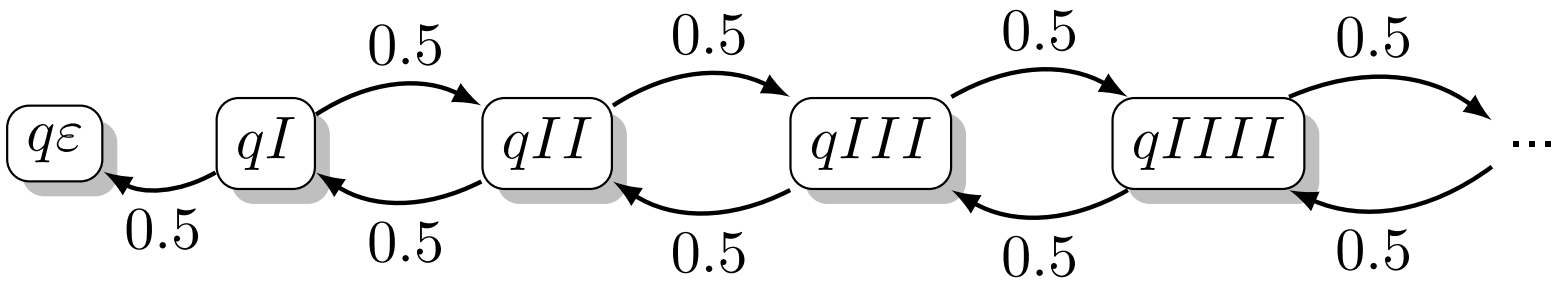


Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

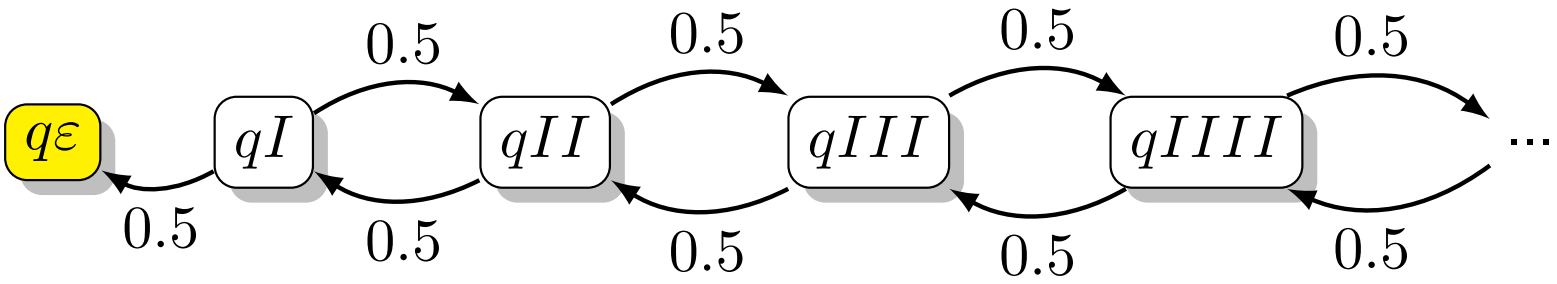


Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$



$\Rightarrow$ Termination on empty stack!

Example pPDA

Example pPDA

A pPDA Δ is given by:

- $Q = \{q\}$
- $\Gamma = \{I\}$
- $qI \overset{0.5}{\hookrightarrow} qII$, $qI \overset{0.5}{\hookrightarrow} q\epsilon$

What could an unbounded random walk look like?

Probabilistic Pushdown Automata

Definition: Regular Configurations

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of configurations.

Probabilistic Pushdown Automata

Definition: Regular Configurations

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of configurations.

We call $\mathcal{C}$ regular if $p\alpha \in \mathcal{C}$ iff the reversed stack of $p\alpha$ is accepted by a deterministic finite-state automaton (DFA) $\mathcal{A}$.

Probabilistic Pushdown Automata

Definition: Regular Configurations

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of configurations.

We call $\mathcal{C}$ regular if $p\alpha \in \mathcal{C}$ iff the reversed stack of $p\alpha$ is accepted by a deterministic finite-state automaton (DFA) $\mathcal{A}$.

We call $\mathcal{C}$ simple if there exists a set $\mathcal{H} \subseteq Q \times \Gamma$ such that $p\alpha \in \mathcal{C}$ iff $\alpha \neq \varepsilon$ and $head(p\alpha) \in \mathcal{H}$.

Paths, Runs

Definition: Paths and Runs

Let $\mathfrak{S}$ be a set of states and $\rightarrow \subseteq \mathfrak{S} \times \mathfrak{S}$ a binary and total relation and $P = (\mathfrak{S}, \rightarrow)$.

Paths, Runs

Definition: Paths and Runs

Let $\mathfrak{S}$ be a set of states and $\rightarrow \subseteq \mathfrak{S} \times \mathfrak{S}$ a binary and total relation and $P = (\mathfrak{S}, \rightarrow)$.

- A path is a word w such that $w(i) \rightarrow w(i+1)$ for every $i \in \{0, \ldots, |w| - 1\}$

Paths, Runs

Definition: Paths and Runs

Let $\mathfrak{S}$ be a set of states and $\rightarrow \subseteq \mathfrak{S} \times \mathfrak{S}$ a binary and total relation and $P = (\mathfrak{S}, \rightarrow)$.

- A path is a word w such that $w(i) \rightarrow w(i+1)$ for every $i \in \{0, \ldots, |w| - 1\}$
- A state t is reachable from s if there is a finite path w with $w(0) = s$ and $w(|w| - 1) = t$.

Paths, Runs

Definition: Paths and Runs

Let $\mathfrak{S}$ be a set of states and $\rightarrow \subseteq \mathfrak{S} \times \mathfrak{S}$ a binary and total relation and $P = (\mathfrak{S}, \rightarrow)$.

- A path is a word w such that $w(i) \rightarrow w(i+1)$ for every $i \in \{0, \ldots, |w| - 1\}$
- A state t is reachable from s if there is a finite path w with $w(0) = s$ and $w(|w| - 1) = t$.
- A run is an infinite path.

Paths, Runs

Definition: Paths and Runs

Let $\mathfrak{S}$ be a set of states and $\rightarrow \subseteq \mathfrak{S} \times \mathfrak{S}$ a binary and total relation and $P = (\mathfrak{S}, \rightarrow)$.

- A path is a word w such that $w(i) \rightarrow w(i+1)$ for every $i \in \{0, \ldots, |w| - 1\}$
- A state t is reachable from s if there is a finite path w with $w(0) = s$ and $w(|w| - 1) = t$.
- A run is an infinite path.
- $Runs(P, w) =$ set of all runs in P that start with w

Reachability

Definition: Reachability

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

Reachability

Definition: Reachability

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

- $Reach(pX, \mathcal{C})$ set of all runs $r \in Runs(\Delta, pX)$ with $r(i) \in \mathcal{C}$, $i \in \mathbb{N}_0$

Reachability

Definition: Reachability

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

- $Reach(pX, \mathcal{C})$ set of all runs $r \in Runs(\Delta, pX)$ with $r(i) \in \mathcal{C}$, $i \in \mathbb{N}_0$
- $\mathscr{P}(Reach(pX, \mathcal{C}))$

Termination

Definition: Termination

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $q \in Q$ a (final) state.

Termination

Definition: Termination

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $q \in Q$ a (final) state.

- $Reach(pX, \{q\varepsilon\})$ set of all runs $r \in Runs(\Delta, pX)$ with $r(i) = q\varepsilon$, $i \in \mathbb{N}_0$.

Termination

Definition: Termination

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $q \in Q$ a (final) state.

- $Reach(pX, \{q\varepsilon\})$ set of all runs $r \in Runs(\Delta, pX)$ with $r(i) = q\varepsilon$, $i \in \mathbb{N}_0$.
- $\mathscr{P}(Reach(pX, \{q\varepsilon\}))$.

Qualitative/Quantitative Reachability/Termination

Definition: Qualitative/Quantitative Reachability/Termination

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

Qualitative/Quantitative Reachability/Termination

Definition: Qualitative/Quantitative Reachability/Termination

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

- Qualitative reachability/termination: $\mathscr{P}(Reach(pX, C)) = 1$?

Qualitative/Quantitative Reachability/Termination

Definition: Qualitative/Quantitative Reachability/Termination

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

- Qualitative reachability/termination: $\mathscr{P}(Reach(pX, C)) = 1$?
- Quantitative reachability/termination: $\mathscr{P}(Reach(pX, C)) \leq d$ for a constant $d \in (0, 1)$?

Probabilistic Guarded Command Language

`skip` | `abort` | `var x` | x:=E | Q;R | `{Q}[]{R}` | `{Q}[p]{R}`
`if(G){Q}else{R}` | `while(G){Q}`

Probabilistic Guarded Command Language

`skip` | `abort` | `var x` | x:=E | Q;R | `{Q}[]{R}` | `{Q}[p]{R}`
`if(G){Q}else{R}` | `while(G){Q}`

Probabilistic Guarded Command Language

skip | abort | var x | x:=E | Q;R | {Q}[]{R} | {Q}[p]{R}
if(G){Q}else{R} | while(G){Q}

```
var flip := true;
flip(){
    while(flip){
        {skip;}[0.5]{flip = false;}
    }
}
```

Probabilistic Guarded Command Language

skip | abort | var x | x:=E | Q;R | {Q}[]{R} | {Q}[p]{R}
if(G){Q}else{R} | while(G){Q}

```
var flip := true;
flip(){
    while(flip){
        {skip;}[0.5]{flip = false;}
    }
}
```

We allow function calls!

Dueling Cowboys

Dueling Cowboys (Orig.)

```
var cowboy := A;
var duel:= true;

{cowboy := A;}[p]{cowboy := B;}

while(duel){
    if(cowboy = A){
        {duel := false;}[a]{cowboy := B;}
    }else{
        {duel := false;}[b]{cowboy := A;}
    }
}
```

Dueling Cowboys (Rec.)

```
var player := A;
duel(){                          //d0
    {player:=A;}[p]{player:=B;}
    shoot();                     //d1
}
```

Dueling Cowboys (Rec.)

```
var player := A;
duel(){                                  //d0
    {player:=A;}[p]{player:=B;}
    shoot();                             //d1
}
```

```
shoot(){                                 //s0
        if(player == A){                 //s1
            {
                kill();                  //s3
            }
            [s]
            {
                player := B;
                shoot();                 //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                  //s5
            }
            [t]
            {
                player := A;
                shoot();                 //s6
            }
        }
}
```

Dueling Cowboys (Rec.)

```
var player := A;
duel(){                                  //d0
    {player:=A;}[p]{player:=B;}
    shoot();                             //d1
}
```

```
kill(){ //k0

}
```

```
shoot(){                                 //s0
        if(player == A){                 //s1
            {
                kill();                  //s3
            }
            [s]
            {
                player := B;
                shoot();                 //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                  //s5
            }
            [t]
            {
                player := A;
                shoot();                 //s6
            }
        }
}
```

Dueling Cowboys (Rec.)

```
var player := A;
duel(){                                  //d0
    {player:=A;}[p]{player:=B;}
    shoot();                             //d1
}
```

```
kill(){ //k0

}
```

How do we analyze this?

```
shoot(){                                 //s0
        if(player == A){                 //s1
            {
                kill();                  //s3
            }
            [s]
            {
                player := B;
                shoot();                 //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                  //s5
            }
            [t]
            {
                player := A;
                shoot();                 //s6
            }
        }
}
```

Translating pGCL to pPDA

2 Steps:

Translating pGCL to pPDA

2 Steps:

1. Translate pGCL program to control flow graph

Translating pGCL to pPDA

2 Steps:

1. Translate pGCL program to control flow graph
2. Translate control flow graph to pPDA

pGCL $\rightarrow$ Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes

pGCL $\rightarrow$ Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next

pGCL $\rightarrow$ Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next
- Label edges with statements and transition probability (if $\neq$ 1)

pGCL → Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next
- Label edges with statements and transition probability (if $\neq$ 1)

```
1  var player := A;
2  duel(){                          //d0
3      {player:=A;}[p]{player:=B;}
4      shoot();                     //d1
5  }
```

pGCL → Control Flow Graph

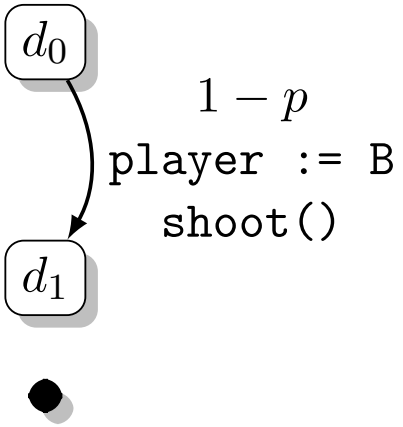
- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next
- Label edges with statements and transition probability (if $\neq$ 1)

```
1  var player := A;
2  duel(){                          //d0
3      {player:=A;}[p]{player:=B;}
4      shoot();                     //d1
5  }
```

pGCL → Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next
- Label edges with statements and transition probability (if $\neq$ 1)

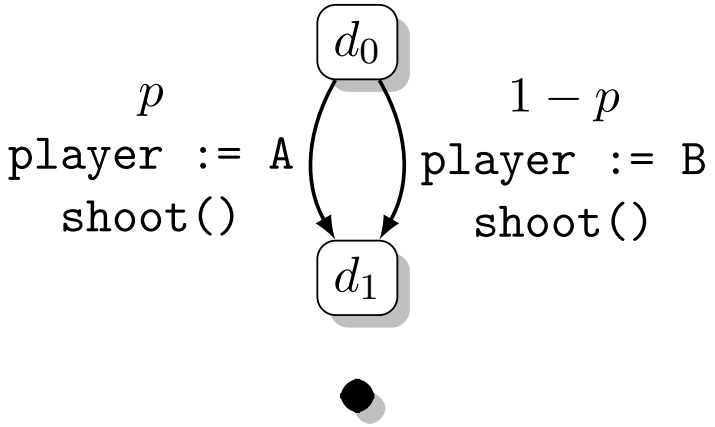
```
1  var player := A;
2  duel(){                          //d0
3      {player:=A;}[p]{player:=B;}
4      shoot();                     //d1
5  }
```



pGCL → Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next
- Label edges with statements and transition probability (if $\neq$ 1)

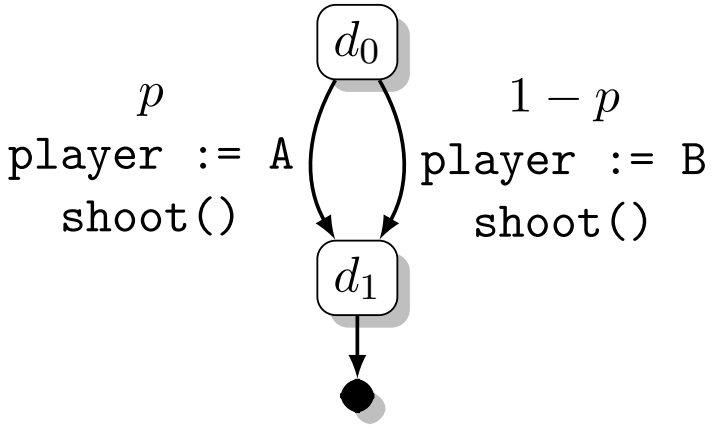
```
1  var player := A;
2  duel(){                          //d0
3      {player:=A;}[p]{player:=B;}
4      shoot();                     //d1
5  }
```



pGCL → Control Flow Graph

- Pick control points ($k0$,$d0$,$d1$,$s0$,...,$s6$) as nodes
- Connect nodes by edges if control flows from one node to the next
- Label edges with statements and transition probability (if $\neq$ 1)

```
1  var player := A;
2  duel(){                          //d0
3      {player:=A;}[p]{player:=B;}
4      shoot();                     //d1
5  }
```



pGCL → Control Flow Graph

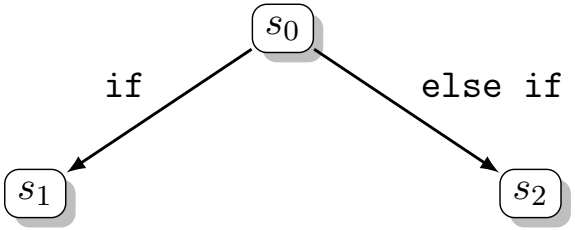
```
shoot(){                                //s0
        if(player == A){                //s1
            {
                kill();                 //s3
            }
            [s]
            {
                player := B;
                shoot();                //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                 //s5
            }
            [t]
            {
                player := A;
                shoot();                //s6
            }
        }
}
```

pGCL → Control Flow Graph

```
shoot(){                                //s0
        if(player == A){                //s1
            {
                kill();                 //s3
            }
            [s]
            {
                player := B;
                shoot();                //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                 //s5
            }
            [t]
            {
                player := A;
                shoot();                //s6
            }
        }
}
```

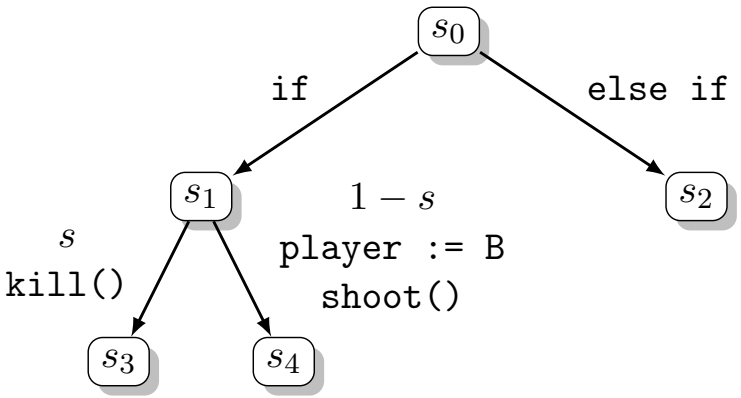
pGCL → Control Flow Graph

```
shoot(){                                //s0
        if(player == A){                //s1
            {
                kill();                 //s3
            }
            [s]
            {
                player := B;
                shoot();                //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                 //s5
            }
            [t]
            {
                player := A;
                shoot();                //s6
            }
        }
}
```



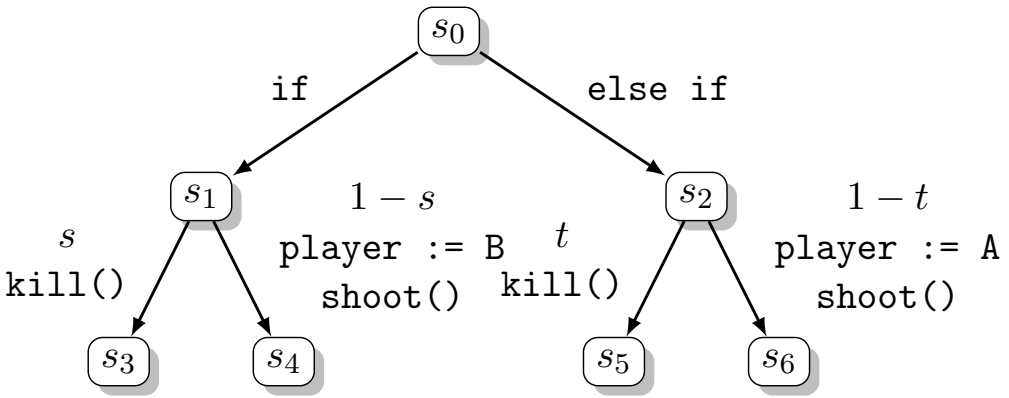
pGCL → Control Flow Graph

```
shoot(){                                //s0
        if(player == A){                //s1
            {
                kill();                 //s3
            }
            [s]
            {
                player := B;
                shoot();                //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                 //s5
            }
            [t]
            {
                player := A;
                shoot();                //s6
            }
        }
}
```



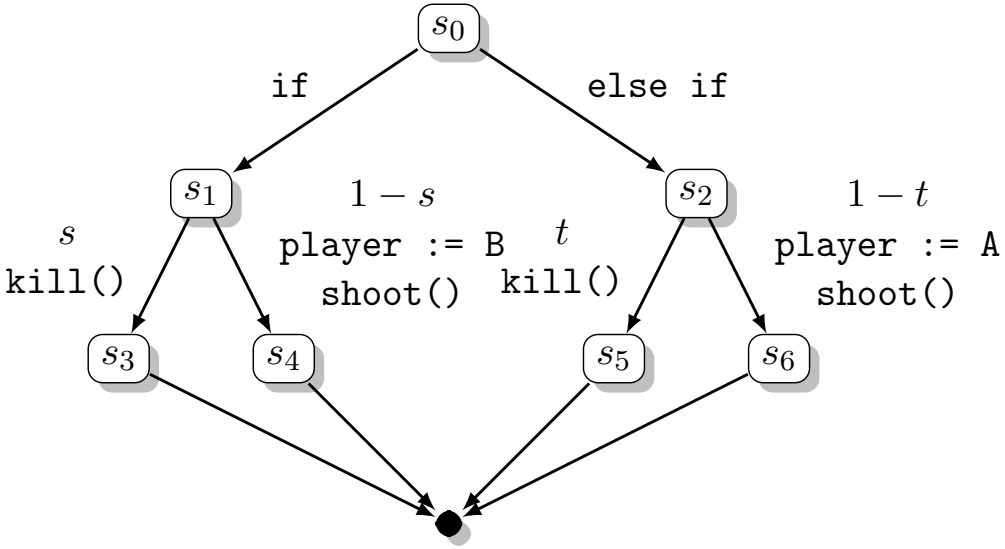
pGCL → Control Flow Graph

```
shoot(){                                //s0
        if(player == A){                //s1
            {
                kill();                 //s3
            }
            [s]
            {
                player := B;
                shoot();                //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                 //s5
            }
            [t]
            {
                player := A;
                shoot();                //s6
            }
        }
}
```



pGCL → Control Flow Graph

```
shoot(){                                //s0
        if(player == A){                //s1
            {
                kill();                 //s3
            }
            [s]
            {
                player := B;
                shoot();                //s4
            }
        }
        else if(player == B){//s2
            {
                kill();                 //s5
            }
            [t]
            {
                player := A;
                shoot();                //s6
            }
        }
}
```



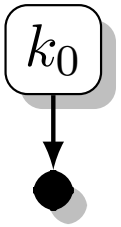
pGCL → Control Flow Graph

```
kill(){ //k0

}
```

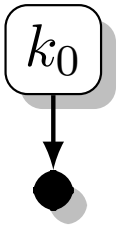
pGCL → Control Flow Graph

```
1  kill(){ //k0
2
3  }
```



pGCL → Control Flow Graph

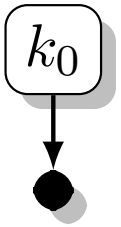
```
1  kill(){ //k0
2
3  }
```



Why do we have this method?

pGCL → Control Flow Graph

```
1  kill(){ //k0
2
3  }
```

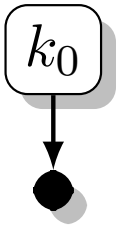


Why do we have this method?

- It yields an “easy” head (pk_0) for analysis

pGCL → Control Flow Graph

```
1  kill(){ //k0
2
3  }
```

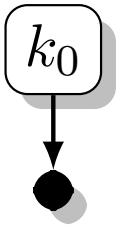


Why do we have this method?

- It yields an "easy" head (pk_0) for analysis
- All configuration with head pk_0 terminate

pGCL → Control Flow Graph

```
1  kill(){ //k0
2
3  }
```

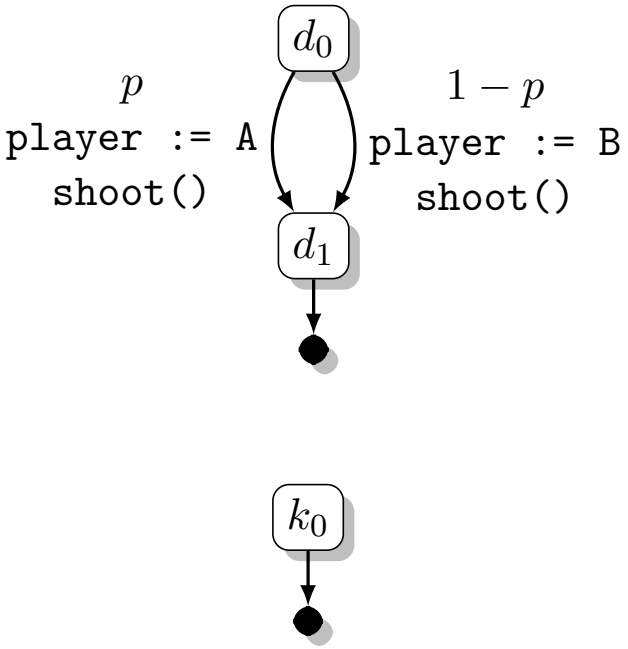


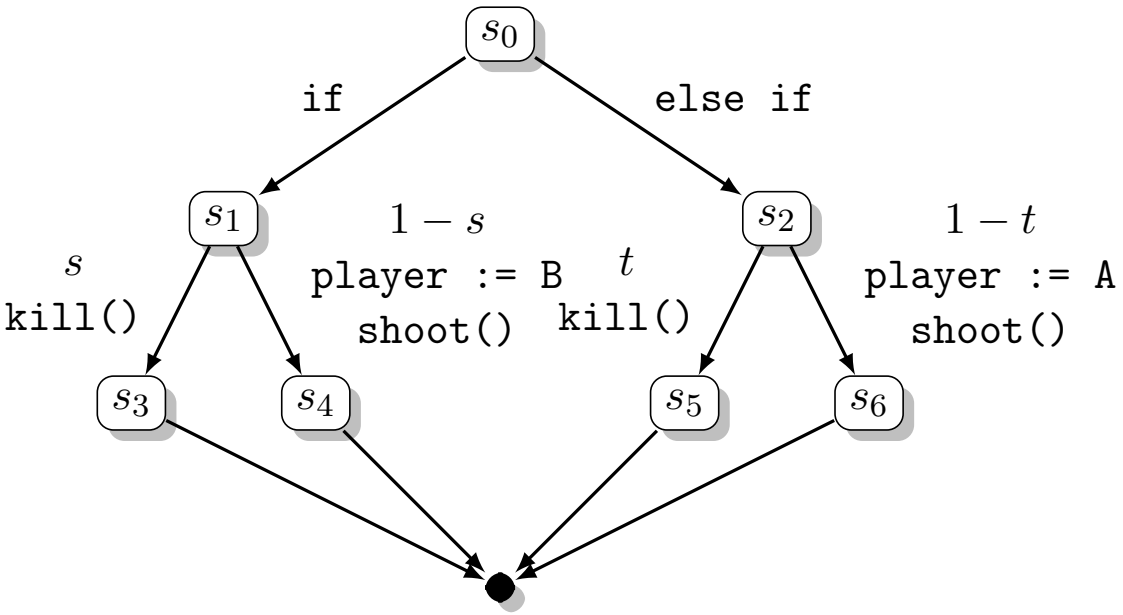
Why do we have this method?

- It yields an "easy" head (pk_0) for analysis
- All configuration with head pk_0 terminate

⇒ Picking good control points and methods is key!

pGCL → Control Flow Graph





Control Flow Graph $\rightarrow$ pPDA

Reminder: pPDA

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.
- Γ is a finite stack alphabet.
- $\hookrightarrow \subseteq (Q \times \Gamma) \times (Q \times \Gamma^*)$ is a set of transitions
- $\mathfrak{P}$ is a function which assigns probabilities to transitions

Control Flow Graph $\rightarrow$ pPDA

Reminder: pPDA

A probabilistic pushdown automaton (pPDA) is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ defined as

- Q is a finite set of states.
- Γ is a finite stack alphabet.
- $\hookrightarrow \subseteq (Q \times \Gamma) \times (Q \times \Gamma^*)$ is a set of transitions
- $\mathfrak{P}$ is a function which assigns probabilities to transitions

How does this relate to control flow graphs?

Control Flow Graph → pPDA

```
Exception in thread "main" java.lang.NullPointerException
        at main.Main.ThereWeAre(Main.java:54)
        at main.Main.ItsComin(Main.java:50)
        at main.Main.inb4NullPointerException(Main.java:46)
        at main.Main.goingToBeLegendary(Main.java:42)
        at main.Main.waitForIt(Main.java:38)
        at main.Main.getReadyForErrors(Main.java:34)
        at main.Main.IsThisEverGonnaStop(Main.java:30)
        at main.Main.Kappa(Main.java:26)
        at main.Main.probabilisticProgramsYEAH(Main.java:22)
        at main.Main.seminarAtI2(Main.java:18)
        at main.Main.onWeGo(Main.java:14)
        at main.Main.letsGo(Main.java:10)
        at main.Main.main(Main.java:6)
```

Control Flow Graph $\rightarrow$ pPDA

- Q: Variable assignments

Control Flow Graph $\rightarrow$ pPDA

- Q: Variable assignments
- Γ: Nodes in the graph

Control Flow Graph $\rightarrow$ pPDA

- Q: Variable assignments
- Γ: Nodes in the graph
- $\mathfrak{P}$: Probabilities of the edges

Control Flow Graph $\rightarrow$ pPDA

- Q: Variable assignments
- Γ: Nodes in the graph
- $\mathfrak{P}$: Probabilities of the edges
- $\hookrightarrow$: ???

Control Flow Graph $\rightarrow$ pPDA

- construct every pair $Q \times \Gamma$

Control Flow Graph $\rightarrow$ pPDA

- construct every pair $Q \times \Gamma$
- for each pair $\langle q, \gamma \rangle \in Q \times \Gamma$ consider edges of the control flow graph:

Control Flow Graph $\rightarrow$ pPDA

- construct every pair $Q \times \Gamma$
- for each pair $\langle q, \gamma \rangle \in Q \times \Gamma$ consider edges of the control flow graph:
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \gamma' \rangle$ if no function call between γ and γ'

Control Flow Graph $\rightarrow$ pPDA

- construct every pair $Q \times \Gamma$
- for each pair $\langle q, \gamma \rangle \in Q \times \Gamma$ consider edges of the control flow graph:
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \gamma' \rangle$ if no function call between γ and γ'
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \varphi\gamma' \rangle$ if function call between γ and γ'

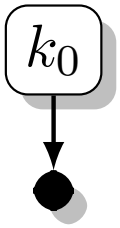
Control Flow Graph $\rightarrow$ pPDA

- construct every pair $Q \times \Gamma$
- for each pair $\langle q, \gamma \rangle \in Q \times \Gamma$ consider edges of the control flow graph:
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \gamma' \rangle$ if no function call between γ and γ'
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \varphi\gamma' \rangle$ if function call between γ and γ'
    - γ' return address
    - φ entry point of function call

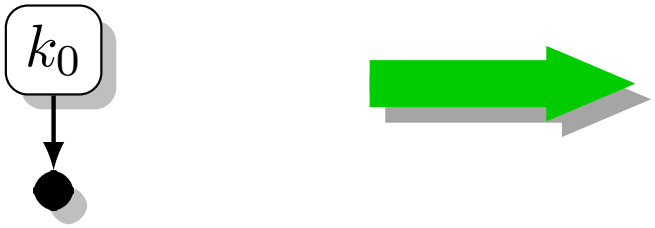
Control Flow Graph $\rightarrow$ pPDA

- construct every pair $Q \times \Gamma$
- for each pair $\langle q, \gamma \rangle \in Q \times \Gamma$ consider edges of the control flow graph:
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \gamma' \rangle$ if no function call between γ and γ'
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \varphi\gamma' \rangle$ if function call between γ and γ'
    - γ' return address
    - φ entry point of function call
  - $\langle q, \gamma \rangle \hookrightarrow \langle q', \varepsilon \rangle$ if function call terminates

Control Flow Graph $\rightarrow$ pPDA

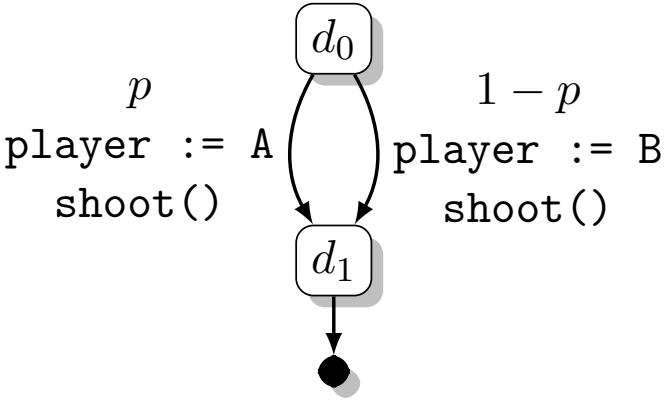


Control Flow Graph $\rightarrow$ pPDA

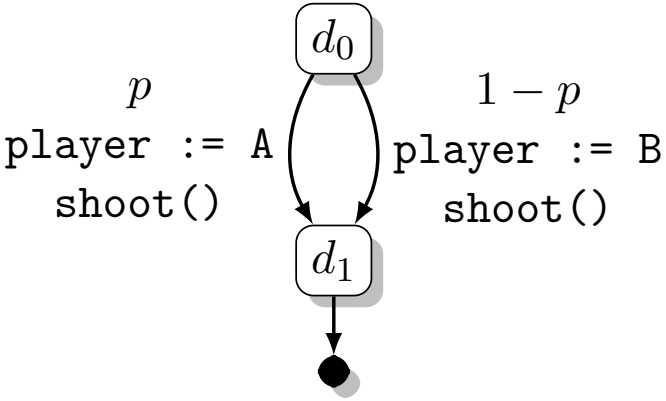


$$\langle Ak_0 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle Bk_0 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$

Control Flow Graph → pPDA

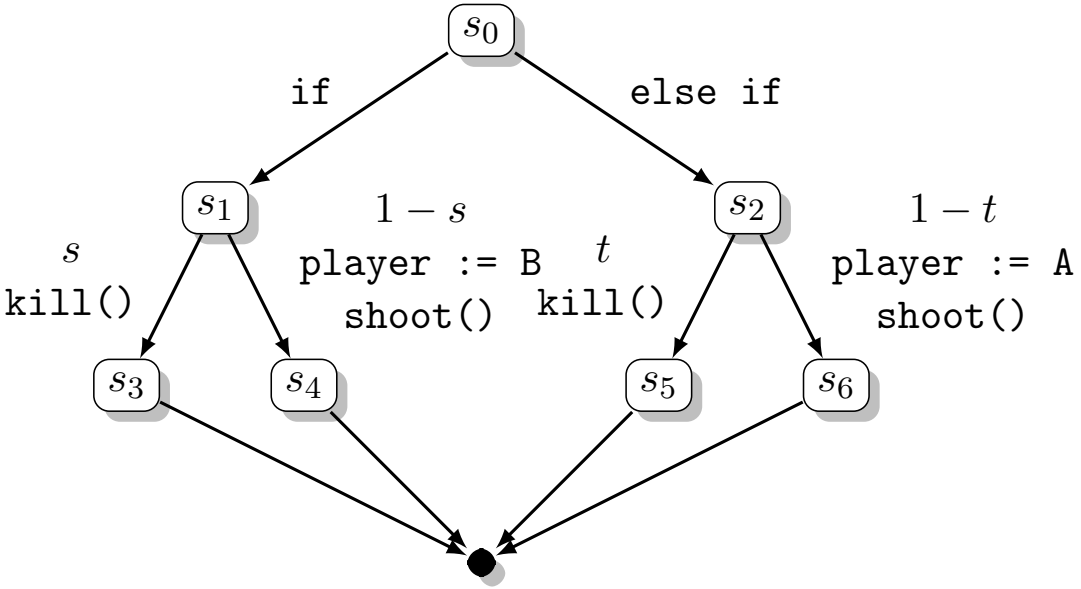


Control Flow Graph → pPDA

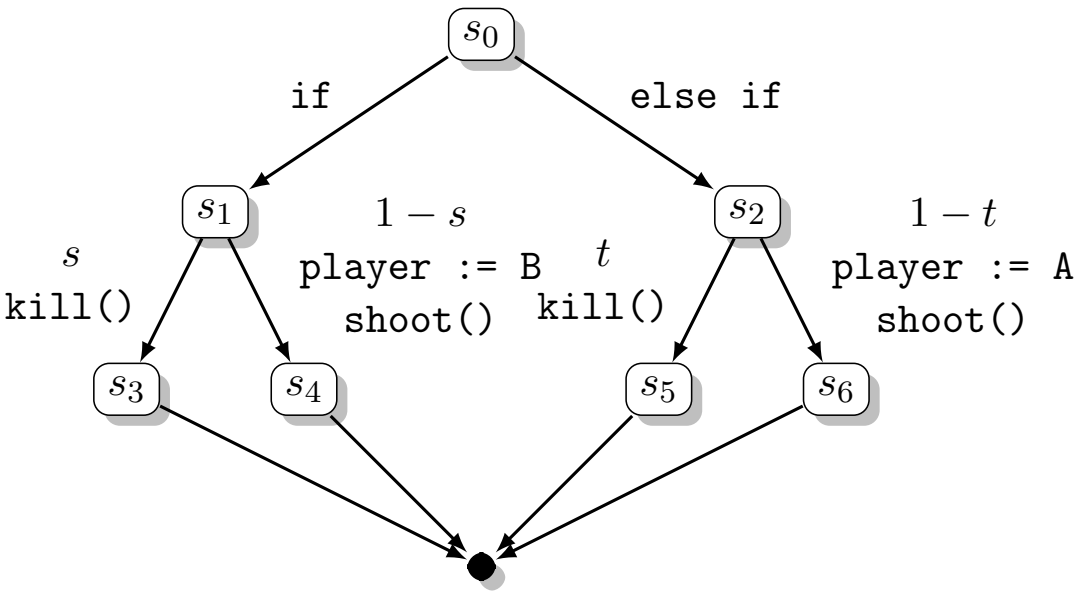


$$\langle Ad_0 \rangle \overset{p}{\hookrightarrow} \langle As_0d_1 \rangle$$
$$\langle Ad_0 \rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1 \rangle$$
$$\langle Ad_1 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle Bd_1 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$

Control Flow Graph → pPDA



Control Flow Graph → pPDA



$$\langle As_0\rangle \overset{1}{\hookrightarrow} \langle As_1\rangle$$
$$\langle Bs_0\rangle \overset{1}{\hookrightarrow} \langle Bs_2\rangle$$
$$\langle As_1\rangle \overset{s}{\hookrightarrow} \langle Ak_0s_3\rangle$$
$$\langle As_1\rangle \overset{1-s}{\hookrightarrow} \langle Bs_0s_4\rangle$$
$$\langle As_3\rangle \overset{1}{\hookrightarrow} \langle A\varepsilon\rangle$$
$$\langle As_4\rangle \overset{1}{\hookrightarrow} \langle A\varepsilon\rangle$$
$$\langle Bs_4\rangle \overset{1}{\hookrightarrow} \langle B\varepsilon\rangle$$
$$\langle Bs_2\rangle \overset{t}{\hookrightarrow} \langle Bk_0s_5\rangle$$
$$\langle Bs_2\rangle \overset{1-t}{\hookrightarrow} \langle As_0s_6\rangle$$
$$\langle Bs_5\rangle \overset{1}{\hookrightarrow} \langle B\varepsilon\rangle$$
$$\langle Bs_6\rangle \overset{1}{\hookrightarrow} \langle B\varepsilon\rangle$$
$$\langle As_6\rangle \overset{1}{\hookrightarrow} \langle A\varepsilon\rangle$$

Control Flow Graph → pPDA

$$\langle Ad_0 \rangle \overset{p}{\hookrightarrow} \langle As_0 d_1 \rangle$$
$$\langle Ad_0 \rangle \overset{1-p}{\hookrightarrow} \langle Bs_0 d_1 \rangle$$
$$\langle Ad_1 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle Bd_1 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$
$$\langle As_0 \rangle \overset{1}{\hookrightarrow} \langle As_1 \rangle$$
$$\langle Bs_0 \rangle \overset{1}{\hookrightarrow} \langle Bs_2 \rangle$$

$$\langle As_1 \rangle \overset{s}{\hookrightarrow} \langle Ak_0 s_3 \rangle$$
$$\langle As_1 \rangle \overset{1-s}{\hookrightarrow} \langle Bs_0 s_4 \rangle$$
$$\langle As_3 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle As_4 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle Bs_4 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$
$$\langle Bs_2 \rangle \overset{t}{\hookrightarrow} \langle Bk_0 s_5 \rangle$$

$$\langle Bs_2 \rangle \overset{1-t}{\hookrightarrow} \langle As_0 s_6 \rangle$$
$$\langle Bs_5 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$
$$\langle Bs_6 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$
$$\langle As_6 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle Ak_0 \rangle \overset{1}{\hookrightarrow} \langle A\varepsilon \rangle$$
$$\langle Bk_0 \rangle \overset{1}{\hookrightarrow} \langle B\varepsilon \rangle$$

Reminder: Properties of pPDA

Properties of pPDA

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $pX \in Q \times \Gamma$ an initial configuration and $\mathcal{C} \subseteq Q \times \Gamma^*$ a set of target configurations.

- Qualitative reachability/termination: $\mathscr{P}(Reach(pX, C)) = 1$?
- Quantitative reachability/termination: $\mathscr{P}(Reach(pX, C)) \leq d$ for a constant $d \in (0, 1)$?

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.
$\Rightarrow$ Extensible to regular sets of target configurations

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.
$\Rightarrow$ Extensible to regular sets of target configurations

Notation

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $\mathcal{C} \subseteq Q \times \Gamma^*$ a simple set of target configurations and $\mathcal{H}$ be the associated set of heads.

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.
$\Rightarrow$ Extensible to regular sets of target configurations

Notation

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $\mathcal{C} \subseteq Q \times \Gamma^*$ a simple set of target configurations and $\mathcal{H}$ be the associated set of heads.

- $[pX\bullet] = \mathscr{P}(Reach(pX, \mathcal{C}))$

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.
$\Rightarrow$ Extensible to regular sets of target configurations

Notation

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $\mathcal{C} \subseteq Q \times \Gamma^*$ a simple set of target configurations and $\mathcal{H}$ be the associated set of heads.

- $[pX\bullet] = \mathscr{P}(Reach(pX, \mathcal{C}))$
- $[pXq] =$ probability of all $r \in Reach(pX, \{q\varepsilon\})$ that do not visit $\mathcal{C}$

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.
$\Rightarrow$ Extensible to regular sets of target configurations

Notation

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $\mathcal{C} \subseteq Q \times \Gamma^*$ a simple set of target configurations and $\mathcal{H}$ be the associated set of heads.

- $[pX\bullet] = \mathscr{P}(Reach(pX, \mathcal{C}))$
- $[pXq] =$ probability of all $r \in Reach(pX, \{q\varepsilon\})$ that do not visit $\mathcal{C}$

For $pX \in \mathcal{H}$:

Qualitative Reachability/Termination

We solve this problem for a simple set of target configurations.
$\Rightarrow$ Extensible to regular sets of target configurations

Notation

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA, $\mathcal{C} \subseteq Q \times \Gamma^*$ a simple set of target configurations and $\mathcal{H}$ be the associated set of heads.

- $[pX\bullet] = \mathscr{P}(Reach(pX, \mathcal{C}))$
- $[pXq] =$ probability of all $r \in Reach(pX, \{q\varepsilon\})$ that do not visit $\mathcal{C}$

For $pX \in \mathcal{H}$:

- $[pX\bullet] = 1$
- $[pXq] = 0$

Qualitative Reachability/Termination

For $pX \notin \mathcal{H}$:

Qualitative Reachability/Termination

For $pX \notin \mathcal{H}$:

$$[pX\bullet] = \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet]$$

Qualitative Reachability/Termination

For $pX \notin \mathcal{H}$:

$$[pX\bullet] = \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet]$$

$$[pXq] = \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rYq] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZq]$$

Qualitative Reachability/Termination

For $pX \notin \mathcal{H}$:

$$[pX\bullet] = \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet]$$

$$[pXq] = \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rYq] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZq]$$

1. Construct equations for all $[pX\bullet]$ and $[pXq]$ needed.

Qualitative Reachability/Termination

For $pX \notin \mathcal{H}$:

$$[pX\bullet] = \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet]$$

$$[pXq] = \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rYq] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZq]$$

1. Construct equations for all $[pX\bullet]$ and $[pXq]$ needed.
2. Replace all $[pX\bullet]$ and $[pXq]$ by fresh random variables $\langle pX\bullet \rangle$ and $\langle pXq \rangle$

Qualitative Reachability/Termination

For $pX \notin \mathcal{H}$:

$$[pX\bullet] = \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot [rY\bullet] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet]$$

$$[pXq] = \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rYq] + \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZq]$$

1. Construct equations for all $[pX\bullet]$ and $[pXq]$ needed.
2. Replace all $[pX\bullet]$ and $[pXq]$ by fresh random variables $\langle pX\bullet \rangle$ and $\langle pXq \rangle$
3. Solve the system of recursive equations for least solution.

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$[Ad_0\bullet] = \sum_{Ad_0 \overset{x}{\hookrightarrow} rY} x \cdot [rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} x \cdot [rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet]$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$[Ad_0\bullet] = \sum_{Ad_0 \overset{x}{\hookrightarrow} rY} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} \sum_{t\in Q} x\cdot[rYt]\cdot[tZ\bullet]$$

$$\langle Ad_0\rangle \overset{p}{\hookrightarrow} \langle As_0d_1\rangle$$
$$\langle Ad_0\rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1\rangle$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$[Ad_0\bullet] = \sum_{Ad_0 \overset{x}{\hookrightarrow} rY} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} \sum_{t\in Q} x\cdot[rYt]\cdot[tZ\bullet]$$
$$= p\cdot[As_0\bullet]$$

$$\langle Ad_0\rangle \overset{p}{\hookrightarrow} \langle As_0d_1\rangle$$
$$\langle Ad_0\rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1\rangle$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$[Ad_0\bullet] = \sum_{Ad_0 \overset{x}{\hookrightarrow} rY} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} \sum_{t\in Q} x\cdot[rYt]\cdot[tZ\bullet]$$
$$= p\cdot[As_0\bullet] + (1-p)\cdot[Bs_0\bullet]$$

$$\langle Ad_0\rangle \overset{p}{\hookrightarrow} \langle As_0d_1\rangle$$
$$\langle Ad_0\rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1\rangle$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$
\begin{aligned}
[Ad_0\bullet] &= \sum_{Ad_0 \overset{x}{\hookrightarrow} rY} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} x\cdot[rY\bullet] + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} \sum_{t\in Q} x\cdot[rYt]\cdot[tZ\bullet] \\
&= p\cdot[As_0\bullet] + (1-p)\cdot[Bs_0\bullet] \\
&\qquad + \sum_{Ad_0 \overset{x}{\hookrightarrow} rYZ} \left(x\cdot[rYA]\cdot[AZ\bullet] + x\cdot[rYB]\cdot[BZ\bullet]\right)
\end{aligned}
$$

$$\langle Ad_0\rangle \overset{p}{\hookrightarrow} \langle As_0d_1\rangle$$
$$\langle Ad_0\rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1\rangle$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$
\begin{aligned}
[Ad_0\bullet] = p\cdot[As_0\bullet] + (1-p)\cdot[Bs_0\bullet] \\
+ (p\cdot[As_0A]\cdot[Ad_1\bullet] + p\cdot[As_0B]\cdot[Bd_1\bullet]) \\
+ ((1-p)\cdot[Bs_0A]\cdot[Ad_1\bullet] + (1-p)\cdot[Bs_0B]\cdot[Bd_1\bullet])
\end{aligned}
$$

$$\langle Ad_0\rangle \overset{p}{\hookrightarrow} \langle As_0d_1\rangle$$
$$\langle Ad_0\rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1\rangle$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$\begin{aligned}[Ad_0\bullet] = p\cdot[As_0\bullet] &+ (1-p)\cdot[Bs_0\bullet] \\ &+ (p\cdot[As_0A]\cdot[Ad_1\bullet] + p\cdot[As_0B]\cdot[Bd_1\bullet]) \\ &+ ((1-p)\cdot[Bs_0A]\cdot[Ad_1\bullet] + (1-p)\cdot[Bs_0B]\cdot[Bd_1\bullet])\end{aligned}$$

$\Rightarrow$ Remove zero summands

$$\langle Ad_0\rangle \overset{p}{\hookrightarrow} \langle As_0d_1\rangle$$
$$\langle Ad_0\rangle \overset{1-p}{\hookrightarrow} \langle Bs_0d_1\rangle$$

Qualitative Term./Reach. Dueling Cowboys

Let $\mathcal{C}$ be a set of target configurations with $\mathcal{H} = \{Ak_0, Bk_0\}$
$\Rightarrow \mathscr{P}(Reach(Ad_0, \mathcal{C})) = [Ad_0\bullet] = 1$ means qualitative termination

$$[Ad_0\bullet] = p\cdot[As_0\bullet] + (1-p)\cdot[Bs_0\bullet]$$

$\Rightarrow$ We need to calculate $[As_0\bullet]$ and $[Bs_0\bullet]$!

Qualitative Term./Reach. Dueling Cowboys

$$[Ad_0\bullet] = p\cdot[As_0\bullet] + (1-p)\cdot[Bs_0\bullet]$$
$$[As_0\bullet] = s + (1-s)\cdot[Bs_0\bullet]$$
$$[Bs_0\bullet] = t + (1-t)\cdot[As_0\bullet]$$

Qualitative Term./Reach. Dueling Cowboys

$$[Ad_0\bullet] = p\cdot[As_0\bullet] + (1-p)\cdot[Bs_0\bullet]$$
$$[As_0\bullet] = s + (1-s)\cdot[Bs_0\bullet]$$
$$[Bs_0\bullet] = t + (1-t)\cdot[As_0\bullet]$$

$$\langle Ad_0\bullet\rangle = p\cdot\langle As_0\bullet\rangle + (1-p)\cdot\langle Bs_0\bullet\rangle$$
$$\langle As_0\bullet\rangle = s + (1-s)\cdot\langle Bs_0\bullet\rangle$$
$$\langle Bs_0\bullet\rangle = t + (1-t)\cdot\langle As_0\bullet\rangle$$

Qualitative Term./Reach. Dueling Cowboys

$$\langle Ad_0\bullet\rangle = p\cdot\langle As_0\bullet\rangle + (1-p)\cdot\langle Bs_0\bullet\rangle$$
$$\langle As_0\bullet\rangle = s + (1-s)\cdot\langle Bs_0\bullet\rangle$$
$$\langle Bs_0\bullet\rangle = t + (1-t)\cdot\langle As_0\bullet\rangle$$

Qualitative Term./Reach. Dueling Cowboys

$$\langle Ad_0\bullet\rangle = p\cdot\langle As_0\bullet\rangle + (1-p)\cdot\langle Bs_0\bullet\rangle$$
$$\langle As_0\bullet\rangle = s + (1-s)\cdot\langle Bs_0\bullet\rangle$$
$$\langle Bs_0\bullet\rangle = t + (1-t)\cdot\langle As_0\bullet\rangle$$

$$p = 1/2$$
$$s = 1/2$$
$$t = 1/3$$

Qualitative Term./Reach. Dueling Cowboys

$$\langle Ad_0\bullet\rangle = p\cdot\langle As_0\bullet\rangle + (1-p)\cdot\langle Bs_0\bullet\rangle$$
$$\langle As_0\bullet\rangle = s + (1-s)\cdot\langle Bs_0\bullet\rangle$$
$$\langle Bs_0\bullet\rangle = t + (1-t)\cdot\langle As_0\bullet\rangle$$

$$p = 1/2$$
$$s = 1/2$$
$$t = 1/3$$

$$\langle Ad_0\bullet\rangle = 1, \langle As_0\bullet\rangle = 1, \langle Bs_0\bullet\rangle = 1$$

Qualitative Term./Reach. Dueling Cowboys

$$\langle Ad_0\bullet\rangle = p\cdot\langle As_0\bullet\rangle + (1-p)\cdot\langle Bs_0\bullet\rangle$$
$$\langle As_0\bullet\rangle = s + (1-s)\cdot\langle Bs_0\bullet\rangle$$
$$\langle Bs_0\bullet\rangle = t + (1-t)\cdot\langle As_0\bullet\rangle$$

$$p = 1/2$$
$$s = 0$$
$$t = 0$$

$$\langle Ad_0\bullet\rangle = 0, \langle As_0\bullet\rangle = 0, \langle Bs_0\bullet\rangle = 0$$

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$
- What is the probability that e.g. A wins the duel?

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$
- What is the probability that e.g. A wins the duel?
- $\Rightarrow [Ad_0A] = ?, \mathcal{C} = \emptyset$

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$
- What is the probability that e.g. A wins the duel?
- $\Rightarrow [Ad_0A] = ?, \mathcal{C} = \emptyset$
- Same equations apply!

$$
\begin{aligned}
[Ad_0A] &= p \cdot [As_0A] + (1-p)[Bs_0A] \\
[As_0A] &= s + (1-s)[Bs_0A] \\
[Bs_0A] &= (1-t)[As_0A]
\end{aligned}
$$

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$
- What is the probability that e.g. A wins the duel?
- $\Rightarrow [Ad_0A] = ?, \mathcal{C} = \emptyset$
- Same equations apply!

$$\langle Ad_0A \rangle = p \cdot \langle As_0A \rangle + (1-p)\langle Bs_0A \rangle$$
$$\langle As_0A \rangle = s + (1-s)\langle Bs_0A \rangle$$
$$\langle Bs_0A \rangle = (1-t)\langle As_0A \rangle$$

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$
- What is the probability that e.g. A wins the duel?
- $\Rightarrow [Ad_0A] = ?, \mathcal{C} = \emptyset$
- Same equations apply!

$p = 1/2, s = 1/2, t = 1/2$

Quantitative Term./Reach Dueling Cowboys

- We know that the duel ends if both $s, t > 0$
- What is the probability that e.g. A wins the duel?
- $\Rightarrow [Ad_0A] = ?, \mathcal{C} = \emptyset$
- Same equations apply!

$p = 1/2, s = 1/2, t = 1/2$

$$\langle Ad_0A \rangle = 1/2$$
$$\langle As_0A \rangle = 2/3$$
$$\langle Bs_0A \rangle = 1/3$$

Conclusion

- pPDA nice way to model and analyze systems that use probability as well as recursion

Conclusion

- pPDA nice way to model and analyze systems that use probability as well as recursion
- 2 formulae enough to study quantitative and qualitative reachability

Conclusion

- pPDA nice way to model and analyze systems that use probability as well as recursion
- 2 formulae enough to study quantitative and qualitative reachability
- Can be done efficiently in polynomial space

Conclusion

- pPDA nice way to model and analyze systems that use probability as well as recursion
- 2 formulae enough to study quantitative and qualitative reachability
- Can be done efficiently in polynomial space
- Construction technique to pPBA allows analysis in polynomial time.

Expected Runtime

Definition: Runtime

Let Δ be a transition system with a set of configurations S and target configurations T.

Expected Runtime

Definition: Runtime

Let Δ be a transition system with a set of configurations S and target configurations T.
Let $f\colon S \to \mathbb{R}_{\geq 0}$ be a function that assigns run times to states.

Expected Runtime

Definition: Runtime

Let Δ be a transition system with a set of configurations S and target configurations T.
Let $f\colon S \to \mathbb{R}_{\geq 0}$ be a function that assigns run times to states.
For every run $r \in Runs(\Delta)$ we define k_r as the least $n \in \mathbb{N}_0$ such that $r(n) \in T$.

Expected Runtime

Definition: Runtime

Let Δ be a transition system with a set of configurations S and target configurations T.
Let $f\colon S \to \mathbb{R}_{\geq 0}$ be a function that assigns run times to states.
For every run $r \in Runs(\Delta)$ we define k_r as the least $n \in \mathbb{N}_0$ such that $r(n) \in T$.

$$Time(\Delta) = \sum_{i=0}^{k_r} f(r(i)) \ .$$

Expected Runtime

Definition: Runtime

Let Δ be a transition system with a set of configurations S and target configurations T.
Let $f\colon S \to \mathbb{R}_{\geq 0}$ be a function that assigns run times to states.
For every run $r \in Runs(\Delta)$ we define k_r as the least $n \in \mathbb{N}_0$ such that $r(n) \in T$.

$$Time(\Delta) = \sum_{i=0}^{k_r} f(r(i)) \ .$$

$\Rightarrow k_r$ depends on probability

Expected Runtime

Definition: Runtime

Let Δ be a transition system with a set of configurations S and target configurations T.
Let $f\colon S \to \mathbb{R}_{\geq 0}$ be a function that assigns run times to states.
For every run $r \in Runs(\Delta)$ we define k_r as the least $n \in \mathbb{N}_0$ such that $r(n) \in T$.

$$Time(\Delta) = \sum_{i=0}^{k_r} f(r(i)) \ .$$

$\Rightarrow k_r$ depends on probability
$\Rightarrow \mathbb{E}(Time(\Delta)),\ \mathbb{E}(Time(\Delta)|Reach(pX, \mathcal{C}))$

Expected Runtime

Definition: Expected Runtime pPDA

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA.
$E_{pXq} = \mathbb{E}(Time(\Delta) | Reach(pX, \{q\varepsilon\}))$ conditional expected runtime

Expected Runtime

Definition: Expected Runtime pPDA

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA.
$E_{pXq} = \mathbb{E}(Time(\Delta) | Reach(pX, \{q\varepsilon\}))$ conditional expected runtime

$$\langle E_{pXq} \rangle = f(pX) + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot \frac{[rYq]}{[pXq]} \cdot \langle E_{rYq} \rangle$$
$$+ \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot \frac{[rYt] \cdot [tZq]}{[pXq]} \cdot (\langle E_{rYt} \rangle + \langle E_{tZq} \rangle)$$

Expected Runtime

Definition: Expected Runtime pPDA

Let $\Delta = (Q, \Gamma, \hookrightarrow, \mathfrak{P})$ be a pPDA.
$E_{pXq} = \mathbb{E}(Time(\Delta) | Reach(pX, \{q\varepsilon\}))$ conditional expected runtime

$$\langle E_{pXq} \rangle = f(pX) + \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot \frac{[rYq]}{[pXq]} \cdot \langle E_{rYq} \rangle$$
$$+ \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} x \cdot \frac{[rYt] \cdot [tZq]}{[pXq]} \cdot (\langle E_{rYt} \rangle + \langle E_{tZq} \rangle)$$

$\Rightarrow$ Here: $f(pX) = 1$ for all $pX \in Q \times \Gamma \Rightarrow$ discrete time

Expected Runtime

$$
\begin{aligned}
\mathscr{P}(Reach(rY, \{q\varepsilon\}) | Reach(pX, \{q\varepsilon\})) &= \frac{\mathscr{P}(Reach(rY, \{q\varepsilon\}) \cap Reach(pX, \{q\varepsilon\}))}{\mathscr{P}(Reach(pX, \{q\varepsilon\}))} \\
&= \frac{\mathscr{P}(Reach(rY, \{q\varepsilon\}))}{\mathscr{P}(Reach(pX, \{q\varepsilon\}))} \\
&= \frac{[rYq]}{[pXq]}
\end{aligned}
$$

Expected Runtime

Expected runtime for all runs that start in Ad_0 and end in $A\varepsilon$:

Expected Runtime

Expected runtime for all runs that start in Ad_0 and end in $A\varepsilon$:

$$\langle E_{As_0A}\rangle = 1 + \frac{[As_1A]}{[As_0A]} \cdot \langle E_{As_1A}\rangle$$

$$\langle E_{As_1A}\rangle = 1 + \frac{s}{[As_1A]} \cdot (\langle E_{Ak_0A}\rangle + \langle E_{As_3A}\rangle) + (1-s) \cdot \frac{[Bs_0A]}{[As_1A]}(\langle E_{Bs_0A}\rangle)$$

$$\langle E_{Bs_0A}\rangle = 1 + \frac{[Bs_2A]}{[Bs_0A]} \cdot \langle E_{Bs_2A}\rangle$$

$$\langle E_{Bs_2A}\rangle = 1 + (1-t) \cdot \frac{[As_0A]}{[Bs_2A]} \cdot (\langle E_{As_0A}\rangle + \langle E_{As_6A}\rangle)$$

$$\langle E_{Ad_1A}\rangle = 1$$

$$\langle E_{Ak_0A}\rangle = 1$$

$$\langle E_{As_3A}\rangle = 1$$

$$\langle E_{As_6A}\rangle = 1$$

Expected Runtime

Let's assume $p = s = t = 1/2$

Expected Runtime

Let's assume $p = s = t = 1/2$

$$
\begin{aligned}
\langle E_{Ad_0A} \rangle &= 2 + \frac{2}{3} \cdot \langle As_0A \rangle + \frac{1}{3} \cdot \langle Bs_0A \rangle \\
\langle E_{As_0A} \rangle &= 1 + \langle E_{As_1A} \rangle \\
\langle E_{As_1A} \rangle &= \frac{5}{2} + \frac{1}{4} \langle E_{Bs_0A} \rangle \\
\langle E_{Bs_0A} \rangle &= 1 + \langle E_{Bs_2A} \rangle \\
\langle E_{Bs_2A} \rangle &= 2 + \langle E_{As_0A} \rangle
\end{aligned}
$$

Expected Runtime

Let's assume $p = s = t = 1/2$

$$
\begin{aligned}
\langle E_{Ad_0A} \rangle &= 2 + \frac{2}{3} \cdot \langle As_0A \rangle + \frac{1}{3} \cdot \langle Bs_0A \rangle \\
\langle E_{As_0A} \rangle &= 1 + \langle E_{As_1A} \rangle \\
\langle E_{As_1A} \rangle &= \frac{5}{2} + \frac{1}{4} \langle E_{Bs_0A} \rangle \\
\langle E_{Bs_0A} \rangle &= 1 + \langle E_{Bs_2A} \rangle \\
\langle E_{Bs_2A} \rangle &= 2 + \langle E_{As_0A} \rangle
\end{aligned}
$$

$\Rightarrow \langle E_{Ad_0A} \rangle = 26/3 = 8.\overline{66}$

Quantitative System Parametrized

$$\langle Ad_0A \rangle = \frac{-p - (1-p) \cdot (1-t)}{2(st - s - t)}$$

$$\langle As_0A \rangle = \frac{-1}{2(st - s - t)}$$

$$\langle Bs_0A \rangle = \frac{-(1-t)}{2(st - s - t)}$$