Unit-6: Model-checking ω-regular properties

B. Srivathsan

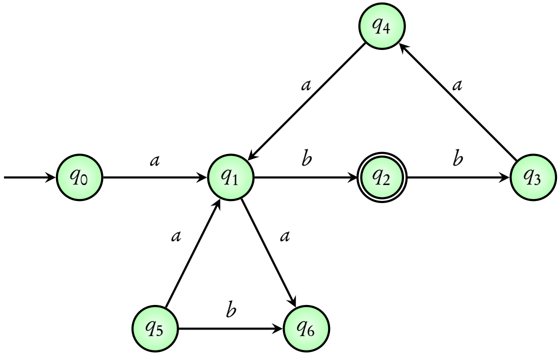
Chennai Mathematical Institute

NPTEL-course

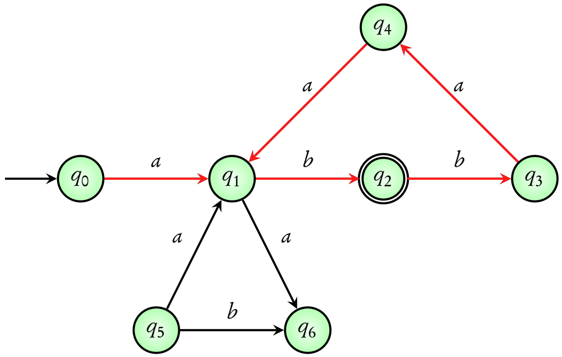
July - November 2015

Module 3:

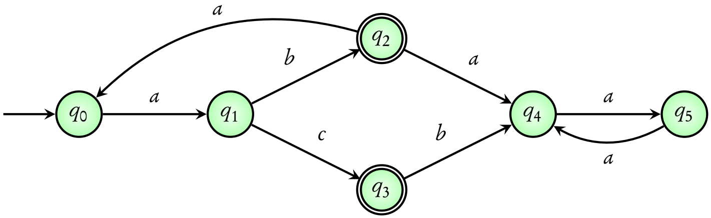
Checking emptiness of Büchi automata



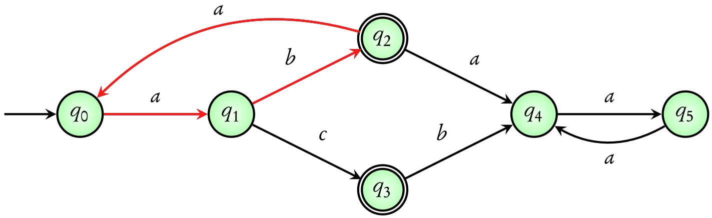
Is the **language** of above NBA **empty**?



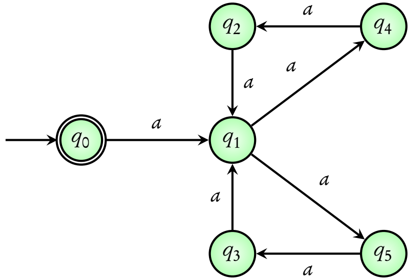
Is the **language** of above NBA **empty**? **No**



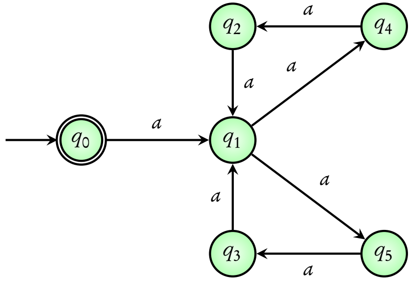
Is the **language** of above NBA **empty**?



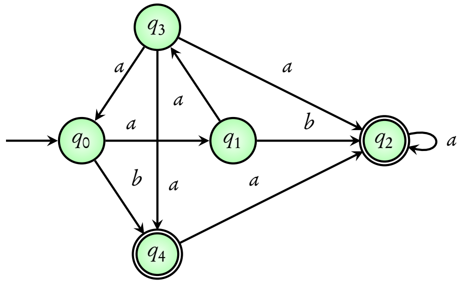
Is the **language** of above NBA **empty**? **No**



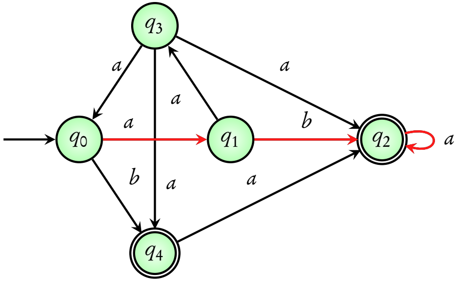
Is the **language** of above NBA **empty**?


$$\begin{tikzcd} & q_2 \arrow[d, "a"] & & q_4 \arrow[ll, "a"'] \\ \arrow[r] q_0 \arrow[r, "a"] & q_1 \arrow[urr, "a"] \arrow[drr, "a"] & & \\ & q_3 \arrow[u, "a"] & & q_5 \arrow[ll, "a"] \end{tikzcd}$$

Is the **language** of above NBA **empty**? **Yes**



Is the **language** of above NBA **empty**?



Is the **language** of above NBA **empty**? **No**

Main idea of algorithm

Find a **reachable cycle** in the automaton that contains an **accepting state**

Main idea of algorithm

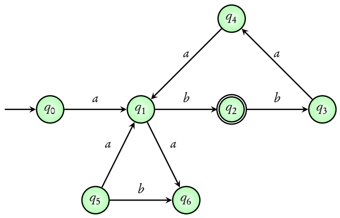
Find a **reachable cycle** in the automaton that contains an **accepting state**

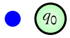
- Do a preliminary DFS to get all **reachable states**
- From every **accepting state**, do a secondary DFS to see if it can **come back to itself**

Coming next: A **nested-DFS** algorithm

Courcoubetis, Vardi, Wolper, Yannakakis. Memory-efficient algorithms for the verification of temporal properties

Formal Methods in System Design, 1992

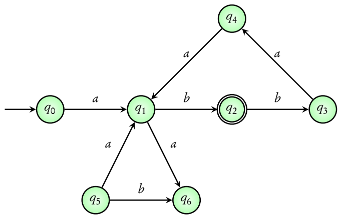


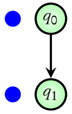


```
procedure $nested\_dfs()$
    call $dfs\_blue\ (s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\ (t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\ (s)$

procedure $dfs\_red\ (s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\ (t)$
        else if $t = seed$
            report cycle
```



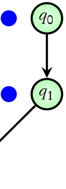


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

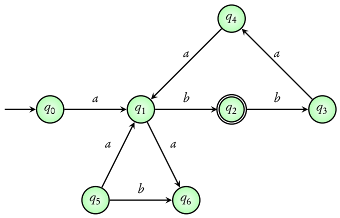
$$\begin{tikzcd} & & & q_4 \arrow[dll, "a"'] & \\ {} \arrow[r] & q_0 \arrow[r, "a"] & q_1 \arrow[r, "b"] \arrow[dr, "a"] & q_2 \arrow[r, "b"] & q_3 \arrow[ul, "a"'] \\ & q_5 \arrow[ur, "a"] \arrow[rr, "b"] & & q_6 & \end{tikzcd}$$

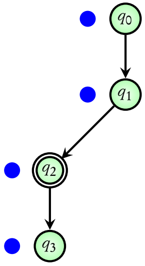


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

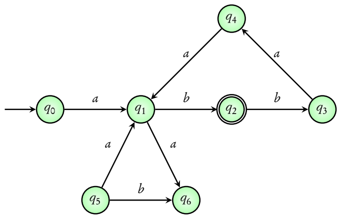


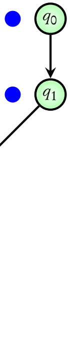


```
**procedure** *nested_dfs*( )
    **call** *dfs_blue* ( $s_0$ )

**procedure** *dfs_blue*( $s$ )
    $s.blue$ := **true**
    **for all** $t \in post(s)$ **do**
        **if** $\neg t.blue$ **then**
            **call** *dfs_blue* ( $t$ )
    **if** $s \in Accept$ **then**
        $seed$ := $s$
        **call** *dfs_red* ( $s$ )

**procedure** *dfs_red* ( $s$ )
    $s.red$ := **true**
    **for all** $t \in post(s)$ **do**
        **if** $\neg t.red$ **then**
            **call** *dfs_red* ( $t$ )
        **else if** $t = seed$
            **report cycle**
```

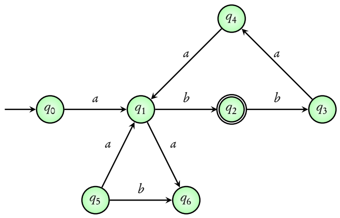


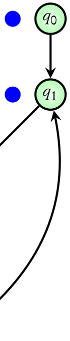


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

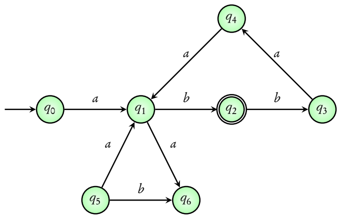


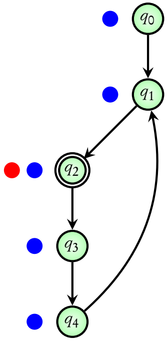


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

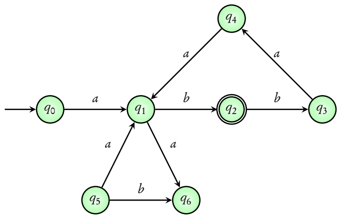


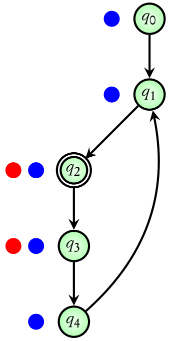


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

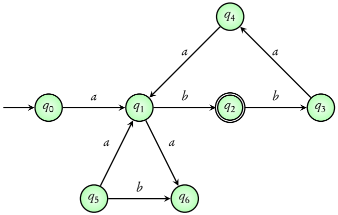


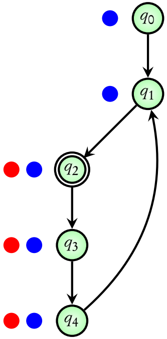


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

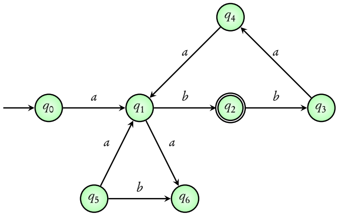


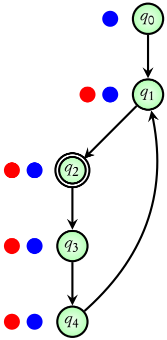


```
procedure nested_dfs()
    call dfs_blue($s_0$)

procedure dfs_blue($s$)
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue($t$)
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red($s$)

procedure dfs_red($s$)
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red($t$)
        else if $t = seed$
            report cycle
```

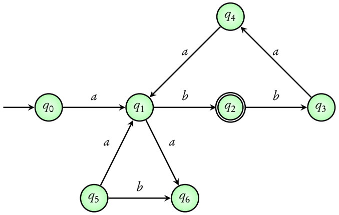


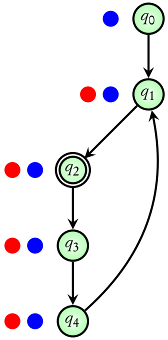


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```



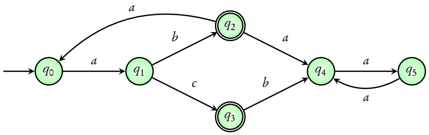


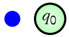
report cycle!

```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

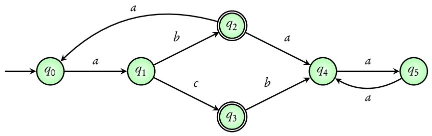


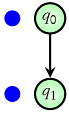


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

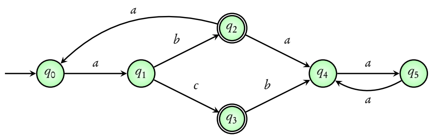


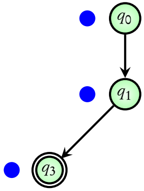


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

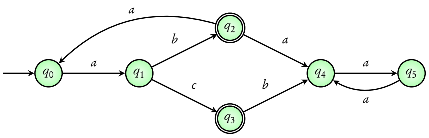


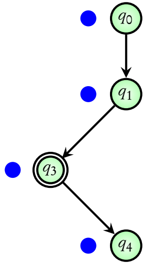


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

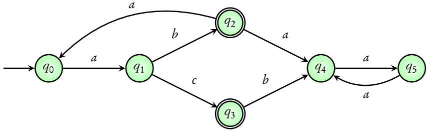


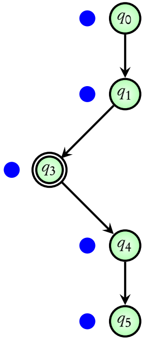


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

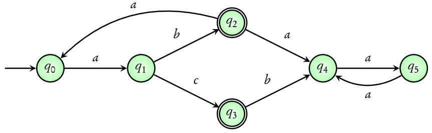


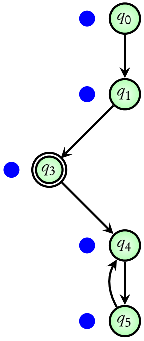


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

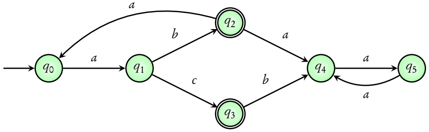


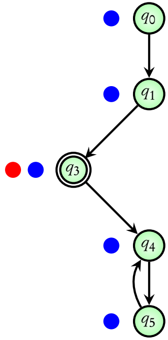


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

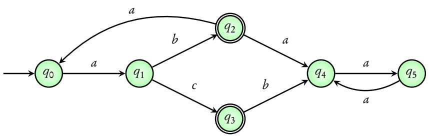


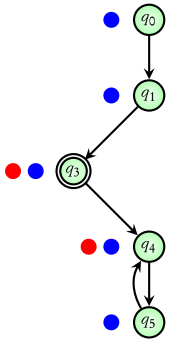


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

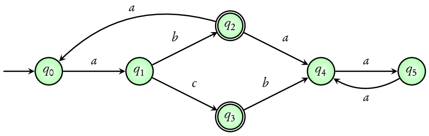


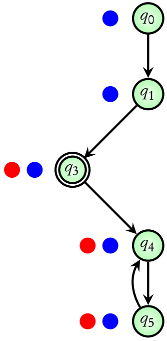


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

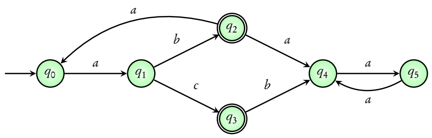


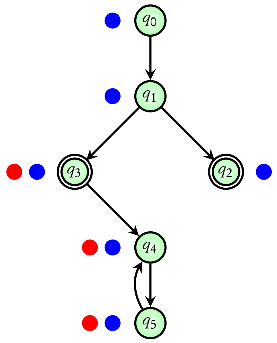


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := \textbf{true}$
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := \textbf{true}$
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

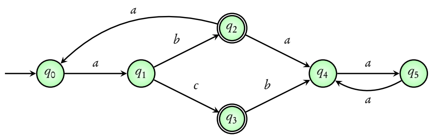


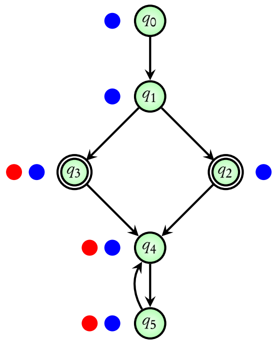


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

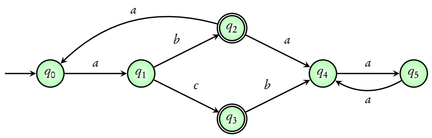


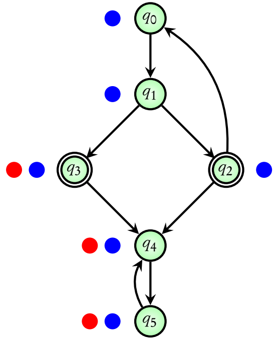


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red ( s )

procedure dfs_red ( s )
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

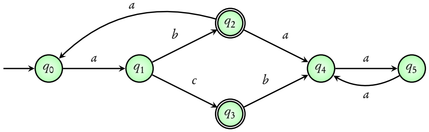


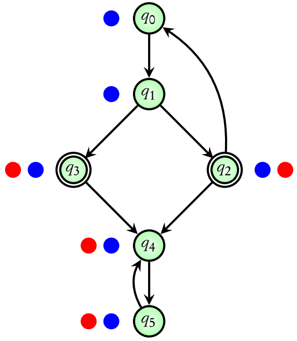


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red ( s )

procedure dfs_red ( s )
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

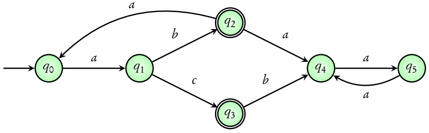


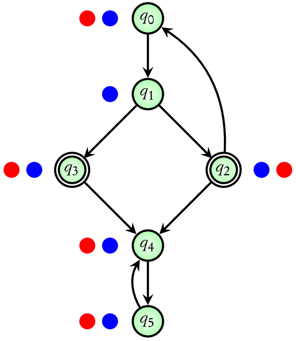


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

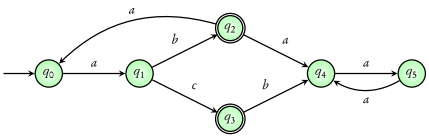


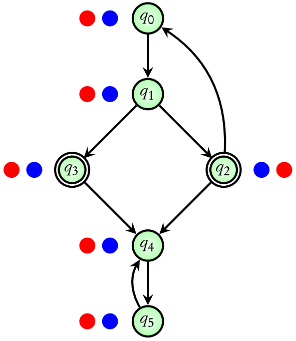


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

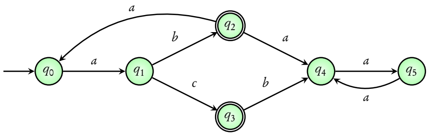


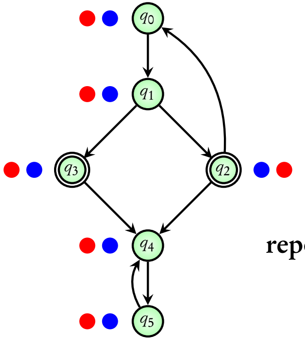


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```



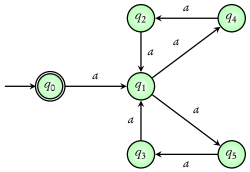


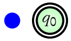
report cycle!

```
procedure nested_dfs( )
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red ( s )

procedure dfs_red ( s )
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

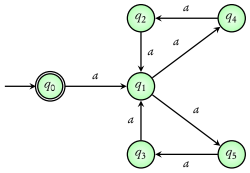


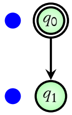


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

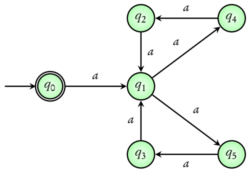


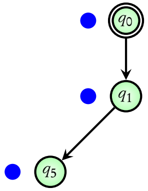


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

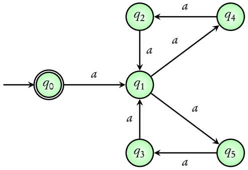


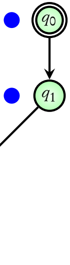


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red ( s )

procedure dfs_red ( s )
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

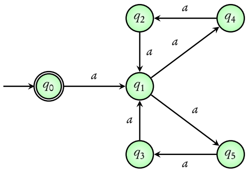


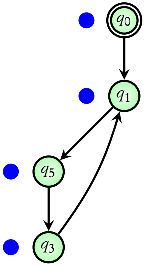


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( $s$ )
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( $t$ )
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red ( $s$ )

procedure dfs_red ( $s$ )
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( $t$ )
        else if $t = seed$
            report cycle
```

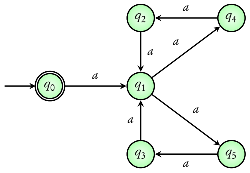


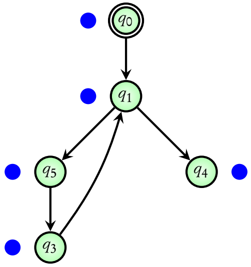


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

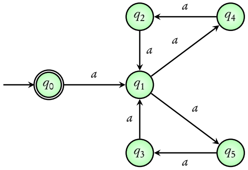


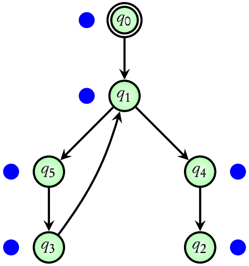


```
procedure nested_dfs()
    call dfs_blue ($s_0$)

procedure dfs_blue(s)
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue (t)
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red (s)

procedure dfs_red (s)
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red (t)
        else if $t = seed$
            report cycle
```

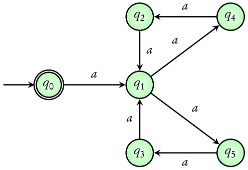


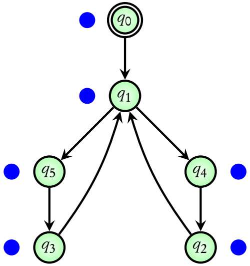


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

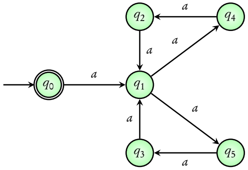


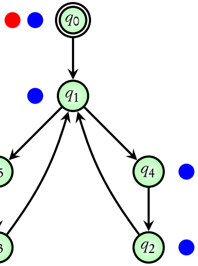


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

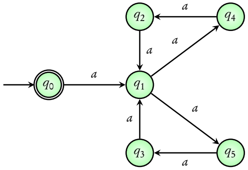


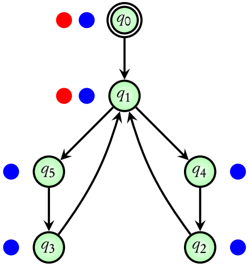


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

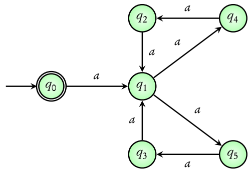


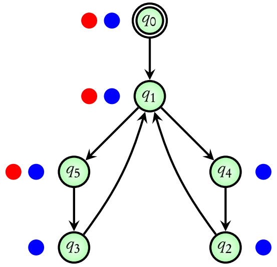


```
procedure $nested\_dfs()$
    call $dfs\_blue(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red(s)$

procedure $dfs\_red(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red(t)$
        else if $t = seed$
            report cycle
```

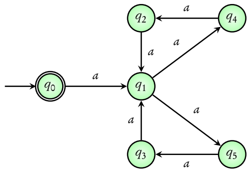


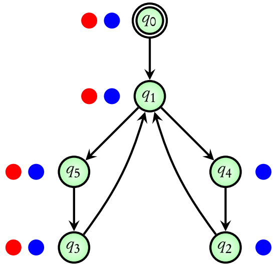


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    s.blue := true
    for all $t \in post( s )$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        seed := s
        call dfs_red ( s )

procedure dfs_red ( s )
    s.red := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

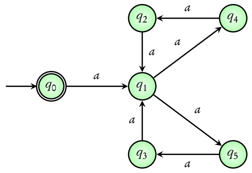


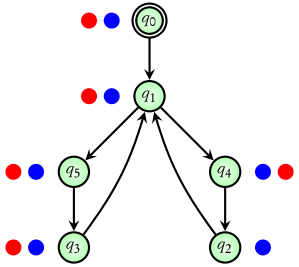


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue := $ true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red := $ true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

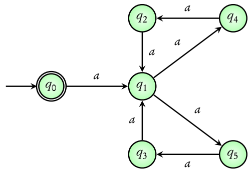


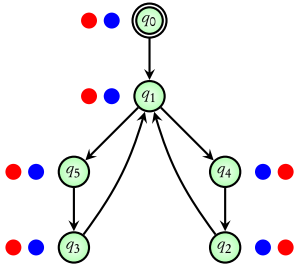


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        $seed$ := $s$
        call dfs_red ( s )

procedure dfs_red ( s )
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

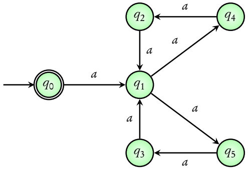


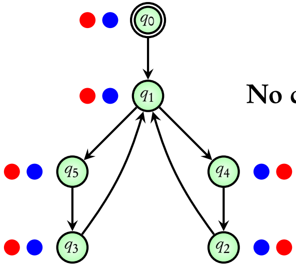


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```



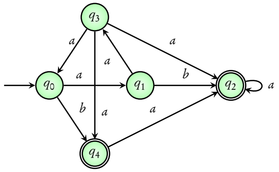


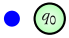
No cycle!

```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

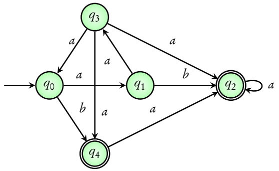


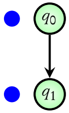


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

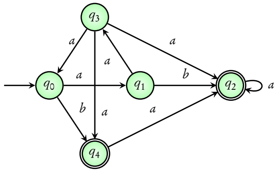


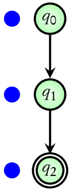


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed := s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

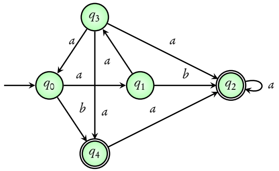


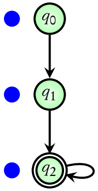


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

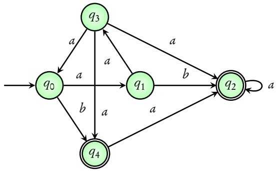


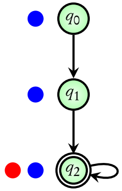


```
procedure nested_dfs()
    call dfs_blue ( $s_0$ )

procedure dfs_blue( s )
    s.blue := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call dfs_blue ( t )
    if $s \in Accept$ then
        seed := s
        call dfs_red ( s )

procedure dfs_red ( s )
    s.red := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call dfs_red ( t )
        else if $t = seed$
            report cycle
```

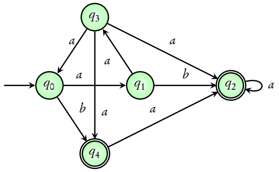


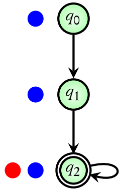


```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```





report cycle!

```
procedure $nested\_dfs()$
    call $dfs\_blue\,(s_0)$

procedure $dfs\_blue(s)$
    $s.blue$ := true
    for all $t \in post(s)$ do
        if $\neg t.blue$ then
            call $dfs\_blue\,(t)$
    if $s \in Accept$ then
        $seed$ := $s$
        call $dfs\_red\,(s)$

procedure $dfs\_red\,(s)$
    $s.red$ := true
    for all $t \in post(s)$ do
        if $\neg t.red$ then
            call $dfs\_red\,(t)$
        else if $t = seed$
            report cycle
```

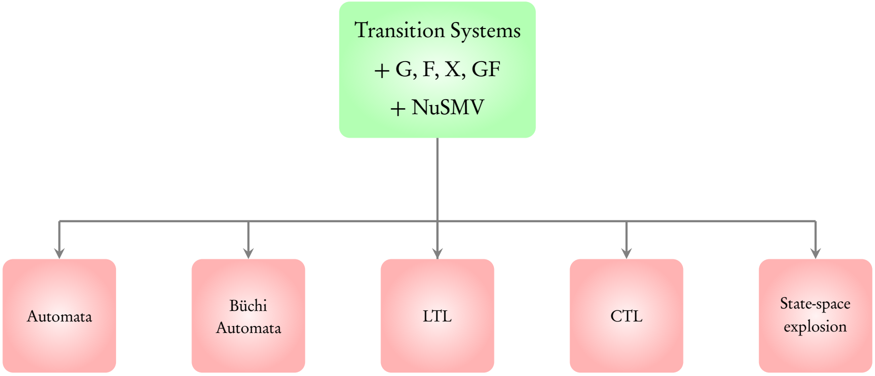
$$\begin{tikzcd}
\text{Does} & \text{Transition system} \arrow[dd] & \text{satisfy} & \omega\text{-regular property ?} \\
 & & & \omega\text{-regular expression } \phi \arrow[d] \\
 & \text{NBA } \mathscr{A}_{T.S} & & \text{NBA } \mathscr{A}_{\phi}
\end{tikzcd}$$

$$\boxed{L(\mathscr{A}_{T.S.}) \subseteq L(\mathscr{A}_{\phi})\ ?}$$

Is $L(\mathscr{A}_{T.S.}) \cap \overline{L(\mathscr{A}_{\phi})}$ empty ?

Is $L(\mathscr{A}_{T.S.}) \cap L(\overline{\mathscr{A}_{\phi}})$ empty ?

Is $L(\mathscr{A}_{T.S.} \times \overline{\mathscr{A}_{\phi}})$ empty ?



Unit: 4 **Unit: 5,6** **Unit: 7,8** **Unit: 9** **Unit: 10**