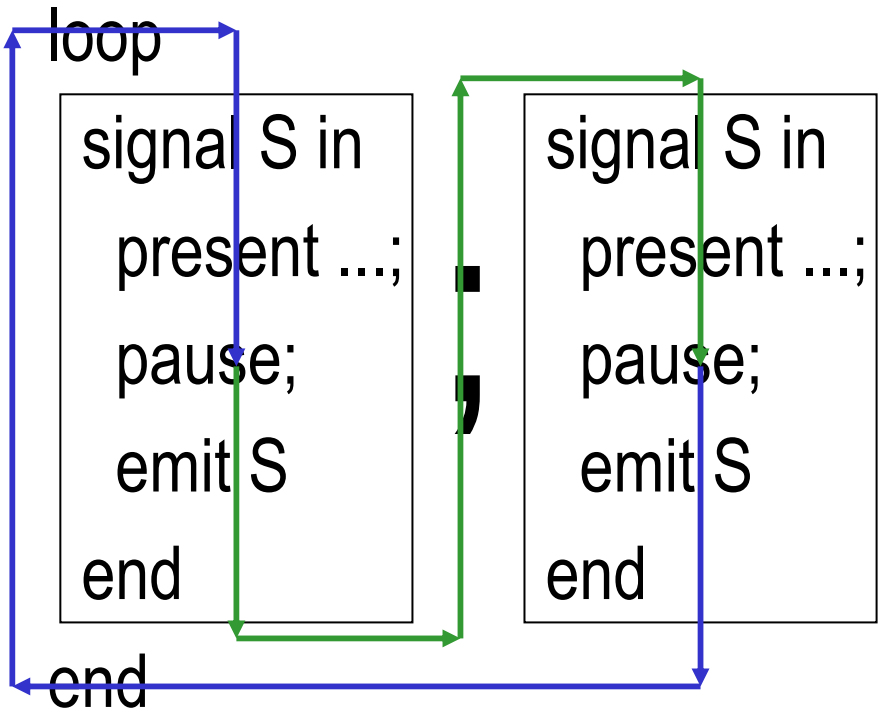
Introducing Safe Jumps in Esterel

INRIA SOPHIA ANTIPOLIS

Olivier Tardieu

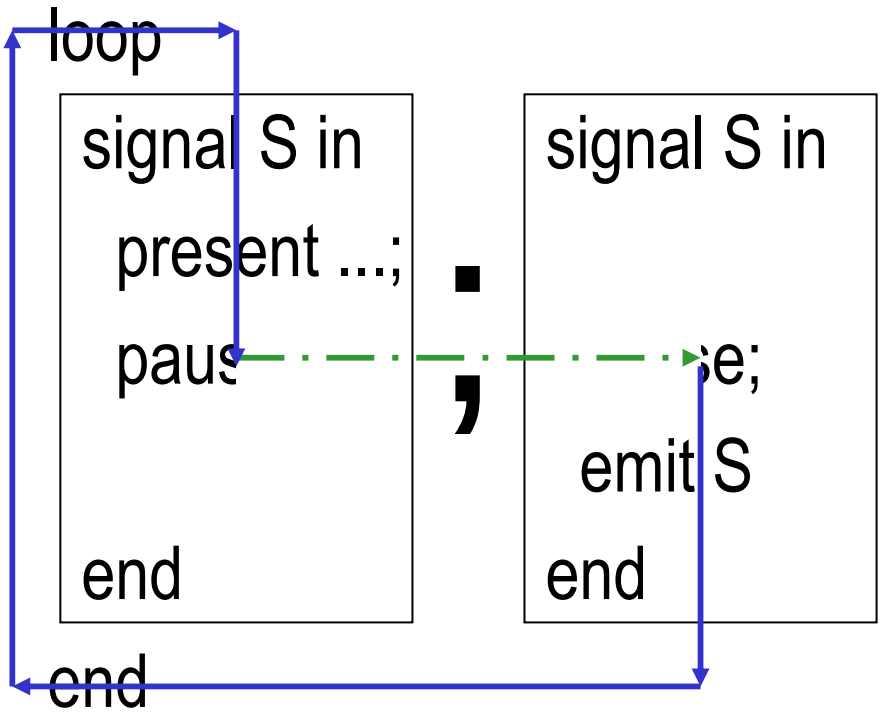
Schizophrenia and Reincarnation

```
loop
  signal S in
    present S then emit O end;
    pause;
    emit S
  end
end
```



Efficient Reincarnation?

```
loop
  signal S in
    present S then emit O end;
    pause;
    emit S
  end
end
```



Non-instantaneous Goto

```
loop
  signal S in
    present S then emit O end;
    pause;
    emit S
  end
end
```

```
loop
  signal S in
    present ...;
    goto 1;
  end
;
  signal S in
    1: pause;
    emit S
  end
end
```

Outline

- Esterel
- Goto
  - formal semantics
  - restrictions
- Applications
  - Automata
  - Schizophrenia

Syntax

- nothing
- pause — *await next instant*
- *p*; *q* — *sequence*
- *p* || *q* — *parallel composition*
- loop *p* end — *infinite loop*
- signal *S* in *p* end — *local declaration*
  - emit *S*
  - present *S* then *p* else *q* end

Logical Semantics

- Reaction:

  $p, E \xrightarrow{b} p', E'$

  - p = program
  - E = inputs + outputs
  - b = terminates?
  - p’ = residual
  - E’ = outputs

- Execution:

  $p \overset{O}{\underset{I}{\Rightarrow}} p' \overset{O'}{\underset{I'}{\Rightarrow}} \bullet$ *iff*

  - $p, I \cup O \xrightarrow{\text{false}} p', O$
  - $p', I' \cup O' \xrightarrow{\text{true}} p'', O'$

Rules – 1/2

$$\text{nothing},E \xrightarrow{\text{true}} \text{nothing},\{\}$$

$$\text{pause},E \xrightarrow{\text{false}} \text{nothing},\{\}$$

$$\frac{S \in E}{\text{emit } S,E \xrightarrow{\text{true}} \text{nothing},\{S\}}$$

$$\frac{p,E \xrightarrow{\text{false}} p',E'}{p;\ q,E \xrightarrow{\text{false}} p';\ q,E'}$$

$$\frac{p,E \xrightarrow{b} p',E' \qquad q,E \xrightarrow{b'} q',F'}{p \parallel q,E \xrightarrow{b \wedge b'} p' \parallel q',E' \cup F'}$$

$$\frac{p,E \xrightarrow{\text{false}} p',E'}{\text{loop } p \text{ end},E \xrightarrow{\text{false}} p';\ \text{loop } p \text{ end},E'}$$

$$\frac{p,E \xrightarrow{\text{true}} p',E' \qquad q,E \xrightarrow{b} q',F'}{p;\ q,E \xrightarrow{b} q',E' \cup F'}$$

Rules – 2/2

$$\frac{S \in E \qquad p,E \xrightarrow{b} p',E'}{\text{present S then p else q end},E \xrightarrow{b} p',E'}$$

$$\frac{S \notin E \qquad q,E \xrightarrow{b} q',E'}{\text{present S then p else q end},E \xrightarrow{b} p',E'}$$

$$\frac{p,E \cup \{S\} \xrightarrow{b} p',E' \qquad S \in E'}{\text{signal S in p end},E \xrightarrow{b} \text{signal S in p' end},E' \setminus \{S\}}$$

$$\frac{p,E \setminus \{S\} \xrightarrow{b} p',E' \qquad S \notin E'}{\text{signal S in p end},E \xrightarrow{b} \text{signal S in p' end},E'}$$

Examples

$$\dfrac{\dfrac{S \in \{S,O\}}{\text{emit } S,\{S,O\} \xrightarrow{\text{true}} \text{nothing},\{S\}} \qquad \dfrac{S \in \{S,O\} \qquad \dfrac{O \in \{S,O\}}{\text{emit } O,\{S,O\} \xrightarrow{\text{true}} \text{nothing},\{O\}}}{\text{present S then emit O end},\{S,O\} \xrightarrow{\text{true}} \text{nothing},\{O\}}}{\dfrac{\text{emit S; present S then emit O end},\{S,O\} \xrightarrow{\text{true}} \text{nothing},\{S,O\} \qquad S \in \{S,O\}}{\text{signal S in emit S;present S then emit O end end},\{O\} \xrightarrow{\text{true}} \text{signal S in nothing end},\{O\}}}$$

$$\dfrac{\dfrac{\dfrac{S \in \{S\}}{\text{emit S}, \{S\} \xrightarrow{\text{true}} \text{nothing}, \{S\}} \qquad \text{pause}, \{S\} \xrightarrow{\text{false}} \text{nothing}, \{\}}{\text{emit S; pause}, \{S\} \xrightarrow{\text{false}} \text{nothing}, \{S\}}}{\text{loop emit S; pause end}, \{S\} \xrightarrow{\text{false}} \text{nothing; loop emit S; pause end}, \{S\}}$$

Goto?

- Syntax
  - $\text{goto}^{\text{label}}$
  - $\text{pause}^{\text{label}}$ (pairwise distinct labels)
- Semantics
  - Collect labels *reached* by the reaction
  - Compute residual by combining:
    - initial statement
    - with labels

$p' \equiv \langle p \mid L \rangle$

Labeled Logical Semantics

$$\text{nothing},E \xrightarrow{\text{true}} \text{nothing},\{\},\{\}$$

$$\text{pause}^{n},E \xrightarrow{\text{false}} \text{nothing},\{\},\{n\}$$

$$\text{goto}^{n},E \xrightarrow{\text{false}} \text{nothing},\{\},\{n\}$$

$$\frac{S \in E}{\text{emit } S,E \xrightarrow{\text{true}} \text{nothing},\{S\},\{\}}$$

$$\frac{p,E \xrightarrow{\text{false}} p',E',L}{p;\ q,E \xrightarrow{\text{false}} p';\ q,E',L}$$

$$\frac{p,E \xrightarrow{b} p',E',L \quad q,E \xrightarrow{b'} q',F',L'}{p \parallel q,E \xrightarrow{b \wedge b'} p' \parallel q',E' \cup F',L \cup L'}$$

$$\frac{p,E \xrightarrow{\text{false}} p',E',L}{\text{loop } p \text{ end},E \xrightarrow{\text{false}} p';\ \text{loop } p \text{ end},E',L}$$

$$\frac{p,E \xrightarrow{\text{true}} p',E',L \quad q,E \xrightarrow{b} q',F',L'}{p;\ q,E \xrightarrow{b} q',E' \cup F',L \cup L'}$$

Example – Part 1

$$\underbrace{\text{loop present I then goto}^2 \text{ end; pause}^1 \text{ end}}_{\textit{loop}}\text{; pause}^2$$

- $\textit{loop}$; pause^2,{} $\xrightarrow{\text{false}}$ nothing; $\textit{loop}$; pause^2,{},{1}
- $\textit{loop}$; pause^2,{I} $\xrightarrow{\text{false}}$ nothing; $\textit{loop}$; pause^2,{},{2}
- We have defined reactions
- How shall we define executions?

State Semantics

- $p \overset{O}{\underset{I}{\Rightarrow}} p' \overset{O'}{\underset{I'}{\Rightarrow}} p'' \Rightarrow \ldots$
  - $p,\ I \cup O \xrightarrow{\text{false}} p',\ O$
  - $p',\ I' \cup O' \xrightarrow{\text{false}} p'',\ O'$

- $p \overset{O}{\underset{I}{\Rightarrow}} p' \overset{O'}{\underset{I'}{\Rightarrow}} p'' \Rightarrow \ldots$
  - $p,\ I \cup O \xrightarrow{\text{false}} p',\ O,\ L$ and $p' \equiv < p \mid L >$
  - $p',\ I' \cup O' \xrightarrow{\text{false}} p'',\ O',\ L'$ and $p'' \equiv < p \mid L' >$

Example – Part 2

$$\underbrace{\text{loop present I then goto}^2 \text{ end; pause}^1 \text{ end}}_{\textit{loop}}\text{; pause}^2$$

- $\textit{loop}$; pause^2,{} $\xrightarrow{\text{false}}$ nothing; $\textit{loop}$; pause^2,{},{1}

< loop present I then goto^2 end; pause^1 end; pause^2 | {1} >

= < loop present I then goto^2 end; pause^1 end | {1} >; pause^2

= < present I then goto^2 end; pause^1 | {1} >; $\textit{loop}$; pause^2

= < pause^1 | {1} >; $\textit{loop}$; pause^2

= nothing; loop present I then goto^2 end; pause^1 end; pause^2

Example – Part 3

$$\underbrace{\text{loop present I then goto}^2 \text{ end; pause}^1 \text{ end}}_{\textit{loop}}\text{; pause}^2$$

- $\textit{loop}$; pause^2,{} $\xrightarrow{\text{false}}$ nothing; $\textit{loop}$; pause^2,{},{1}

  $< \textit{loop}$; pause^2 | {1} > = nothing; $\textit{loop}$; pause^2

- $\textit{loop}$; pause^2,{I} $\xrightarrow{\text{false}}$ nothing; $\textit{loop}$; pause^2,{},{2}

  $<$ $\textit{loop}$; pause^2 | {2} > = < pause^2 | {2} > = nothing

State Expansion

- <p | {}> = nothing
- <pause^n | {n}> = nothing
- <signal S in p end | L> = signal S in <p | L> end
- <p || q | L> = <p | L∩L(p)> || <q | L∩L(q)>
- <loop p end | L> = <p | L>; loop p end
- <present S then p else q end | L> = <p | L> if L⊂L(p)
- <present S then p else q end | L> = <q | L> if L⊂L(q)
- <p; q | L> = <p | L>; q if L⊂L(p)
- <p; q | L> = p; <q | L> if L⊂L(q)

Well-formedness – Part 1

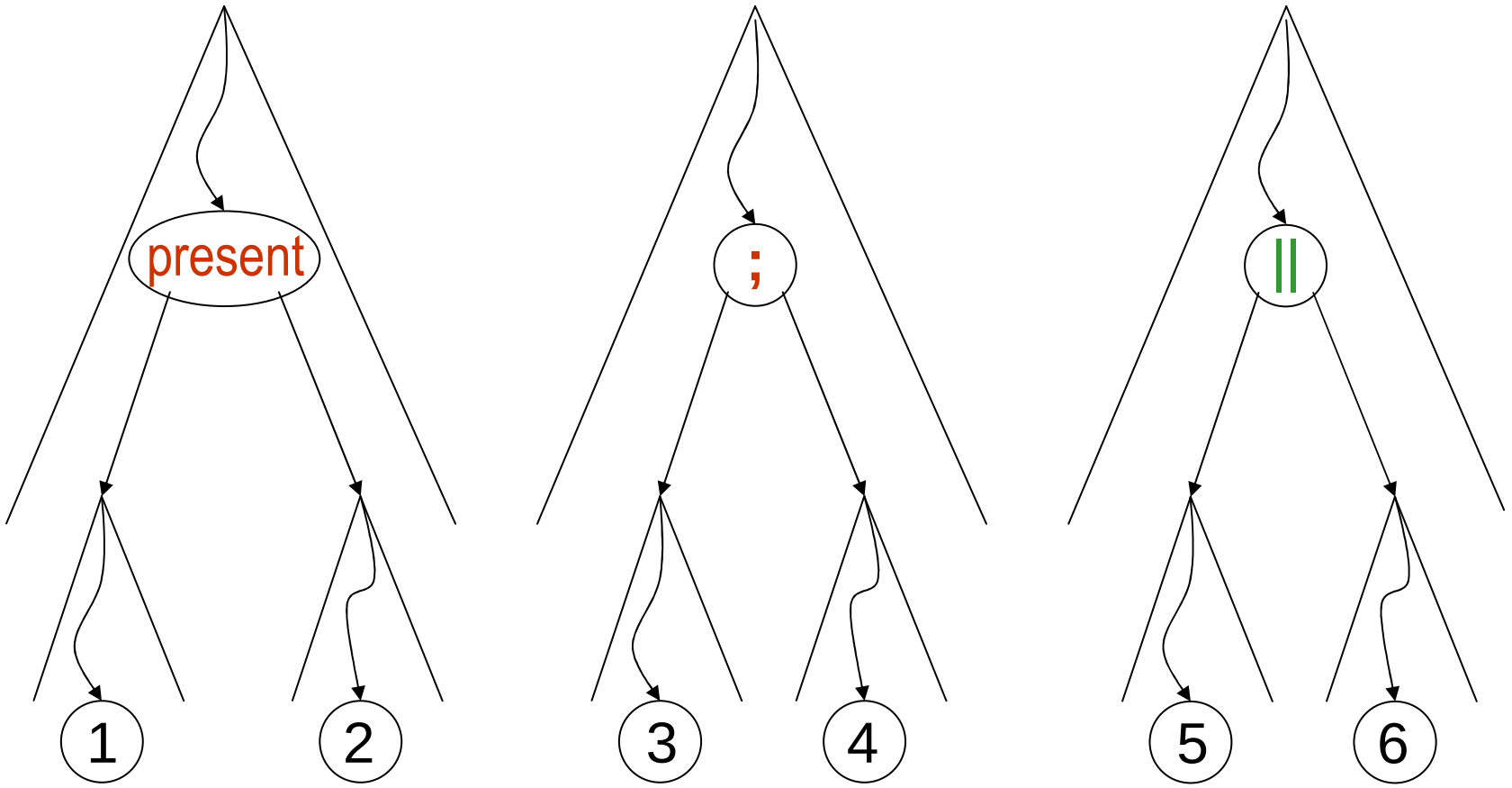
Simultaneous activation of *compatible* pauses are correct

- [goto^1 || goto^2]; [pause^1 || pause^2]
- [goto^1; emit O; pause^1 || pause^2]

Simultaneous activation *of exclusive* pauses are forbidden

- [goto^1 || goto^2]; pause^1; pause^2
- [goto^1 || goto^2]; present S then pause^1 else pause^2 end
- [goto^1 || pause^2]; pause^1

Exclusive or Compatible Statements?



EXCLUSIVE EXCLUSIVE COMPATIBLE

Well-formedness – Part 2

- In Esterel, exclusive pause statements cannot be simultaneously reached;

- In Esterel + goto, exclusive pause statements cannot be simultaneously reached provided that the program is *well-formed*:
  - $\text{goto}^1 \,//\, \text{goto}^2 \Rightarrow \text{pause}^1 \,//\, \text{pause}^2$
  - $\text{goto}^1 \,//\, \text{pause}^2 \Rightarrow \text{pause}^1 \,//\, \text{pause}^2$

Summary

- A sound semantics of *goto* in Esterel
- Constraints
  - *non-instantaneity*
    $\Rightarrow$ no (causality) cycles
  - *well-formedness*
    $\Rightarrow$ preserve Esterel semantics
- But powerful!

Automata

present tick else 1: pause end;

action 1;

present A then goto 2 end;

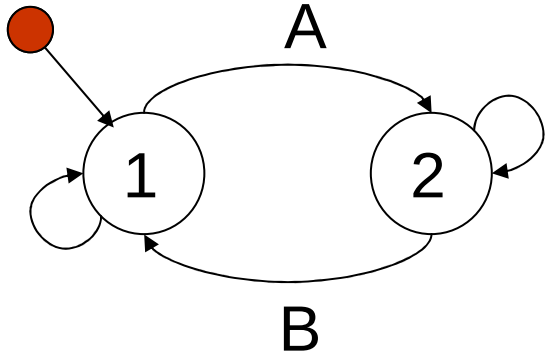
goto 1;

2: pause;

action 2;

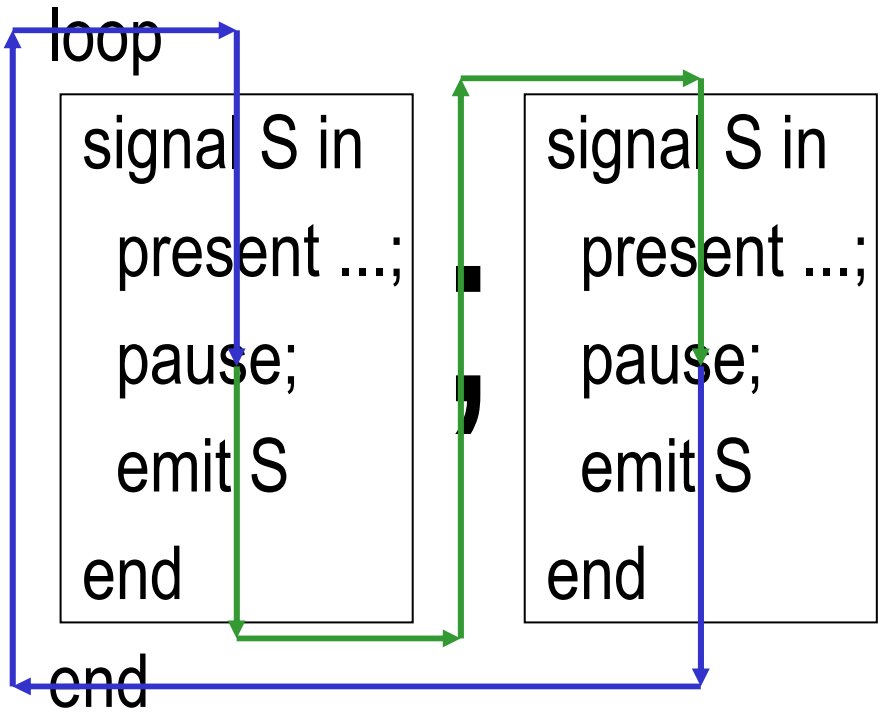
present B then goto 1 end;

goto 2

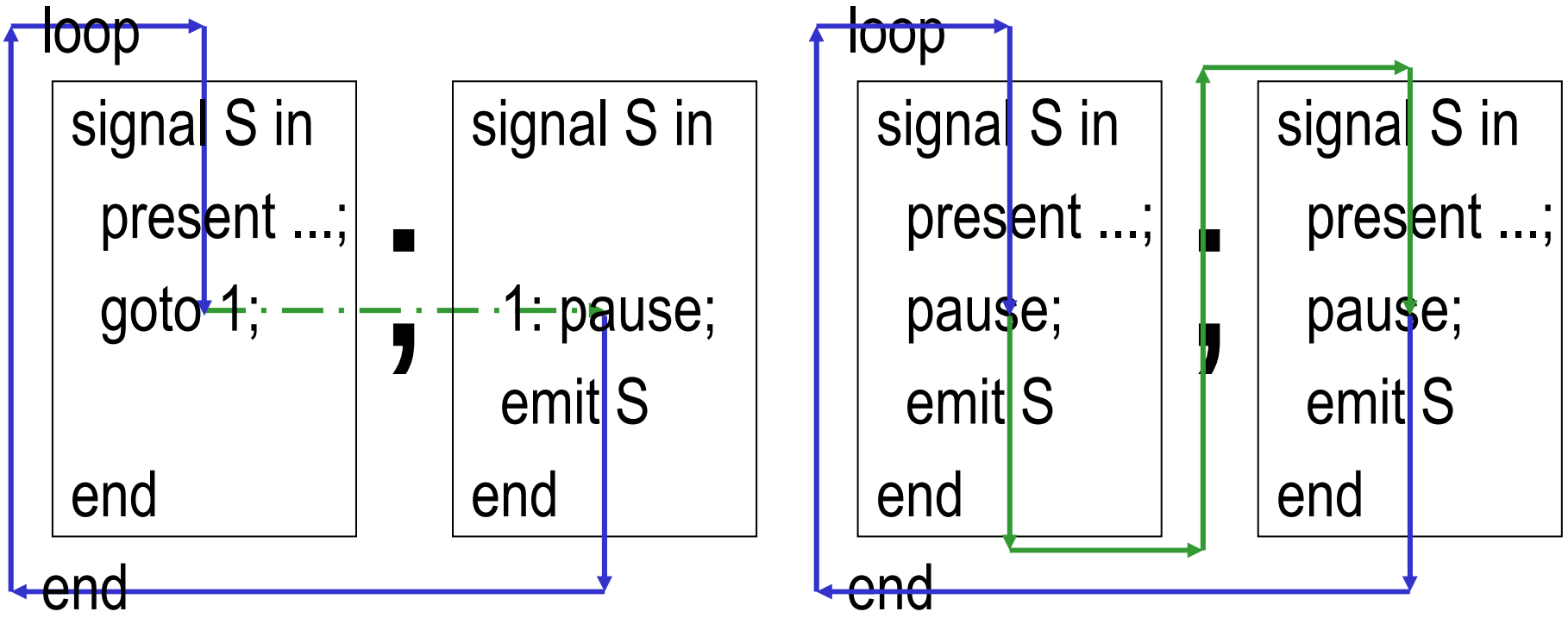


Schizophrenia and Reincarnation

```
loop
  signal S in
    present S then emit O end;
    pause;
    emit S
  end
end
```



To jump or not to jump?

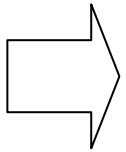


Algorithms

- Original algorithm
  - expand(loop p end) = loop expand(p); expand(p) end
- Our algorithm
  - expand(loop p end) = loop surface(p); expand(p) end
  - surface(pause^n) = goto^n
  - surface(loop p end) = surface(p)
  - surface(p;q) = surface(p) if p cannot instantly terminate
  - surface(p;q) = surface(p);surface(q) otherwise

Example

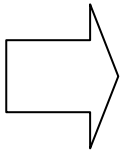
```
loop
  signal S in
    present S then emit O end;
    pause;
    emit S
  end
end
```



```
loop
  signal S in
    present ...;
    goto 1;
  end;
  signal S in
    present ...;
    1: pause;
    emit S
  end;
end
```

Multiple Reincarnation – No Goto

```
loop
  signal S in
    loop
      signal T in
        ...
        pause;
        ...
      end
    end;
    pause;
    ...
  end
end
```



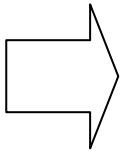
```
loop
  signal S in
    loop
      signal T in
        ...
        pause;
        ...
      end
    end;
    pause;
    ...
  end
end
```

```
loop
  signal S in
    loop
      signal T in
        ...
        pause;
        ...
      end
    ;
      signal T in
        ...
        pause;
        ...
      end
    end;
    pause;
    ...
  end
;
  signal S in
    loop
      signal T in
        ...
        pause;
        ...
      end
    ;
      signal T in
        ...
        pause;
        ...
      end
    end;
    pause;
    ...
  end
end
```

2^n !!!

Multiple Reincarnation – Goto

```
loop
  signal S in
    loop
      signal T in
        ...
        pause;
        ...
      end
    end;
    pause;
    ...
  end
end
```



```
loop
  signal S in
    loop
      signal T in
        ...
        goto 1;
        ...
      end
    end;
    goto 2;
    ...
  end
  ||
  signal S in
    loop
      signal T in
        ...
        goto 1;
        ...
      end
    end;
    2: pause;
    ...
  end
  ;
  signal T in
    ...
    1: pause;
    ...
  end
end
```

n^2 !!!

Conclusion

- Introduction of *safe jumps* in Esterel
- A preprocessor for schizophrenia
  - get rid of schizophrenia using ‘‘goto’’
  - simple, efficient, extends to full Esterel

If you do not understand schizophrenia, that is OK, because you no longer have to!