Rationals of Model checking algorithm

1. translate the initial formula with connectors $\{\bot, \top, \wedge, AF, EU, EX\}$
2. for each subformula ψ of φ, we evaluate the states satisfying ψ :
   - $\varphi = \bot$: no states are labeled with $\bot$
   - $\varphi = p$: state s is labeled with p **if** $p \in L(s)$
   - $\varphi = \psi \wedge \psi'$: s is labeled with $\psi \wedge \psi'$ **if** it's already labeled with ψ and with ψ'
   - $\varphi = \neg\psi$: s is labeled with $\neg\psi$ **if** s is not labeled with ψ
   - $\varphi = AF\psi$:
     1. If a state is labeled with ψ, it's also labeled with $AF\psi$
     2. one labels each state with $AF\psi$ s.t. all its successors are already labeled with $AF\psi$
     3. redo (2) until no labels do change.
   - $\varphi = E[\psi U \psi']$:
   - $\varphi = EX\psi$: one labels with $EX\psi$ each state which has one successor labeled with ψ
3. return the states satisfying φ

Rationals of Model checking algorithm

IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choise : AF, EU, EX
AF, EU, EX
Pseudo-code

1. translate the initial formula with connectors $\{\bot, \top, \wedge, AF, EU, EX\}$
2. for each subformula ψ of φ, we evaluate the states satisfying ψ :
   - $\varphi = \bot$: no states are labeled with $\bot$
   - $\varphi = p$: state s is labeled with p **if** $p \in L(s)$
   - $\varphi = \psi \wedge \psi'$: s is labeled with $\psi \wedge \psi'$ **if** it's already labeled with ψ and with ψ'
   - $\varphi = \neg\psi$: s is labeled with $\neg\psi$ **if** s is not labeled with ψ
   - $\varphi = AF\psi$:
   - $\varphi = E[\psi U \psi']$:
     1. If a state is labeled with ψ', it's also labeled with $E[\psi U \psi']$
     2. one labels with $E[\psi U \psi']$ each state which is labeled with ψ **and** which has at least one successor already labeled with $E[\psi U \psi']$
     3. redo (2) until no labels do change.
   - $\varphi = EX\psi$: one labels with $EX\psi$ each state which has one successor labeled with ψ
3. return the states satisfying φ

Complexity

IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choise : AF, EU, EX
AF, EU, EX
Pseudo-code

Let f be the connector numbers in the formula ; V the number of states, and E the number of transitions

1. Function *translate* is linear with the number of connectors ($O(f)$).
2. Search of states with label φ : in $O(V)$.
3. The labelling of connector $\bot$: constant time
4. the labelling of $p, \psi \wedge \psi', \neg\psi$: in $O(V)$ because one have to go through each state
5. AF is in $O(V.(V+E))$:
   - Labeling of state s with $AF\varphi$ when φ is a label of s, is in $O(V)$.
   - For each state s, one has to enumerate all its successors, and if they are all labeled with $AF\varphi$, one labels s with $AF\varphi$. $\Longrightarrow O(V+E)$.
   - One starts again as long as some states are labeled, at worst V times.

   Then, the labeling process for $AF\varphi$ is in $O(V.(V+E))$.
6. $E[\varphi U \varphi']$ is in $O(V+E)$:
7. $EF\varphi$ is in $O(V+E)$:

Complexity

IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choise : AF, EU, EX
AF, EU, EX
Pseudo-code

Let f be the connector numbers in the formula ; V the number of states, and E the number of transitions

1. Function *translate* is linear with the number of connectors ($O(f)$).
2. Search of states with label φ : in $O(V)$.
3. The labelling of connector $\bot$: constant time
4. the labelling of $p, \psi \wedge \psi', \neg\psi$: in $O(V)$ because one have to go through each state
5. AF is in $O(V.(V+E))$:
6. $E[\varphi U \varphi']$ is in $O(V+E)$:
   - initialisation : Label s with $E[\varphi U \varphi']$ if φ' is already a label : $O(V)$.
   - Reverse the transitions (in $O(E)$)
   - Depth-first search in $O(V+E)$. While the current state is labeled with φ, it's labeled with $E[\varphi U \varphi']$.

   Then, the labeling process for $E[\varphi U \varphi']$ is in $O(V+E)$.
7. $EF\varphi$ is in $O(V+E)$:

Complexity

Let f be the connector numbers in the formula ; V the number of states, and E the number of transitions

1. Function *translate* is linear with the number of connectors ($O(f)$).
2. Search of states with label φ : in $O(V)$.
3. The labelling of connector $\perp$: constant time
4. the labelling of $p, \psi \wedge \psi', \neg\psi$: in $O(V)$ because one have to go through each state
5. AF is in $O(V.(V+E))$:
6. $E[\varphi U \varphi']$ is in $O(V+E)$:
7. $EF\varphi$ is in $O(V+E)$: By a similar method, one can show that the labeling process for $EF\varphi$ is in $O(V+E)$.

CÔTE D'AZUR
IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choise : AF, EU, EX
AF, EU, EX
Pseudo-code

Pseudo-Code **SAT**

function SAT$(KS = (S, \rightarrow, L), \phi)$:
Switch(φ) :

φ = True	: **return** S
φ = False	: **return** $\emptyset$
φ is atomic	: **return** $\{s \in S \mid \varphi \in L(s)\}$
φ is $\neg\varphi_1$	: **return** S $\setminus$ **SAT** (φ_1)
φ is $\varphi_1 \wedge \varphi_2$	: **return SAT**$(\varphi_1) \cap$ **SAT**(φ_2)
φ is $\varphi_1 \vee \varphi_2$	: **return SAT**$(\varphi_1) \cup$ **SAT** (φ_2)
φ is $\varphi_1 \rightarrow \varphi_2$	: **return SAT** $(\neg\varphi_1 \vee \varphi_2)$
φ is AX φ_1	: **return SAT** $(\neg$**EX**$\neg\varphi_1)$
φ is EX φ_1	: **return SAT**$_{\mathbf{EX}}(\varphi_1)$
φ is A$(\varphi_1 U \varphi_2)$	: **return SAT**$(\neg($**E**$[\neg\varphi_2$**U**$(\neg\varphi_1 \wedge \neg\varphi_2)] \vee$ **EG**$(\neg\varphi_2)))$
φ is E$(\varphi_1 U \varphi_2)$	: **return SAT**$_{\mathbf{EU}}(\varphi_1; \varphi_2)$
φ is EF φ_1	: **return SAT**$($**E**$(\top$**U**$\varphi_1))$
φ is EG φ_1	: **return SAT**$(\neg AF \neg\varphi_1)$
φ is AF φ_1	: **return SAT**$_{\mathbf{AF}}(\varphi_1)$
φ is AG φ_1	: **return SAT** $(\neg$**EF**$\neg\varphi_1)$

CÔTE D'AZUR
IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choise : AF, EU, EX
AF, EU, EX
Pseudo-code

Pseudo-Code **SAT**$_{\mathbf{EX}}$

```
function SAT_EX(φ)
""" determines the set of states satisfying EX(φ) """
local var X, Y

begin
   X = SAT (φ);
   Y = {s0 ∈ S | s0 → s1 for some s1 ∈ X};
   return Y
end
```


CÔTE D'AZUR
IA symbolique & réseaux Biol.
Jean-Paul Comet
Introduction to model checking
Reminder of CTL
Equivalences
Choise : AF, EU, EX
AF, EU, EX
Pseudo-code

Pseudo-Code **SAT**$_{\mathbf{AF}}$

```
function SAT_AF(φ)
""" determines the set of states satisfying AF(φ)"""
local var X, Y

begin
   X = S;
   Y = SAT(φ);
   repeat until X = Y
   begin
      X = Y;
      Y = Y ∪{s | for all s' with s → s' we have s' ∈ Y}
   end
   return Y
end
```

Pseudo-Code $\mathbf{SAT_{EU}}$

function $\mathbf{SAT_{EU}}(\varphi, \psi)$
""" determines the set of states satisfying E[φ U ψ] """
local var W, X, Y

begin
  W = SAT (φ)
  X = S
  Y = SAT (ψ)
  repeat until X = Y
  begin
    X = Y
    Y = Y $\cup(W \cap \{s \mid \text{exists} s' \text{ such that } s \rightarrow s' \text{ and } s' \in Y\})$
  end
  return Y
end