Abstract Interpretation of CTL Properties

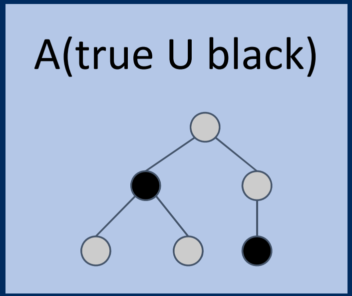
Caterina Urban, Samuel Ueltschi, Peter Müller

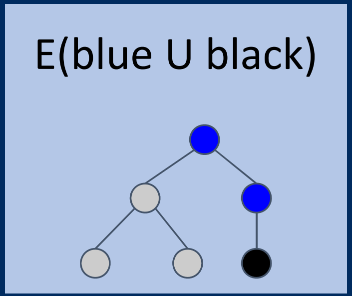
ETH zürich

Computation Tree Logic

- Branching-time logic

ϕ ::= p | $\neg\phi$ | $\phi\wedge\phi$ | $\phi\vee\phi$
| AX ϕ | EX ϕ
| A(ϕ U ϕ) | E(ϕ U ϕ)
| AG ϕ | EG ϕ





- Goal: Automatically check CTL properties of programs
  - Infer sufficient preconditions
  - Handle existential properties

Example

```
while( rand() ) {
  x := 1
  y := y + 1
  x := 0
}
while( true ) { }
```

- CTL specification

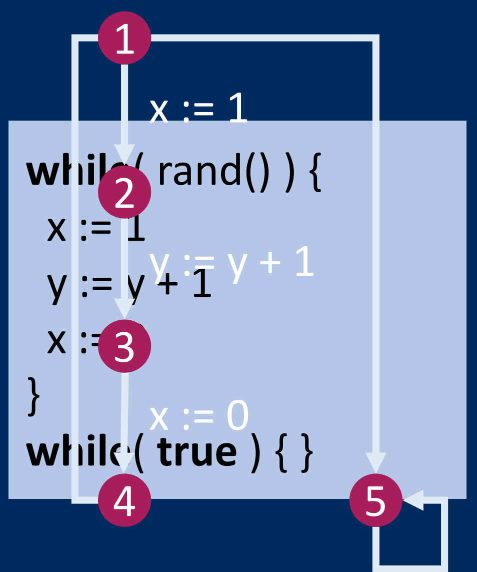
  A(true U x = 0)

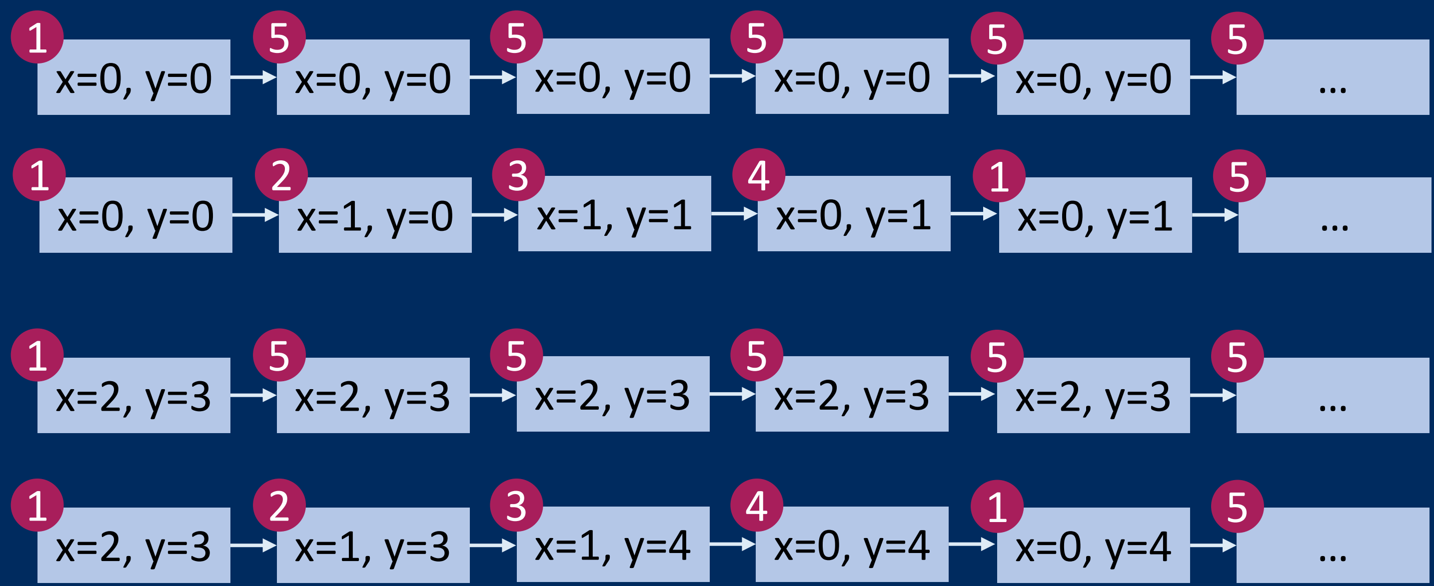
- Inferred precondition

  x = 0

Maximal Trace Semantics

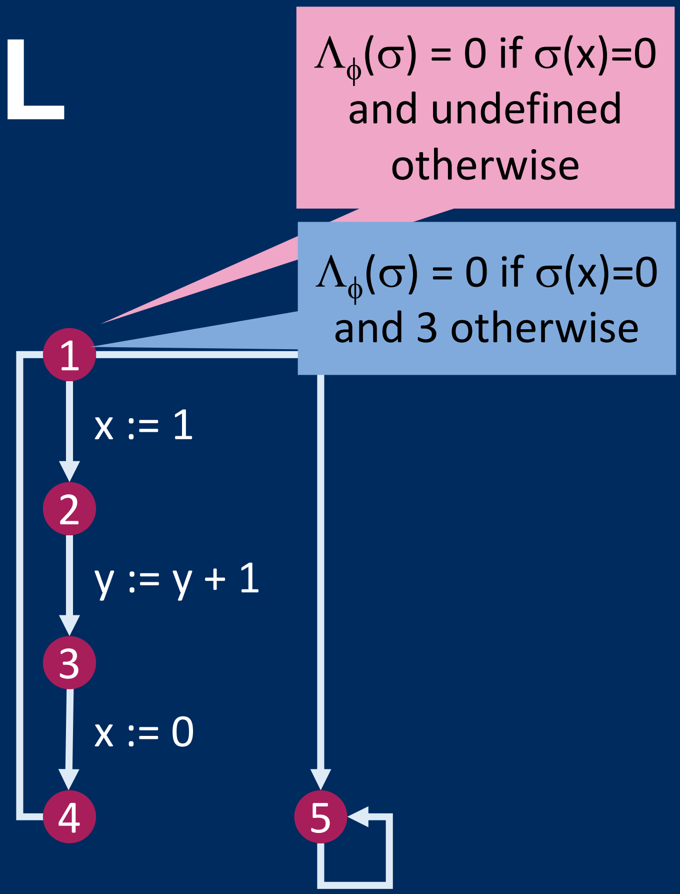
- Contains all finite and infinite traces of a program





Program Semantics for CTL

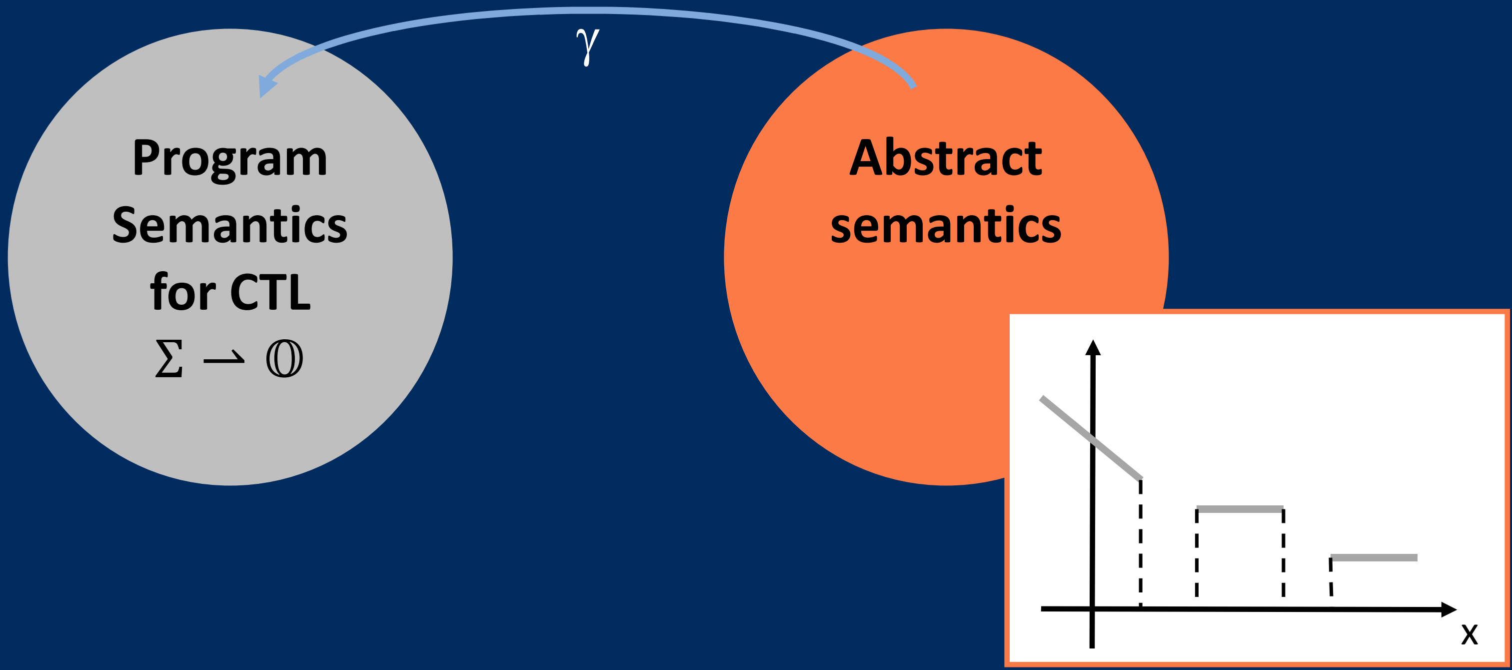
- For a given CTL formula ϕ and a set of program traces, define a partial function Λ_ϕ from states to ordinals
- A program satisfies a CTL formula ϕ for all traces starting from an initial state σ if and only if $\sigma \in \mathrm{dom}(\Lambda_\phi)$
- If defined for an until-formula ϕ_1 U ϕ_2, $\Lambda_\phi(\sigma)$ yields the number of steps until ϕ_2 holds (ranking function)



A(true U x = 0)

E(true U x = 0)

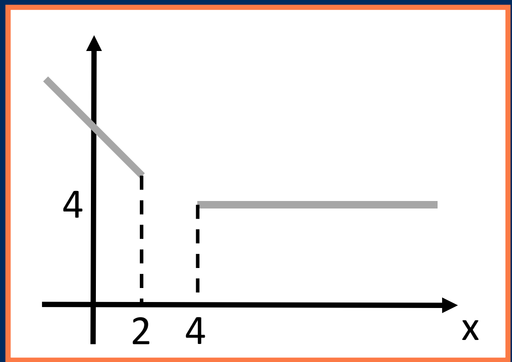
Piecewise-defined Ranking Functions



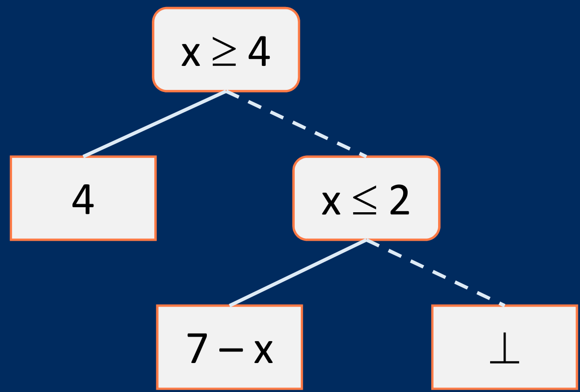
Earlier work by Caterina Urban and Antoine Miné [SAS’13, SAS’14, ESOP’14]

Abstract Domain: Decision Trees

- Piecewise-defined functions are represented as decision trees



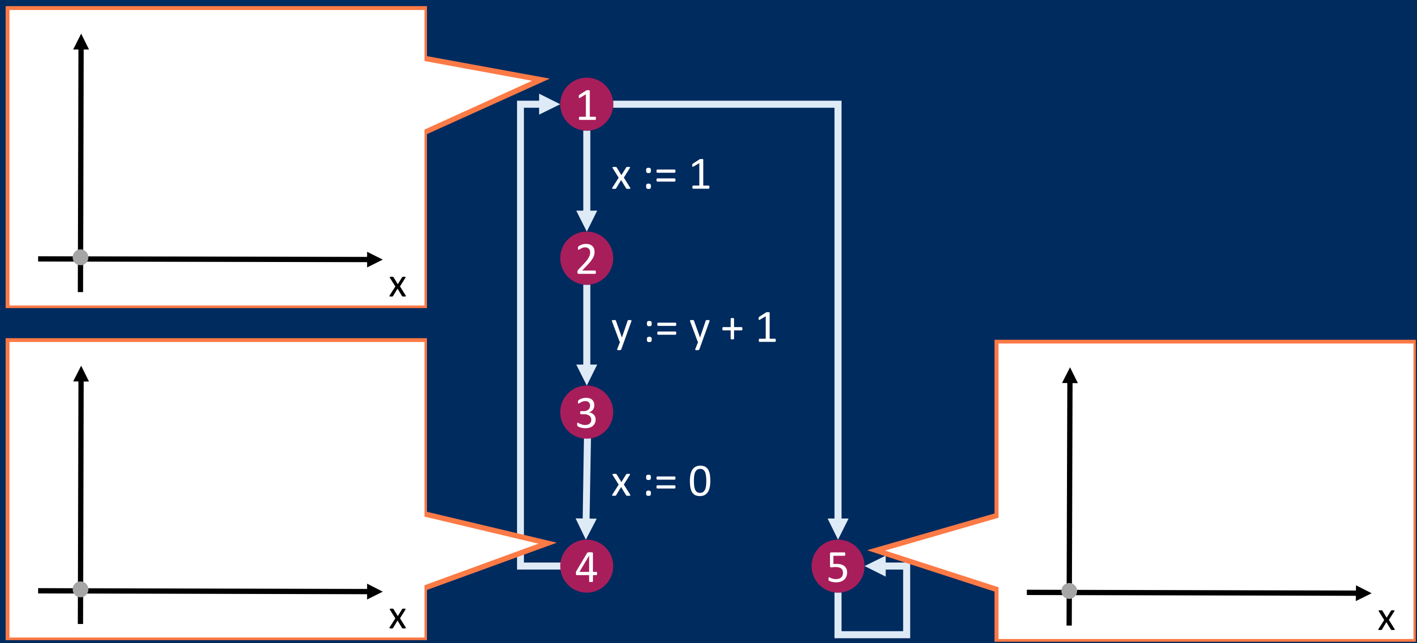
$$f(x) = \begin{cases} 4 & \text{if } x \geq 4 \\ 7 - x & \text{if } x \leq 2 \\ \bot & \text{otherwise} \end{cases}$$



Static Analysis

- Map each point to a function over-approximating concrete semantics
- Analysis is performed backward for each constituent formula

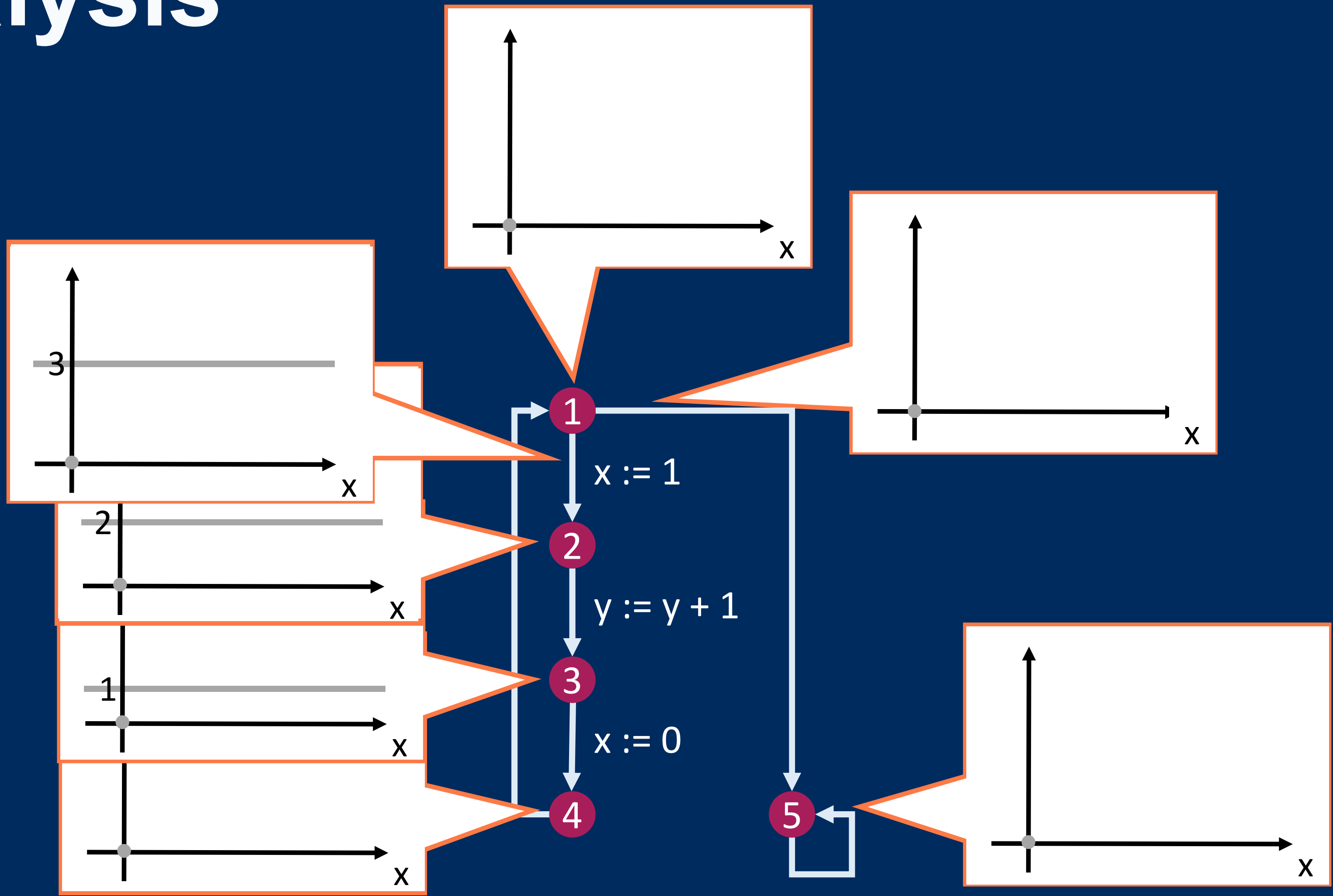
A(true U x = 0)



Static Analysis

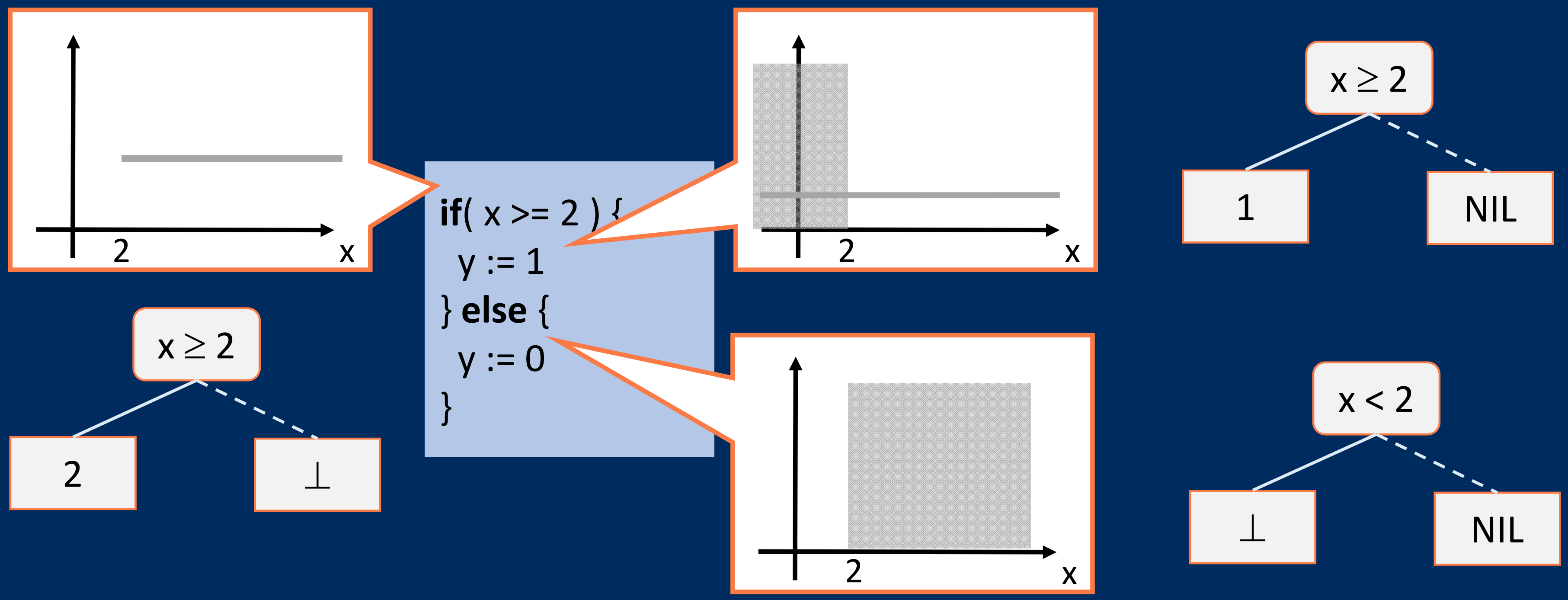
A(true U x = 0)

For universal formulas, merge preserves undefinedness



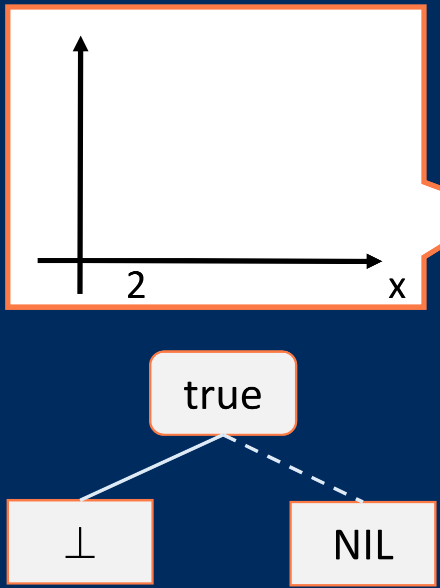
Conditional Statements

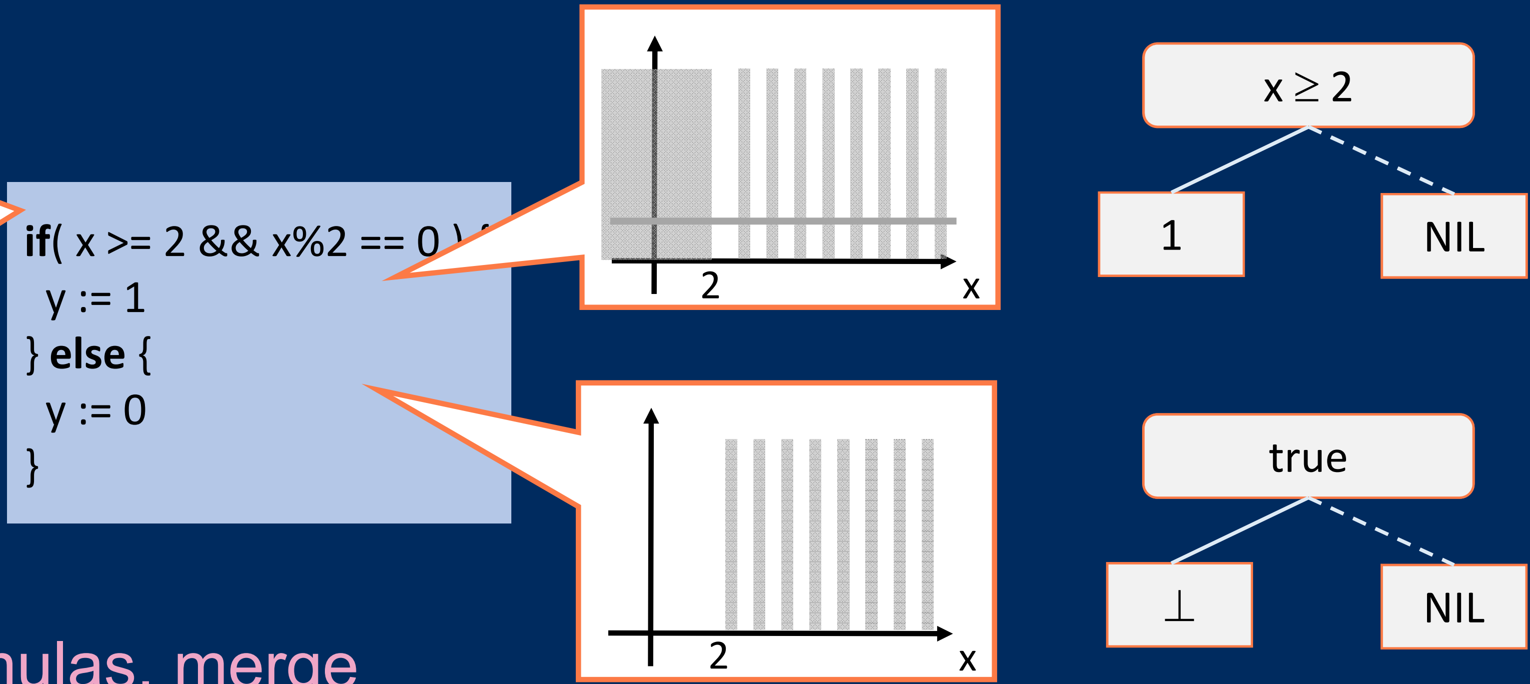
A(true U y = 1)



Conditional Statements

A(true U y = 1)

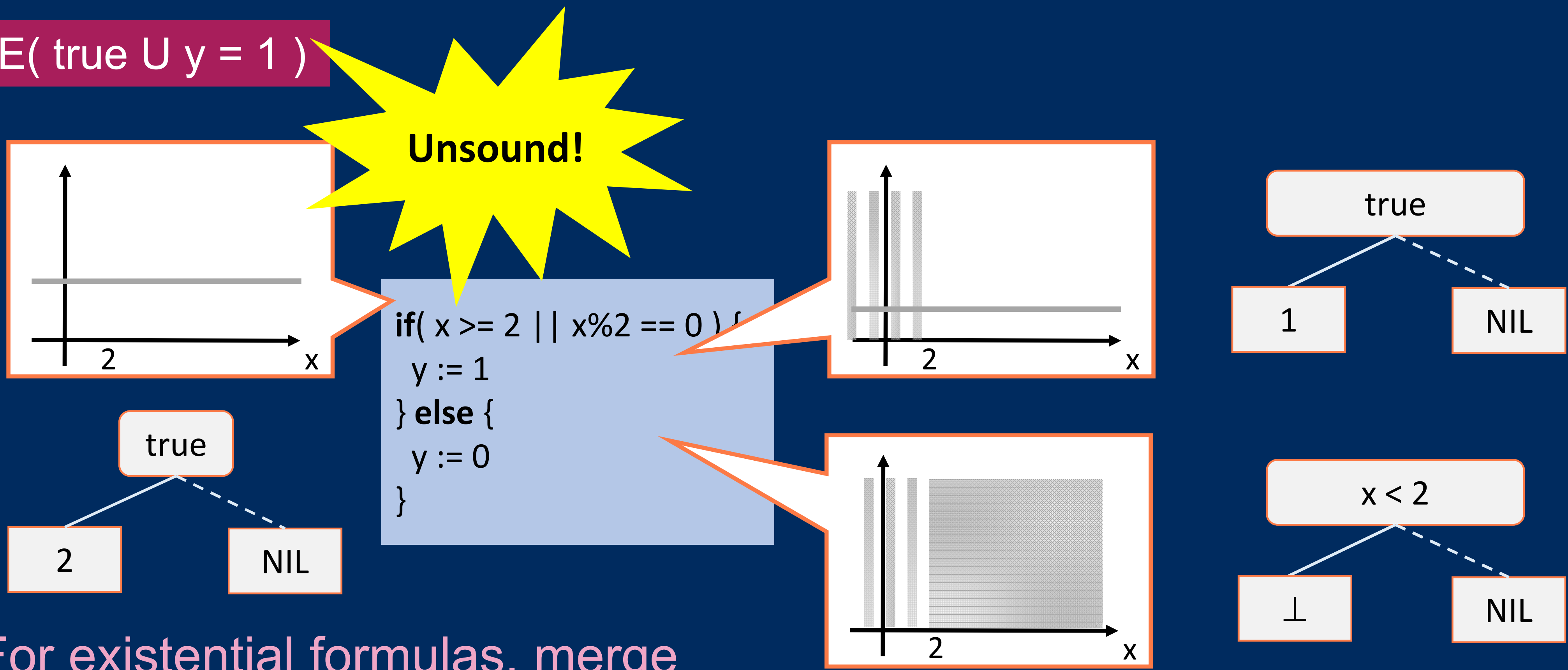




For universal formulas, merge preserves undefinedness

Conditional Statements

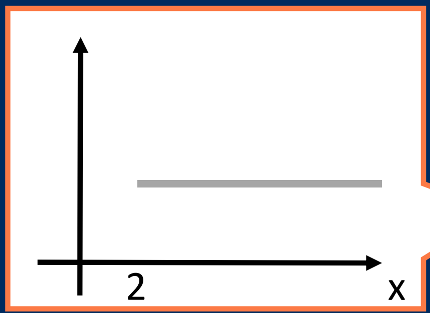
E(true U y = 1)

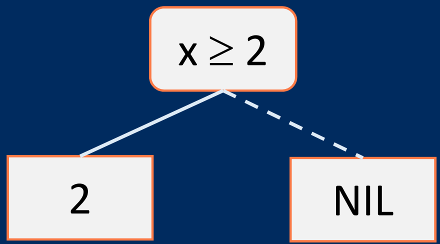


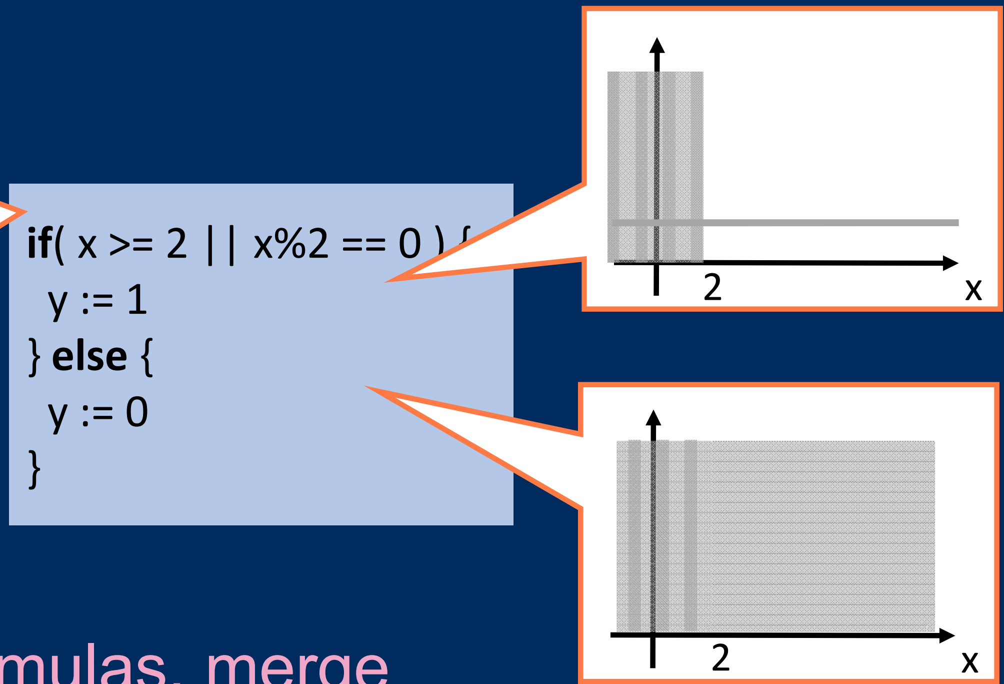
For existential formulas, merge preserves definedness

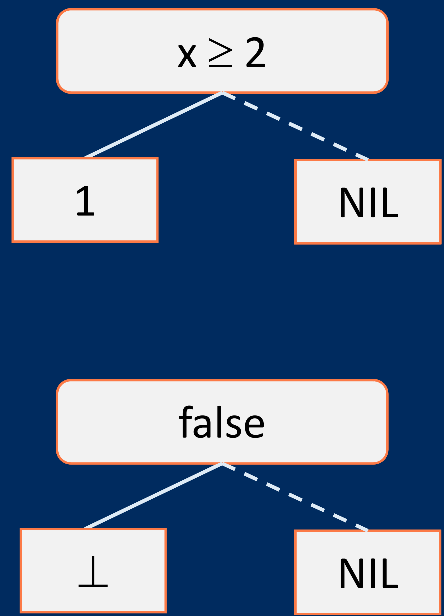
Conditional Statements

E(true U y = 1)









For existential formulas, merge preserves definedness

Soundness

A program satisfies a CTL formula ϕ for all traces starting from an initial state σ if $\sigma \in \text{dom}(\ \gamma(\Lambda^{\#}{}_{\phi})\ )$

Evaluation

- Implementation in FuncTion static analyzer
  - C-like input language
  - Available at https://github.com/caterinaurban/function
- Evaluated on test cases and benchmarks from the literature and SV-COMP competition
- Abstract domains
  - Polyhedra for constraints
  - Affine functions and ordinals for leaves of decision trees

Experimental Results

CTL Property	Result	Time	T2	Ultimate LTL Automizer
$\mathsf{AGAF}(n = 1) \wedge \mathsf{AF}(n = 0)$	✓	0.05s	✗	✓
$\mathsf{EGAF}(n = 1)$	✓	0.05s	✗	-
$\mathsf{AGEF}(x \leq -10)$	✓	0.15s	✗	-
$\mathsf{AFEG}(x < -100) \vee \mathsf{AF}(x = 20)$	✓	0.05s	✗ (error)	-
$\mathsf{EF}(\text{exit} : \mathit{true})$	✗	-	✓	-
$\mathsf{A}(x \geq y\ \mathsf{U}\ x = y)$	✓	0.03s	✗	✓
$\mathsf{EGEF}(n = 1)$	✓	0.04s	✗	-
$\mathsf{E}(x \geq y\ \mathsf{U}\ x = y)$	✓	0.03s	✗ (no implementation)	-
$\mathsf{AFAG}(\text{WItemsNum} \geq 1)$	✓	0.15s	✗	✓
$(c \leq 5 \wedge c > 0) \vee \mathsf{AF}(\text{resp} > 5)$	✗	-	✗	✓
$\mathsf{A}(i = 0\ \mathsf{U}\ (\mathsf{A}(i = 1\ \mathsf{U}\ \mathsf{AG}(i = 3)) \vee \mathsf{AG}(i = 1)))$	✗	-	-	✓
$\neg\mathsf{AG}(\text{timer} = 0 \Rightarrow \mathsf{AF}(\text{output} = 1))$	✗	-	-	✓
$\mathsf{AG}(\mathsf{AF}(t = 1) \wedge \mathsf{AF}(t = 0))$	✗	-	-	✓
$\mathsf{EF}(x < 0)$	✗	-	✓	-
$i < 5 \Rightarrow \mathsf{AF}(\text{exit} : \mathit{true})$	✓	0.04s	✗ (out of memory)	✓
$\mathsf{AFEG}(i = 0)$	✓	0.1s	✗	-
$\mathsf{EF}(\mathsf{AF}(j \geq 21) \wedge i = 100)$	✓	0.3s	✗ (error)	-
$\mathsf{AF}(x = 7 \wedge \mathsf{EFAG}(x = 2))$	✓	0.02s	✗	-

Summary

- Theory for analyzing CTL properties with abstract interpretation
- Automatic inference of sufficient preconditions
- Implementation in FuncTion static analyzer: https://github.com/caterinaurban/function
- Future work: extension to LTL