SAS 2019
3rd World Congress on Formal Methods, Porto, Portugal
Friday, October 11th, 2019

Responsibility Analysis by Abstract Interpretation

Chaoqiang Deng Patrick Cousot

Computer Science, Courant Institute of Mathematics, New York University
deng@cs.nyu.edu pcousot@cs.nyu.edu

What is Responsibility?

What is responsibility?

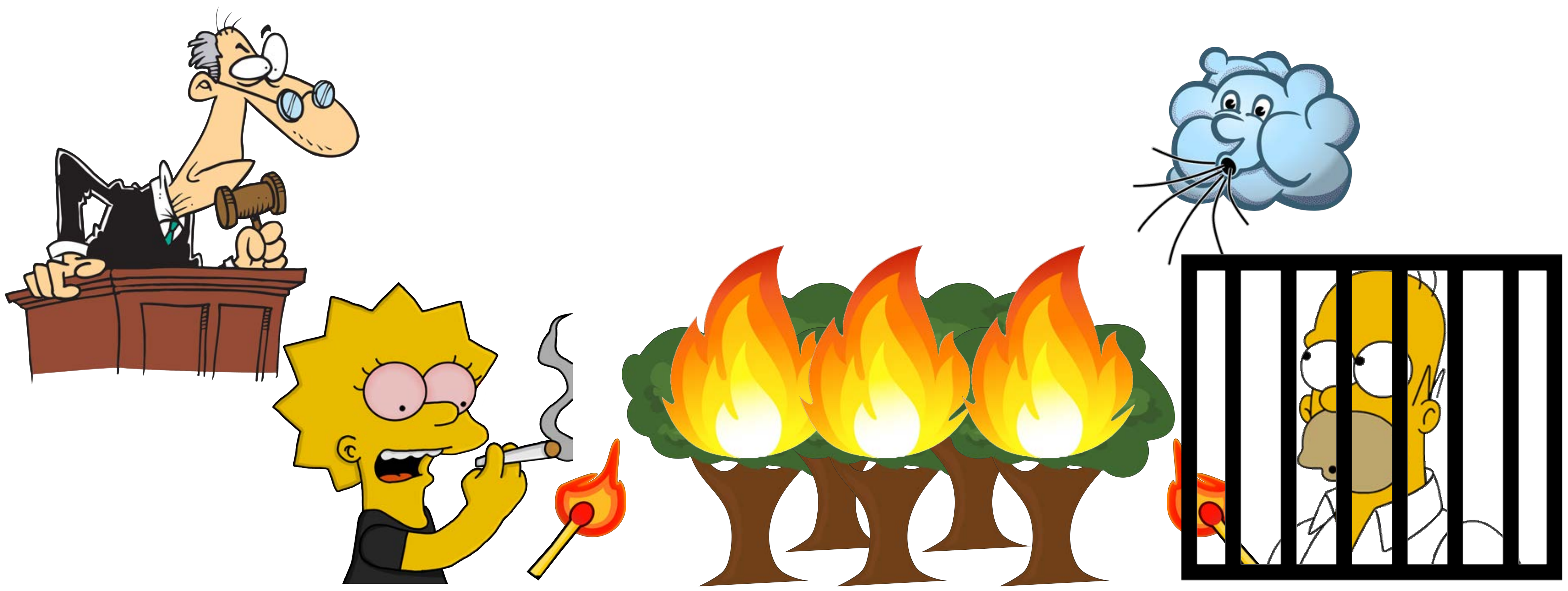
$$\langle\wp(\mathrm{E}^{*\infty}),\ \subseteq\rangle \xleftrightarrows[\alpha_R(\mathcal{S}^{\mathsf{Max}},\mathcal{L}^{\mathsf{Max}},\mathbb{C},\mathcal{B})]{\gamma_R(\mathcal{S}^{\mathsf{Max}},\mathcal{L}^{\mathsf{Max}},\mathbb{C},\mathcal{B})} \langle\wp(\mathrm{E}^{*\infty}\times\mathrm{E}\times\mathrm{E}^{*\infty}),\ \subseteq\rangle$$

where

$$\begin{aligned}
&\alpha_R \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\wp(\mathrm{E}^{*\infty})) \mapsto (\mathrm{E}^{*\infty} \mapsto \wp(\mathrm{E}^{*\infty}))\\
&\qquad\quad \mapsto \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty}\times\mathrm{E}\times\mathrm{E}^{*\infty})\\
&\alpha_R(\mathcal{S}^{\mathsf{Max}},\mathcal{L}^{\mathsf{Max}},\mathbb{C},\mathcal{B},\mathcal{T}) \triangleq\\
&\quad \mathit{let}\ \alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P}) = \{\sigma\in\mathsf{Pref}(\mathcal{P}) \mid \forall\sigma'\in\mathcal{S}.\ \sigma\preceq\sigma' \Rightarrow \sigma'\in\mathcal{P}\}\ \mathit{in}\\
&\qquad \mathit{let}\ \mathbb{I}(\mathcal{S},\mathcal{L},\sigma) = \Cap\{\mathcal{P}\in\mathcal{L} \mid \sigma\in\alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P})\}\ \mathit{in}\\
&\qquad\quad \mathit{let}\ \mathbb{O}(\mathcal{S},\mathcal{L},\mathbb{C},\sigma) = \Cup\{\mathbb{I}(\mathcal{S},\mathcal{L},\sigma') \mid \sigma'\in\mathbb{C}(\sigma)\}\ \mathit{in}\\
&\qquad\qquad \{\langle\sigma_{\mathrm{H}},\ \sigma_{\mathrm{R}},\ \sigma_{\mathrm{F}}\rangle \mid \sigma_{\mathrm{H}}\sigma_{\mathrm{R}}\sigma_{\mathrm{F}}\in\mathcal{T} \wedge |\sigma_{\mathrm{R}}| = 1 \wedge\\
&\qquad\qquad\quad \emptyset \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}},\mathcal{L}^{\mathsf{Max}},\mathbb{C},\sigma_{\mathrm{H}}\sigma_{\mathrm{R}}) \subseteq \mathcal{B} \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}},\mathcal{L}^{\mathsf{Max}},\mathbb{C},\sigma_{\mathrm{H}})\}
\end{aligned}$$

Who is responsible?

- Forest Fire Example [Halpern & Pearl, 2001]
  - Disjunctive Scenario: one lit match is sufficient to burnt down the forest.
    - Homer throws a lit match to the forest, which guarantees to burn down the forest.
    - Later, Lisa drops another lit match in the forest.
    - The wind can influence the speed of fire.
    - The forest is burnt down, and who is responsible?



Who is responsible?

- Forest Fire Example [Halpern & Pearl, 2001]

```
1:  homer = input_1( ); // T or F (light a match or not)
2:  m1 = homer; // 1st match
3:  lisa = input_2( ); // T or F (light a match or not)
4:  m2 = lisa; // 2nd match
5:  wind = input_3( ); // 1 or 2 (weak or strong wind)
6:  forest = (m1 || m2) ? wind : 0;
7:  // Undesired behavior if forest > 0
```

- Dependency
  - The value of forest at 7 depends on wind at 6, m2 at 5, lisa at 4, m1 at 3, homer at 2.
  - By dependency analysis, the forest's status depends on not only Homer and Lisa's decisions, but also the wind and two matches.

- Actual Cause
  - Use a structural equations model (SEM) to represent the system (built by hand).
  - An event C is an actual cause of another event E, iff E counterfactually depends on C under certain contingencies.
  - In the context where input_1=T, input_2=T and input_3=1 (i.e. Homer and Lisa lit matches, and the wind is weak), the actual causes of "forest > 0" are "homer = T", "m1=T", "lisa = T" and "m2=T".
  - By actual causality analysis, both Homer and Lisa are causes of fire, as well as two matches, but not the wind.

Who is responsible?

- Forest Fire Example [Halpern & Pearl, 2001]

```
homer = input_1( ); // T or F (light a match or not)
m1 = homer; // 1st match
lisa = input_2( ); // T or F (light a match or not)
m2 = lisa; // 2nd match
wind = input_3( ); // 1 or 2 (weak or strong wind)
forest = (m1 || m2) ? wind : 0;
// Undesired behavior if forest > 0
```

- Responsibility
    - Lisa shall not be responsible.
        - Keep the temporal ordering of events, and only the first event that guarantees the behavior of interest is responsible.
    - Matches shall not be responsible.
        - Only entities that are free to make choices can possibly be responsible.
    - Specify "to whose cognizance / knowledge".
        - All the reasoning above is implicitly based on the cognizance of an omniscient observer (i.e. everything occurred is known). How about the cognizance of a non-omniscient observer?
            - If the observer observes only Lisa's action and does not know that Homer already started a fire, the observer would determine Lisa to be responsible for the fire.
        - E.g. in program security, the cognizance can represent attackers' capabilities.

Informal Definition of Responsibility

Def. To the cognizance of an observer, the entity E_R is responsible for a behavior B of interest in a certain execution, iff according to the observer's cognizance, E_R is free to choose its value, and such a choice is the first one that guarantees the occurrence of B in that execution.

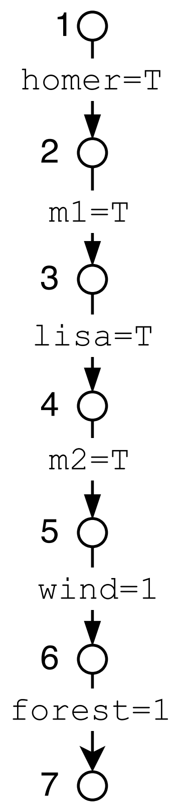
- Remark
  - In every single execution where B occurs, there is one and only one entity responsible for B.
  - For the whole system, there may exist more than one entities that are responsible for B.
  - Responsibility is NOT a trace property.
    - To decide which entity in an execution is responsible, the execution alone is not sufficient, and it is necessary to reason on the whole semantics to exhibit the entity's "free choice" and its guarantee of B.

Framework of Responsibility Analysis

Framework of Responsibility Analysis

- Three Components
  1. System Semantics
     - The set of all possible executions, each of which can be analyzed individually.

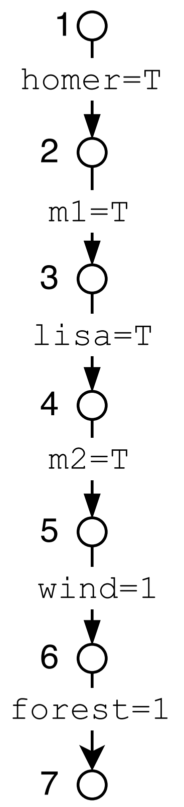
System Semantics



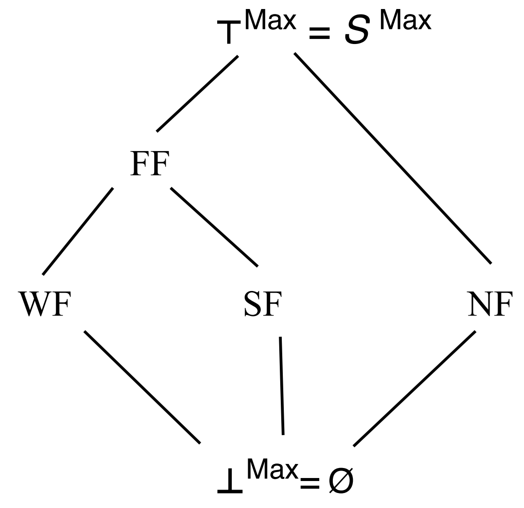
Framework of Responsibility Analysis

- Three Components
  1. System semantics
  2. Lattice of system behaviors of interest
     - The stronger a behavior is, the lower its position in the lattice is.

System Semantics



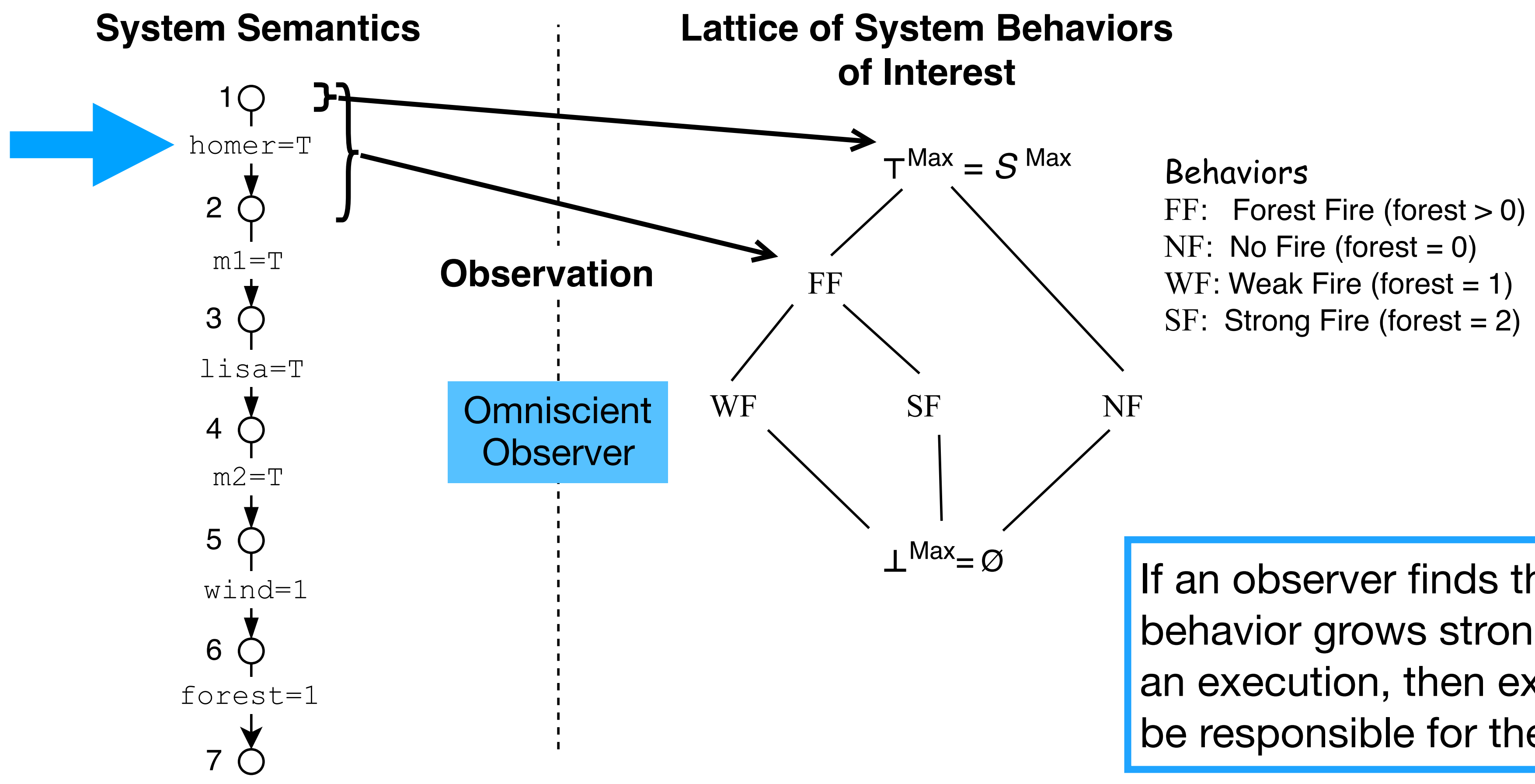
Lattice of System Behaviors of Interest



Behaviors
FF: Forest Fire (forest > 0)
NF: No Fire (forest = 0)
WF: Weak Fire (forest = 1)
SF: Strong Fire (forest = 2)

Framework of Responsibility Analysis

- Three Components
  1. System semantics
  2. Lattice of system behaviors of interest
  3. An observation function for each observer
     - Maps every (probably unfinished) execution to a behavior that is guaranteed to occur.

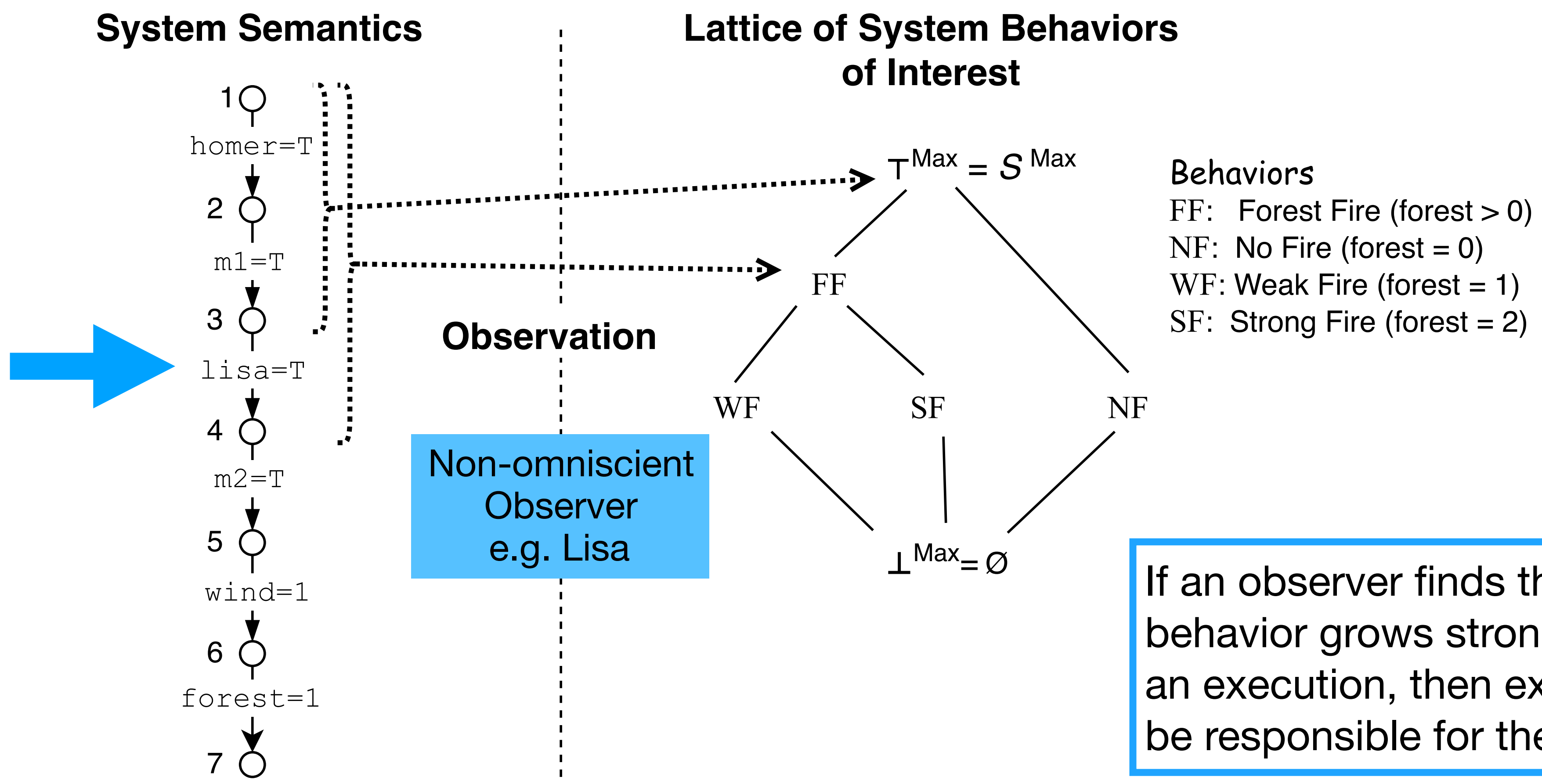


Behaviors
FF: Forest Fire (forest > 0)
NF: No Fire (forest = 0)
WF: Weak Fire (forest = 1)
SF: Strong Fire (forest = 2)

If an observer finds that the guaranteed behavior grows stronger after extending an execution, then extension part must be responsible for the stronger behavior.

Framework of Responsibility Analysis

- Three Components
  1. System semantics
  2. Lattice of system behaviors of interest
  3. An observation function for each observer
     - Maps every (probably unfinished) execution to a behavior that is guaranteed to occur.



If an observer finds that the guaranteed behavior grows stronger after extending an execution, then extension part must be responsible for the stronger behavior.

Formal Definition of Responsibility

1) System Semantics

- **Event**: any action in the system
  - E.g. assignment, Boolean test, skip
- **Trace**: a sequence of events

$e \in \mathrm{E}$	event
$\sigma \in \mathrm{E}^{+\infty} \triangleq \bigcup_{n \geqslant 1} \{[0, n-1] \mapsto \mathrm{E}\} \cup \{\mathbb{N} \mapsto \mathrm{E}\}$	nonempty trace
$\sigma \in \mathrm{E}^{*\infty} \triangleq \{\varepsilon\} \cup \mathrm{E}^{+\infty}$	empty or nonempty trace
$\sigma \preceq \sigma' \triangleq \lvert\sigma\rvert \leqslant \lvert\sigma'\rvert \wedge \forall 0 \leqslant i \leqslant \lvert\sigma\rvert - 1 : \sigma_i = \sigma'_i$	prefix ordering of traces

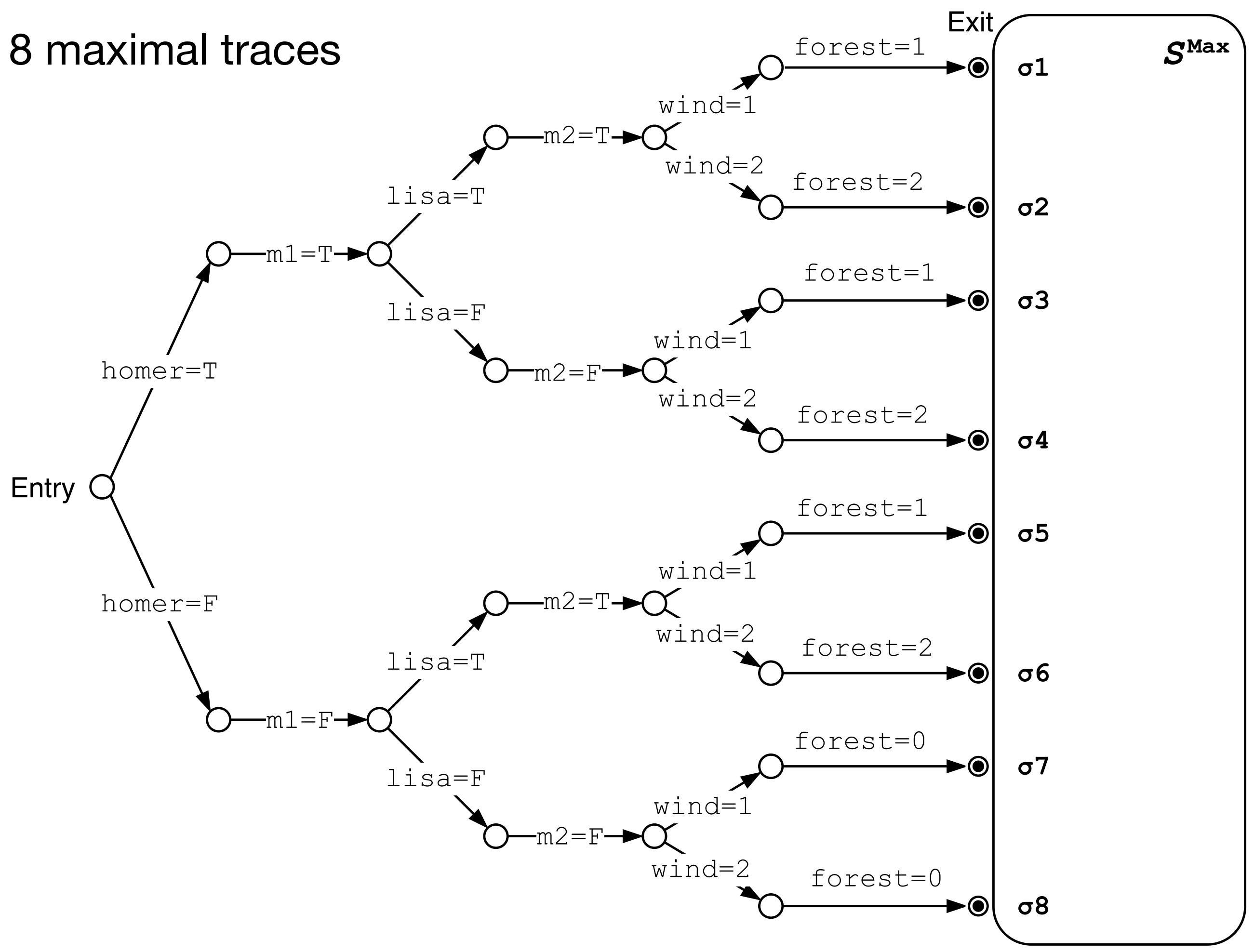
  - E.g. $\sigma 1 =$ `homer=T` $\triangleright$ `m1=T` $\triangleright$ `lisa=T` $\triangleright$ `m2=T` $\triangleright$ `wind=1` $\triangleright$ `forest=1`

- **Trace Prefix**

$\mathsf{Pref} \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty})$	prefixes of traces
$\mathsf{Pref}(P) \triangleq \{\sigma' \in \mathrm{E}^{*\infty} \mid \exists \sigma \in P.\ \sigma' \preceq \sigma\}$	

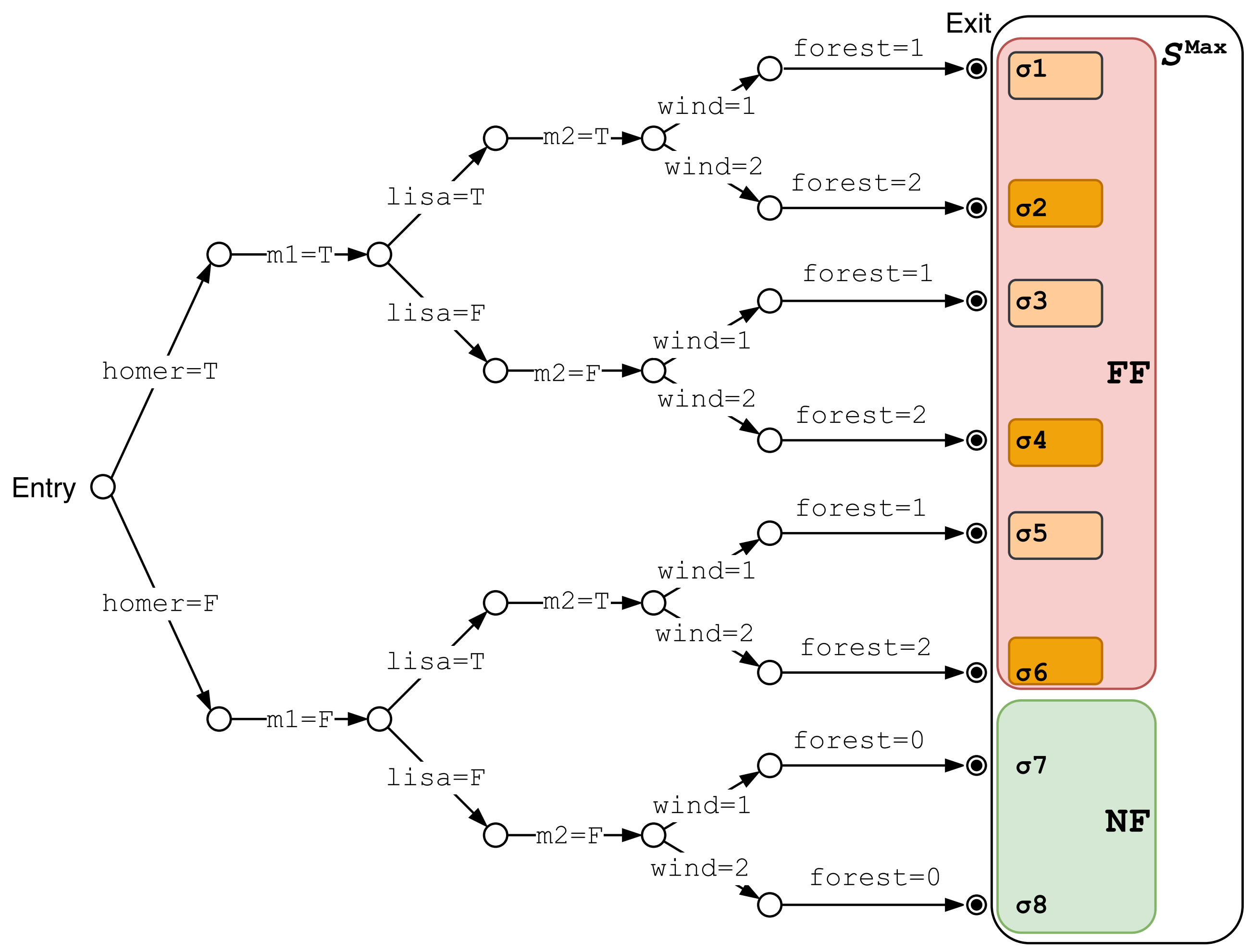
1) System Semantics

- **Maximal Trace Semantics** $\mathcal{S}^{\text{Max}} \in \wp(\mathrm{E}^{*\infty})$
  - the set of all valid maximal traces
- **Prefix Trace Semantics** $\mathcal{S}^{\text{Pref}} \in \wp(\mathrm{E}^{*\infty})$
  - the set of all valid prefix traces
  - an abstraction of maximal trace semantics $\mathcal{S}^{\text{Pref}} = \text{Pref}(\mathcal{S}^{\text{Max}})$
- E.g. Forest Fire Program
  - consists of 8 maximal traces



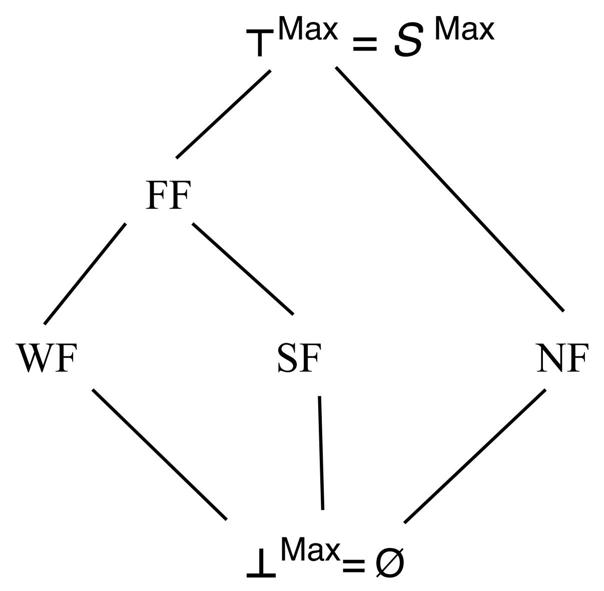
2) Lattice of System Behaviors of Interest

- **Trace Property**
  - A trace property is a set of traces in which a given property holds.
  - Every system behavior is represented as a maximal trace property $\mathcal{P} \in \wp(\mathcal{S}^{\text{Max}})$
  - E.g. "Forest Fire" `FF` is a set of maximal traces whose last event is `forest=1` or `forest=2`. Similarly, "No forest Fire" `NF(forest=0)`, "Weak forest Fire" `WF(forest=1)`, and "Strong forest Fire" `SF(forest=2)`.



2) Lattice of System Behaviors of Interest

- **Complete Lattice of Maximal Trace Properties of Interest**
  $\langle \mathcal{L}^{\mathsf{Max}}, \subseteq, \top^{\mathsf{Max}}, \bot^{\mathsf{Max}}, \cup\!\!\!\cdot, \cap\!\!\!\cdot \rangle$
  - $\mathcal{L}^{\mathsf{Max}} \in \wp(\wp(\mathrm{E}^{*\infty}))$ is a set of system behaviors of interest.
  - $\top^{\mathsf{Max}} = \mathcal{S}^{\mathsf{Max}}$ is the weakest maximal trace property which holds in every valid maximal trace.
  - $\bot^{\mathsf{Max}} = \emptyset$ is the strongest property such that no valid trace has this property, hence it is used to represent the property of invalidity.
  - E.g. Forest Fire Program



$\top^{\mathsf{Max}} = \mathcal{S}^{\mathsf{Max}}$
FF
WF
SF
NF
$\bot^{\mathsf{Max}} = \emptyset$

Behaviors
FF: Forest Fire (forest > 0)
NF: No Fire (forest = 0)
WF: Weak Fire (forest = 1)
SF: Strong Fire (forest = 2)

2) Lattice of System Behaviors of Interest

- **Prediction Abstraction**
  - Along a maximal trace σ in a property P, where is the point from which P is guaranteed to hold later in the execution?
    - E.g. in $\sigma 1 = \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T} \triangleright \texttt{m2=T} \triangleright \texttt{wind=1} \triangleright \texttt{forest=1}$, the property “Forest Fire” `FF` is guaranteed immediately after the event `homer=T`.
  - Abstraction from Maximal Trace Property P into Prediction Trace Property Q.
    - P is isomorphically abstracted into a set Q of prefixes, excluding those whose maximal prolongation may not satisfy P.

$$\alpha_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!] \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty}) \qquad \text{prediction abstraction}$$

$$\alpha_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!](\mathcal{P}) \triangleq \{\sigma \in \mathsf{Pref}(\mathcal{P}) \mid \forall \sigma' \in \mathcal{S}^{\mathsf{Max}}.\ \sigma \preceq \sigma' \Rightarrow \sigma' \in \mathcal{P}\}$$

$$\gamma_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!] \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty}) \qquad \text{prediction concretization}$$

$$\gamma_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!](\mathcal{Q}) \triangleq \{\sigma \in \mathcal{Q} \mid \sigma \in \mathcal{S}^{\mathsf{Max}}\} = \mathcal{Q} \cap \mathcal{S}^{\mathsf{Max}}$$

- E.g. for the property “Forest Fire” `FF`, the corresponding prediction trace property is
  $\alpha_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!](\texttt{FF}) = \{\sigma \in \mathcal{S}^{\mathsf{Pref}} \mid \texttt{homer=T} \preceq \sigma \vee \texttt{homer=F} \triangleright \texttt{m1=F} \triangleright \texttt{lisa=T} \preceq \sigma\}$
  i.e. for any valid prefix trace, after `homer=T` or `lisa=T` occurs, `FF` is guaranteed.
- E.g. $\alpha_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!](\texttt{NF}) = \{\sigma \in \mathcal{S}^{\mathsf{Pref}} \mid \texttt{homer=F} \triangleright \texttt{m1=F} \triangleright \texttt{lisa=F} \preceq \sigma\}$
  i.e. for any valid prefix trace, only after `homer=F` and `lisa=F`, `NF` is guaranteed.

3) Observation of System Behaviors

- **Inquiry Function** (strongest guaranteed property on a trace σ)
  - Given $\mathcal{S}^{\mathsf{Max}}$ and $\mathcal{L}^{\mathsf{Max}}$, an inquiry function $\mathbb{I}$ maps a trace σ to the strongest maximal trace property in $\mathcal{L}^{\mathsf{Max}}$ that σ guarantees.

$$\mathbb{I} \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\wp(\mathrm{E}^{*\infty})) \mapsto \mathrm{E}^{*\infty} \mapsto \wp(\mathrm{E}^{*\infty}) \qquad \text{inquiry } (2)$$

$$\begin{aligned}&\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \sigma) \triangleq\\ &\quad \mathit{let}\ \alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P}) = \{\sigma \in \mathsf{Pref}(\mathcal{P}) \mid \forall \sigma' \in \mathcal{S}.\ \sigma \preceq \sigma' \Rightarrow \sigma' \in \mathcal{P}\}\ \mathit{in}\\ &\qquad \Cap\{\mathcal{P} \in \mathcal{L}^{\mathsf{Max}} \mid \sigma \in \alpha_{\mathsf{Pred}}[\![\mathcal{S}^{\mathsf{Max}}]\!](\mathcal{P})\}\end{aligned}$$

  - The inquiry function is decreasing: the greater (longer) σ is, the stronger property it guarantees.
  - E.g. $\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \varepsilon) = \top^{\mathsf{Max}}$ At the entry, it is not guaranteed if there will be fire or not.

$\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=T}) = \texttt{FF}$ After Homer lit a match, the forest fire is guaranteed.

$\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=F}) = \mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=F} \triangleright \texttt{m1=F}) = \top^{\mathsf{Max}}$ If Homer does not light a match, it is not guaranteed if there will be fire or not.

$\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=F} \triangleright \texttt{m1=F} \triangleright \texttt{lisa=T}) = \texttt{FF}$ After Lisa lit a match, the fire is guaranteed.

$\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=F} \triangleright \texttt{m1=F} \triangleright \texttt{lisa=F}) = \texttt{NF}$ If neither Homer nor Lisa lit a match, it is guaranteed to have no fire.

3) Observation of System Behaviors

- **Cognizance Function** (Observable Abstraction of the Semantics)
  - An observer may not distinguish a trace σ from some other traces.

$$\mathbb{C} \in \mathrm{E}^{*\infty} \mapsto \wp(\mathrm{E}^{*\infty}) \qquad \text{cognizance (3)}$$
$$\mathbb{C}(\sigma) \triangleq \{\sigma' \in \mathrm{E}^{*\infty} \mid \text{observer cannot distinguish } \sigma' \text{ from } \sigma\}$$

  - Specially, for an omniscient observer, every trace is unambiguous.

$$\forall \sigma \in \mathrm{E}^{*\infty}.\ \mathbb{C}_o(\sigma) = \{\sigma\}$$

  - E.g. in program security, the observer's cognizance can be used to represent attackers' capabilities (e.g. what they can learn from the program execution).
  - E.g. in the forest fire example, if the observer observes only Lisa, but not Homer:

$$\mathbb{C}(\texttt{homer=T} \triangleright \texttt{m1=T}) = \mathbb{C}(\texttt{homer=F} \triangleright \texttt{m1=F}) = \{\texttt{homer=T} \triangleright \texttt{m1=T}\ ,\ \texttt{homer=F} \triangleright \texttt{m1=F}\}$$

3) Observation of System Behaviors

- **Observation Function**
  - What can an observer learn from a trace σ?
    1. Get the set $\mathbb{C}(\sigma)$ of traces that the observer cannot distinguish from σ;
    2. For each trace σ' in $\mathbb{C}(\sigma)$, apply the inquiry function $\mathbb{I}$ to get a trace property guaranteed by σ';
    3. By joining all trace properties returned by $\mathbb{I}$, we get the property learned from σ.

$$\mathbb{O} \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\wp(\mathrm{E}^{*\infty})) \mapsto (\mathrm{E}^{*\infty} \mapsto \wp(\mathrm{E}^{*\infty})) \mapsto \mathrm{E}^{*\infty} \mapsto \wp(\mathrm{E}^{*\infty})$$

$$\mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma) \triangleq \qquad \text{observation } (4)$$
$$\quad let\ \alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P}) = \{\sigma \in \mathsf{Pref}(\mathcal{P}) \mid \forall \sigma' \in \mathcal{S}.\ \sigma \preceq \sigma' \Rightarrow \sigma' \in \mathcal{P}\}\ in$$
$$\quad\quad let\ \mathbb{I}(\mathcal{S}, \mathcal{L}, \sigma) = \Cap\{\mathcal{P} \in \mathcal{L} \mid \sigma \in \alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P})\}\ in$$
$$\quad\quad\quad \Cup\{\mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \sigma') \mid \sigma' \in \mathbb{C}(\sigma)\}.$$

- For an omniscient observer, the observation function = the inquiry function.
- The observation function is decreasing: the greater (longer) σ is, the stronger property is learned.
- E.g. in the forest fire example, if the observer is Lisa who is unaware of Homer's action:

$$\mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T}) = \mathbb{I}\left(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=T} \triangleright \texttt{m1=T}\right) \Cup \mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=F} \triangleright \texttt{m1=F}) = \texttt{FF} \Cup \top^{\mathsf{Max}} = \top^{\mathsf{Max}}$$

$$\mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T}) = \mathbb{I}\left(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T}\right) \Cup \mathbb{I}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \texttt{homer=F} \triangleright \texttt{m1=F} \triangleright \texttt{lisa=T}) = \texttt{FF} \Cup \texttt{FF} = \texttt{FF}$$

4) Formal Definition of Responsibility

- To an observer's cognizance, in a trace $\sigma_H\sigma_R\sigma_F$, σ_R is responsible for behavior B, iff: $\emptyset \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma_{\mathrm{H}}\sigma_{\mathrm{R}}) \subseteq \mathcal{B} \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma_{\mathrm{H}})\}$
  - B cannot be learned from σ_H
  - B can be learned from $\sigma_H\sigma_R$

- Responsibility Abstraction

$$\alpha_R \in \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\wp(\mathrm{E}^{*\infty})) \mapsto (\mathrm{E}^{*\infty} \mapsto \wp(\mathrm{E}^{*\infty}))$$
$$\mapsto \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty}) \mapsto \wp(\mathrm{E}^{*\infty} \times \mathrm{E} \times \mathrm{E}^{*\infty})$$

$$\begin{array}{l}
\alpha_R(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \mathcal{B}, \mathcal{T}) \triangleq \\
\quad \mathit{let}\ \alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P}) = \{\sigma \in \mathsf{Pref}(\mathcal{P}) \mid \forall \sigma' \in \mathcal{S}.\ \sigma \preceq \sigma' \Rightarrow \sigma' \in \mathcal{P}\}\ \mathit{in} \\
\quad\quad \mathit{let}\ \mathbb{I}(\mathcal{S}, \mathcal{L}, \sigma) = \Cap\{\mathcal{P} \in \mathcal{L} \mid \sigma \in \alpha_{\mathsf{Pred}}[\![\mathcal{S}]\!](\mathcal{P})\}\ \mathit{in} \\
\quad\quad\quad \mathit{let}\ \mathbb{O}(\mathcal{S}, \mathcal{L}, \mathbb{C}, \sigma) = \Cup\{\mathbb{I}(\mathcal{S}, \mathcal{L}, \sigma') \mid \sigma' \in \mathbb{C}(\sigma)\}\ \mathit{in} \\
\quad\quad\quad\quad \{\langle \sigma_{\mathrm{H}},\ \sigma_{\mathrm{R}},\ \sigma_{\mathrm{F}} \rangle \mid \sigma_{\mathrm{H}}\sigma_{\mathrm{R}}\sigma_{\mathrm{F}} \in \mathcal{T} \wedge |\sigma_{\mathrm{R}}| = 1 \wedge \\
\quad\quad\quad\quad\quad \emptyset \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma_{\mathrm{H}}\sigma_{\mathrm{R}}) \subseteq \mathcal{B} \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma_{\mathrm{H}})\}
\end{array}$$

$$\langle \wp(\mathrm{E}^{*\infty}),\ \subseteq \rangle \underset{\alpha_R(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \mathcal{B})}{\overset{\gamma_R(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \mathcal{B})}{\leftrightarrows}} \langle \wp(\mathrm{E}^{*\infty} \times \mathrm{E} \times \mathrm{E}^{*\infty}),\ \subseteq \rangle$$

- Maximal trace semantics: $\mathcal{S}^{\mathsf{Max}}$
- Lattice of system behaviors: $\mathcal{L}^{\mathsf{Max}}$
- Observer's cognizance: $\mathbb{C}$
- Behavior of interest: $\mathcal{B}$
- Set of analyzed traces: $\mathcal{T}$

- An variant: $\emptyset \subsetneq \mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma_{\mathrm{H}}\sigma_{\mathrm{R}}) \subseteq \mathcal{B} \not\supseteq \mathbb{O}(\mathcal{S}^{\mathsf{Max}}, \mathcal{L}^{\mathsf{Max}}, \mathbb{C}, \sigma_{\mathrm{H}})$

4) Formal Definition of Responsibility

- To an observer's cognizance, in a trace σ_Hσ_Rσ_F, σ_R is responsible for behavior B, iff: $\emptyset \subsetneq \mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \sigma_{\text{H}}\sigma_{\text{R}}) \subseteq \mathcal{B} \subsetneq \mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \sigma_{\text{H}})\}$
  - E.g. in the forest fire example, analyze $\sigma 1 = \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T} \triangleright \texttt{m2=T} \triangleright \texttt{wind=1} \triangleright \texttt{forest=1}$
    - For the omniscient observer:

$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \varepsilon) = \top^{\text{Max}}$$
$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T}) = \texttt{FF}$$

$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T} \triangleright \texttt{m2=T}) = \texttt{FF}$$
$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T} \triangleright \texttt{m2=T} \triangleright \texttt{wind=1}) = \texttt{WF}$$

    - For a non-omniscient observer of Lisa who is unaware of Homer's action:

$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T}) = \top^{\text{Max}}$$
$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T}) = \texttt{FF}$$

$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T} \triangleright \texttt{m2=T}) = \texttt{FF}$$
$$\mathbb{O}(\mathcal{S}^{\text{Max}}, \mathcal{L}^{\text{Max}}, \mathbb{C}, \texttt{homer=T} \triangleright \texttt{m1=T} \triangleright \texttt{lisa=T} \triangleright \texttt{m2=T} \triangleright \texttt{wind=1}) = \texttt{WF}$$

Responsibility Analysis

- Responsibility analysis
    - Collect the system's trace semantics $\mathcal{S}^{\text{Max}}$;
    - Build the lattice $\mathcal{L}^{\text{Max}}$ of maximal trace properties of interest;
    - Derive an inquiry function $\mathbb{I}$ from $\mathcal{L}^{\text{Max}}$,

      define a cognizance function $\mathbb{C}$ for each observer,

      create the corresponding observation function $\mathbb{O}$;
    - Specify the behavior $\mathcal{B}$ of interest and the analyzed traces $\mathcal{T}$,

      apply the responsibility abstraction α_R.
- Applications
    - E.g. buffer overflow analysis, information leakage analysis
    - E.g. permissions configuration/authorization in a social network

More Work in the Appendix

- Abstract Responsibility Analysis
  - Abstract system semantics by Floyd-Hoare Automaton
    - Leverage trace partitioning to increase precision
  - Specify system behaviors as program invariants
  - Define the cognizance function as an equivalence relation on invariants
  - Build the observation function on automaton nodes
    - Utilize iterative forward/backward invariance analysis, and CTL property analysis
  - Find “under conditions” responsibility that is a sound abstraction of concrete responsibility analysis

Ongoing Work

- A lattice of responsibility abstractions
    - Cope with possible alternative weaker or stronger definitions of responsibility
- Generalization of cognizance function
    - Not only an equivalence relation, but also an abstraction of system semantics

Conclusion

- Responsibility based on the abstract interpretation of event trace semantics
    - More precise than dependency, actual cause, etc.
    - The observer's cognizance is new in responsibility analysis.

The End. Thank You!