

Abstract Detection of Computer Viruses

APPSEM II

Guillaume Bonfante, Matthieu Kaczmarek and Jean-Yves Marion

LORIA – Mines de Nancy – INPL



What is a Computer Virus



Cohen (1986)

A computer virus wrt a Turing Machine is a word that replicates itself when it is executed.



Adleman (1988)

In a programming environment, a computer virus is recursive function that maps each program to an infected form. And infected forms produce others infected forms.



Programming Environment

* Programming Language.

A domain of computation \mathcal{D} .

A programming language $\varphi : \mathcal{D} \rightarrow (\mathcal{D} \rightarrow \mathcal{D})$.

For all $\mathbf{p} \in \mathcal{D}$, $\varphi_{\mathbf{p}}$ is the function computed by \mathbf{p} .

* Acceptable Programming Language.

Turing complete.

A universal program \mathbf{u} , $\varphi_{\mathbf{u}}(\mathbf{p}, x) = \varphi_{\mathbf{p}}(x)$.

The iteration function (s-m-n) S , $\varphi_{S(\mathbf{p}, x)}(y) = \varphi_{\mathbf{p}}(x, y)$.



Definition of a Computer Virus

- * A propagation function \mathcal{B} .

$\mathcal{B}(\mathbf{v}, \mathbf{p})$ spreads the code \mathbf{v} through \mathbf{p} .

A vulnerability, an infection vector.

- * A computer virus $(\mathcal{B}, \mathbf{v})$ is such that the program \mathbf{v} propagates its own code wrt the function \mathcal{B} .

$$\forall \mathbf{p}, x : \varphi_{\mathbf{v}}(\mathbf{p}, x) = \varphi_{\mathcal{B}(\mathbf{v}, \mathbf{p})}(x).$$

$\mathcal{B}(\mathbf{v}, \mathbf{p})$ is the infected form of \mathbf{p} wrt \mathbf{v} .

Example

- * \vec{p} a program structure, $\vec{p} = (p_1, \dots, p_n)$.
- * A duplication function $\Delta(\mathbf{v}, \vec{p}) = (p_1 \cdot \mathbf{v}, \dots, p_n \cdot \mathbf{v})$.
- * Bash Code (E. Filiol)

```
for FName in $(ls *.infect.sh); do
  if [ ./$FName != $0 ]; then
    echo [$0 infects ./$FName]
    tail $0 -n 6 | cat >> ./$FName
  fi
done
```

- * Concrete example, the virus Jerusalem (1988).



Basic theorems

* Iteration theorem (Kleene 56).

There is a function S such that

$$\varphi_{S(p,q)}(x) = \varphi_p(q, x) .$$

* Recursion theorem (Kleene 56).

For all semi-computable function f , there is e such that

$$\varphi_e(x) = f(e, x) .$$

e is a fixed point of f .

Example

- * The duplication function $\Delta(y, \vec{p}) = (\mathbf{p}_1 \cdot y, \dots, \mathbf{p}_n \cdot y)$.

$$\varphi_{\mathbf{q}}(y, \vec{p}, x) = \Delta(y, \vec{p})$$

\mathbf{q} computes Δ .

$$\varphi_{S(\mathbf{q}, y, \vec{p})}(x) = \varphi_{\mathbf{q}}(y, \vec{p}, x)$$

Iteration theorem.

$$\varphi_{\mathbf{v}}(\vec{p}, x) = \varphi_{\mathbf{q}}(\mathbf{v}, \vec{p}, x)$$

Recursion theorem.

- * With $\mathcal{B}(\mathbf{v}, \vec{p}) = S(\mathbf{q}, \mathbf{v}, \vec{p})$,

$$\varphi_{\mathbf{v}}(\vec{p}, x) = \varphi_{\mathcal{B}(\mathbf{v}, \vec{p})}(x) \quad \mathbf{v} \text{ is a virus}$$

$$= \Delta(\mathbf{v}, \vec{p}) = (\mathbf{p}_1 \cdot \mathbf{v}, \dots, \mathbf{p}_n \cdot \mathbf{v}) .$$

- * The virus $(\mathcal{B}, \mathbf{v})$ adds its own code at the end of each program.



Computability of Virus Detection

- * A detection strategy: identify all viruses for each propagation function.
- * The set of the viruses associated to propagation function \mathcal{B} .

$$V_{\mathcal{B}} = \{ \mathbf{v} \mid \varphi_{\mathbf{v}} \approx \lambda \mathbf{p} . \varphi_{\mathcal{B}(\mathbf{v}, \mathbf{p})} \} .$$

- * **Theorem.**
There is a \mathcal{B} such that the set of viruses is not computable (Π_2 -complete).
- * **Theorem.**
There is a \mathcal{B} such that the set of viruses is computable.

Computability of Virus Detection

- * Signature matching is used by most anti-virus softwares.
- * Detect all infected wrt a known virus $(\mathcal{B}, \mathbf{v})$: the infection set.

$$I_{\mathcal{B}, \mathbf{v}} = \{ \mathcal{B}(\mathbf{v}, \mathbf{p}) \mid \mathbf{p} \in \mathcal{D} \} .$$

- * **Theorem.**

There is a computer virus such that the infection set is not computable (Σ_1 -complete).

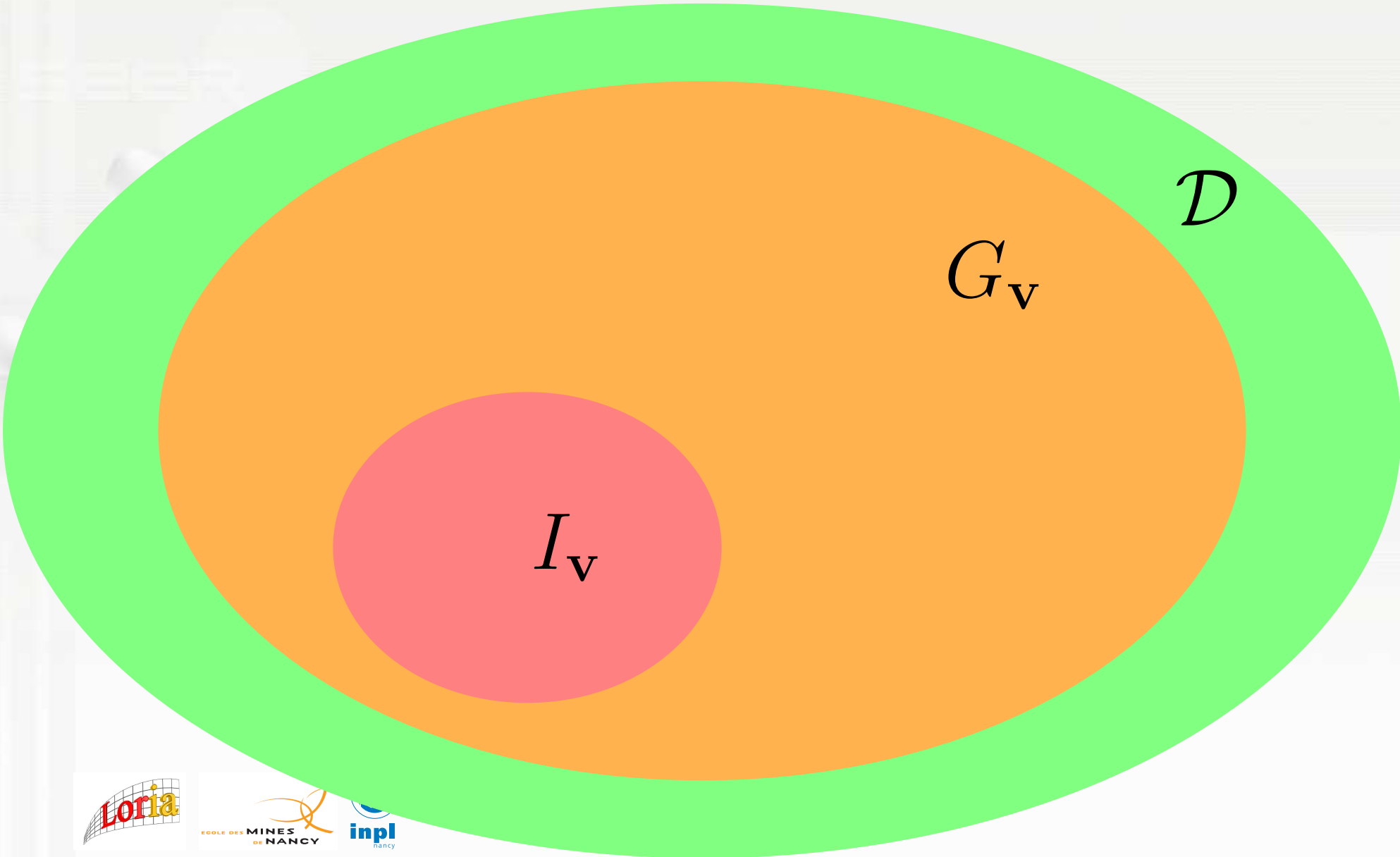
Computability of Virus Detection

- * Spectral analysis, code emulation... Detecting a viral behavior.
- * The set of all programs that behave as infected forms, the germ.

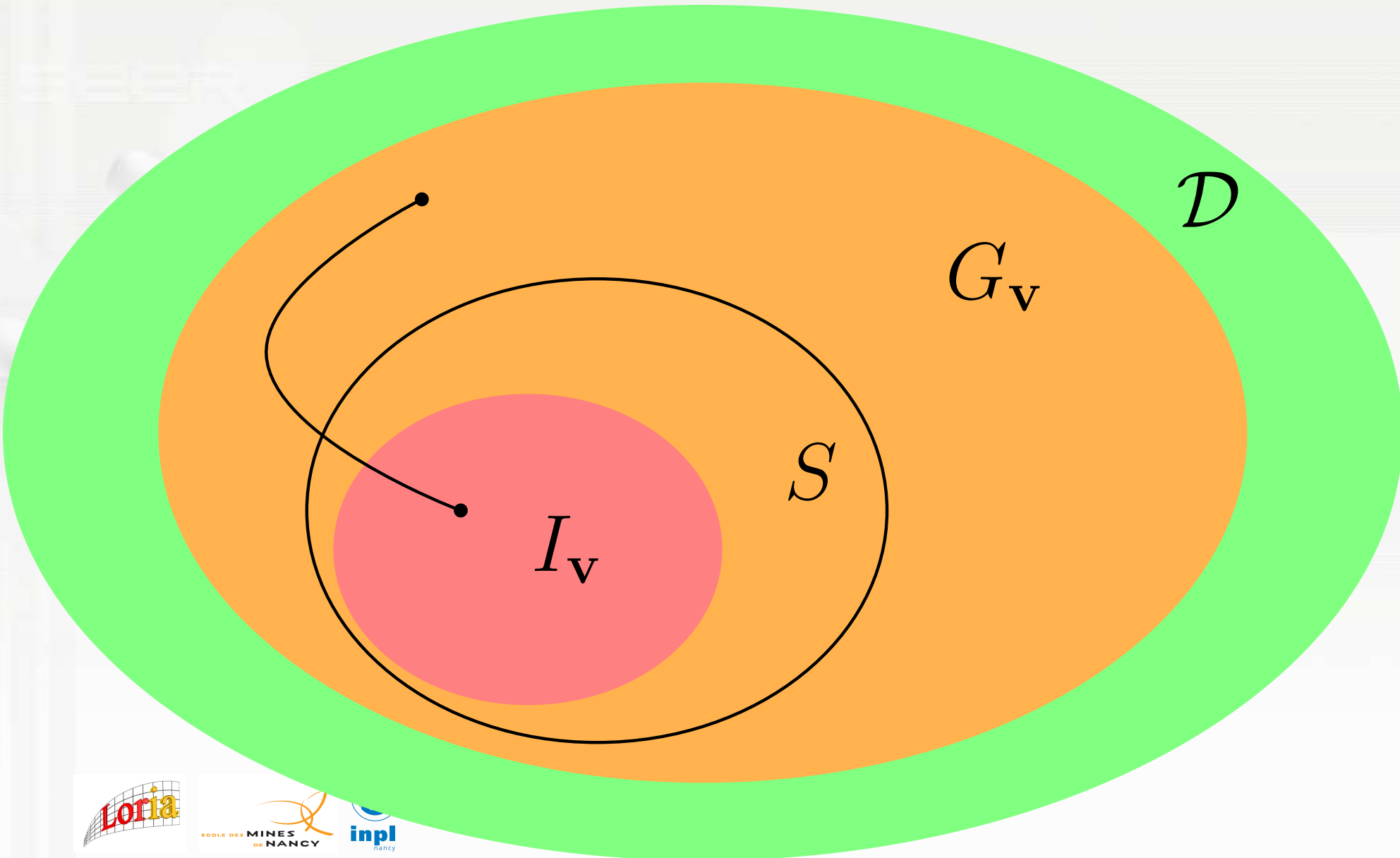
$$G_{\mathcal{B},v} = \{p \mid \exists q \in I_{\mathcal{B},v} : \varphi_p \approx \varphi_q\} .$$

$$\varphi_v \approx \lambda p. \varphi_{\mathcal{B}(v,p)}$$

Isolate within the germ



Isolate within the germ



Isolate within the germ

* **Theorem.**

There is a computer virus which cannot be isolated within its germ.


$$\varphi_v \approx \lambda p \cdot \varphi_B(v, p)$$

Further Research

* By the way... what are the properties of a propagation function?



$$\varphi_v \approx \lambda p \cdot \varphi_B(v, p)$$



$$\varphi_v \approx \lambda p \cdot \varphi_B(v, p)$$

