

Mocos, Fallas y Fallutadas

Todo lo que no nos gusta ver en el software

Pedro R. D'Argenio

Universidad Nacional de Córdoba – CONICET (AR)

Saarland University (DE)

<http://www.cs.famaf.unc.edu.ar/~dargenio/>



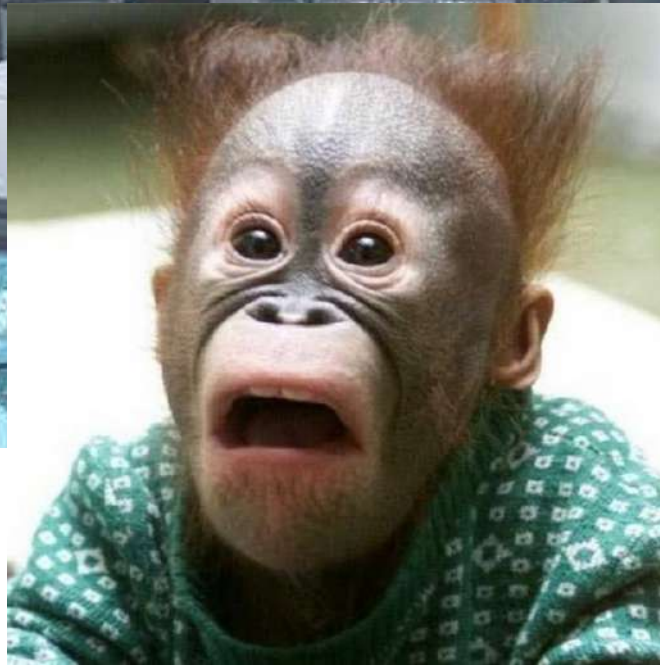
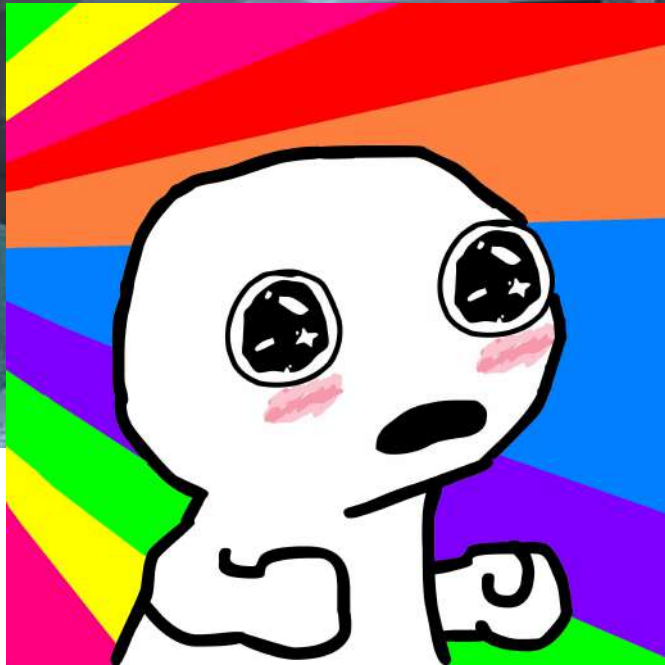
LCC 2018 - Rosario



El software parece hacer maravillas...



El software parece hacer maravillas...



... pero debajo de esa cáscara de maravilla
nos encontramos con una pila de ...



¿De donde sale toda esa porquería?

- ❖ Errores en el desarrollo del software
- ❖ Fallas externas al software pero que son parte del sistema
- ❖ Programación malintencionada

¿De donde sale toda esa porquería?

- ❖ Errores en el desarrollo del software

Bugs

- ❖ Fallas externas al software pero que son parte del sistema

- ❖ Programación malintencionada

¿De donde sale toda esa porquería?

- ❖ Errores en el desarrollo del software
- ❖ Fallas externas al software pero que son parte del sistema
- ❖ Programación malintencionada

MOCOS

FALLAS

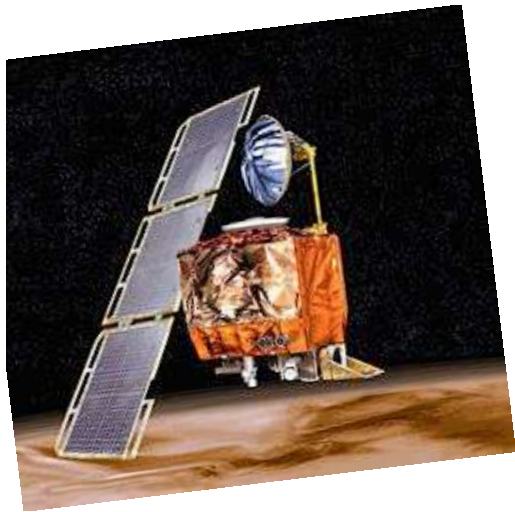
FALLUTADAS

Mocos famosos

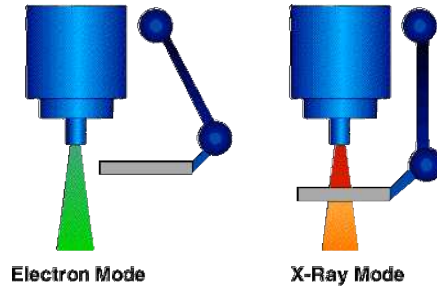


Pentium:
FDIV

Ariane 5:
64 bits fp
vs 16 bits int



Mars Climate
Orbiter:
Métrico vs Imperial



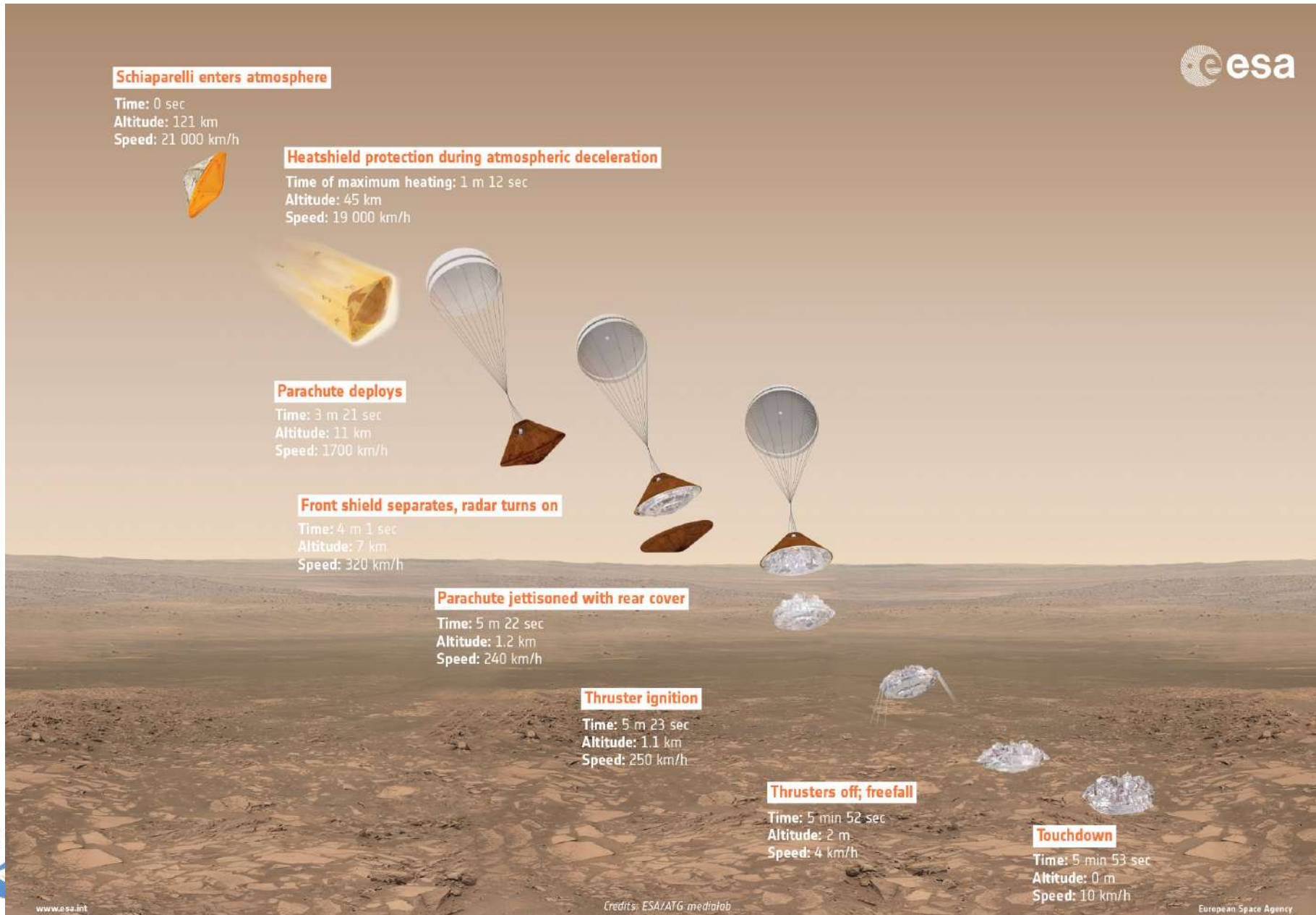
Therac-25:
Condición de
carrera

Northeast blackout
en 2003:
Condición de
carrera

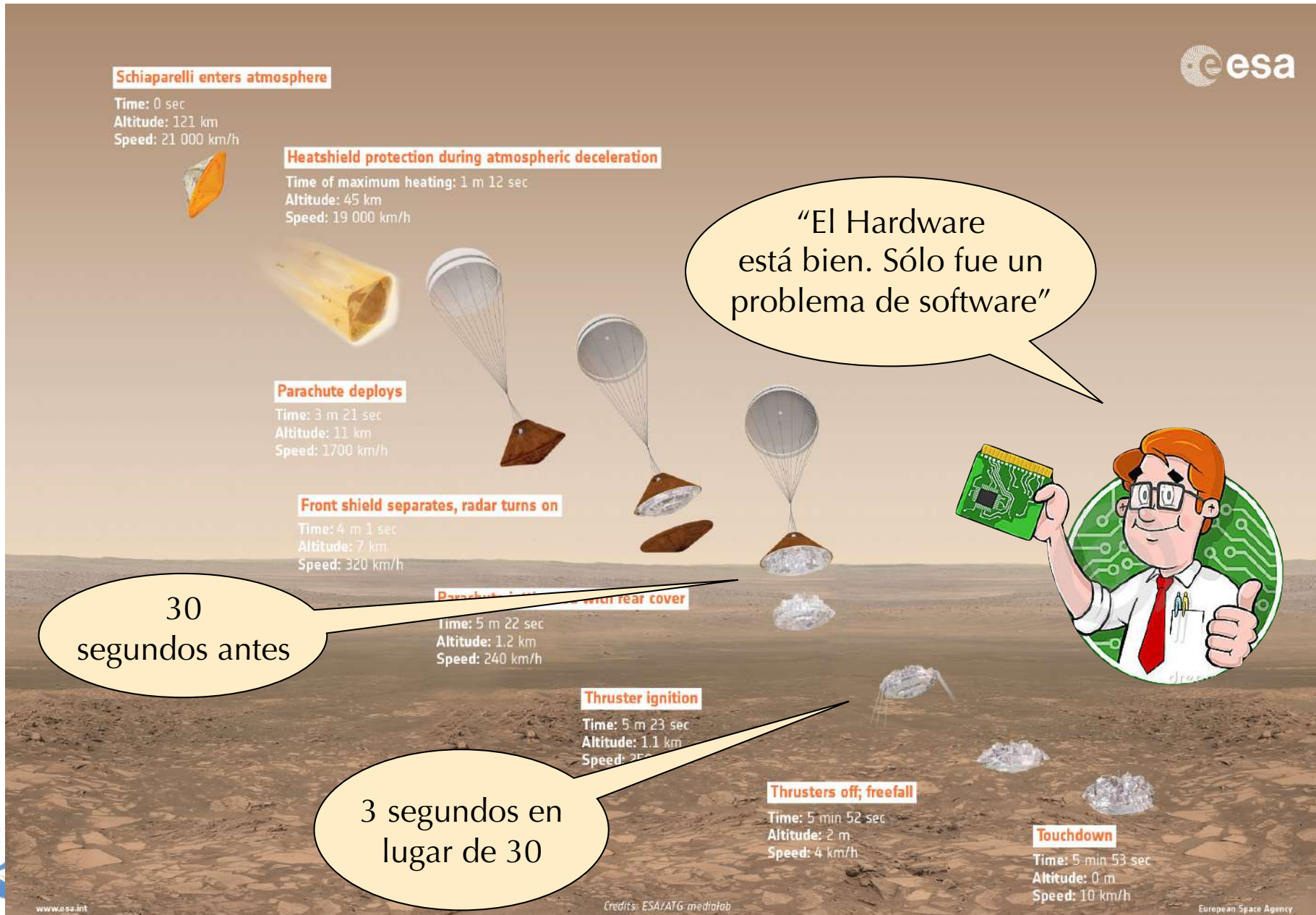


Heartbleed:
Integridad/Confidencialidad

El problema de llamarlo *Bug*

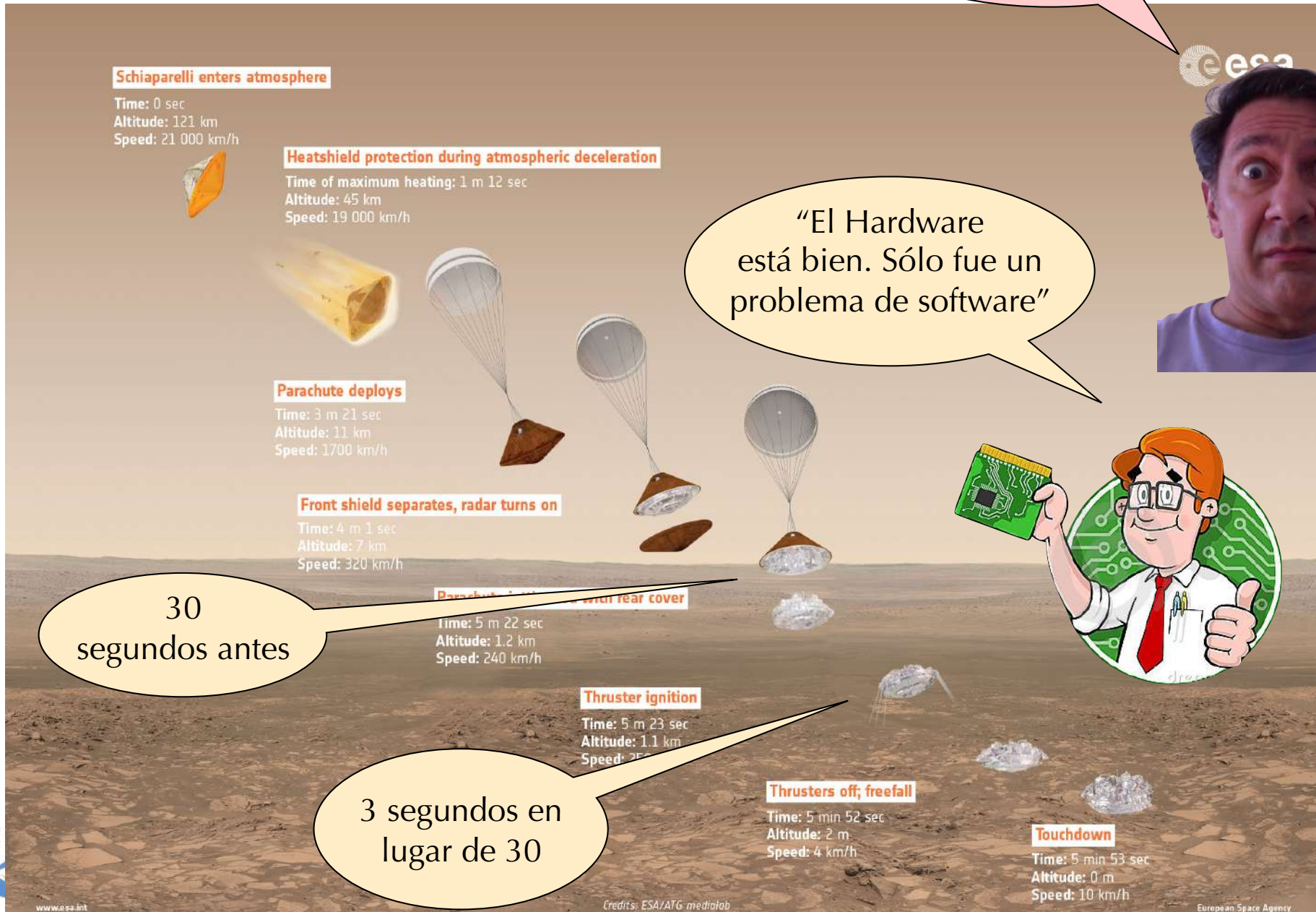


El problema de llamarlo *Bug*

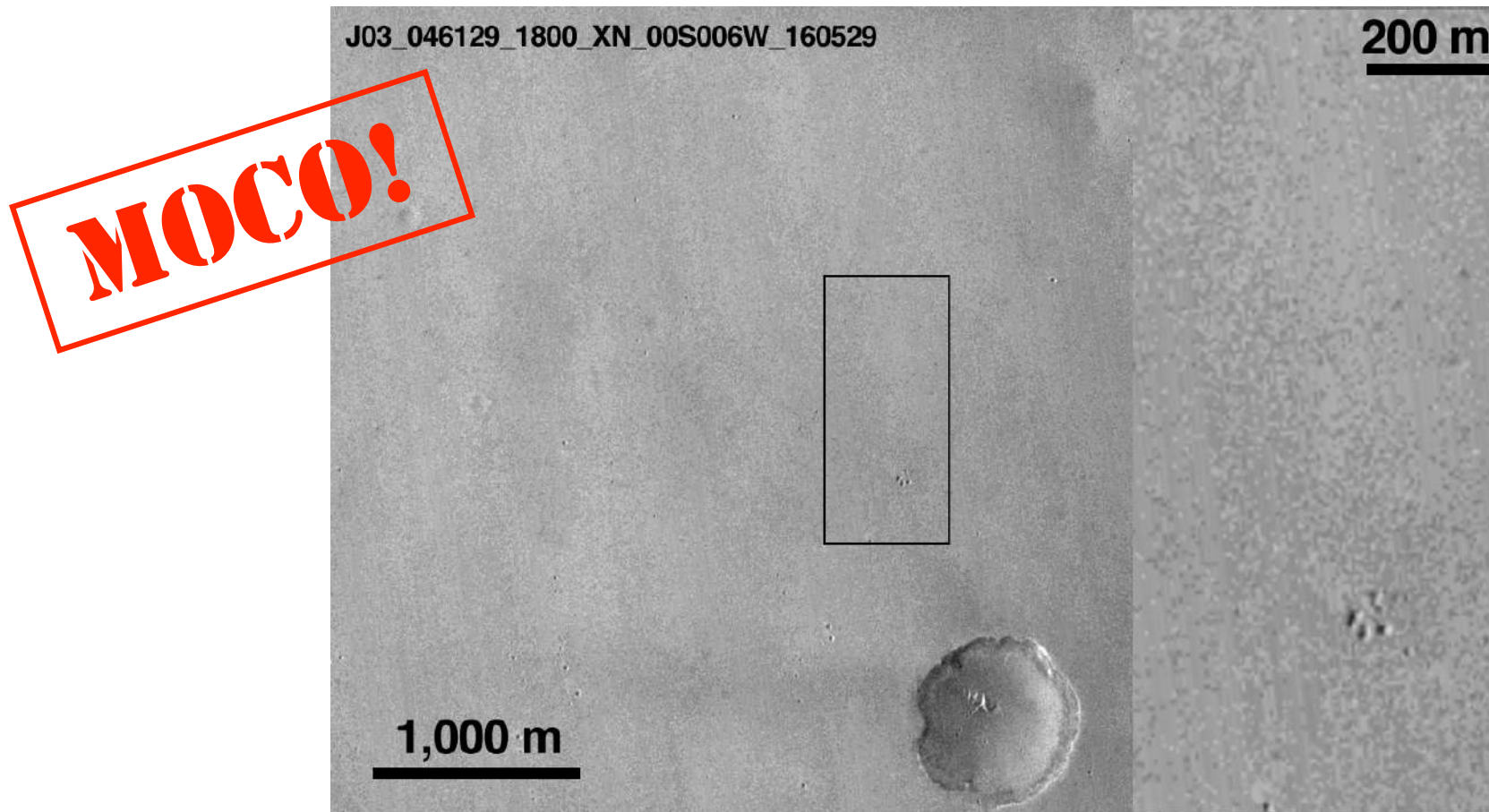


El problema de llamo

WTF!!!!



El problema de llamarlo *Bug*



https://en.wikipedia.org/wiki/Schiaparelli_EDM_lander

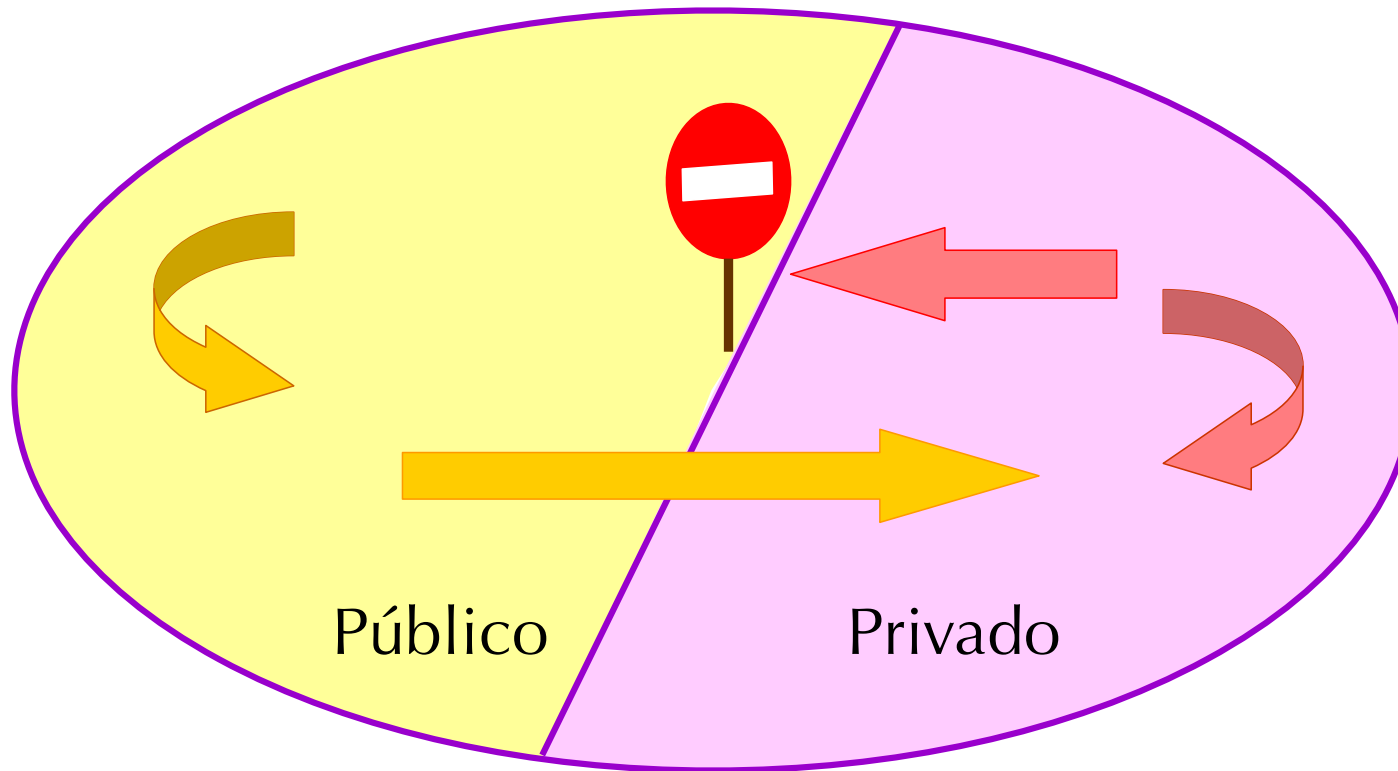
El problema de la corrección

Sistema \models Propiedad

Usualmente una
abstracción que describe su
comportamiento

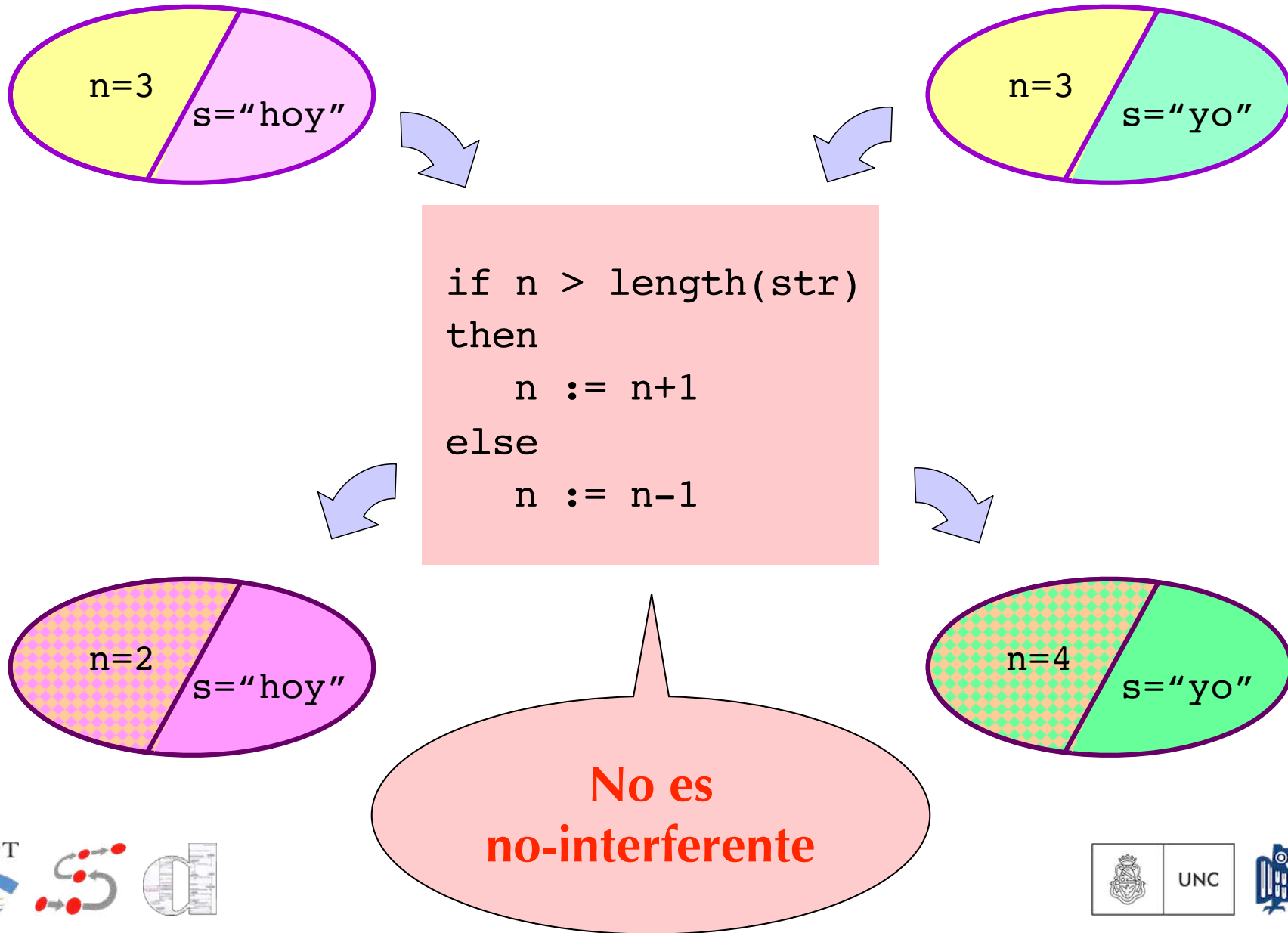
Describe lo que se
espera del sistema
(el criterio de corrección)

Ejemplo: Confidencialidad

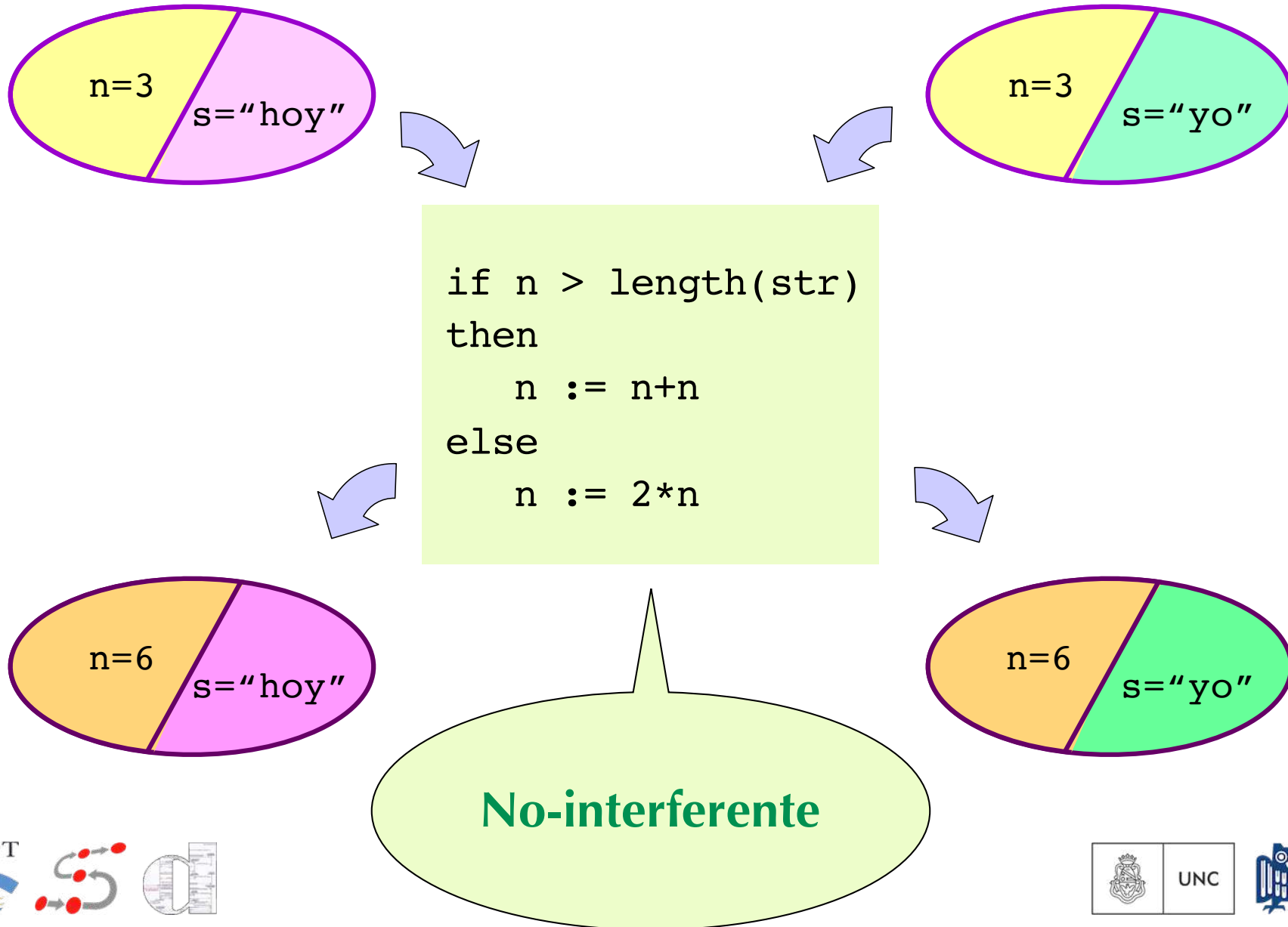


Espacio de datos

No-interferencia



No-interferencia



No-interferencia

definición formal

❖ Memoria: $\mu : \text{Variables} \rightarrow \text{Valores}$

Estado inicial

❖ Un programa es un transformador de memoria: $(S, \mu) \Downarrow \mu'$

Estado final

No-interferencia

definición formal

- ❖ Memoria: $\mu : \text{Variables} \rightarrow \text{Valores}$
- ❖ Un programa es un transformador de memoria: $(S, \mu) \Downarrow \mu'$
- ❖ Variables:
Públicas: $l \in \text{Variables}$ Privadas: $h \in \text{Variables}$
- ❖ No-interferencia

S es *no-interferente* si para todas μ_1, μ_2, μ'_1 y μ'_2 ,

$$\left. \begin{array}{l} \mu_1(l) = \mu_2(l) \\ (S, \mu_1) \Downarrow \mu'_1 \\ (S, \mu_2) \Downarrow \mu'_2 \end{array} \right\} \Rightarrow \mu'_1(l) = \mu'_2(l)$$

No-interferencia

definición formal

Hiperpropiedad:
ve más de una ejecución
al mismo tiempo

¿Se puede analizar
con lógica de Hoare?

❖ No-interferencia

S es *no-interferente* si para todas μ_1, μ_2, μ'_1 y μ'_2 ,

$$\left. \begin{array}{l} \mu_1(l) = \mu_2(l) \\ (S, \mu_1) \Downarrow \mu'_1 \\ (S, \mu_2) \Downarrow \mu'_2 \end{array} \right\} \Rightarrow \mu'_1(l) = \mu'_2(l)$$

Self-Composition

$$\{l = l'\} S ; S[\vec{x}/\vec{x}'] \{l = l'\}$$

Self-Composition

$$\{l = l'\} S ; S[\vec{x}/\vec{x}'] \{l = l'\}$$

Secure Information Flow by Self-Composition

Gilles Barthe^{1*} Pedro R. D'Argenio^{2†} Tamara Rezk^{3*}
¹INRIA Sophia-Antipolis, Project EVEREST,
2004, Route de Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France
{Gilles.Barthe, Tamara.Rezk}@inria.fr
²Laboratoire d'Informatique Fondamentale
39, Rue Joliot Curie, 13453 Marseille
dargenio@cmi.univ-paris1.fr

Abstract

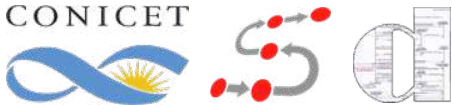
Non-interference is a high level security property that guarantees the absence of illicit information leaks through executing programs. More precisely, non-interference for a program assumes a separation between secret inputs and public outputs on the one hand, and secret outputs and public outputs on the other hand, such that the value of public outputs does not depend on the value of secret inputs.

Math. Struct. in Comp. Science (2011), vol. 21, pp. 1207–1252. © Cambridge University Press 2011
doi:10.1017/S0960129511000193

Secure information flow by self-composition[†]

GILLES BARTHE[‡], PEDRO R. D'ARGENIO[§] and TAMARA REZK^{*}
[‡]IMDEA Software Institute, Madrid, Spain
Email: gilles.barthe@imdea.org
[§]FaMAF, Universidad Nacional de Córdoba – CONICET, Córdoba, Argentina
Email: dargenio@famaf.unc.edu.ar
^{*}INRIA Sophia-Antipolis, INDES Project, France
Email: tamara.rezk@inria.fr

CONICET



Model Checking

$?\mathcal{M} \models \phi?$

```

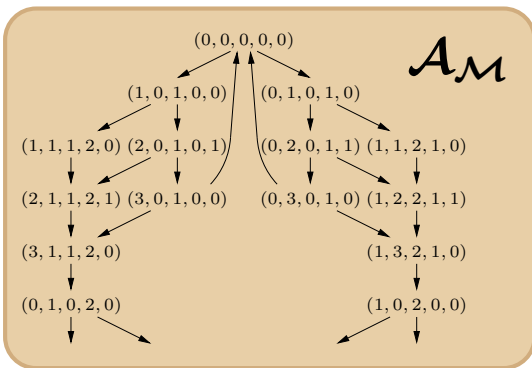
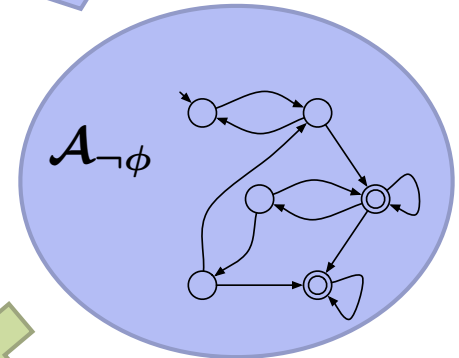
int y1 = 0;
int y2 = 0;
short in_critical = 0;

active proctype process_1() {
do
:: true ->
0:   y1 = y2+1;
1:   ((y2==0) || (y1<=y2));
    in_critical++;
2:   in_critical--;
3:   y1 = 0;
od
}

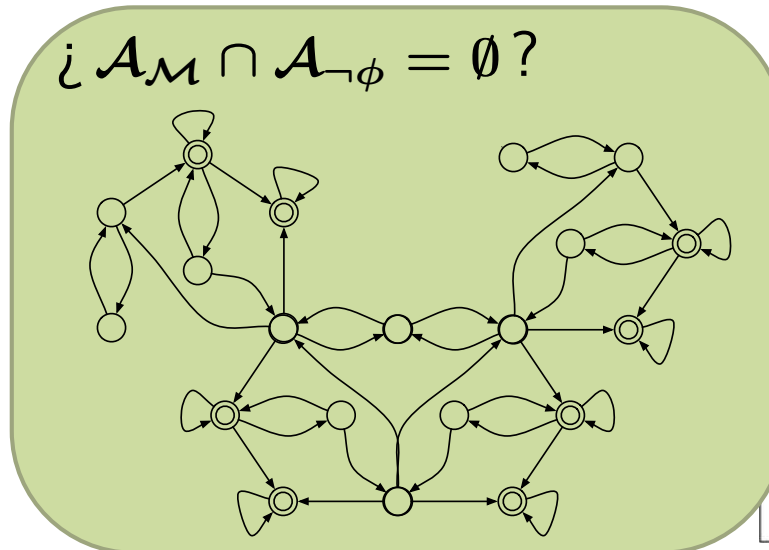
active proctype process_2() {
do
:: true ->
0:   y2 = y1+1;
1:   ((y1==0) || (y2<y1));
    in_critical++;
2:   in_critical--;
3:   y2 = 0;
od
}
    
```

\mathcal{M}

$\phi : \square \diamond crit_1 \wedge \square \diamond crit_2$



$?\mathcal{A}_M \cap \mathcal{A}_{\neg\phi} = \emptyset?$



Model Checking

$¿\mathcal{M} \models \phi?$

```

int y1 = 0;
int y2 = 0;
short in_critical = 0;

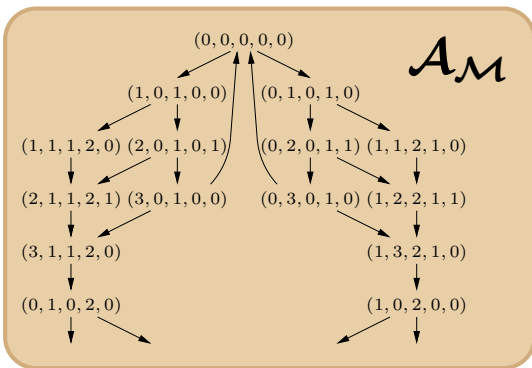
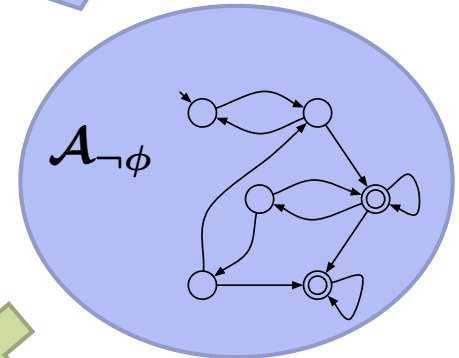
active proctype process_1() {
do
:: true ->
0: y1 = y2+1;
1: ((y2==0) || (y1<=y2));
   in_critical++;
2: in_critical--;
3: y1 = 0;
od
}

active proctype process_2() {
do
:: true ->
0: y2 = y1+1;
1: ((y1==0) || (y2<y1));
   in_critical++;
2: in_critical--;
3: y2 = 0;
od
}
    
```

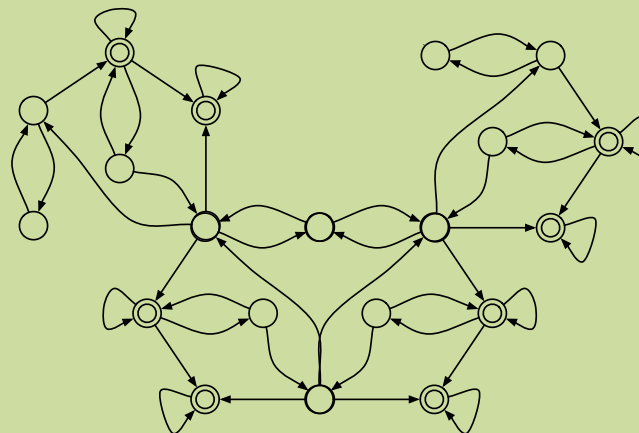
\mathcal{M}

$\phi : \square \diamond crit_1 \wedge \square \diamond crit_2$

El problema se reduce a análisis de grafos



$¿\mathcal{A}_{\mathcal{M}} \cap \mathcal{A}_{\neg\phi} = \emptyset?$



Limitaciones del Model Checking

- ❖ Muchos algoritmos **proponen (mejores) soluciones** utilizando **aleatoriedad**.
 - ❖ Leader Election Protocol en IEEE 1394 “Firewire”
 - ❖ Binary Exponential Backoff en IEEE 802.3 “Ethernet”
- ❖ Muchas veces **no se puede establecer corrección** con una **lógica bivaluada**. Sin embargo la validez de la propiedad **puede cuantificarse probabilísticamente**.
 - ❖ Bounded Retransmission Protocol en Philips RC6
 - ❖ Binary Exponential Backoff en IEEE 802.3 “Ethernet”

Model Checking **Cuantitativo**

$?\mathcal{M} \models \phi?$

```

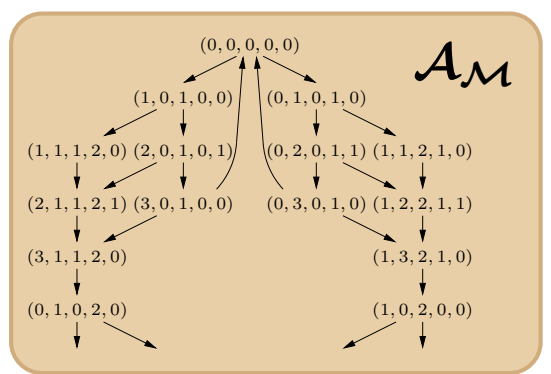
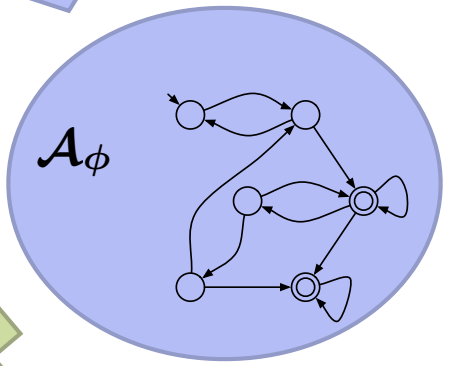
int y1 = 0;
int y2 = 0;
short in_critical = 0;

active proctype process_1() {
do
:: true ->
0: y1 = y2+1;
1: ((y2==0) || (y1<=y2));
   in_critical++;
2: in_critical--;
3: y1 = 0;
od
}

active proctype process_2() {
do
:: true ->
0: y2 = y1+1;
1: ((y1==0) || (y2<y1));
   in_critical++;
2: in_critical--;
3: y2 = 0;
od
}
    
```

\mathcal{M}

$\phi : \square \diamond crit_1 \wedge \square \diamond crit_2$



$P(\phi) > 0.95$

Incluye primitivas de aleatoriedad

Model Checking Cuantitativo

$¿\mathcal{M} \models \phi?$

```

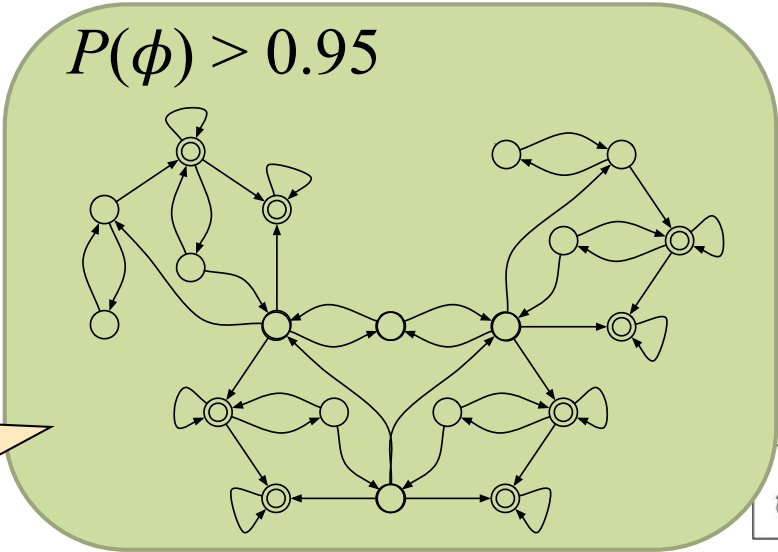
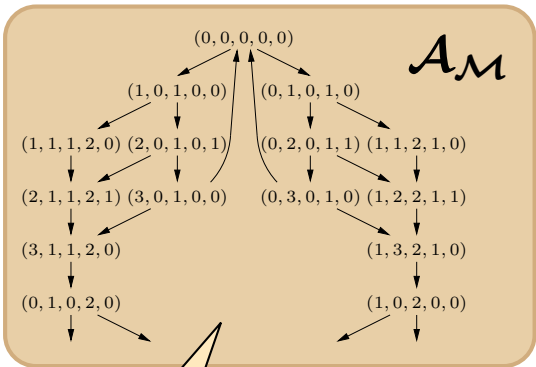
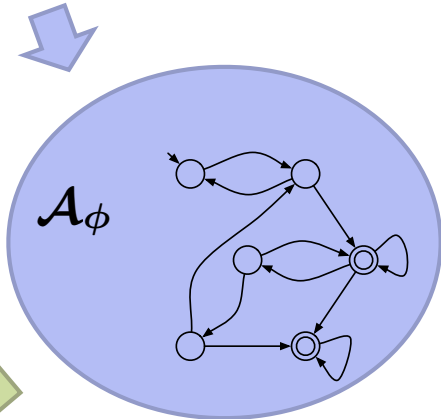
int y1 = 0;
int y2 = 0;
short in_critical = 0;

active proctype process_1() {
do
:: true ->
0:   y1 = y2+1;
1:   ((y2==0) || (y1<=y2));
    in_critical++;
2:   in_critical--;
3:   y1 = 0;
od
}

active proctype process_2() {
do
:: true ->
0:   y2 = y1+1;
1:   ((y1==0) || (y2<y1));
    in_critical++;
2:   in_critical--;
3:   y2 = 0;
od
}

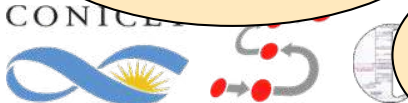
```

$\phi : \square \diamond crit_1 \wedge \square \diamond crit_2$



Proceso de decisión de Markov

Proceso de decisión de Markov



Incluye primitivas de aleatoriedad

Model Checking Cuantitativo

$¿\mathcal{M} \models \phi?$

```

int y1 = 0;
int y2 = 0;
short in_critical = 0;

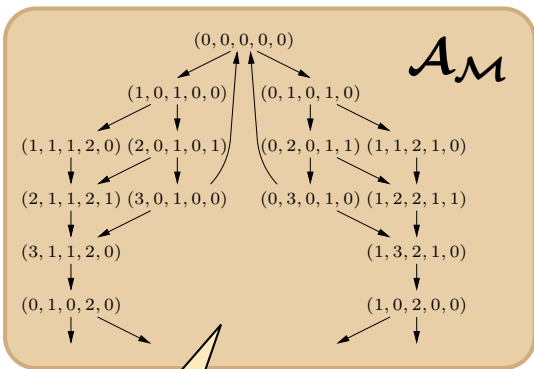
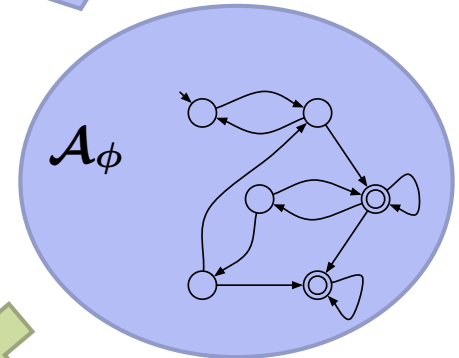
active proctype process_1() {
do
:: true ->
0: y1 = y2+1;
1: ((y2==0) || (y1<=y2));
   in_critical++;
2: in_critical--;
3: y1 = 0;
od
}

active proctype process_2() {
do
:: true ->
0: y2 = y1+1;
1: ((y1==0) || (y2<y1));
   in_critical++;
2: in_critical--;
3: y2 = 0;
od
}

```

$\phi : \square \diamond crit_1 \wedge \square \diamond crit_2$

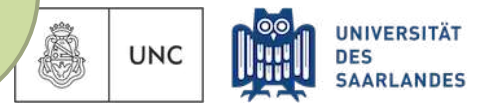
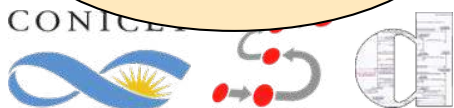
Se reduce a un problema de optimización lineal



$P(\phi) > 0.95$

$$\begin{aligned}
 x_s &= \max_{a \in A} \sum_{t \in S} P_a(s, t) \cdot x_t && \text{if } s \in Pr^{>0}(B) \setminus B \\
 x_s &= 1 && \text{if } s \in B \\
 x_s &= 0 && \text{if } s \notin Pr^{>0}(B)
 \end{aligned}$$

Proceso de decisión de Markov



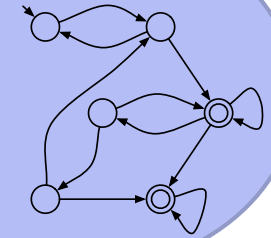
Model Checking Cuantitativo

$?\mathcal{M} \models \phi?$

$\phi : \square \diamond crit_1 \wedge \square \diamond crit_2$

Se reduce a un problema de optimización lineal

\mathcal{A}_ϕ



$P(\phi) > 0.95$

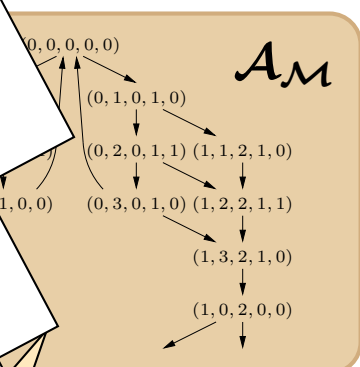
$$P_a(s, t) \cdot x_t \text{ if } s \in Pr^{>0}(B) \setminus B$$

Formalism for Hard and Softly Timed Systems
 MODEST: A Compositional Modeling
 Probabilistic Transition System Specification
 Congruence and Full Abstraction of Bisimulation
 Henrik Bohme, Jörg Hoffmann, and Frank Hees

```

0: active proctype process_2() {
  do
  :: true ->
  0: y2 = y1+1;
  1: ((y1=0) || (y2<y1));
  in_critical++;
  2: in_critical--;
  3: y2 = 0;
  } od
    
```

SOS rule formats for
 convex and abstract probabilistic bisimulation
 Axiomatizing Bisimulation Equivalences and
 Metrics from Probabilistic Transition Systems
 Pedro R. D'Argenio, Daniel C. Gorla, and Daniel Varacca



A general SOS theory for the specification of probabilistic transition systems
 R. D'Argenio, Daniel Gebler, Matias David Lee, and Daniel Varacca

Reachability Analysis of Probabilistic Systems by Successive Refinements
 Reduction and Refinement Strategies for Probabilistic Programs
 Partial Order Reduction on Concurrent Probabilistic Programs

Partial Order Reduction for Probabilistic Systems: A Revisited Approach
 Significant Diagnostic Counterexamples in Probabilistic Model Checking
 Miguel E. Andrés, Pedro D'Argenio, and Peter van Rossum

Fallas



Un sistema es **resiliente** si ...

... tiene la habilidad de **proveer y mantener un nivel de servicio aceptable** aún **bajo la presencia de fallas** y otros inconvenientes que puedan surgir y presentar un desafío al funcionamiento normal del sistema.

Cómo enfrentar las fallas

Redundancia
Redundancia
Redundancia
Redundancia
Redundancia
Redundancia

Cómo enfrentar las fallas

1. **Failover:** Varias componentes idénticas de respaldo. Cuando la componente principal falla el sistema lo detecta y cambia a una de las de respaldo.
2. **Votación:** Varias componentes idénticas activas. La información correcta se decide por votación.
3. **Detección y corrección de errores:** Redundancia de información en los datos.
4. **Reconocimientos (Acks) y timeouts:** Reconocimiento de entrega y repetición de la información sospechada perdida.

Eventos

Los eventos pueden cuantificarse probabilísticamente

Ejemplos:

- ❖ Probabilidad de pérdida de un mensaje
- ❖ Tiempo esperado de vida de una fuente de alimentación
- ❖ Tiempo esperado de reparación del disco rígido
- ❖ Tiempo esperado de transmisión tierra-satélite
- ❖ Probabilidad de alteración de un bit bajo radiación
- ❖ Tiempo esperado de refrescado de memoria

(Algunas) Técnicas de análisis

- ❖ Model checking cuantitativo

Ya lo vimos

- ❖ Simulación por eventos discretos

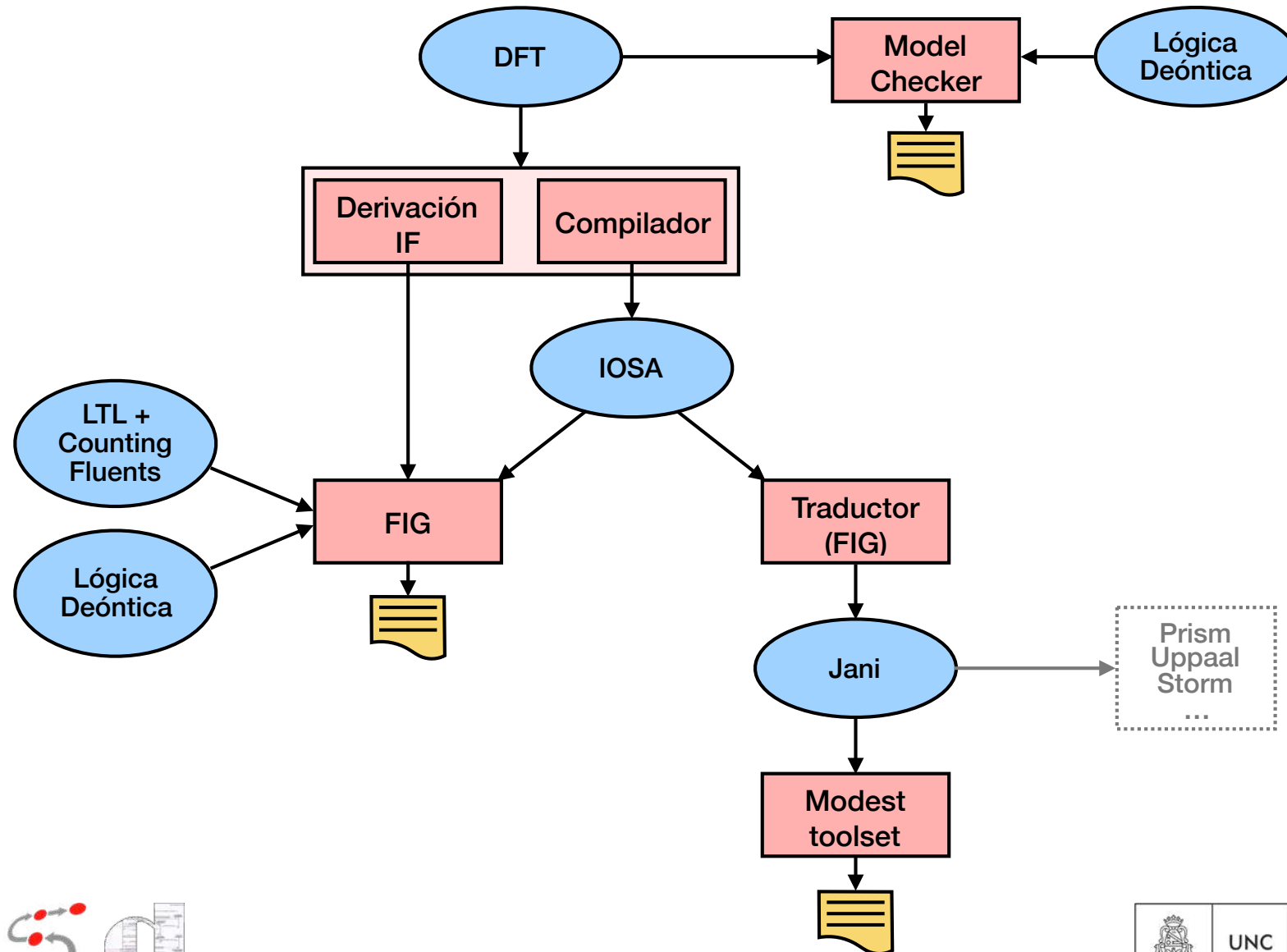
En particular nos interesa la simulación de eventos raros

- ❖ Model checking estadístico

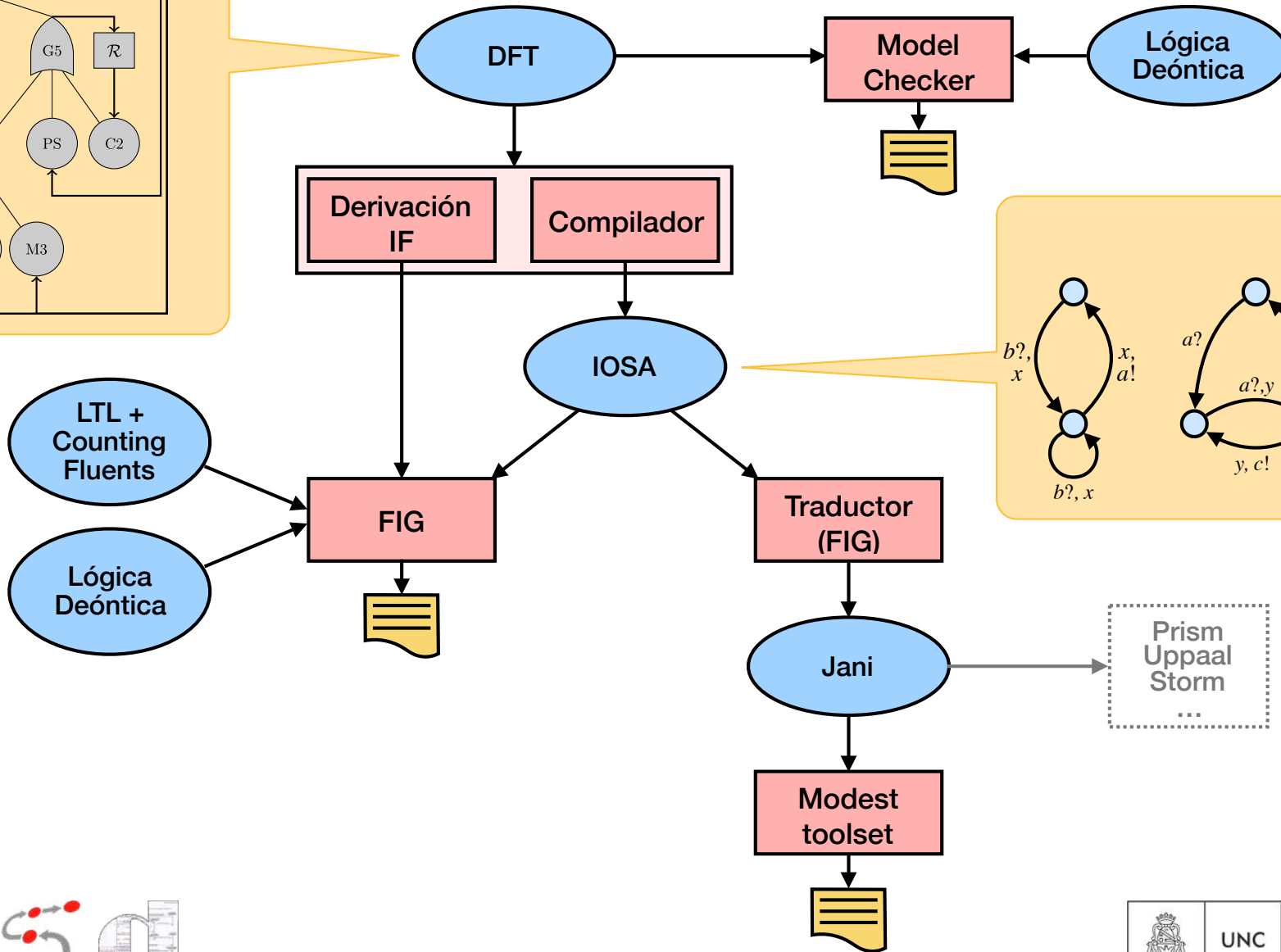
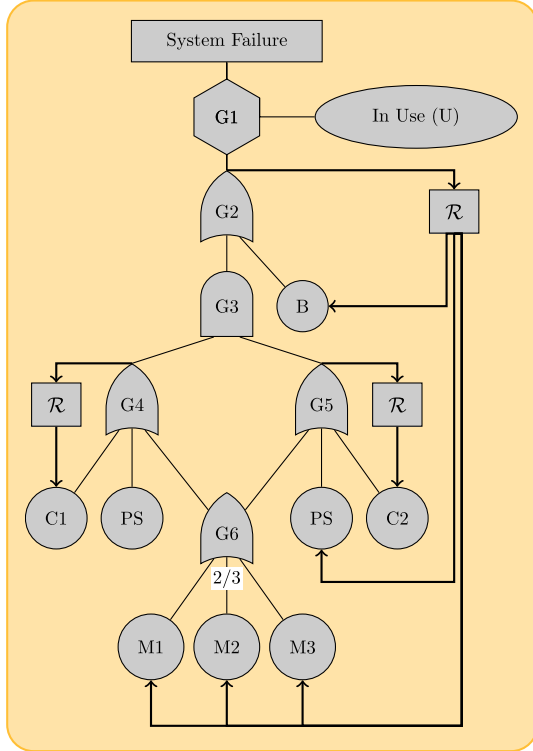
Es una variante específica de la simulación

*es decir,
de muy baja probabilidad
+ no-determinismo*

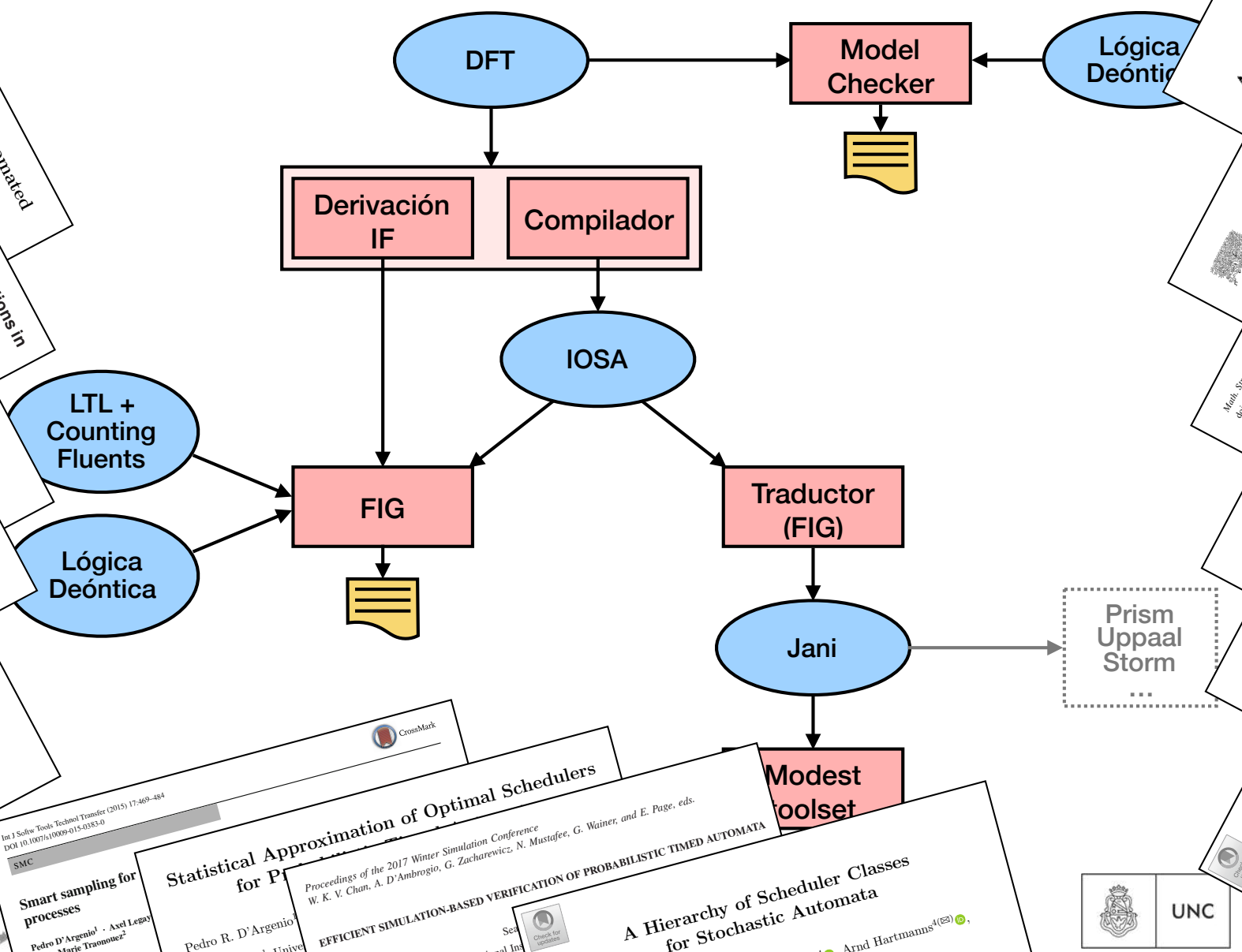
Proyectos RAFTSys y ARES



Proyectos RAFTSys y ARES



Proyectos RAFTSys y ARES



Rare Event Simulation with Fully Automated Importance Splitting

Compositional Construction of Importance Splitting Fully Automated

The Road from Stochastic Automata to Better Automated Importance Splitting for Nondeterministic Rare Events

A Statistical Model Checker for Nondeterminism and Rare Events

Pedro R. D'Argenio, Arnd Hartmanns

An Algebraic Approach to the Logic of Stochastic Automata

Information and Computation

Available online at www.sciencedirect.com

Science @ Elsevier

Information and Computation 2011, 20(1), 1-28

© Cambridge University Press 2011

First published online 26 September 2011

Bisimulations for non-deterministic Markov Decision Processes

A Theory for the Semantics of Stochastic and Non-deterministic Continuous Systems

Input/Output Stochastic Automata with Urgency: Confluence and Determinism

Compositional and Determinism

Input/Output Stochastic Automata with Urgency: Confluence and Determinism

Pedro R. D'Argenio, Arnd Hartmanns

1 Universidad Nacional de Córdoba

2 Universidad de Buenos Aires

3 Saarland University



Statistical Approximation of Optimal Schedulers for P

Proceedings of the 2017 Winter Simulation Conference

W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

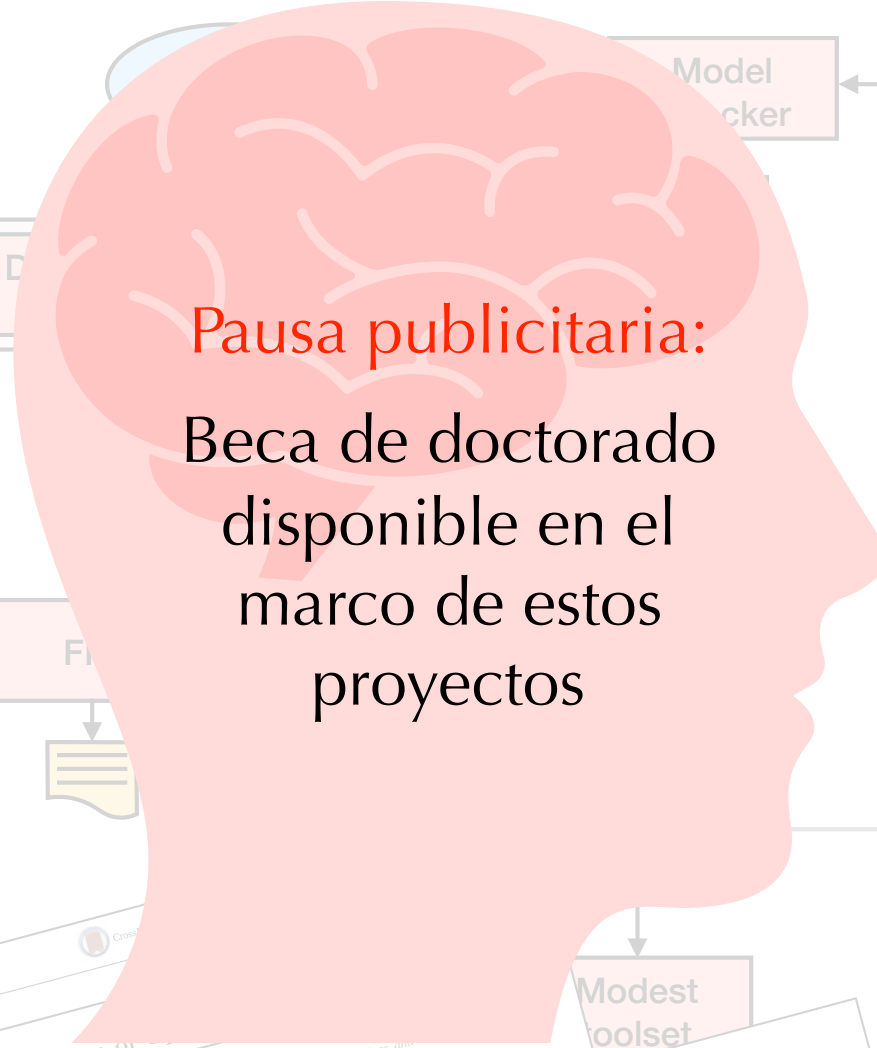
EFFICIENT SIMULATION-BASED VERIFICATION OF PROBABILISTIC TIMED AUTOMATA

A Hierarchy of Scheduler Classes for Stochastic Automata

Marcus Gerhold, Arnd Hartmanns



Proyectos RAFTSys y ARES



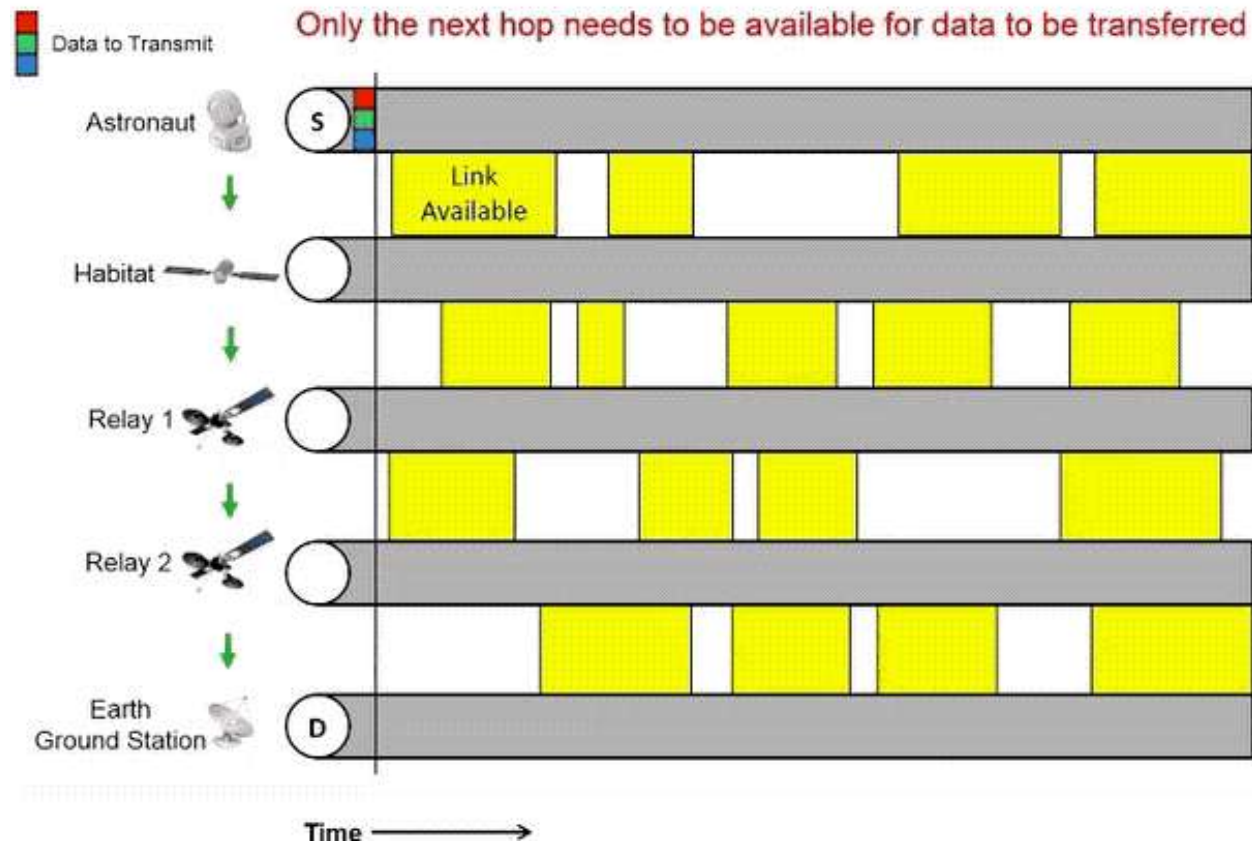
Pausa publicitaria:
Beca de doctorado
disponible en el
marco de estos
proyectos



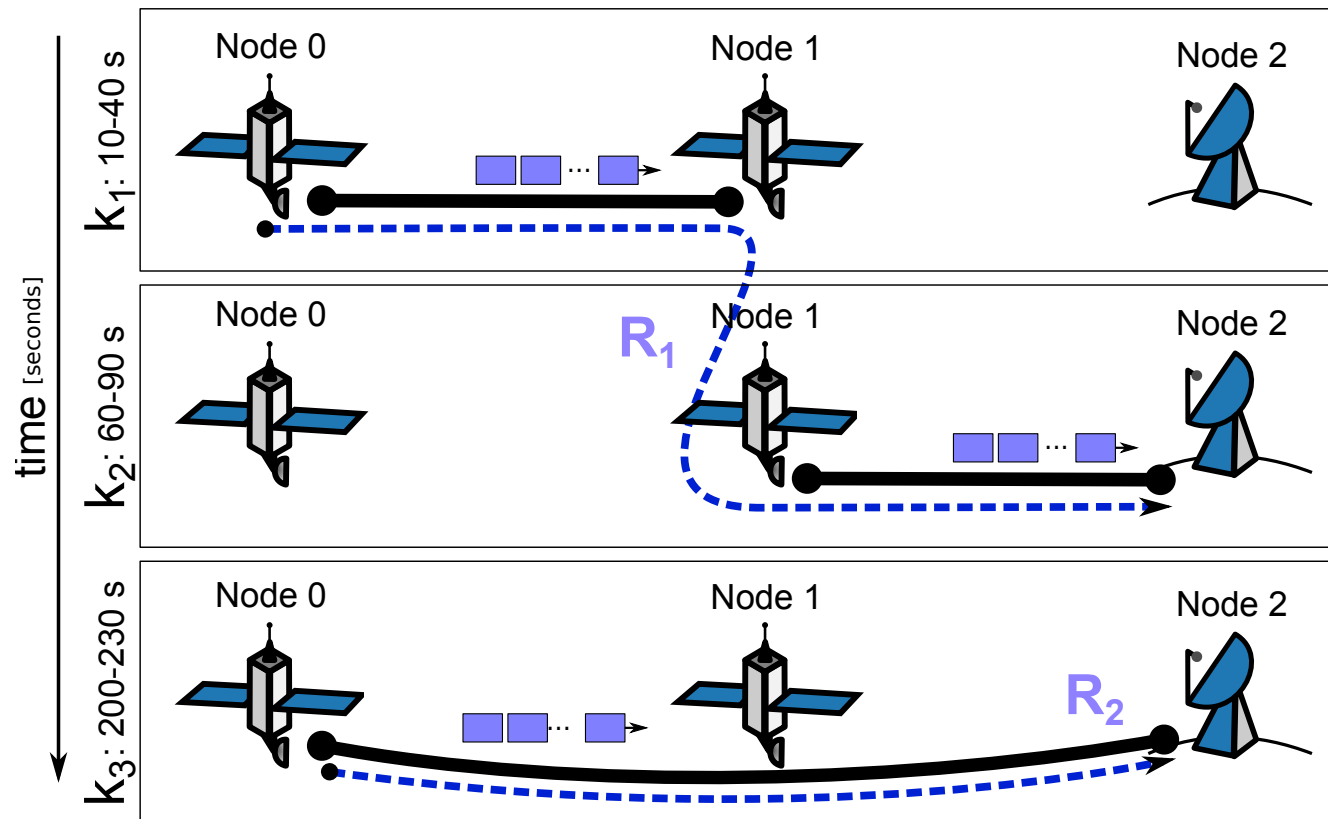
dargenio@famaf.unc.edu.ar

Redes Tolerantes a Demora

Sample Scenario Using DTN-Capable Nodes

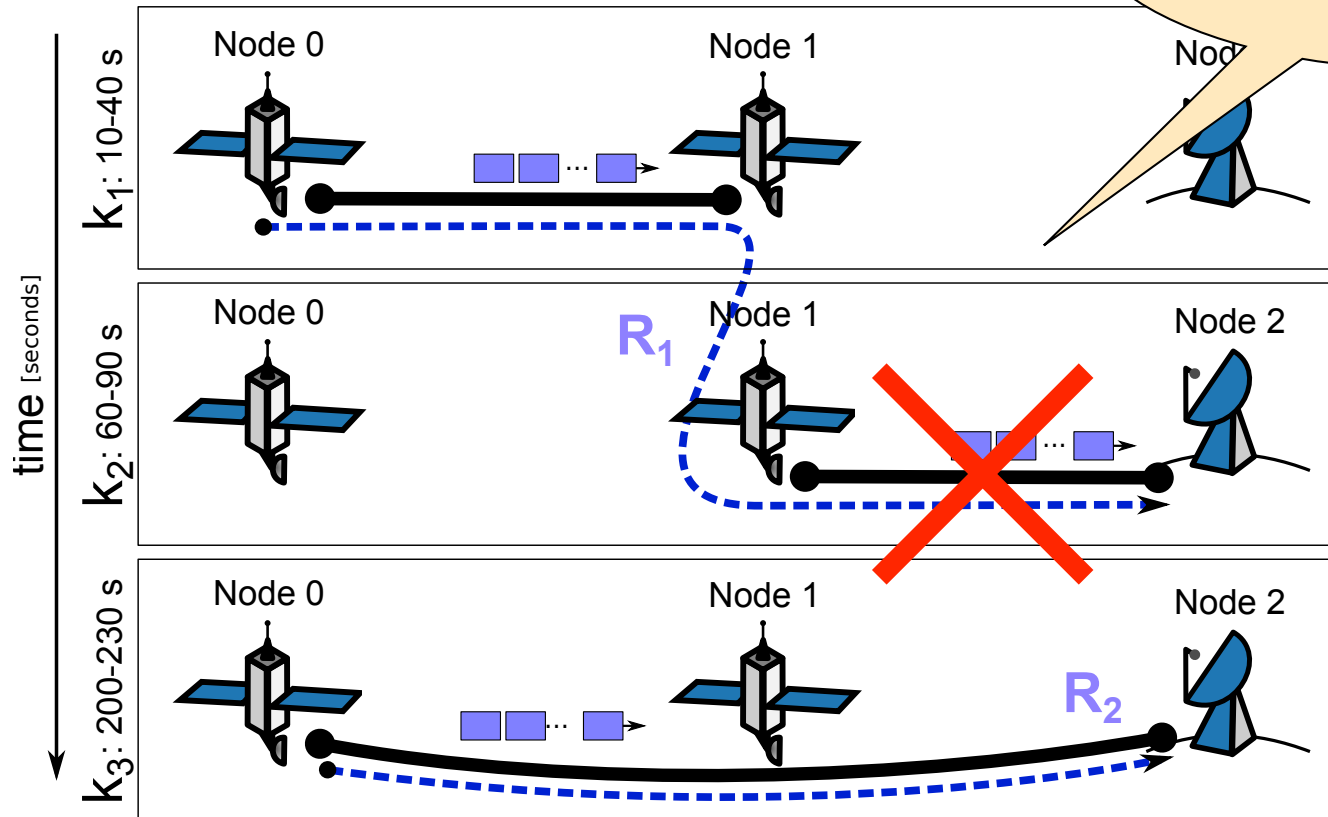


Redes Tolerantes a Demora

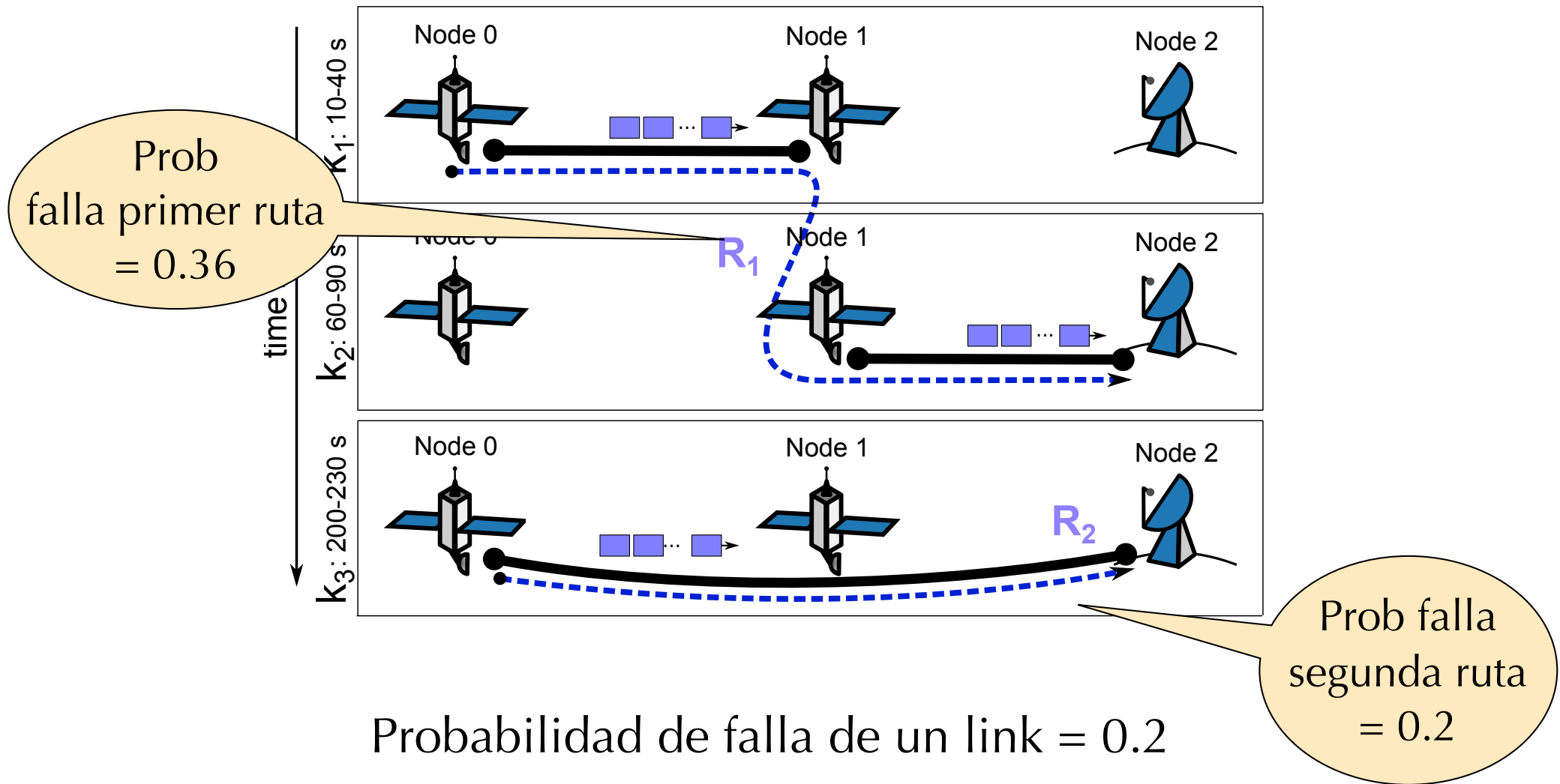


Redes Tolerantes a Demoras

El mensaje no puede ser transmitido

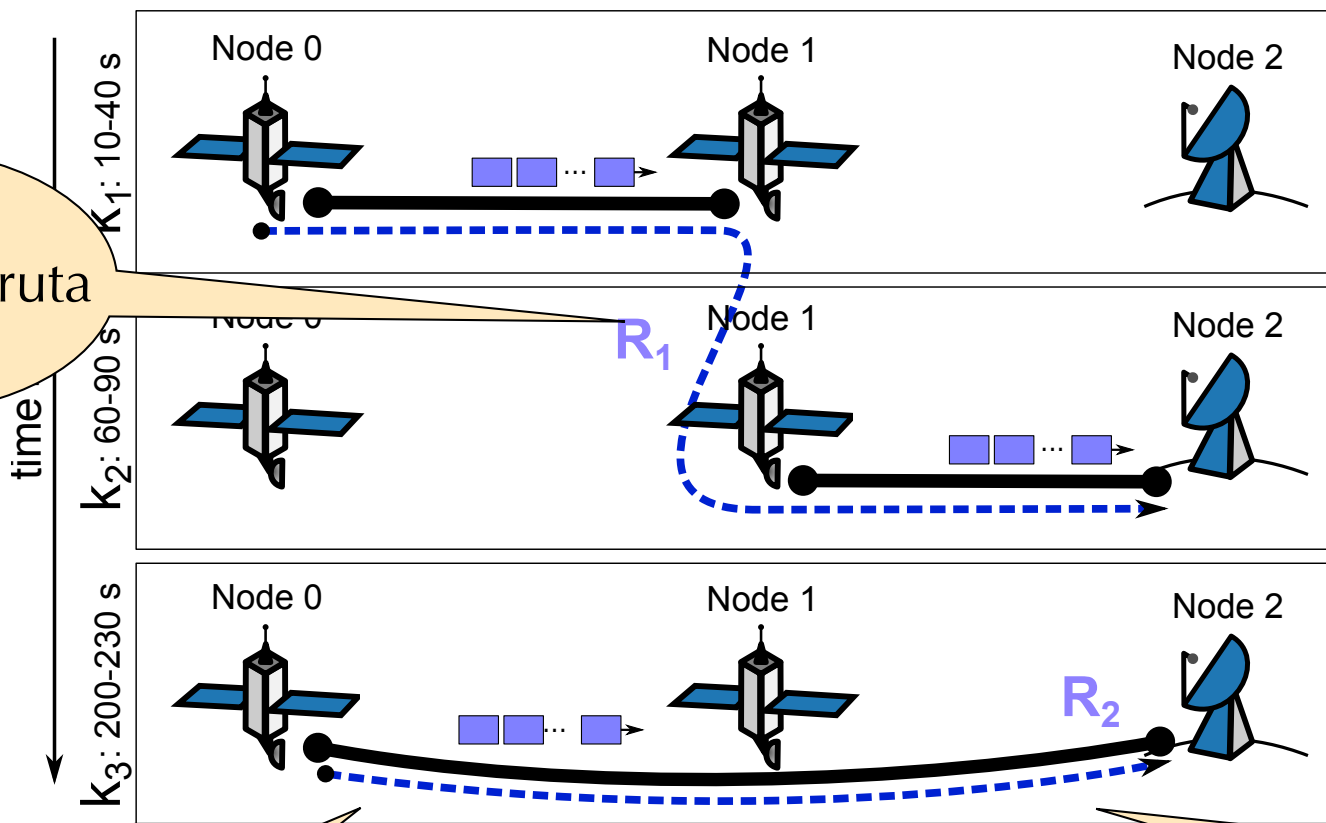


Redes Tolerantes a Demora



Redes Tolerantes a Demora

Prob
falla primer ruta
= 0.36



Prob falla
segunda ruta

Síntesis de ruteo
usando Model Checking
Cuantitativo

Probabilidad de falla de un link = 0.2

A Markov Decision Process for Routing in Space
DTNs with Uncertain Contact Plans
Fernando D. Raverta^{*†}, Ramiro Demasi^{**‡}, Pablo G. Madoery^{*†}, Juan A. Fraire^{*‡§},
Jorge M. Finochietto^{*†}, Pedro R. D'Argenio^{*‡§}
*CONICET, Córdoba, Argentina
†Instituto de Investigaciones Científicas y Técnicas (FCEFN), UNC, Córdoba, Argentina
‡Instituto de Física y Computación (FAMAF), UNC, Córdoba, Argentina
§Saarland University, Saarbrücken, Germany

Fallutadas abusivas



Estrategias de
“lock in”

Fallutadas famosas



Fallutadas famosas



International Business Times

Economy | Companies | Markets | Finance | Regulation

Business | Companies

VW scandal: Carmaker was warned by Bosch about test-rigging software in 2007

By Karthick Arvinth
Updated September 28, 2015 05:53 BST

Fallutadas famosas

Home > Lifestyle > Cars > News
Diesel emissions scandal: Fiat under investigation



International Business Times

Economy | Companies | Markets | Finance | Regulation
Business | Companies

VW scandal: Carmaker was warned by Bosch about test-rigging software in 2007



By Karthick Arvinth

Updated September 28, 2015 05:53 BST



Fallutadas famosas

Home > Lifestyle > Cars > News
Diesel emissions scandal: Fiat under investigation



Germany orders Porsche recall over diesel emissions cheating

Economic May 18, 2018

Business | Companies
International Business Times
Markets | Finance | Regulation

VW scandal: Carmaker was warned by Bosch about test-rigging software in 2007

By Karthick Arvinth
Updated September 28, 2015 05:53 BST



Fallutadas famosas

Home > Lifestyle > Cars > News

Diesel emission investigation

Scandal: Fiat under

Nissan found guilty of using diesel emissions cheat device in South Korea

Nissan has denied any wrongdoing, but the South Korean government has ruled the Renault-

Germany orders Porsche recall over diesel

Economic May 18, 2018

Business | Companies | Markets | Finance | Regulation
International Business Times

VW scandal: Carmaker was warned by Bosch about test-rigging software in 2007

By Karthick Arvinth

Updated September 28, 2015 05:53 BST



Fallutadas famosas

Home > Lifestyle > Cars > News

Diesel emission investigation

Scandal: Fiat under

Nissan found guilty of using diesel emissions cheat device in South Korea

Nissan has denied any wrongdoing, but the South Korean government has ruled the Renault-

Germany orders Porsche recall over diesel

Economic May 18, 2018

Business | Companies

International Business Times

VW scandal: Carnage about test-

Daimler forced to recall Mercedes with defeat devices



By Karthick Arvir

Updated September

11 June 2018

Diesel emissions scandal



sch



Fallutada

Peugeot suspected of fraud in diesel scandal

France
PSA Peugeot Citroën | French economy | auto industry

Share 1
Tweet
Share

in Share

Diesel emission investigation

Nissan found guilty of using diesel emission cheat device in South Korea

Nissan has denied any wrongdoing, but the South Korean government has ruled the Renault-

Germany orders Porsche recall over diesel

Economic May 18, 2018

Business Times
Business | Companies | Markets | Finance | Regulation

VW scandal: Carnage about test-

Daimler forced to recall Mercedes with defeat devices

By Karthick Arvir
Updated September

11 June 2018

Diesel emissions scandal

f
Share



Fallutada

France Peugeot suspected of fraud in diesel scandal

PSA Peugeot Citroën | French economy | auto industry

Share 1

Tweet

Share

Share

Diesel emission investigation

Nissan found guilty of using diesel emission cheat device in South Korea

Nissan has denied any wrongdoing, but the South Korean government has ruled the Renault-

Germany orders Porsche recall over diesel

Economic May 18, 2018

Business | Companies Markets | Finance | Regulation

VW scandal: Car maker to recall Mercedes with defeat about test

By Karthick Arvir Updated September

Renault 'cheated on 25 years of pollution tests'

11 June 2018

Diesel emissions scandal

Fallutada

France Peugeot suspected of fraud in diesel scandal

PSA Peugeot Citroën | French economy | auto industry

Share 1

Tweet

Share

in Share

Diesel emission investigation

Nissan found guilty of using diesel emission cheat device in South Korea

Nissan has denied any wrongdoing, but the South Korean government has ruled...

Germany orders Porsche recall over diesel

Economic May 18, 2018

Citroën may have breached emissions rules: report

A model tested by the European Commission recorded pollution levels more than seven times higher than labeled

Mercedes to recall Mercedes with defeat

VW scandal: Car...

Renault 'cheated on 25 years of pollution tests'



By Karthick Arvir Updated September

Dai de

© 11 June 2018

Diesel emissions scandal

Fallutada

France
PSA Peugeot Citroën | French economy | auto industry
Peugeot suspected of fraud in diesel scandal
Share 1 Tweet Share

Lifestyle Cars News
Diesel emission investigation

Scandal: Fiat und
Nissan found guilty of cheat device in S
Nissan has denied any wrongdoing, but the South Korean government has ruled

GM Accused of Cheating on Diesel Emissions

Germany orders Porsche recall over diesel
Econ May 18, 2018

Business | Companies
Citroën may have breached emissions rules: report
A model tested by the European Commission recorded pollution levels more than seven times higher than labeled
to recall Mercedes with defeat

VW scandal: Car
about test

Renault 'cheated on 25 years of pollution tests'
Share

By Karthick Arvi
Updated September

© 11 June 2018
Diesel emissions scandal

Fallutadas imperdonables



<https://www.mintpressnews.com/214505-2/214505/>

Software doping

❖ Es un problema **ético** y hasta **legal**.

❖ Un software está “**dopado**” si...

... el fabricante incluyó una **funcionalidad oculta** de manera tal que el comportamiento resultante **favorezca intencionalmente a una parte previamente designada**, en contra de los intereses de la sociedad o el licenciataria del software

Software doping

❖ Es un problema **ético** y hasta **legal**.

Pero ... voy a proponer una **solución técnica** (formal)

❖ Un software está “**dopado**” si...

... el fabricante incluyó una **funcionalidad oculta** de manera tal que el comportamiento resultante **favorezca intencionalmente a una parte previamente designada**, en contra de los intereses de la sociedad o el licenciatarario del software

No es posible de formalizar

Software doping

definición formal

- ❖ Memoria: $\mu : \text{Variables} \rightarrow \text{Valores}$
- ❖ Un programa es un transformador de memoria: $(S, \mu) \Downarrow \mu'$
- ❖ Variables:
Entrada de interés: $i \in \text{Variables}$ Salida de interés: $o \in \text{Variables}$
- ❖ Software doping

S *no está dopado* si para todas μ_1, μ_2, μ'_1 y μ'_2 ,

$$\left. \begin{array}{l} \mu_1(i) \approx \mu_2(i) \\ (S, \mu_1) \Downarrow \mu'_1 \\ (S, \mu_2) \Downarrow \mu'_2 \end{array} \right\} \Rightarrow \mu'_1(o) \approx \mu'_2(o)$$

“se parece”

Self-composition otra vez!!

$$\{i \approx i'\} S ; S[\vec{x}/\vec{x}'] \{o \approx o'\}$$

Self-composition otra vez!!

$$\{i \approx i'\} S ; S[\vec{x}/\vec{x}'] \{o \approx o'\}$$

Facets of Software Doping
Gilles Barthe¹, Pedro R. D'Argenio²(✉), Bernd Finkbeiner³, and
¹ IMDEA
² FaMAF, Universidad Nacional de Córdoba
³ Saarland University – Computer Science

Is Your Software on Dope?
Formal Analysis of Surreptitiously “enhanced” Programs
Pedro R. D'Argenio^{1,2}(✉), Gilles Barthe¹, and Bernd Finkbeiner², and
¹ FaMAF, Universidad Nacional de Córdoba
² Computer Science, Saarland Informatics Campus
³ IMDEA Software Engineering

Cyber-Physical Doping Tests
2018 3rd Workshop on Monitoring and Testing of Cyber-Physical Systems
Sebastian Biewer, Holger Hermanns
Saarland University – Computer Science
Saarland Informatics Campus
Saarbrücken, Germany

Verification, Testing, and Runtime Monitoring of Automotive Exhaust Emissions
Holger Hermanns¹, Sebastian Biewer¹, Pedro R. D'Argenio^{2,3,1}, and Maximilian A. Köhl¹
¹ Saarland Informatics Campus, Germany
² FaMAF, Universidad Nacional de Córdoba, Argentina
³ IMDEA Software Engineering

EPiC Series in Computing
Volume 57, 2018, Pages 1–17
LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning

CO

TÄT
ES

¿Qué conclusión sacan ustedes?

Las que saco yo:

- ❖ No está bueno echarse moco
 - ➔ hacer lo posible por evitarlos y eliminarlos
- ❖ Las fallas inevitablemente ocurren
 - ➔ tratar efectivamente con ellas y de manera eficiente
- ❖ Las fallutadas son una mala costumbre de la disciplina
 - ➔ No sólo contrarrestarlas ética y legalmente sino también técnicamente

Las técnicas formales
(matemáticas) son cruciales
para todo esto

Epílogo perturbador



Preferencias de Materias Optativas (estudiantes)

Departamento de Computación
FCEyN, UBA

Introducción al Procesamiento del Lenguaje Natural	29
Programación de Sistemas Operativos	29
Aprendizaje Automático	27
Ciencia de Datos	25
Aprendizaje Profundo / Redes Neuronales Profundas	20
Programación Concurrente	20
Algoritmos y Estructuras de Datos Avanzadas / Problemas de Grafos y Tratabilidad Computacional	19
Seguridad de la Información	19
Arquitectura de Aplicaciones Web	17
Programación Orientada a Objetos/Diseño Avanzado de Objetos	17
Computación Móvil	13
Investigación Operativa	12
Generación Automática de Casos de Test	10
Problemas, Algoritmos y Programación	10
Redes Neuronales	9
Redes, Sociedad y Economía	9
Inferencia Bayesiana	8
Simulación de Eventos Discretos	8
Computación, Ciencia y Sociedad en la Argentina	7
Computación Gráfica	7
Introducción a la Robótica Móvil	7
Metaheurísticas	7

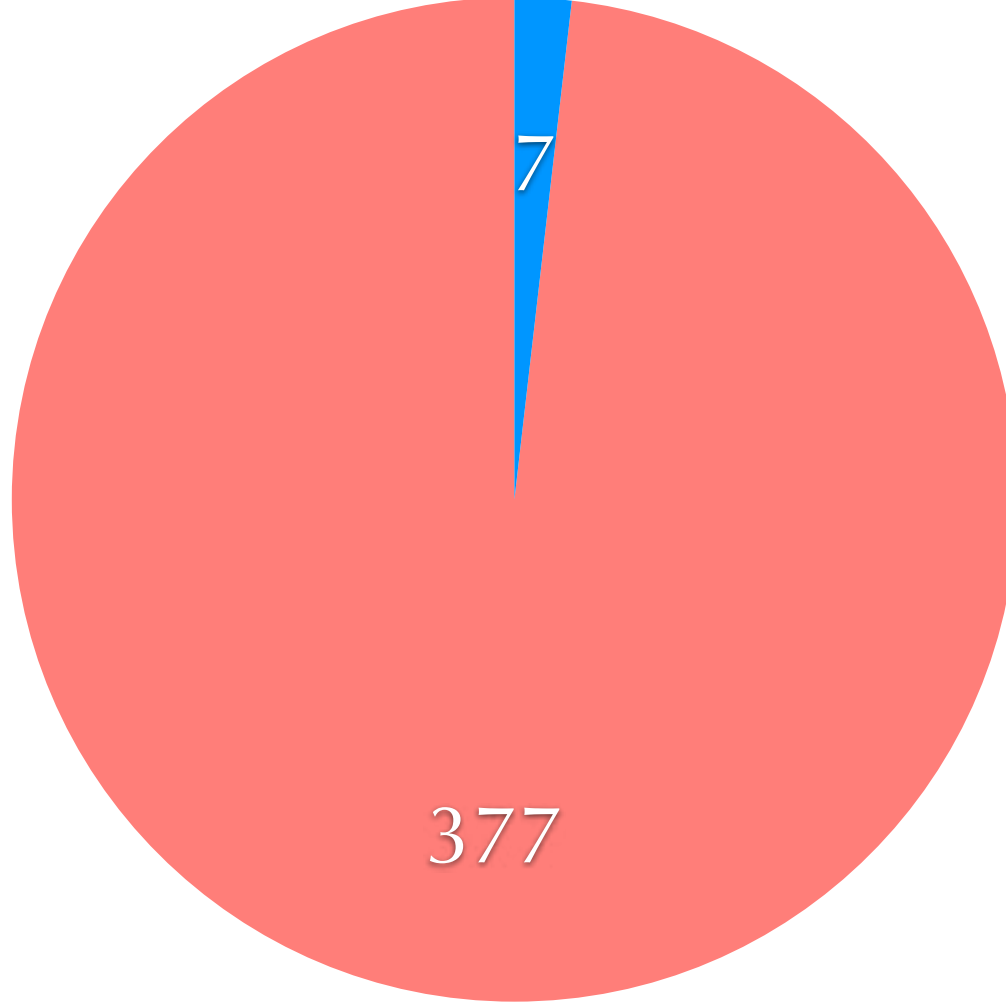
Preferencias de Materias Optativas (estudiantes)

Departamento de Computación FCEyN, UBA

Simulación de Eventos Discretos	8
Computación, Ciencia y Sociedad en la Argentina	7
Computación Gráfica	7
Introducción a la Robótica Móvil	7
Metaheurísticas	7
Introducción al Procesamiento Digital de Imágenes	6
Reconocimiento de Patrones	6
Reescritura (Cálculo Lambda)	6
Teoría de la computabilidad	6
Arquitectura y Comunicación de Datos	4
Visión en Robótica	4
Introducción al Análisis Formal de Normas Legales	3
Seminario sobre Algoritmos de Análisis de Secuencias Biológicas	3
Sistemas Complejos	3
Validación y Verificación de Programas	3
Visión por Computadora	3
Seminario Avanzado de Programación Lineal Entera -	2
Análisis y Síntesis Automática de Programas	1
Fundamentos de Especificación de Software	1
Reglas de Asociación y Patrones Secuenciales -	1
Seminario Avanzado de Análisis de Programas	1
Seminario Avanzado sobre Modelos y Algoritmos para el Análisis de Sistemas	1
Integración de Bases de Conocimiento	1

Preferencias de Materias Optativas
(estudiantes)
Departamento de Computación
FCEyN, UBA

Sólo el 1.8% pondera la importancia de la corrección y la confiabilidad



Preferencias de Materias Optativas
(estudiantes)
Departamento de Computación
FCEyN, UBA

Sólo el 1.8% pondera la importancia de la corrección y la confiabilidad

