Mocos, Fallas y Fallutadas

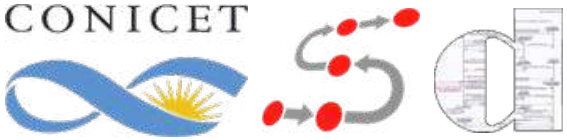
Todo lo que no nos gusta ver en el software

Pedro R. D'Argenio

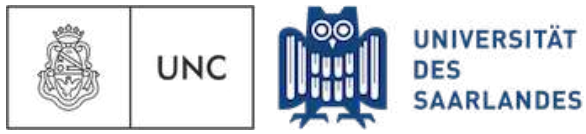
Universidad Nacional de Córdoba – CONICET (AR)

Saarland University (DE)

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



LCC 2018 - Rosario



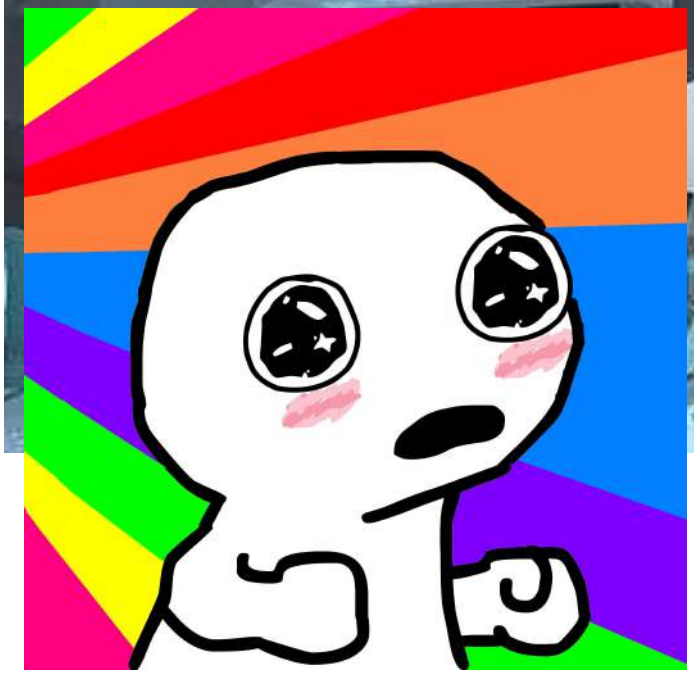
El software parece hacer maravillas…

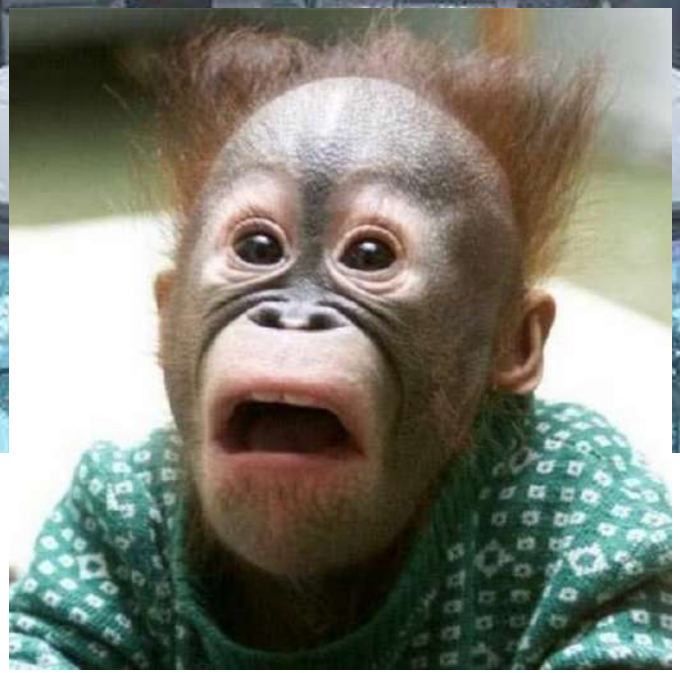
CONICET

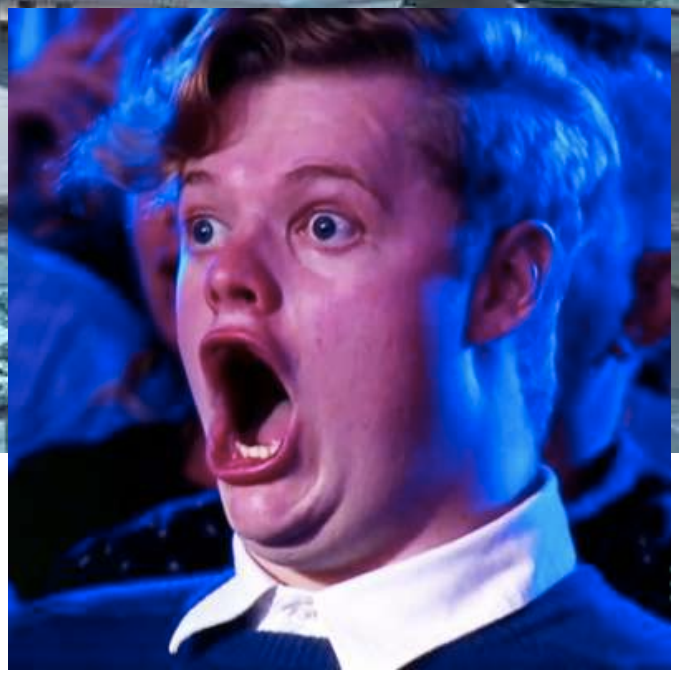
UNC

UNIVERSITÄT DES SAARLANDES

El software parece hacer maravillas…







CONICET

UNC

UNIVERSITÄT DES SAARLANDES

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

CONICET

UNC

UNIVERSITÄT DES SAARLANDES

¿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

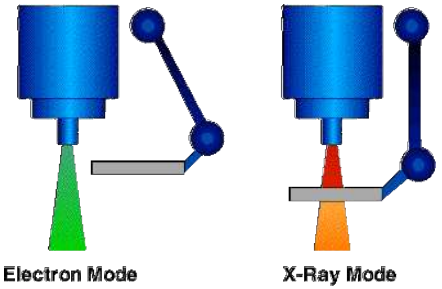
- Fallas externas al software pero que son parte del sistema

- Programación malintencionada

Mocos famosos

Pentium:
FDIV

Ariane 5:
64 bits fp
vs 16 bits int



Therac-25:
Condición de
carrera

Mars Climate
Orbiter:
Métrico vs Imperial

Northeast blackout
en 2003:
Condición de
carrera

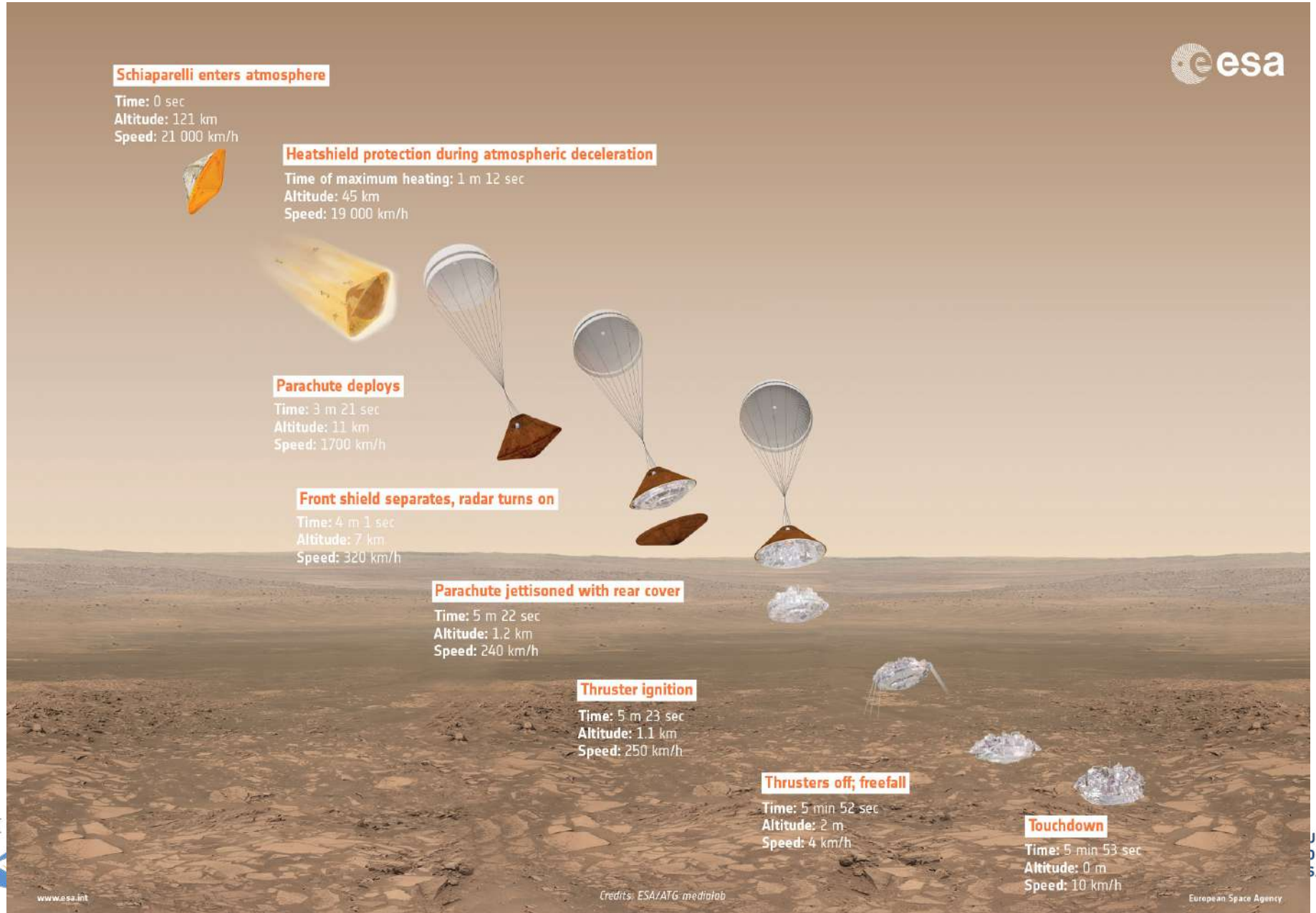
Heartbleed:
Integridad/Confidencialidad

CONICET

UNC

UNIVERSITÄT
DES
SAARLANDES

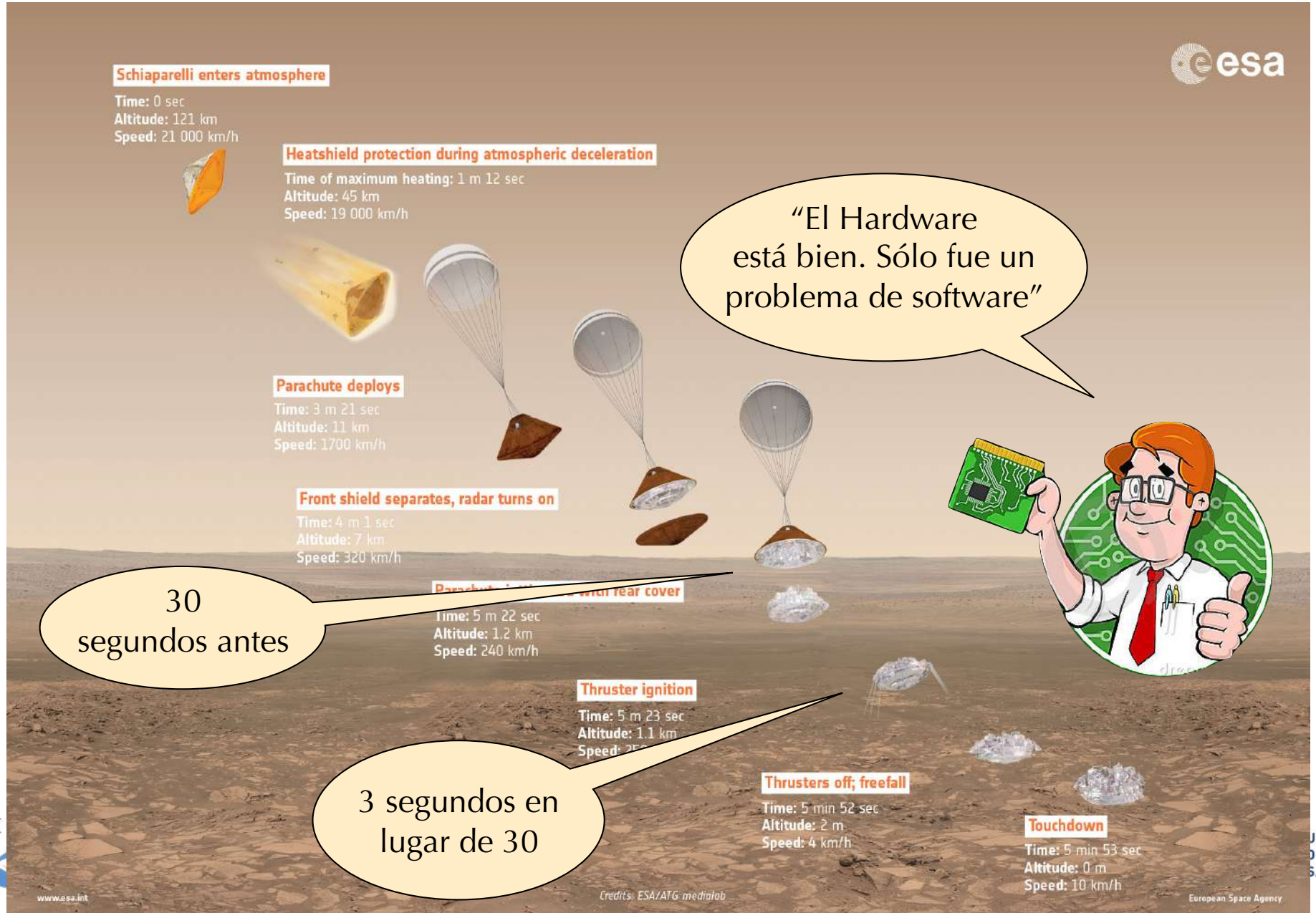
El problema de llamarlo *Bug*



CONI

UNIVERSITÄT DES SAARLANDES

El problema de llamarlo *Bug*

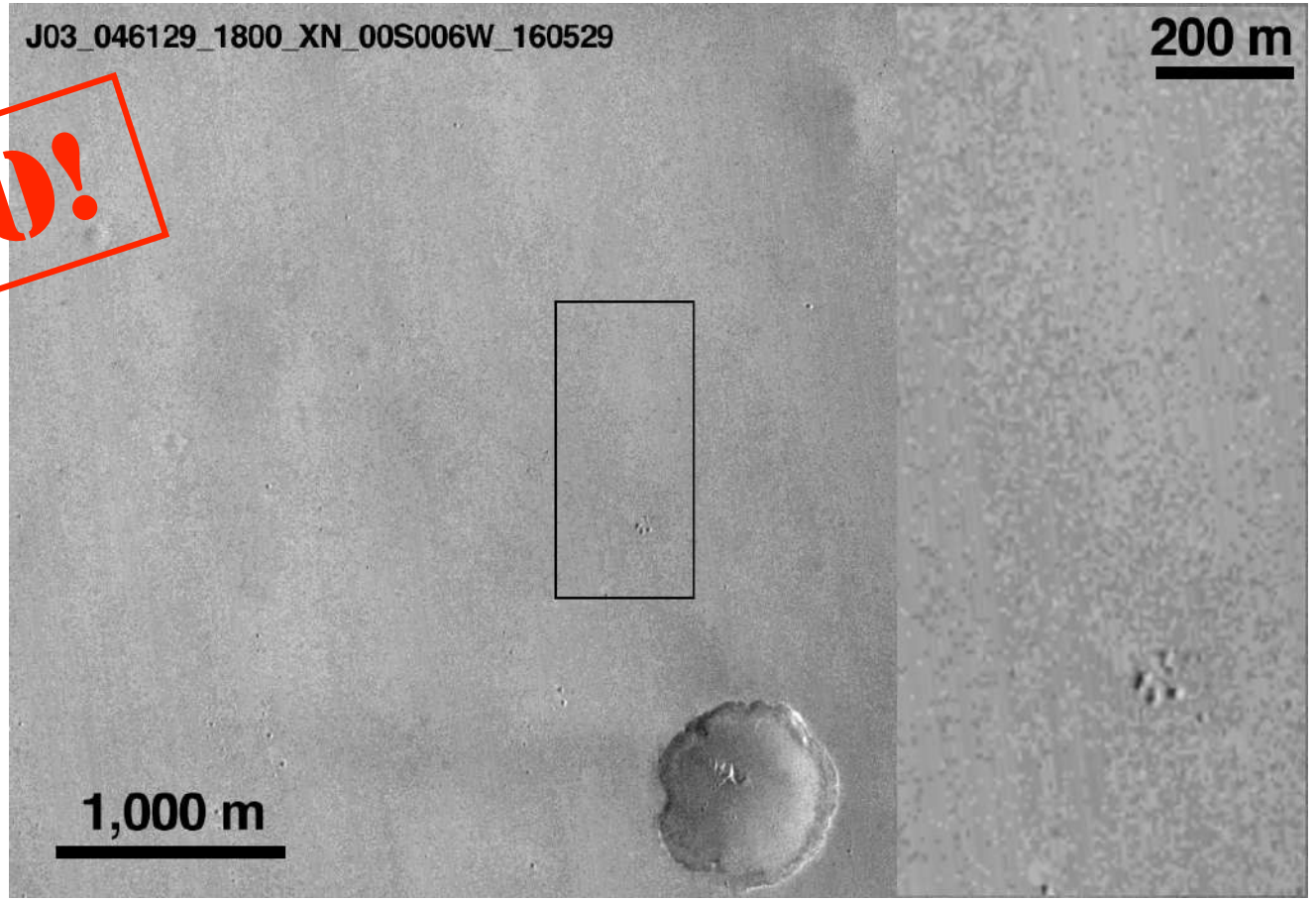


CONI

UNIVERSITÄT DES SAARLANDES

El problema de llam

El problema de llamarlo *Bug*



MOCO!

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

CONICET

UNC

UNIVERSITÄT DES SAARLANDES

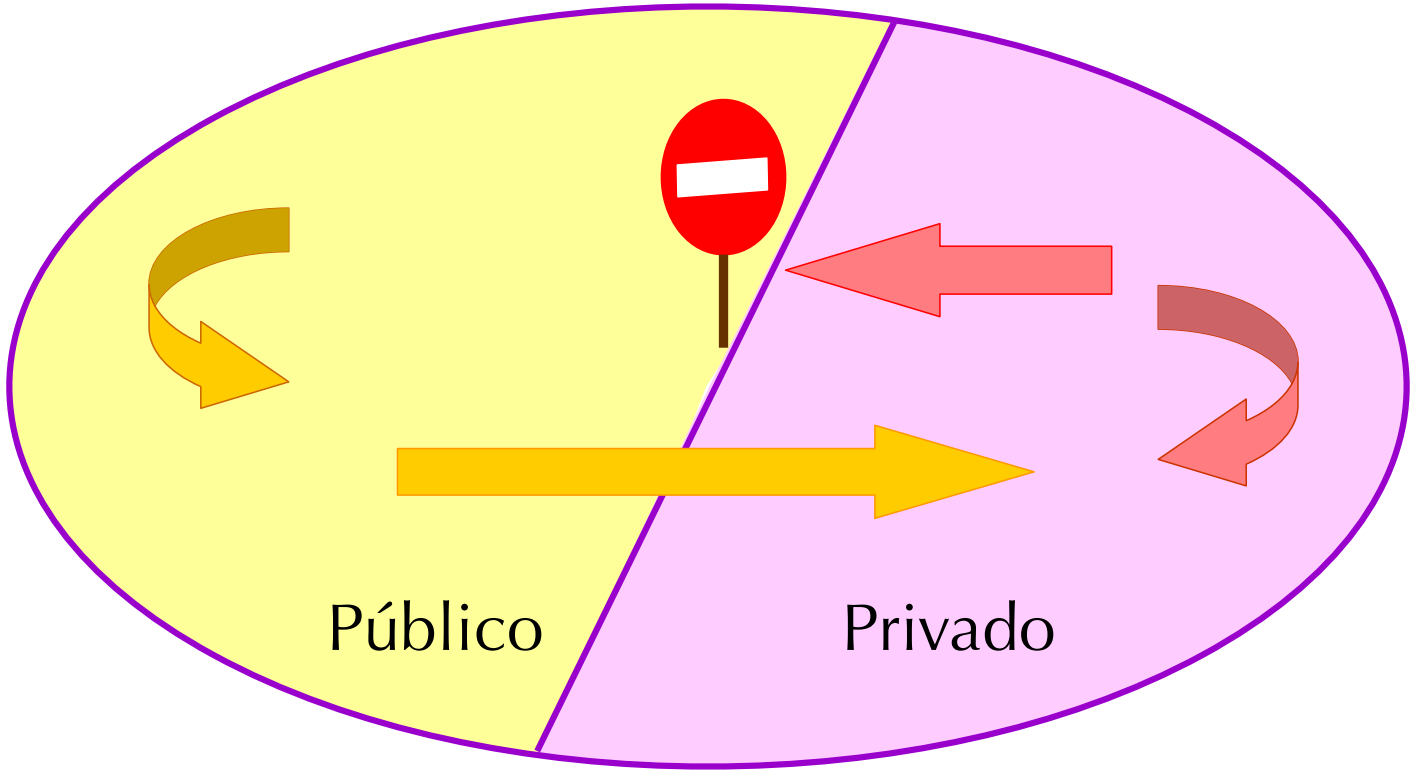
El problema de la corrección

$$Sistema \models Propiedad$$

Usualmente una abstracción que describe su comportamiento

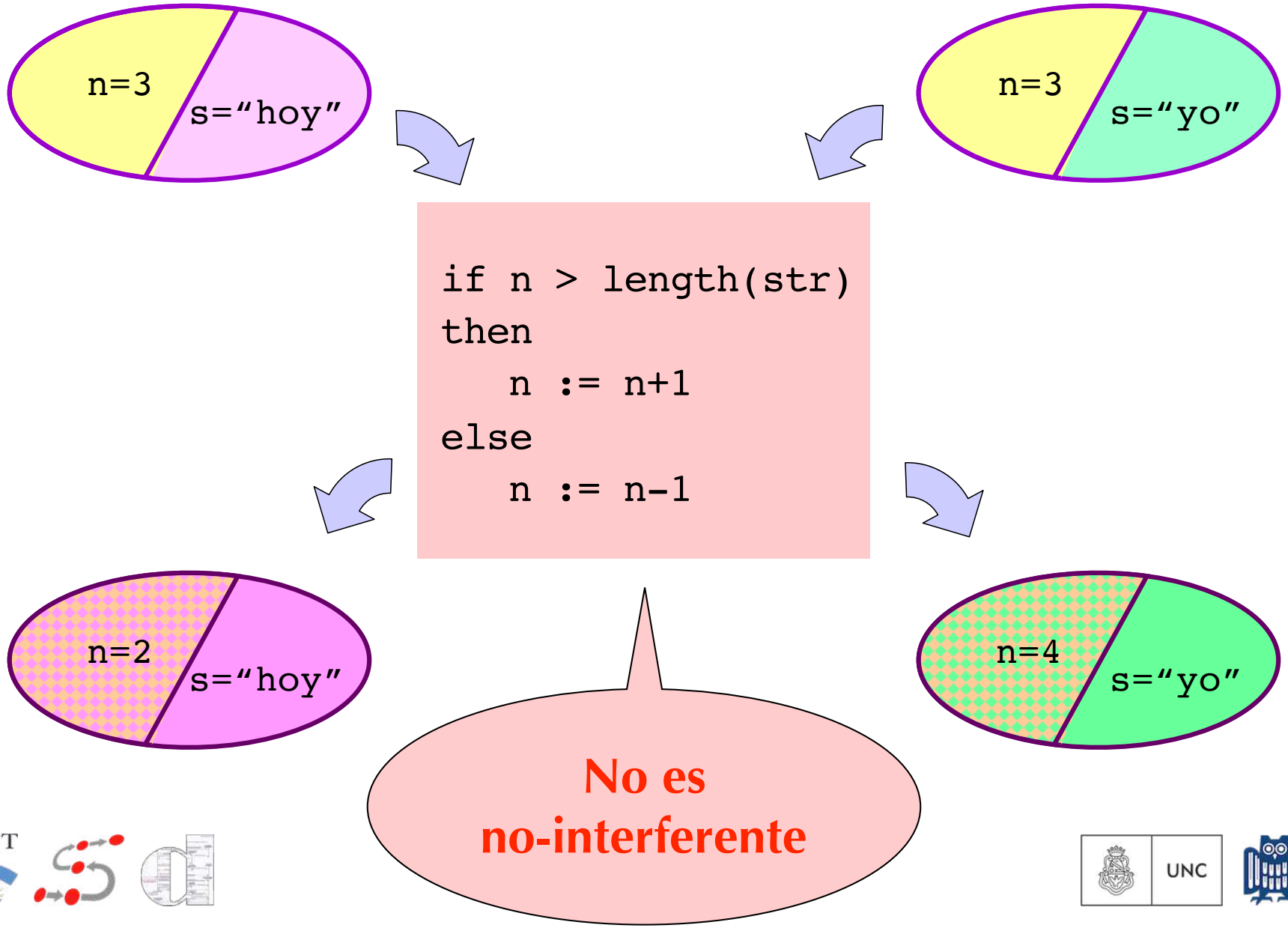
Describe lo que se espera del sistema (el criterio de corrección)

Ejemplo: Confidencialidad

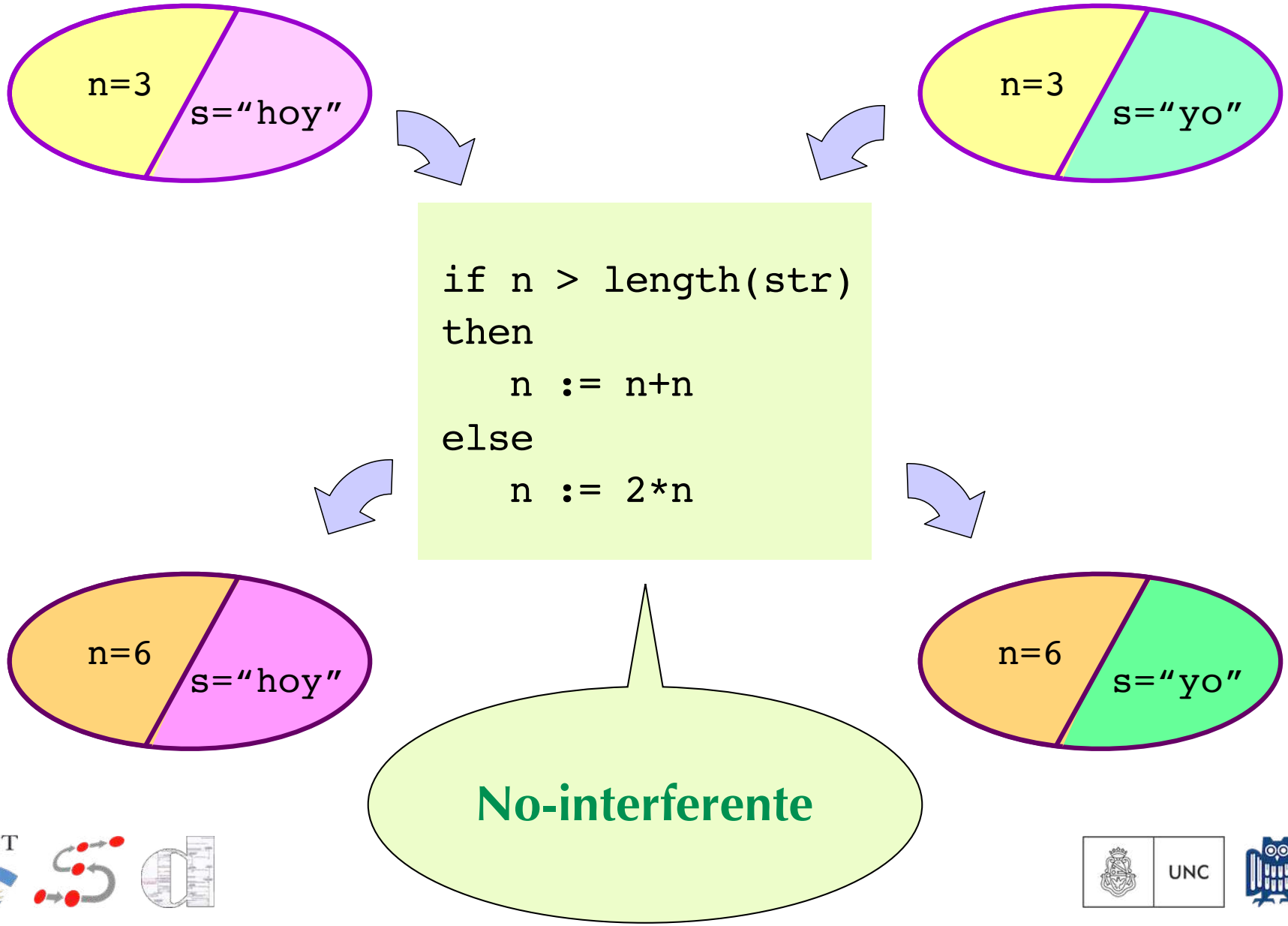


Espacio de datos

No-interferencia



No-interferencia



No-interferencia

definición formal

- Memoria: $\mu : \mathit{Variables} \to \mathit{Valores}$

Estado inicial

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

Estado final

No-interferencia

definición formal

- Memoria: $\mu : \mathit{Variables} \to \mathit{Valores}$
- Un programa es un transformador de memoria: $(S, \mu) \Downarrow \mu'$
- Variables:

  Públicas: $l \in \mathit{Variables}$ Privadas: $h \in \mathit{Variables}$

- No-interferencia

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

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

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,

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

CONICET

UNC

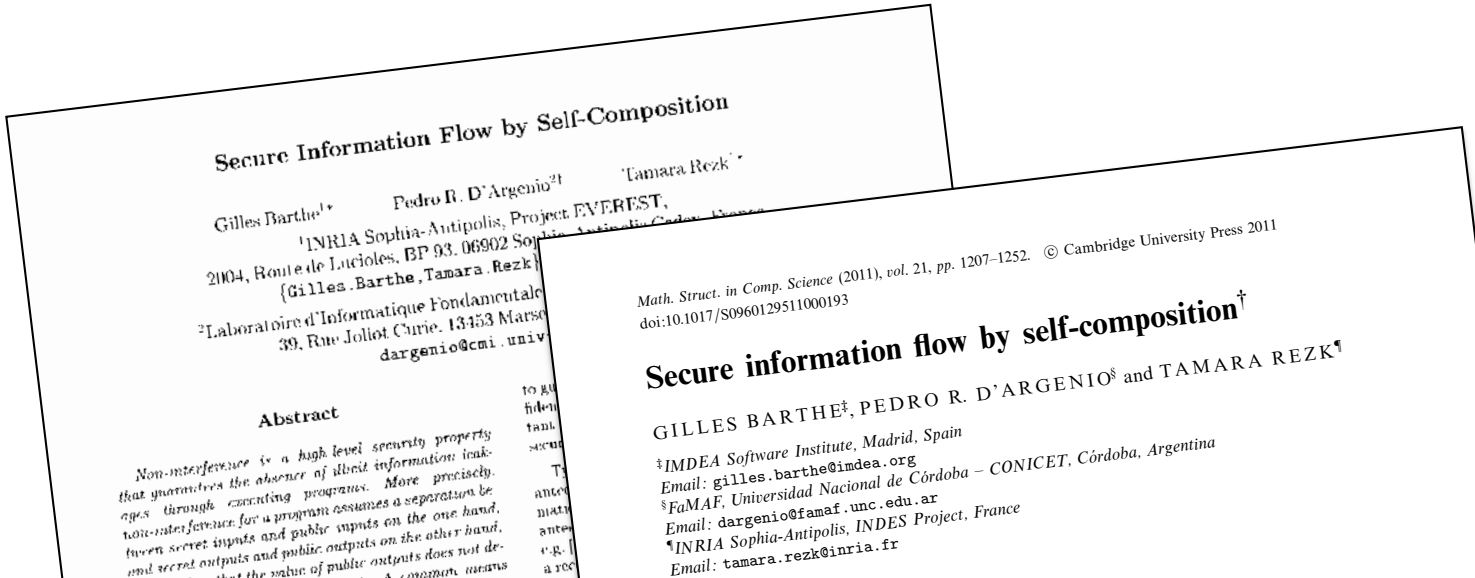
UNIVERSITÄT DES SAARLANDES

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[1]*

[1]INRIA Sophia-Antipolis, Project EVEREST,
2004, Route de Lucioles, BP 93, 06902 Sop
{Gilles.Barthe,Tamara.Rezk}

[2]Laboratoire d'Informatique Fondamentale
39, Rue Joliot Curie, 13453 Marse
dargenio@cmi.univ

Abstract

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

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

Model Checking

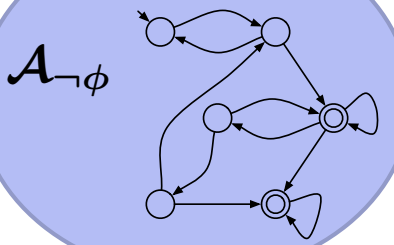
$¿\mathcal{M} \models \phi?$

$\mathcal{M}$

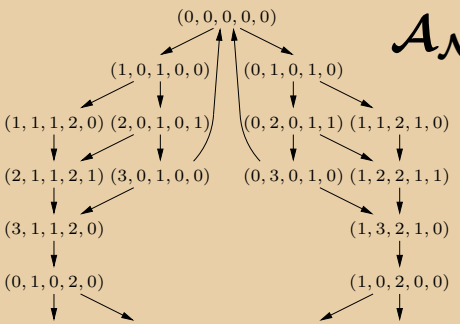
```
int y1 = 0;
int y2 = 0;
short in_critical = 0;

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
}
```

ϕ : $\Box\Diamond crit_1 \wedge \Box\Diamond crit_2$

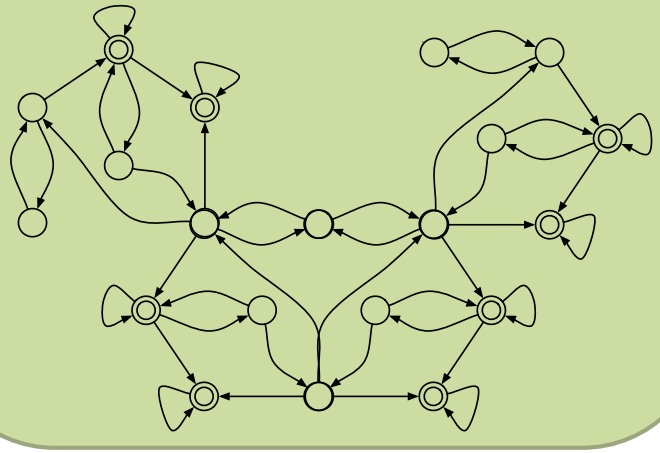
$\mathcal{A}_{\neg\phi}$



$\mathcal{A}_{\mathcal{M}}$



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



CONICET

UNC

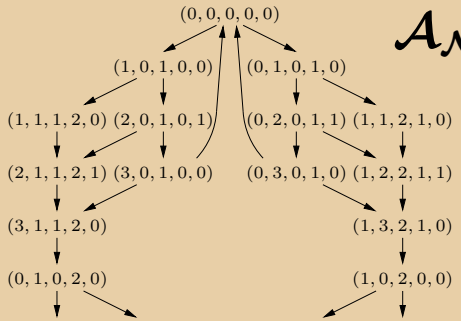
UNIVERSITÄT DES SAARLANDES

Model Checking

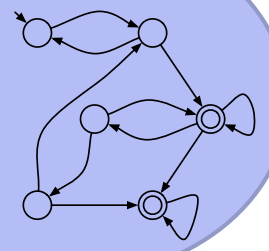
$\mathcal{M}$

```
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
}
```

$\mathcal{A}_{\mathcal{M}}$



$\mathcal{A}_{\neg\phi}$

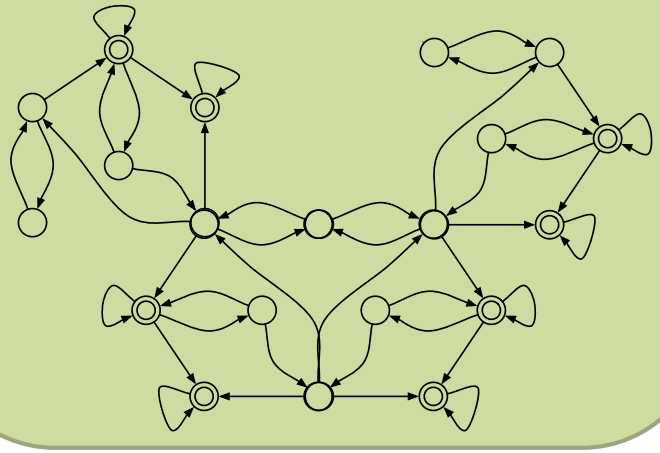


ϕ : $\Box\Diamond crit_1 \wedge \Box\Diamond crit_2$

¿ $\mathcal{M} \models \phi$?

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

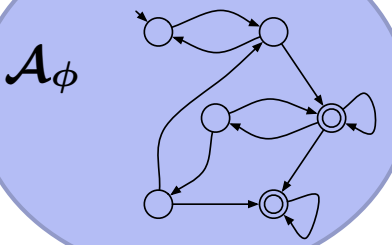
$¿\mathcal{M} \models \phi?$

$\mathcal{M}$

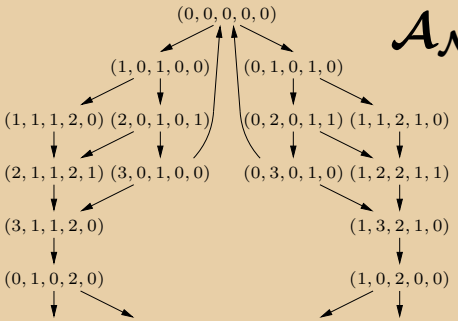
```
int y1 = 0;
int y2 = 0;
short in_critical = 0;

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
}
```

ϕ : $\Box \Diamond crit_1 \wedge \Box \Diamond crit_2$

$\mathcal{A}_\phi$



$\mathcal{A}_\mathcal{M}$



$P(\phi) > 0.95$

CONICET

UNC

UNIVERSITÄT DES SAARLANDES

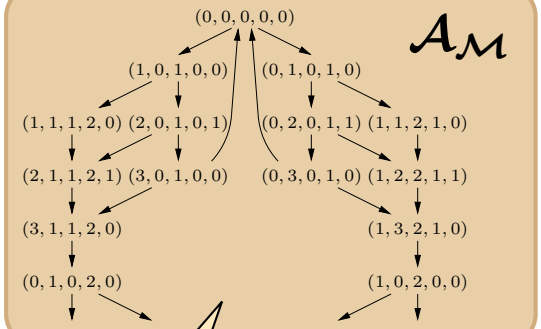
Model Checking Cuantitativo

Incluye primitivas de aleatoriedad

$\mathcal{M}$

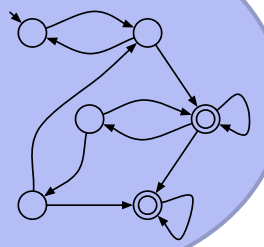
```
int y1 = 0;
int y2 = 0;
short in_critical = 0;

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
}
```

$\mathcal{A}_{\mathcal{M}}$



Proceso de decisión de Markov

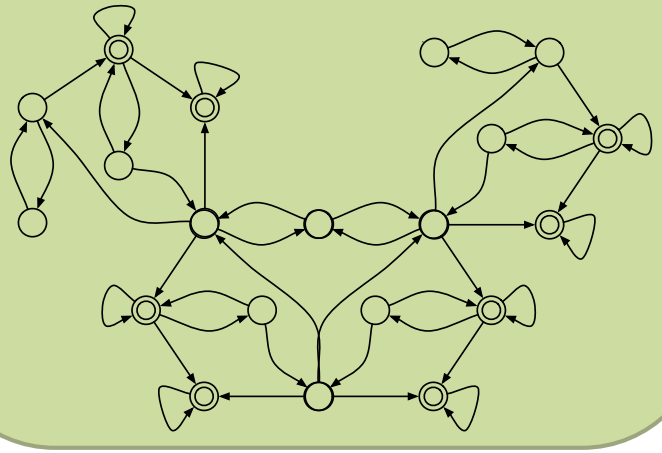
$\mathcal{A}_{\phi}$



ϕ : $\Box \Diamond crit_1 \wedge \Box \Diamond crit_2$

ϕ : $\Box \Diamond crit_1 \wedge \Box \Diamond crit_2$

$P(\phi) > 0.95$



Proceso de decisión de Markov

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

¿ $\mathcal{M} \models \phi$?

CONICET

UNC

UNIVERSITÄT DES SAARLANDES

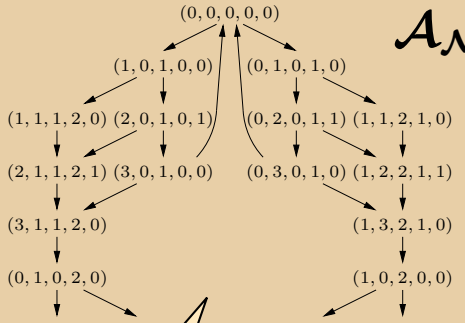
Model Checking Cuantitativo

Incluye primitivas de aleatoriedad

$\mathcal{M}$

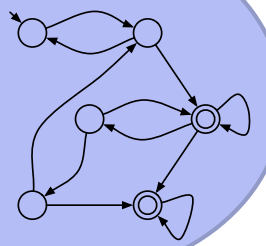
```
int y1 = 0;
int y2 = 0;
short in_critical = 0;

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
}
```

$\mathcal{A}_\mathcal{M}$



Proceso de decisión de Markov

$\mathcal{A}_\phi$



ϕ : $\Box\Diamond crit_1 \wedge \Box\Diamond crit_2$

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

Se reduce a un problema de optimización lineal

$$P(\phi) > 0.95$$

$$x_s = \max_{a \in A} \sum_{t \in S} \mathbf{P}_a(s,t) \cdot x_t \quad \text{if } s \in Pr^{>0}(B) \backslash B$$

$$x_s = 1 \quad \text{if } s \in B$$

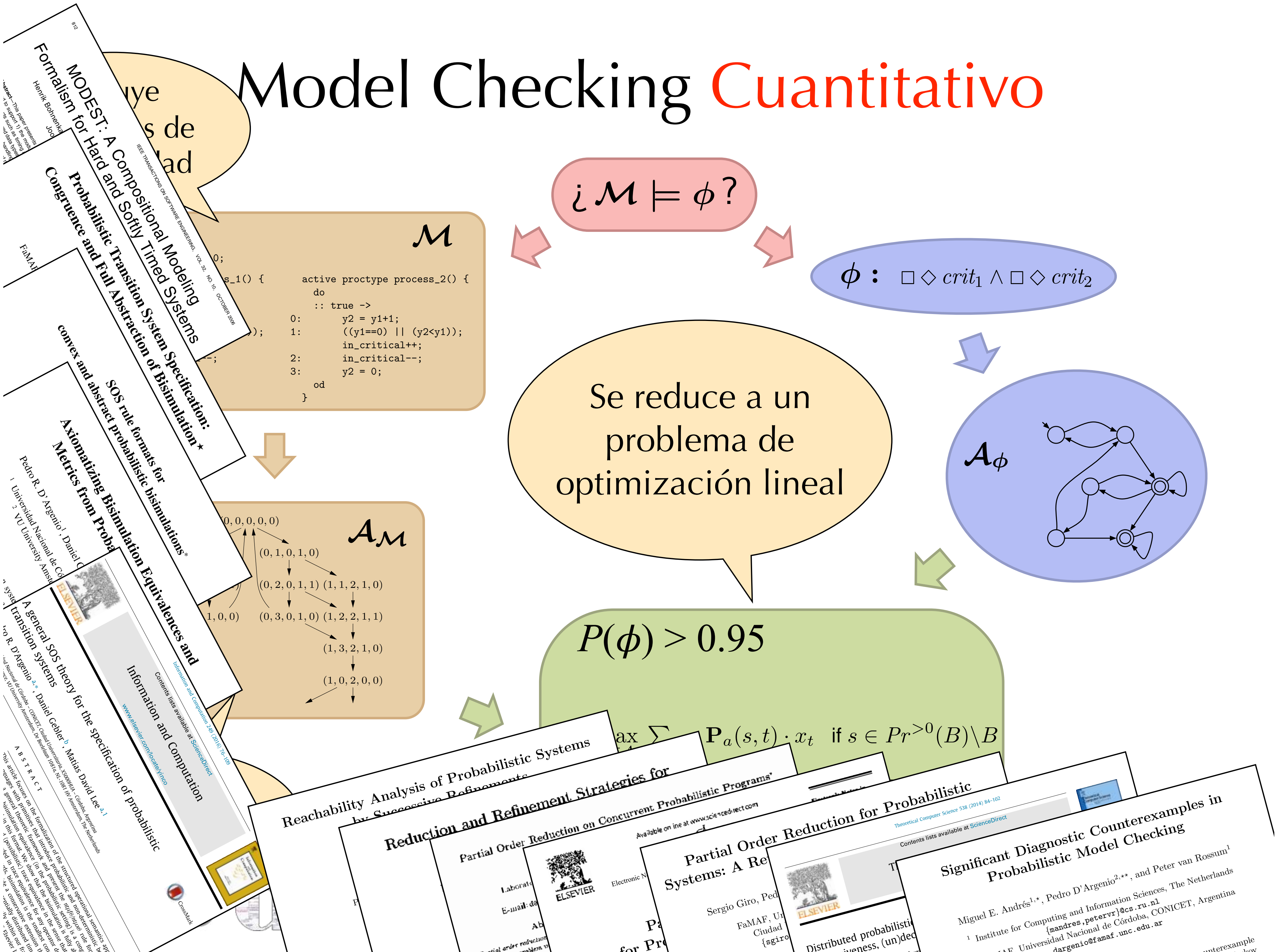
$$x_s = 0 \quad \text{if } s \notin Pr^{>0}(B)$$

CONICET

UNC

UNIVERSITÄT DES SAARLANDES

Model Checking Cuantitativo



Fallas

CONICET

UNC

UNIVERSITÄT DES SAARLANDES

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

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

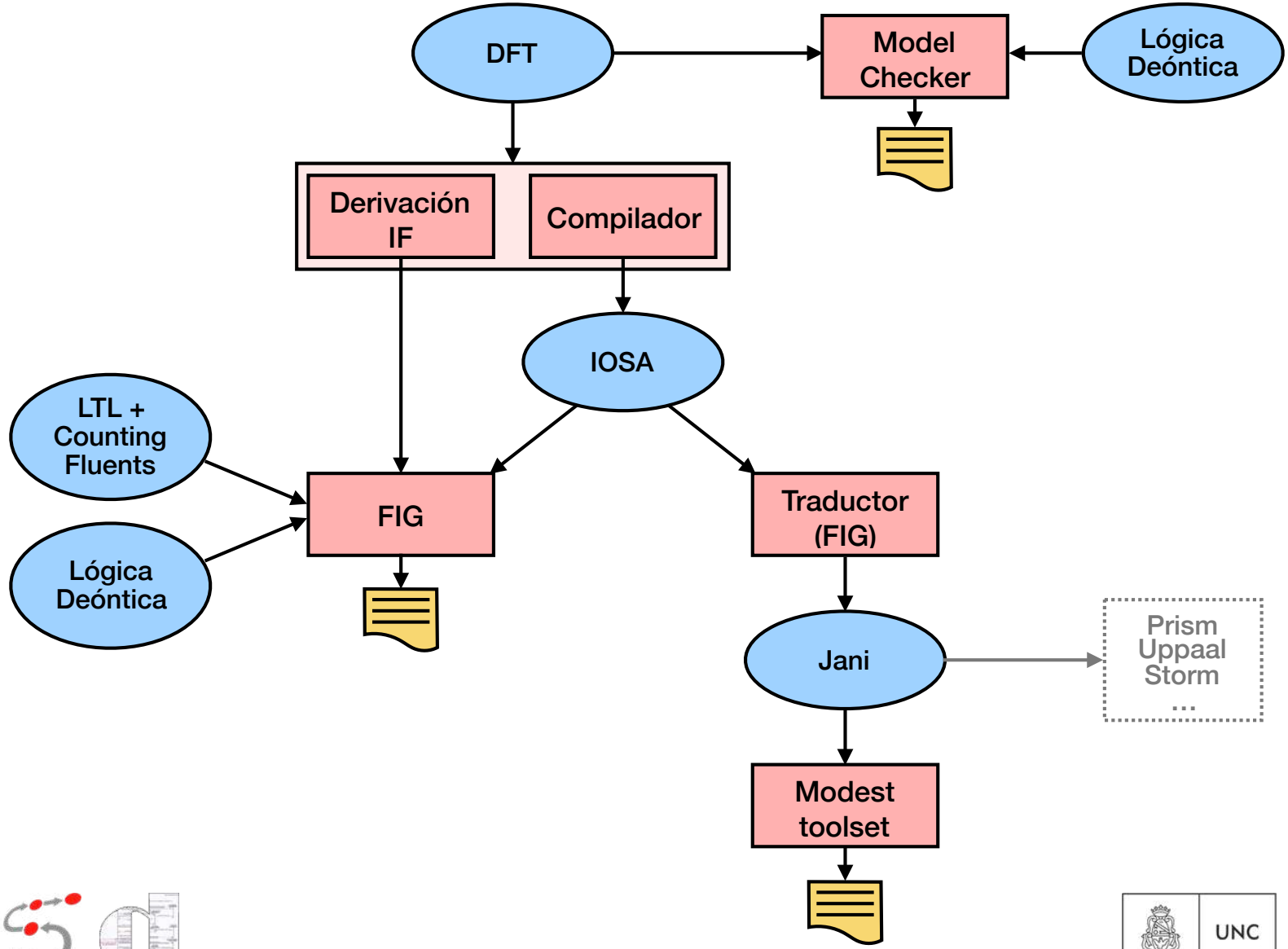
  es decir, de muy baja probabilidad

- Model checking estadístico

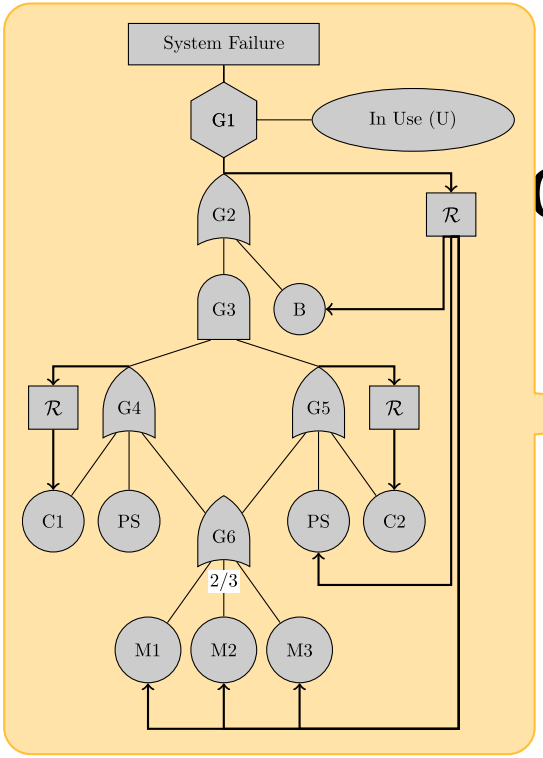
  Es una variante específica de la simulación

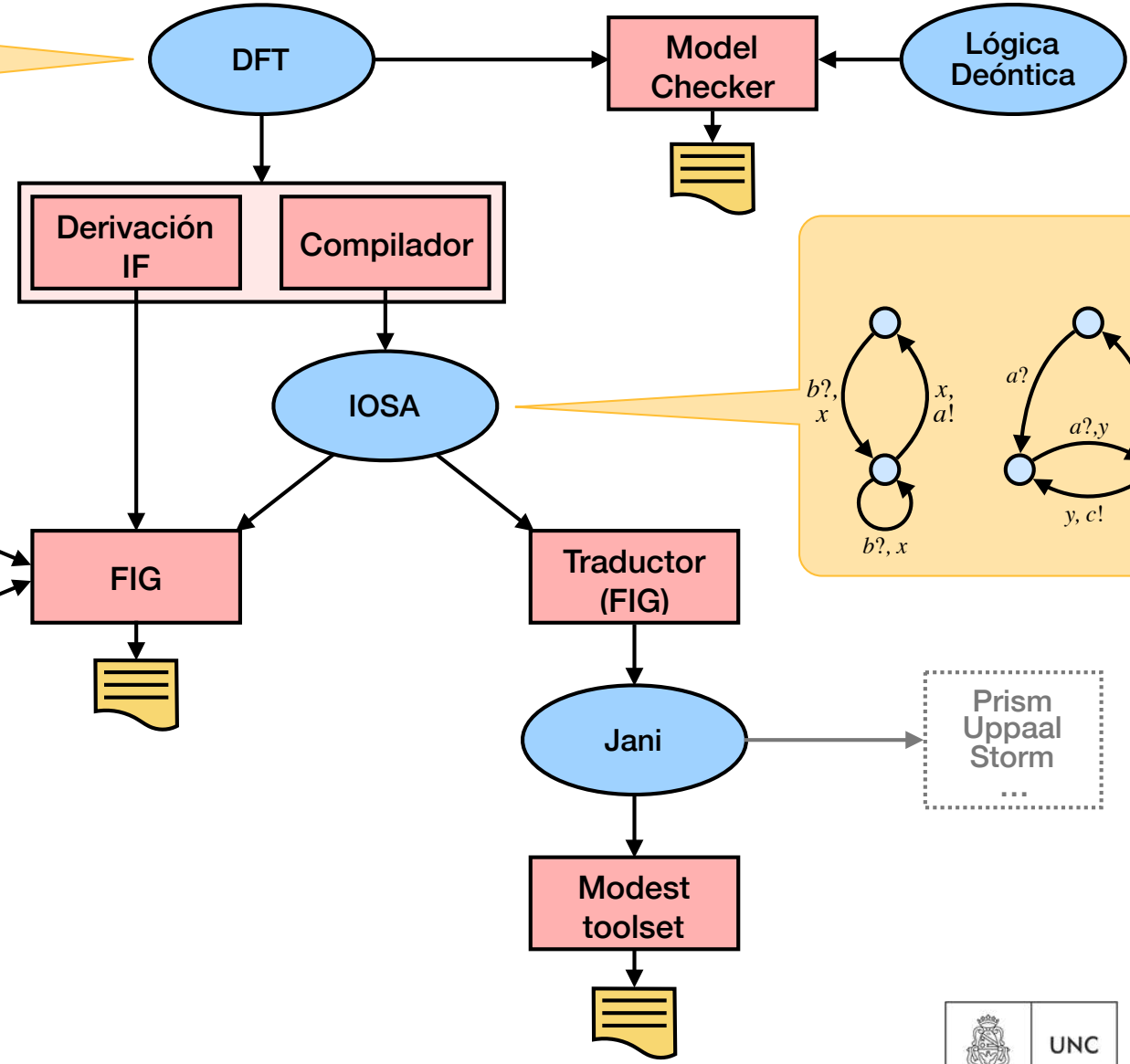
  + no-determinismo

Proyectos RAFTSys y ARES

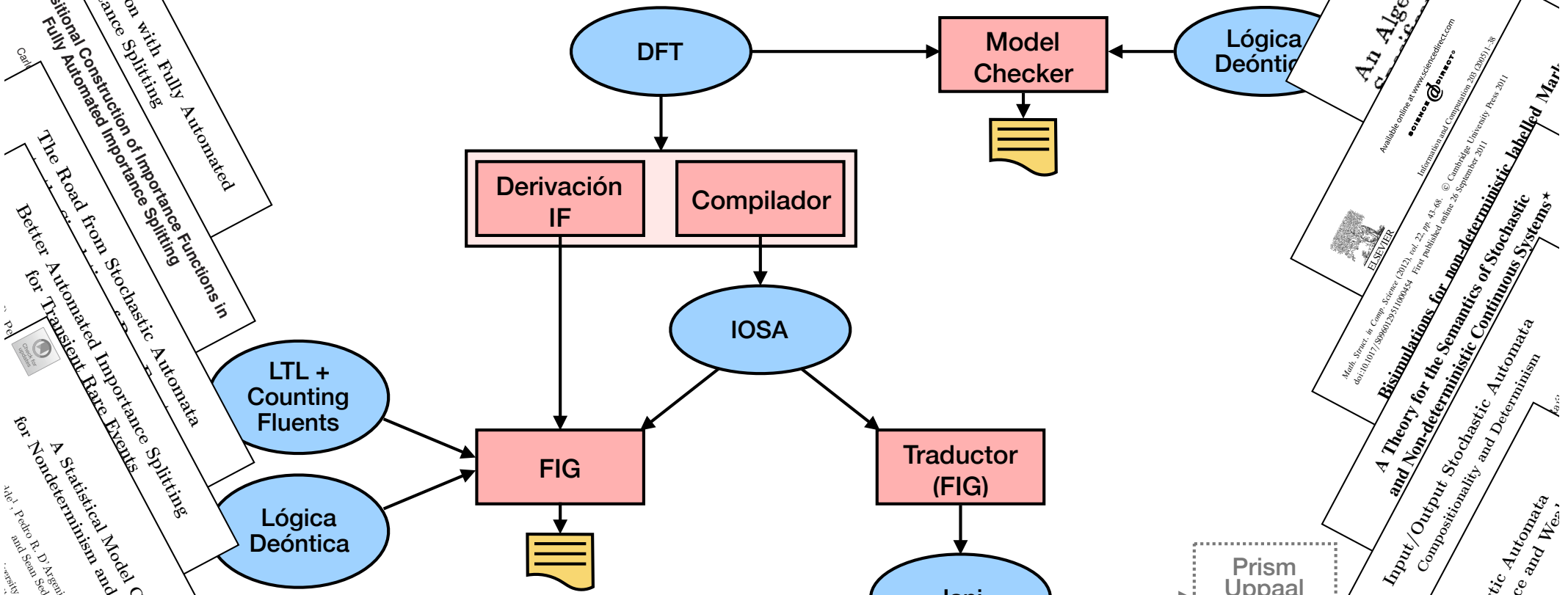


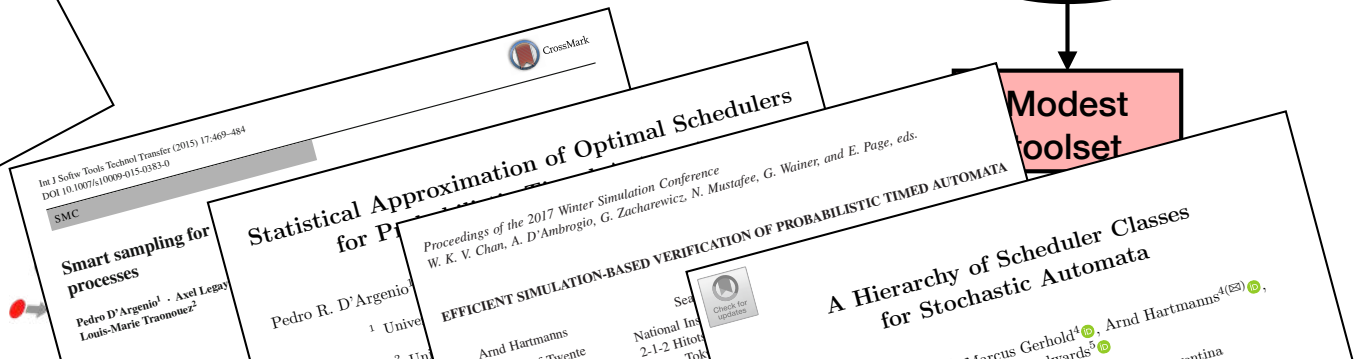
oyectos RAFTSys y ARES





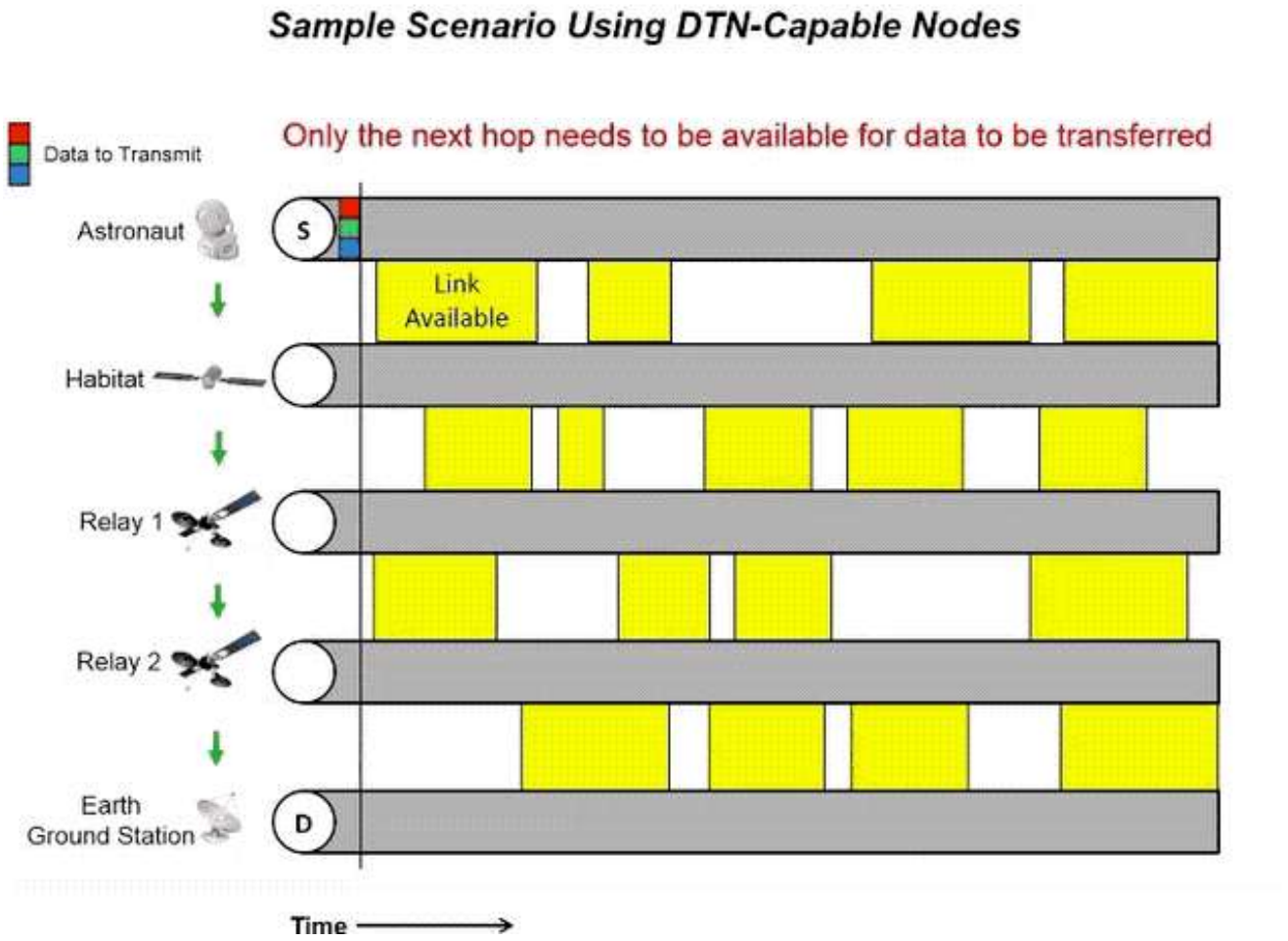
Proyectos RAFTSys y ARES



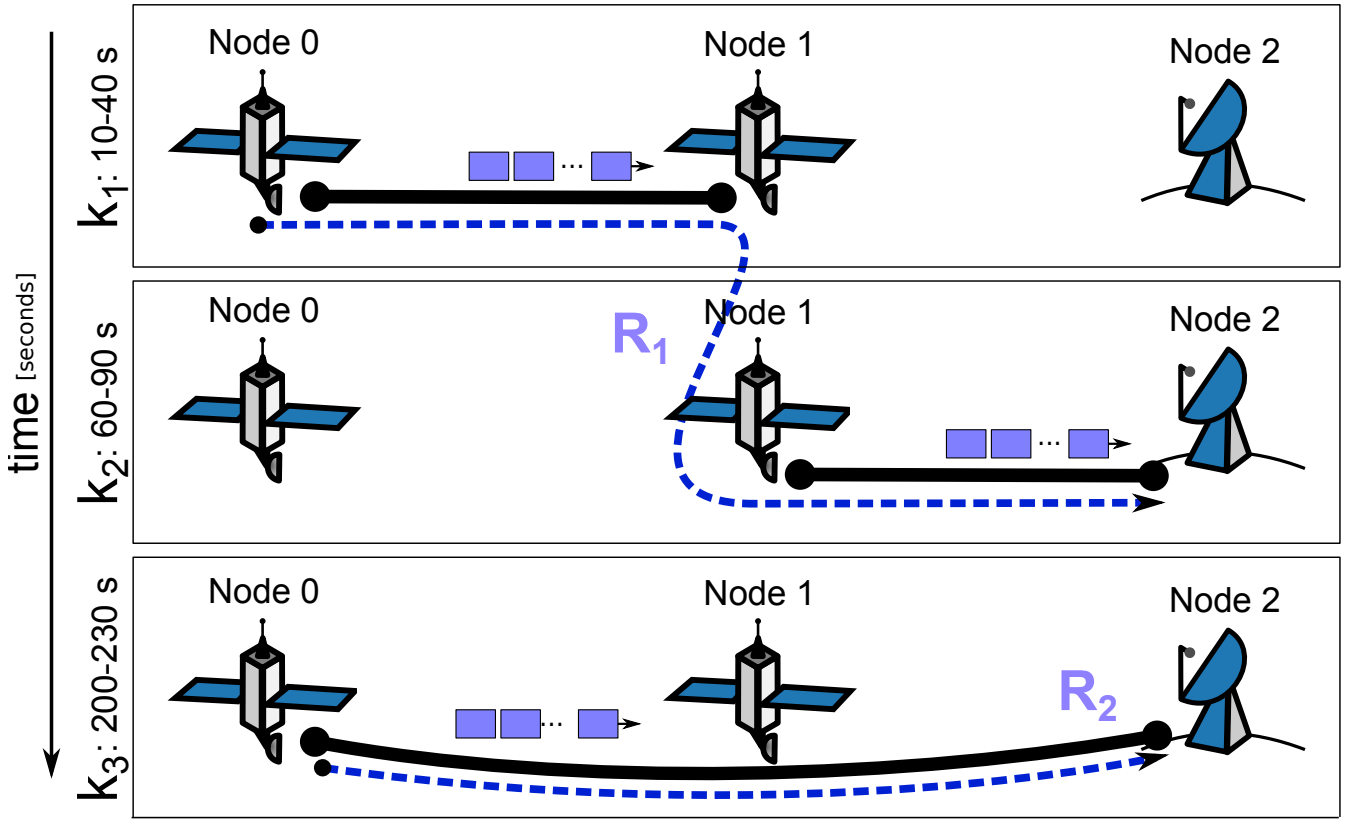


Proyectos RAFTSys y ARES

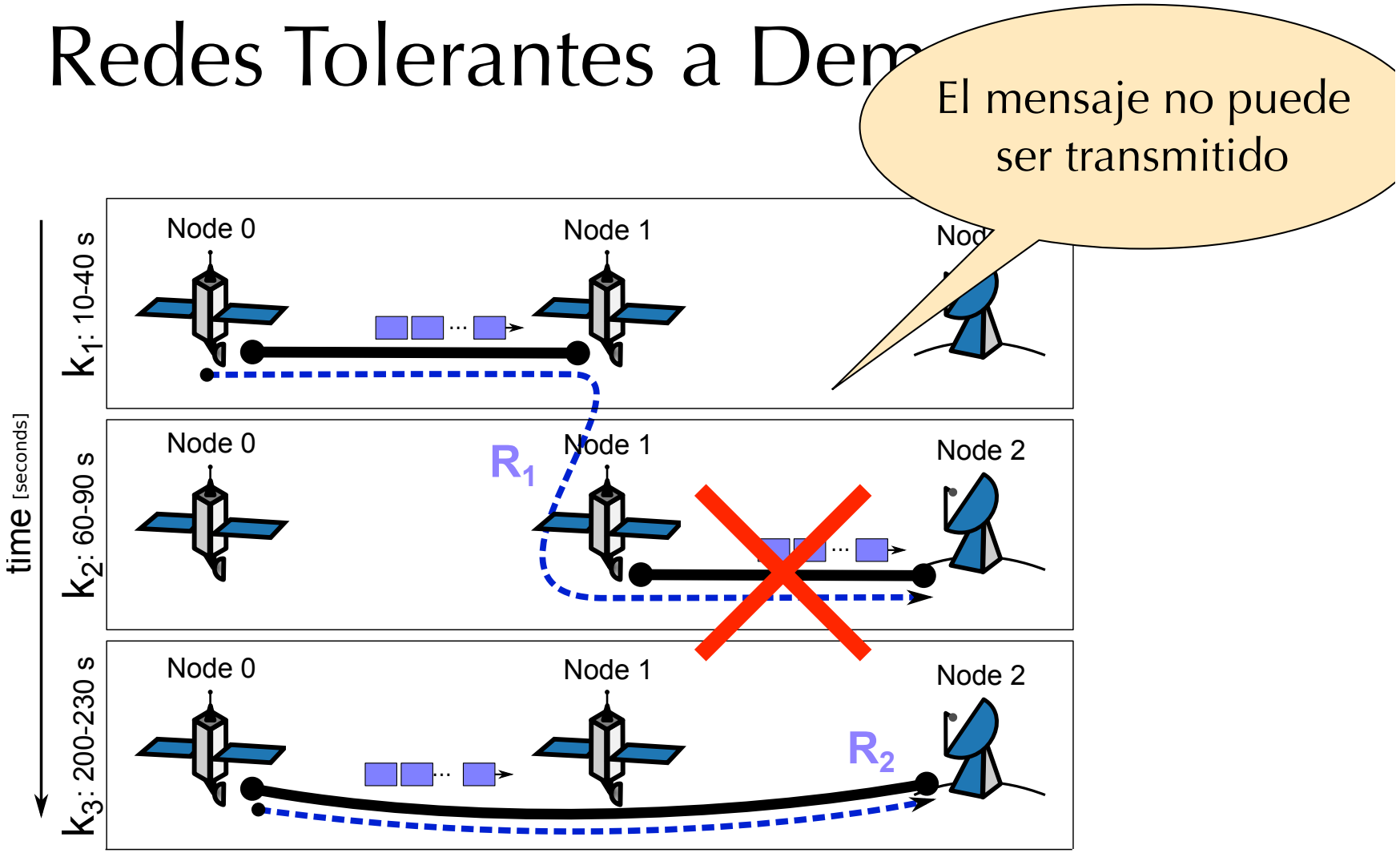
Redes Tolerantes a Demora



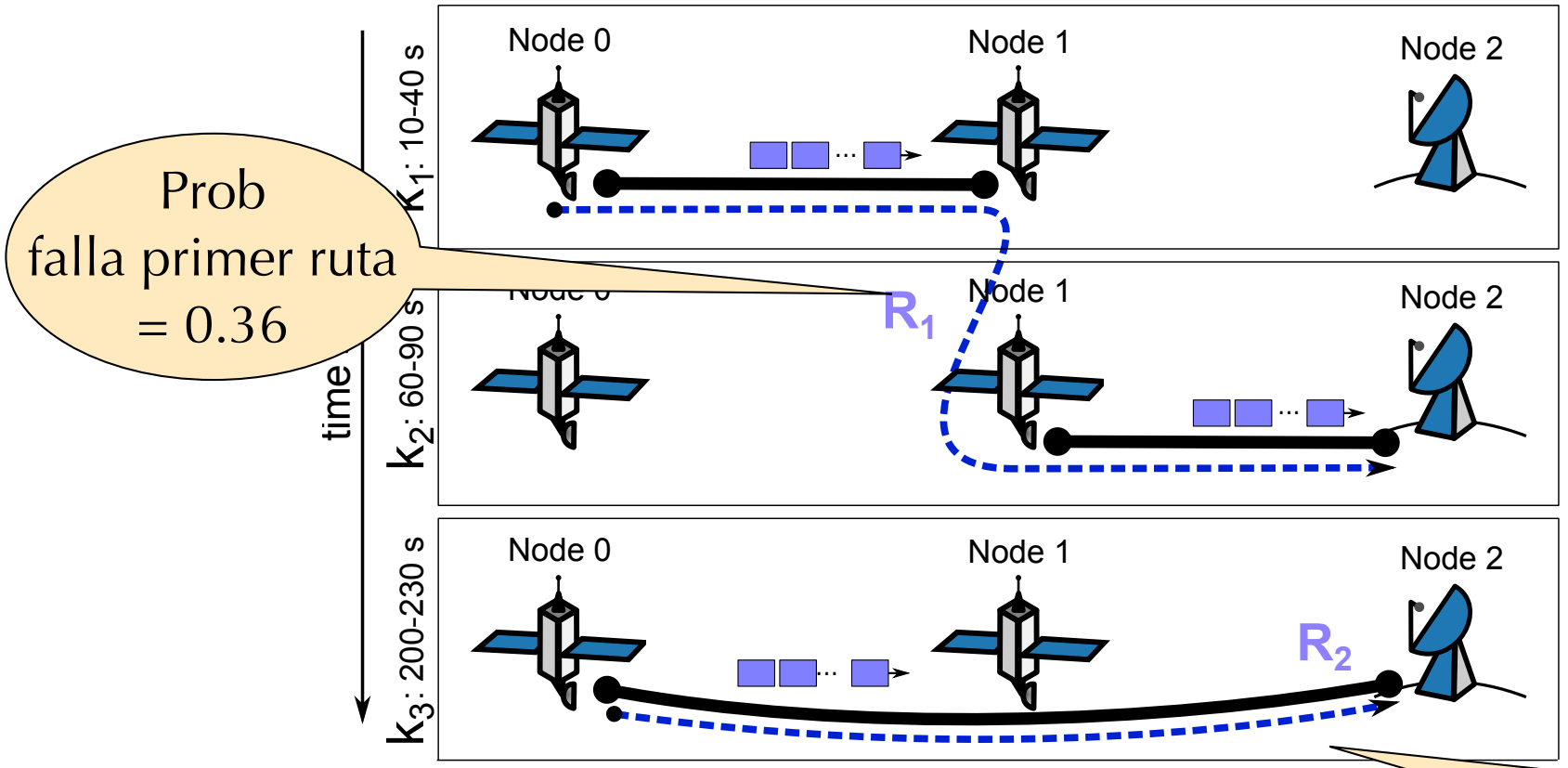
Redes Tolerantes a Demora



Redes Tolerantes a Dem

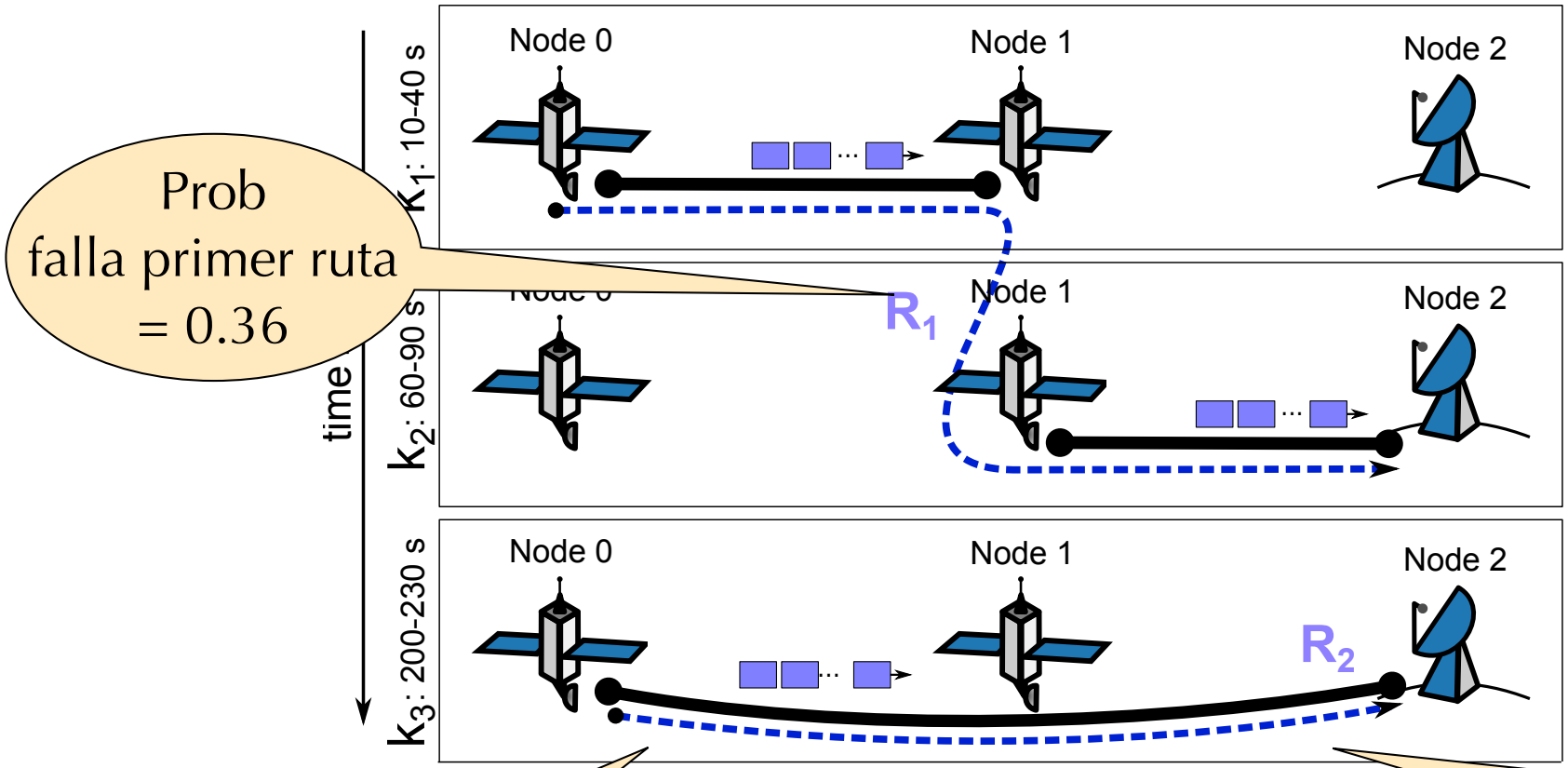


Redes Tolerantes a Demora



Probabilidad de falla de un link = 0.2

Redes Tolerantes a Demora



Prob falla primer ruta = 0.36

Síntesis de ruteo usando Model Checking Cuantitativo

bilidad de falla de un link = 0.2

Prob falla segunda ruta

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*‡§
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Córdoba, Argentina
†Universidad Nacional de Córdoba, Facultad de Ciencias Exactas, Físicas y Naturales (FCEFyN), UNC. Córdoba, Argentina
‡Universidad Nacional de Córdoba, Facultad de Matemática, Astronomía y Computación (FAMAF), UNC, Córdoba, Argentina
§Saarland University, Saarbrücken, Germany

Fallutadas abusivas

Estrategias de
"lock in"

Fallutadas famosas

Fallutadas famosas

UNC

UNIVERSITÄT DES SAARLANDES

Fallutadas famosas

UNC

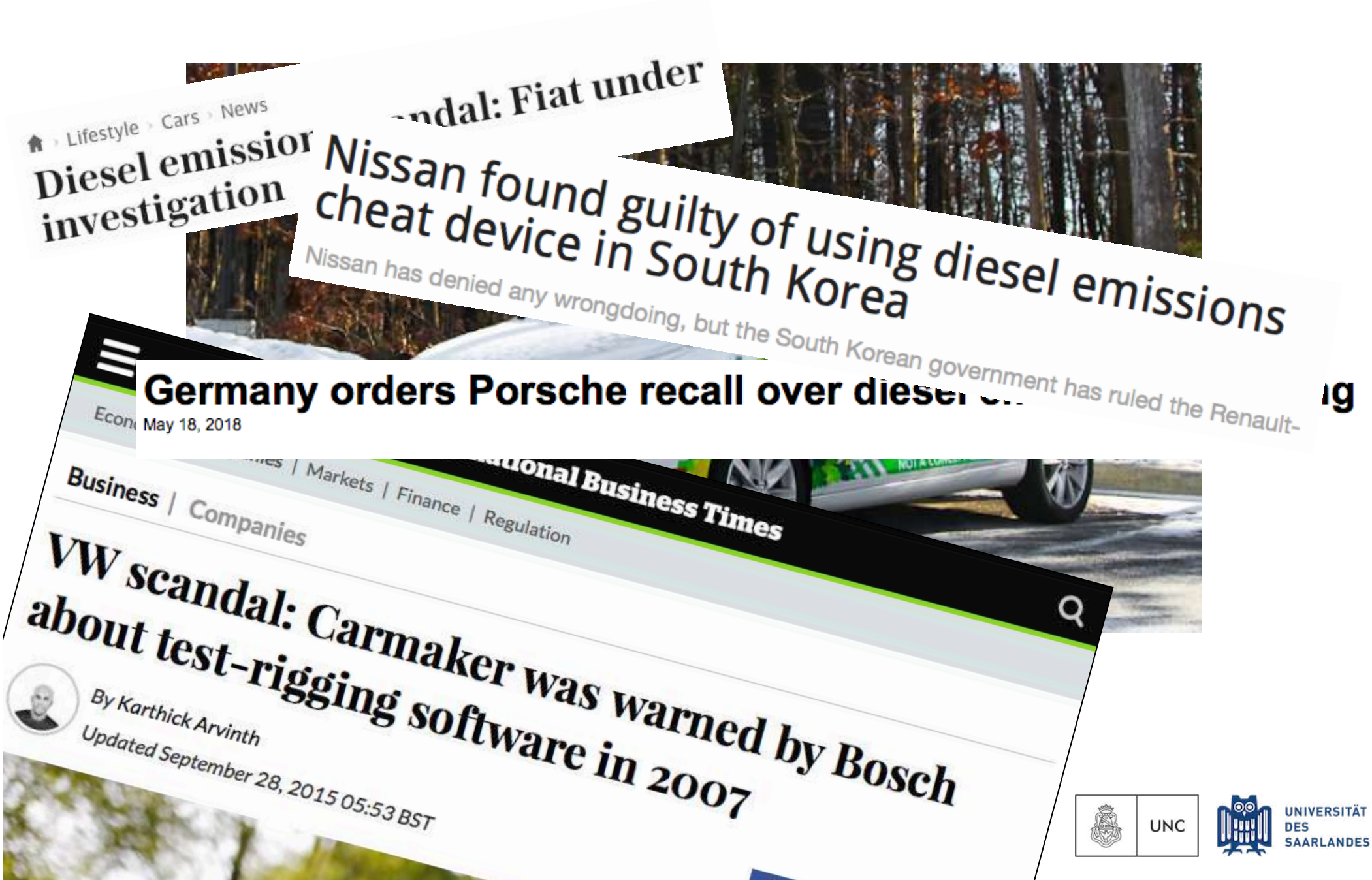
UNIVERSITÄT DES SAARLANDES

Fallutadas famosas

UNC

UNIVERSITÄT DES SAARLANDES

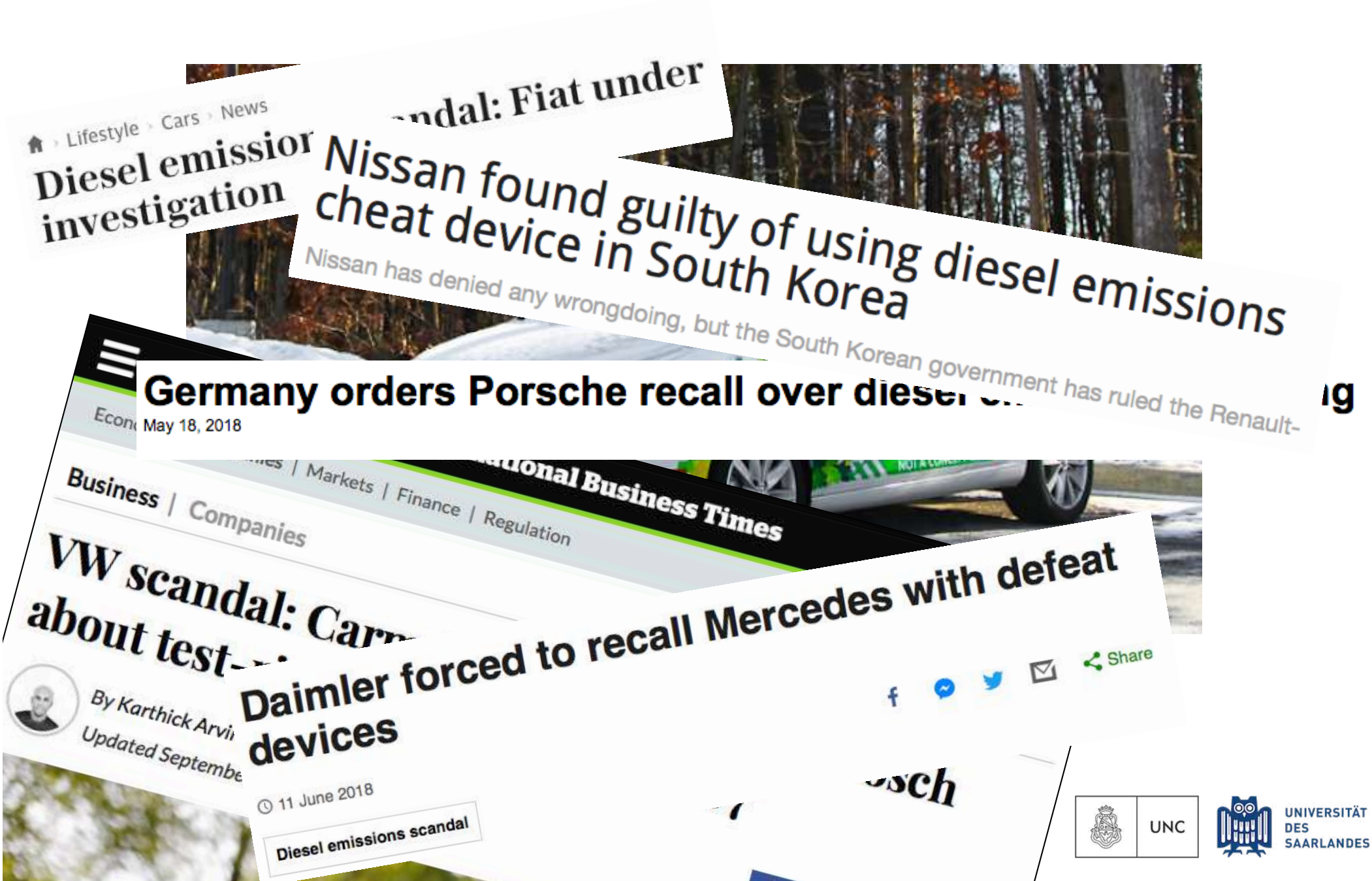
Fallutadas famosas



UNC

UNIVERSITÄT DES SAARLANDES

Fallutadas famosas



UNC

UNIVERSITÄT DES SAARLANDES

Fallutada

UNC

UNIVERSITÄT DES SAARLANDES

Fallutada

France

Peugeot suspected of fraud in diesel scandal

PSA Peugeot Citroën | French economy | auto industry

Share 1 Tweet Share Share

Lifestyle › Cars › News

Diesel emissior investigation

Nissan found guilty of using diesel emissi 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

May 18, 2018

Business | Companies | Markets | Finance | Regulation

National Business Times

VW scandal: Carm to recall Mercedes with defeat about test-

By Karthick Arvi

Updated Septembe

Share

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 Share

Lifestyle › Cars › News

Diesel emissior investigation

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

Nissan has denied any wrongdoing, but the South Korean government has rule

Germany orders Porsche recall over diesel

May 18, 2018

Business | Companies

VW scandal: Carm about test-

By Karthick Arvi

Updated Septembe

Citroën may have breached emissions rules: report

A model tested by the European Commission recorded pollution levels more t

d to recall Mercedes with defeat

Share

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
Lifestyle › Cars › News
Diesel emissior
investigation
Nissan found gu
cheat device in S
Nissan has denied any wrongdoing, but the South Korean government has
GM Accused of Cheating on Diesel Emissions
Germany orders Porsche recall over diesel
May 18, 2018
Business | Companies
VW scandal: Carm
about test-
By Karthick Arvi
Updated Septembe
Citroën may have breached emissions rules: report
A model tested by the European Commission recorded pollution levels more t
to recall Mercedes with defeat
Share
Renault 'cheated on 25 years of pollution tests'
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 licenciatario 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 licenciatario del software

No es posible de formalizar

Software doping

definición formal

- Memoria: $\mu : \mathit{Variables} \to \mathit{Valores}$
- Un programa es un transformador de memoria: $(S, \mu) \Downarrow \mu'$
- Variables:

  Entrada de interés: $i \in \mathit{Variables}$ Salida de interés: $o \in \mathit{Variables}$

- Software doping

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

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

"se parece"

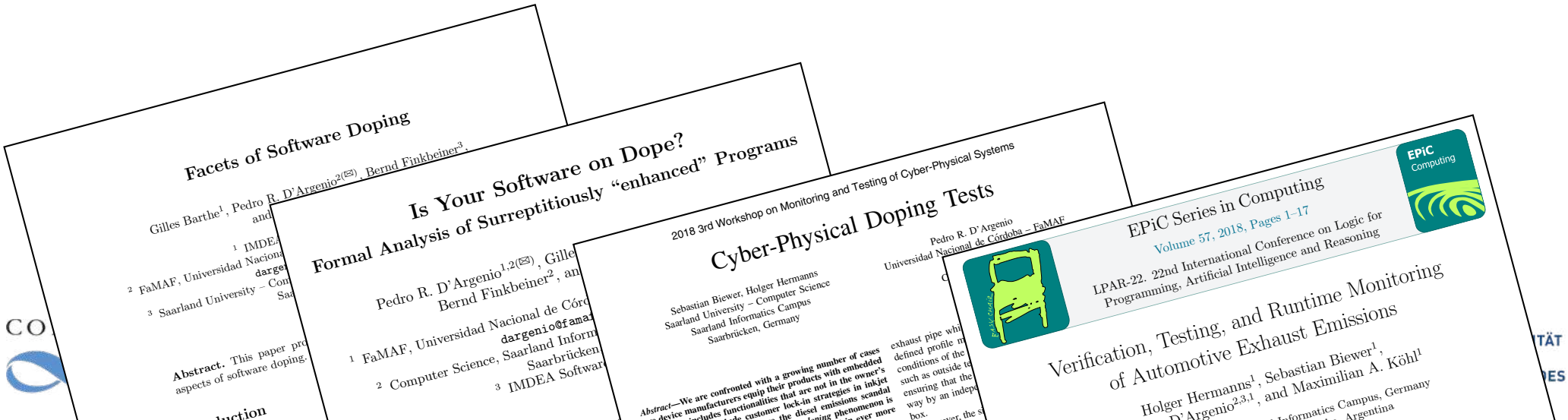
CONICET UNC UNIVERSITÄT DES SAARLANDES

Self-composition otra vez!!

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

Self-composition otra vez!!

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



¿Qué conclusión sacan ustedes?

Las que saco yo:

Las técnicas formales (matemáticas) son cruciales para todo esto

- 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

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

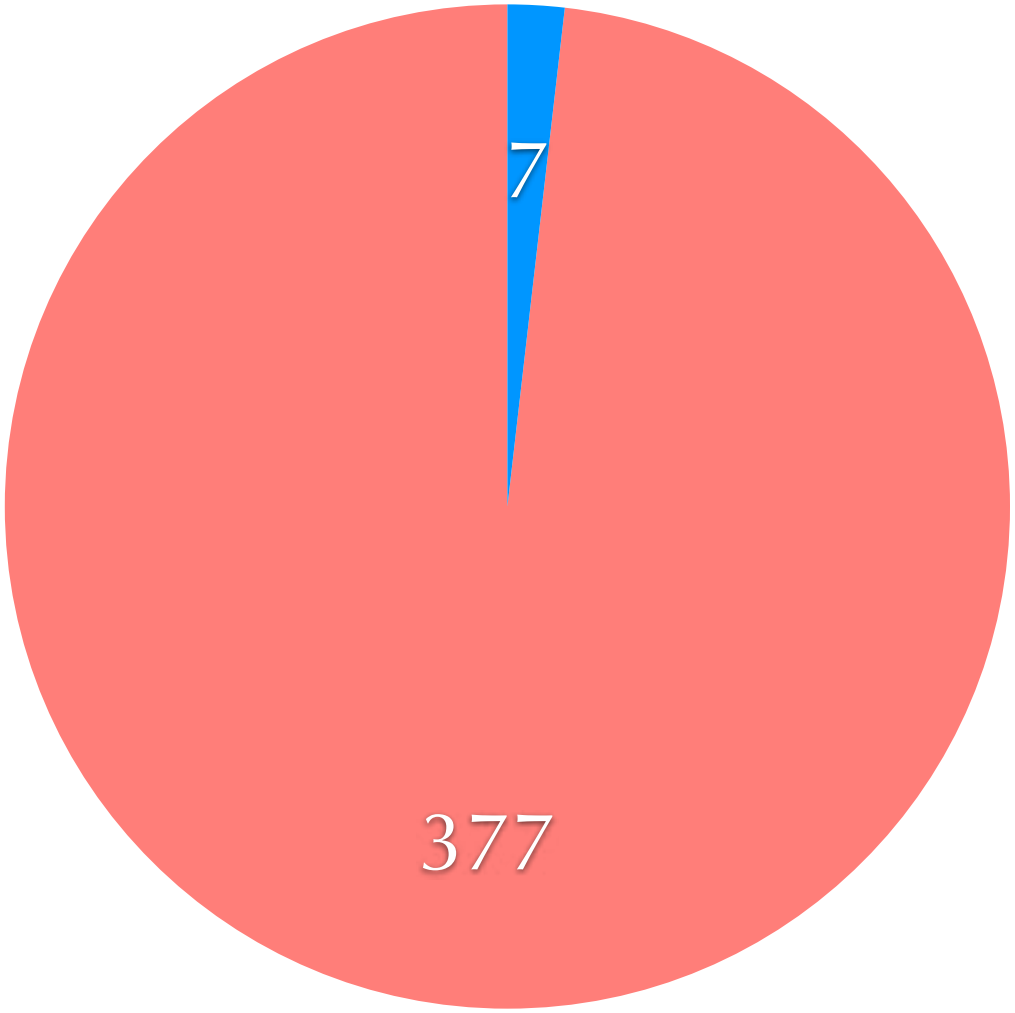
Simulación de Eventos Discretos	8
Computación, Ciencia y Sociedad en la Argentina	7
Computación Gráfica	7
Introducción a la Robótica Móvil	7
Metaheurísticas	7
Introducción al Procesamiento Digital de Imágenes	6
Reconocimiento de Patrones	6
Reescritura (Cálculo Lambda)	6
Teoría de la computabilidad	6
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

CONICET

SAARLANDES

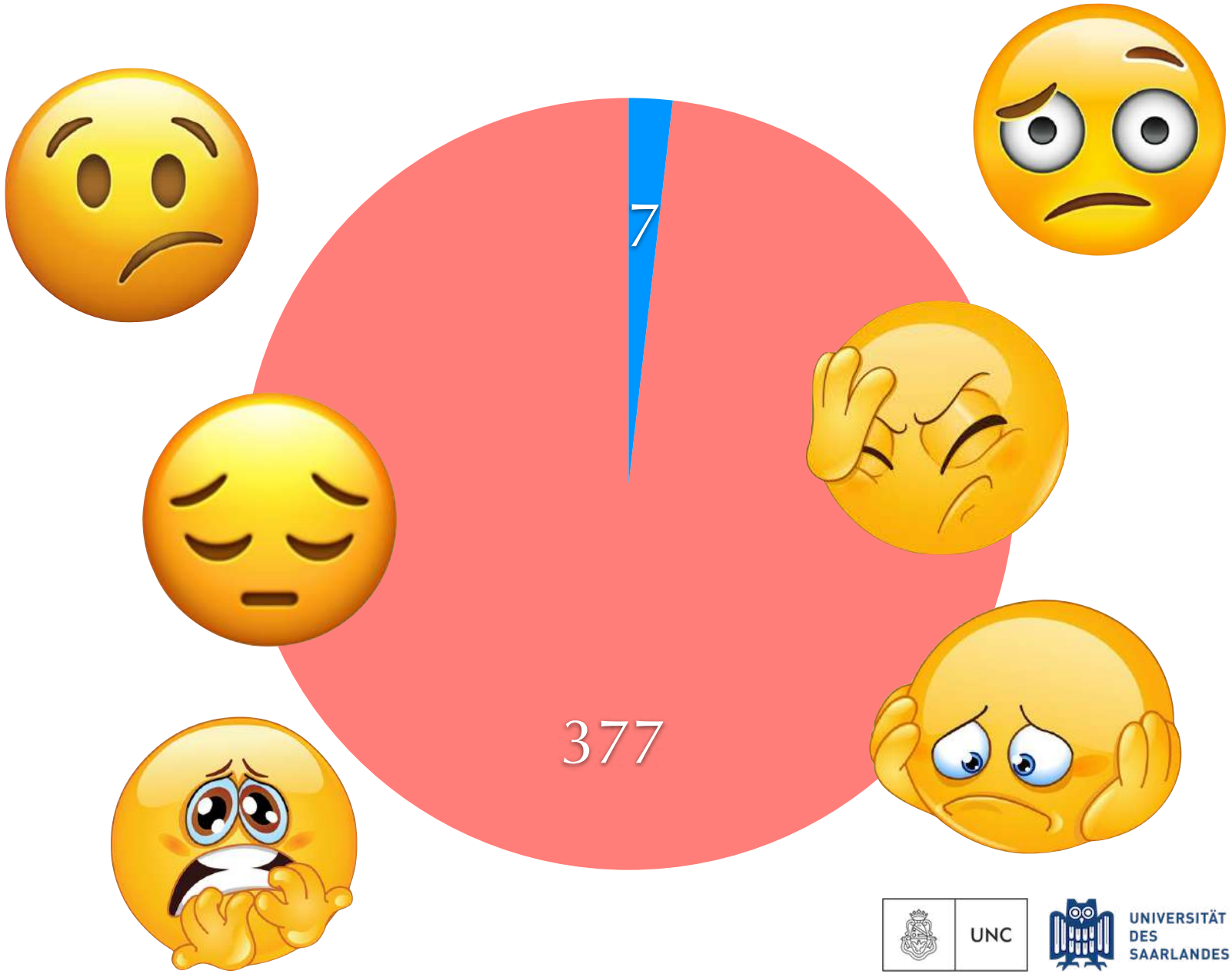
Preferencias de Materias Optativas
(estudiantes)
Departamento de Computación
FCEyN, UBA

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



CONICET

UNC

UNIVERSITÄT DES SAARLANDES