Luchando contra Errores, Fallas y Chantadas para construir Software Confiable

Pedro R. D'Argenio

Grupo de Sistemas Confiables
Universidad Nacional de Córdoba – CONICET (AR)

CONICET

UNC

Luchando contra Errores, Fallas y Chantadas para construir Software Confiable

Pedro R. D'Argenio

Grupo de Sistemas Confiables
Universidad Nacional de Córdoba – CONICET (AR)

CONICET

UNC

Luchando contra Errores, Fallas y Chantadas para construir Sistemas Confiables

Pedro R. D'Argenio

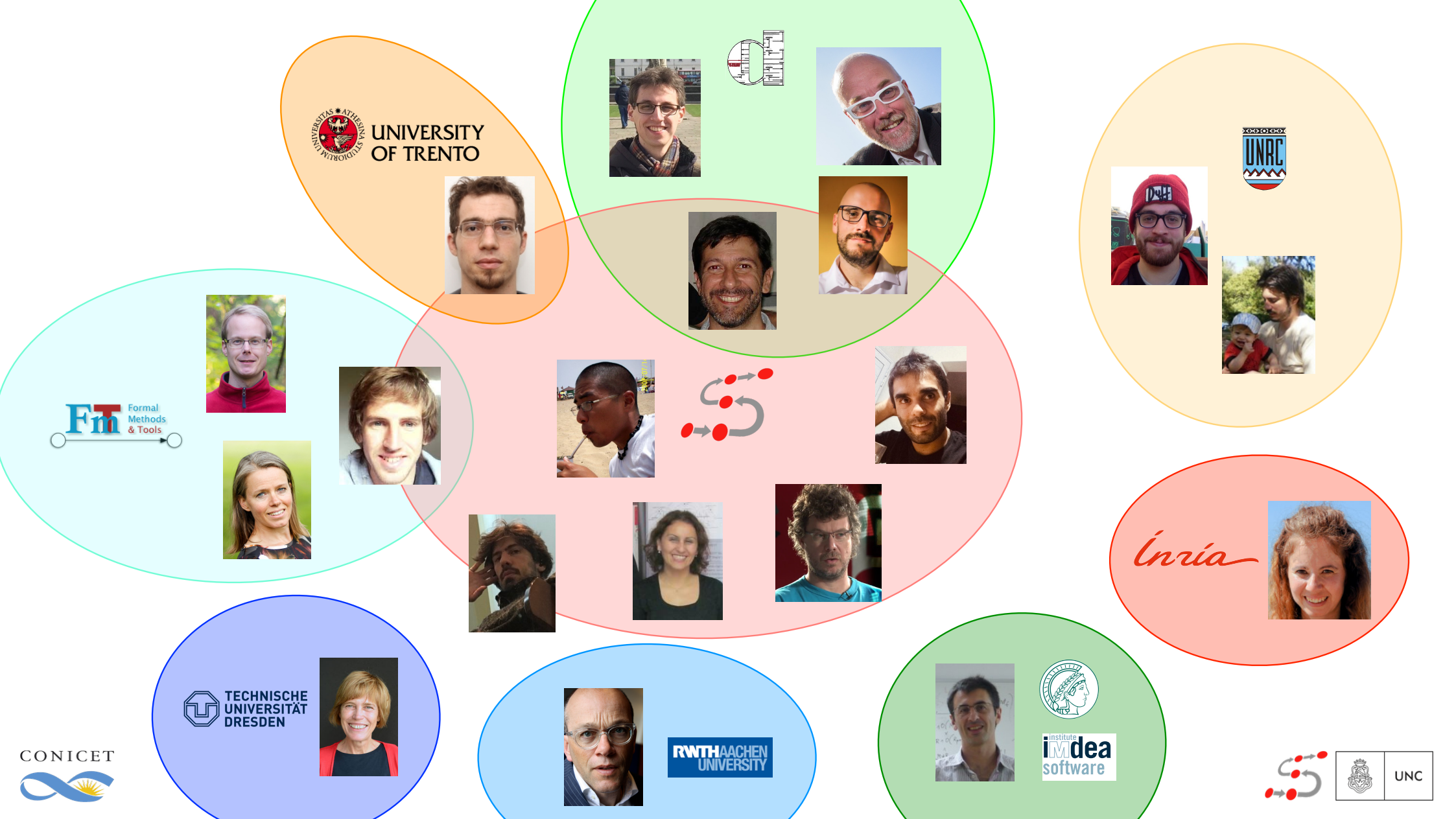
Grupo de Sistemas Confiables
Universidad Nacional de Córdoba – CONICET (AR)

CONICET

UNC

CONICET

UNC

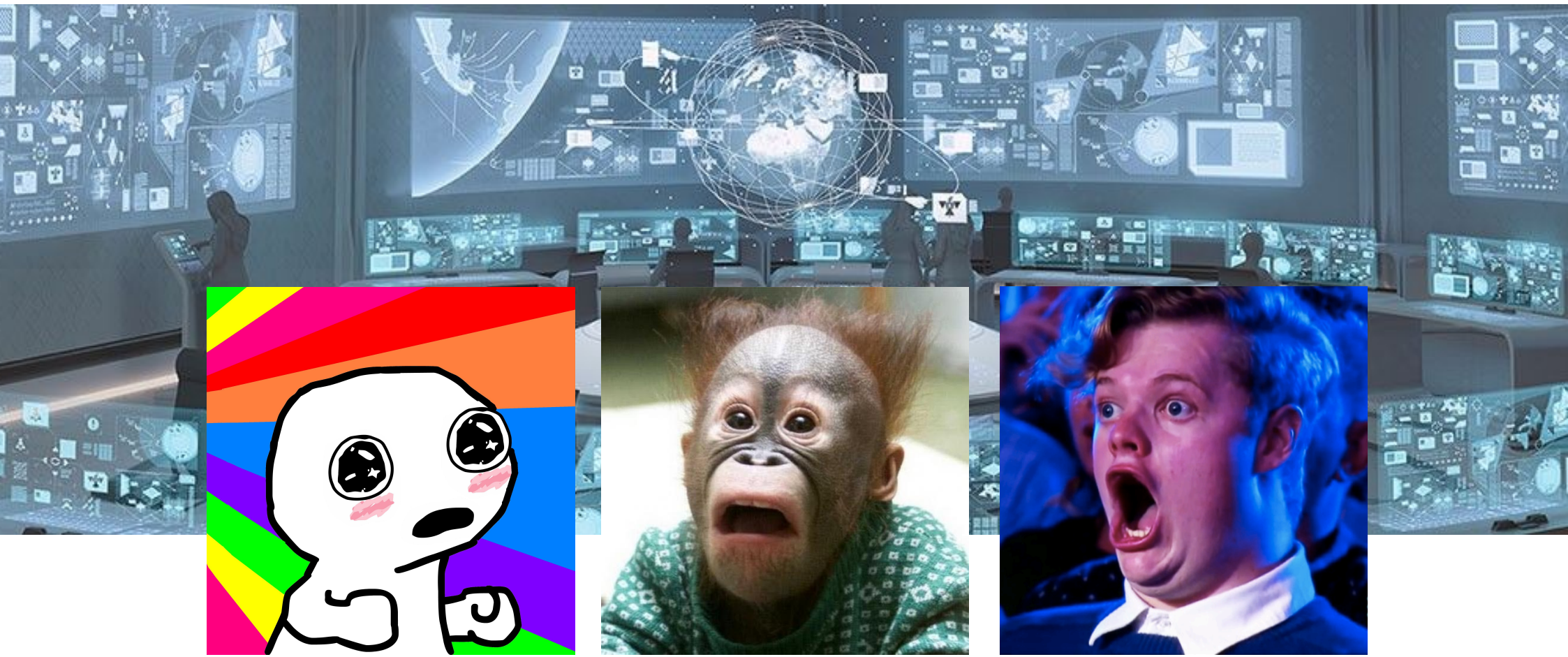

UNIVERSITY OF TRENTO
UNRC
Formal Methods & Tools
Ínría
TECHNISCHE UNIVERSITÄT DRESDEN
RWTH AACHEN UNIVERSITY
institute IMdea software
CONICET
UNC

El software parece hacer maravillas…

CONICET

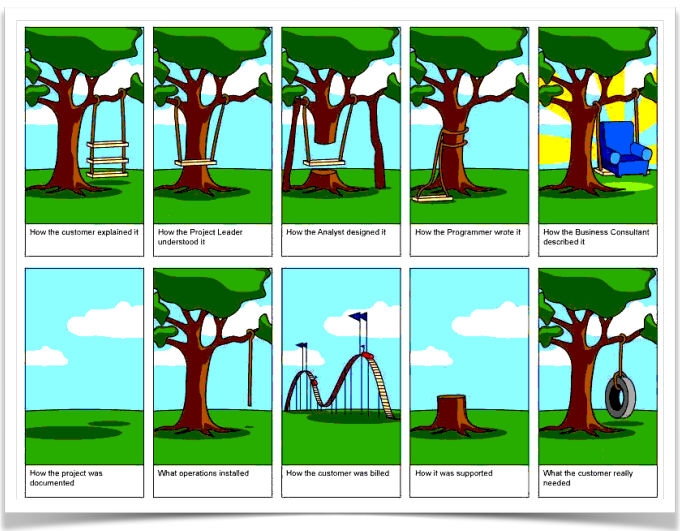
UNC

El software parece hacer maravillas…

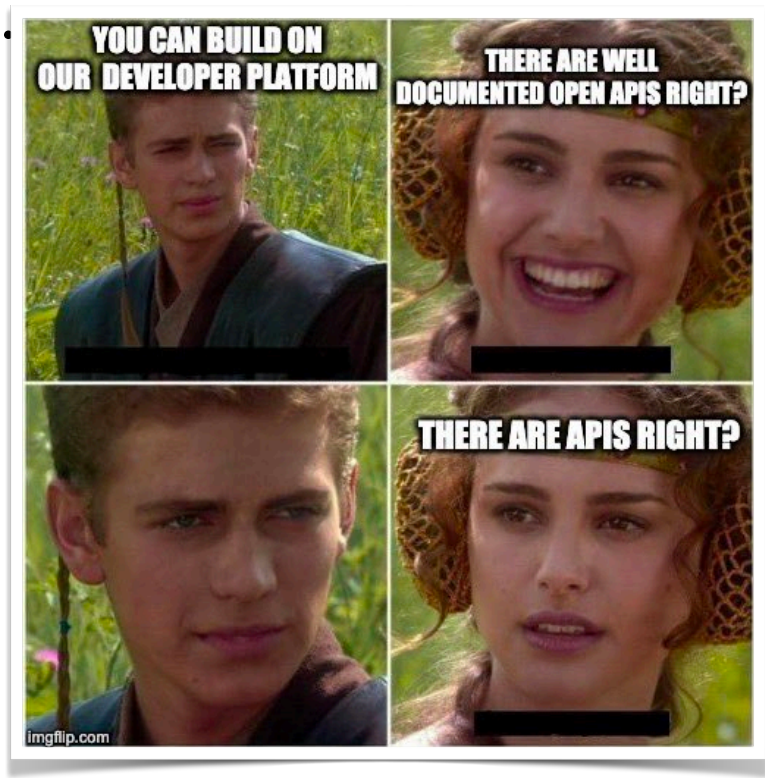


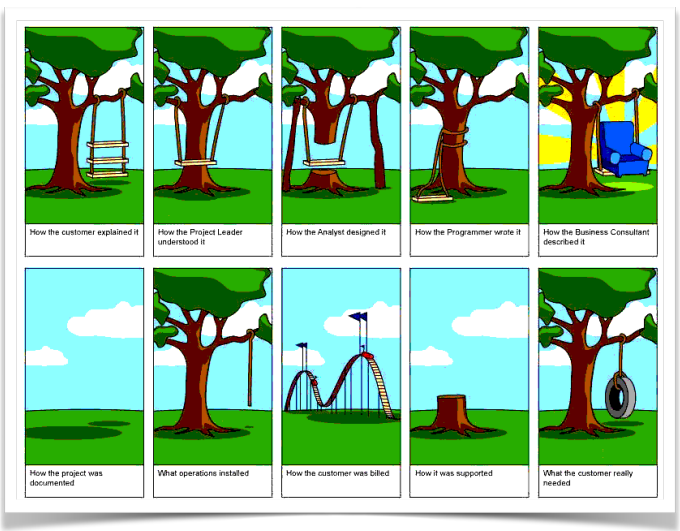
… pero debajo de esa cáscara de maravilla
sabemos muy bien que …

… pero debajo de esa cáscara de maravilla sabemos muy bien que …



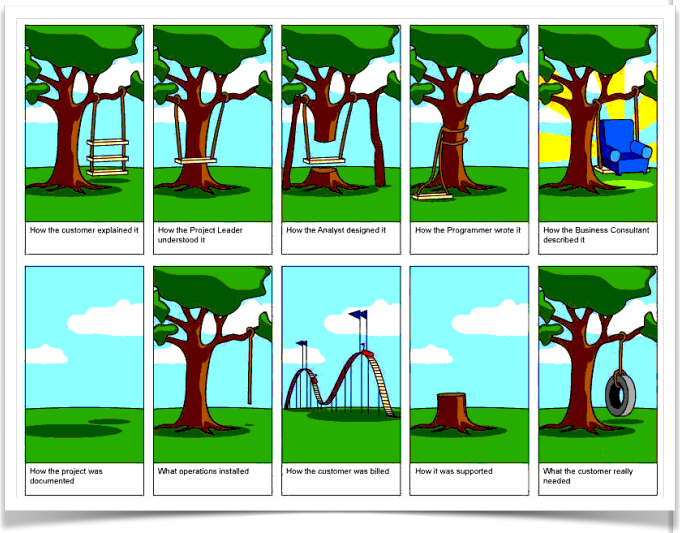
… pero debajo de esa cáscara de maravilla sabemos muy bien que …

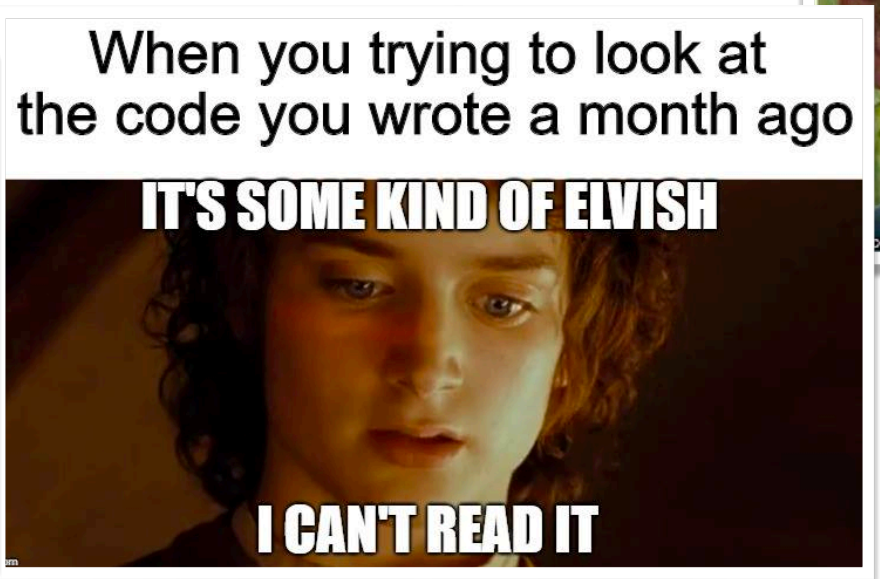


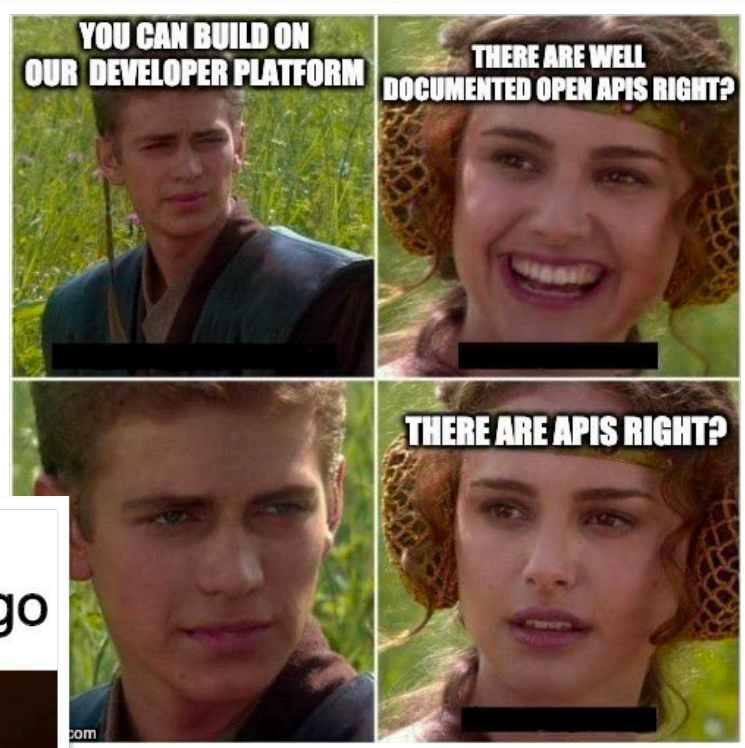


UNC

… pero debajo de esa cáscara de maravilla sabemos muy bien que …

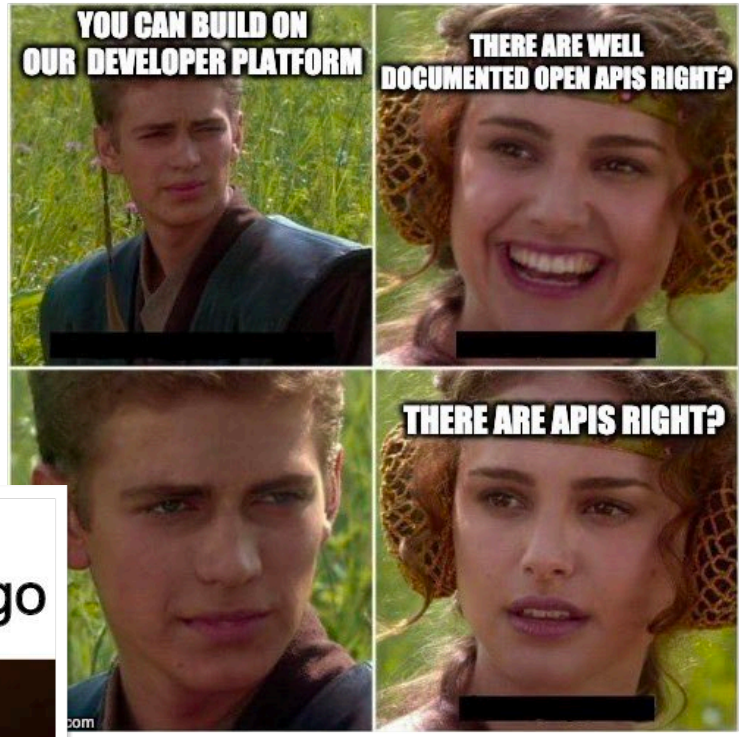


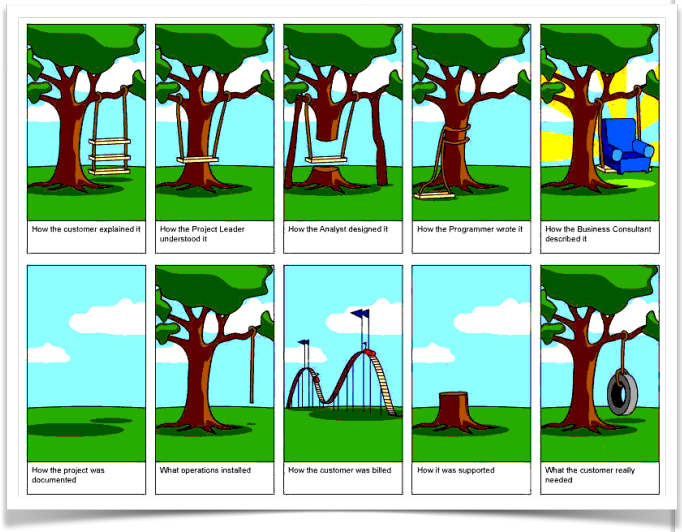


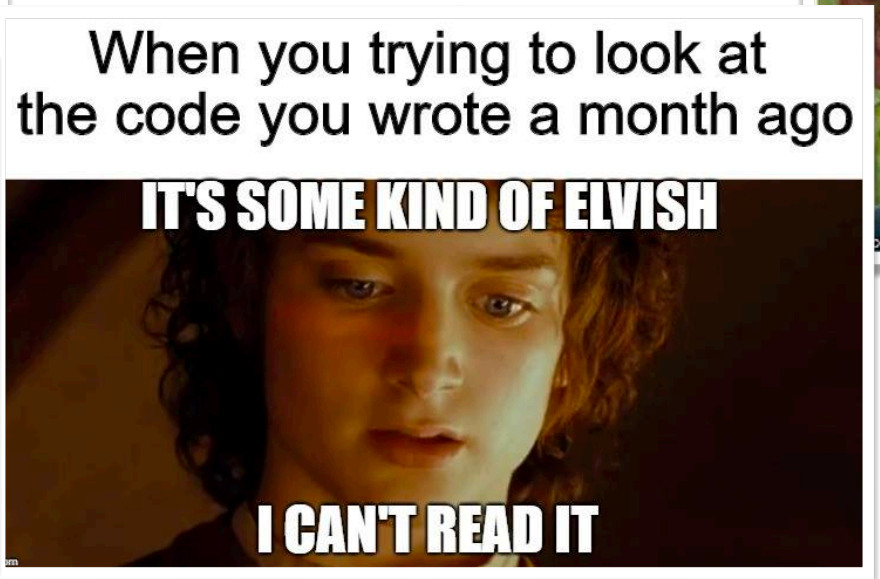


UNC

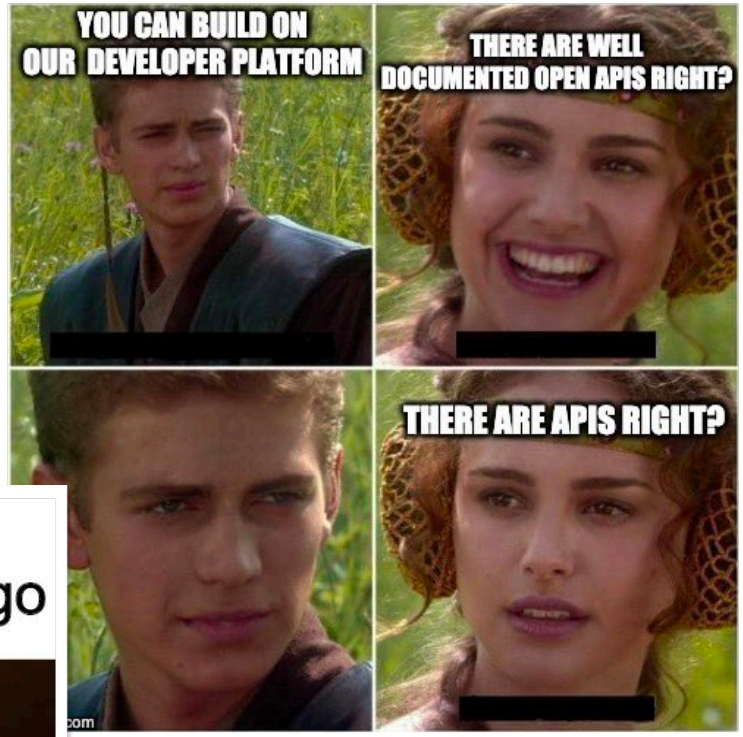
… pero debajo de esa cáscara de maravilla sabemos muy bien que …

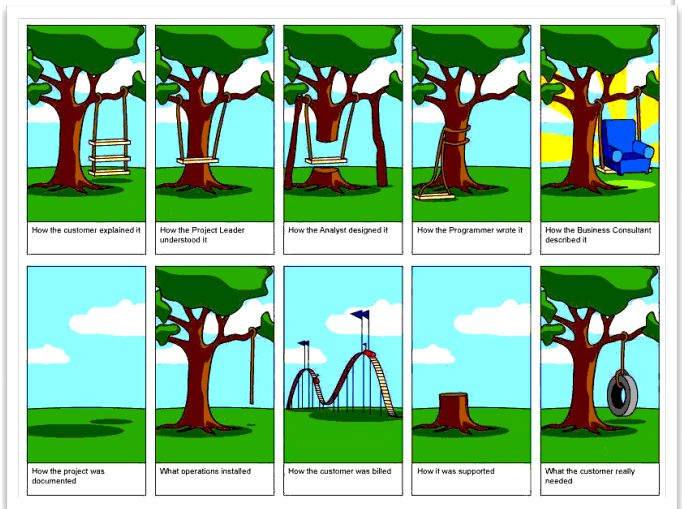


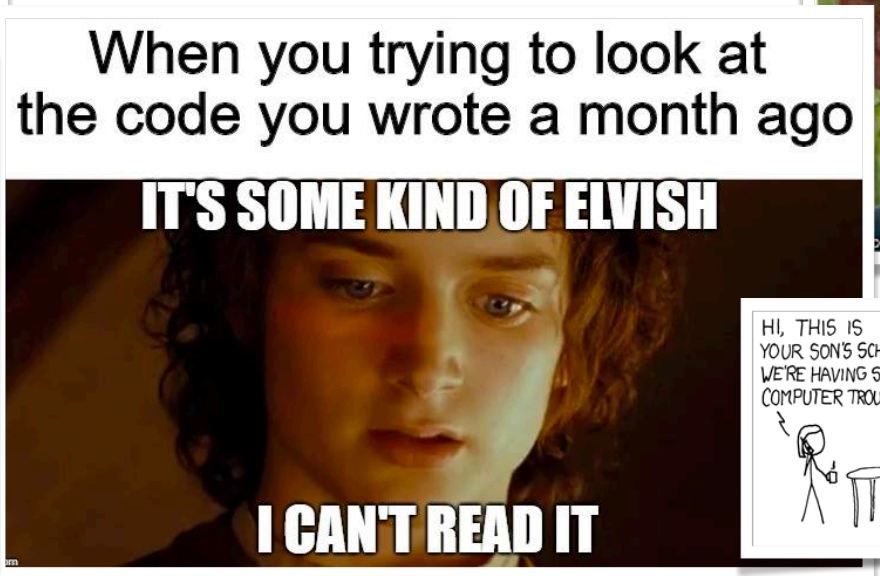


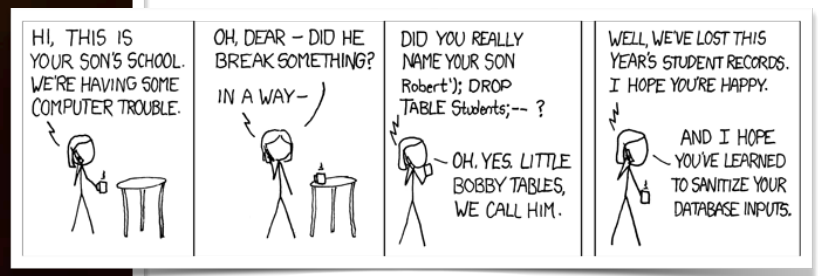


… pero debajo de esa cáscara de maravilla sabemos muy bien que …

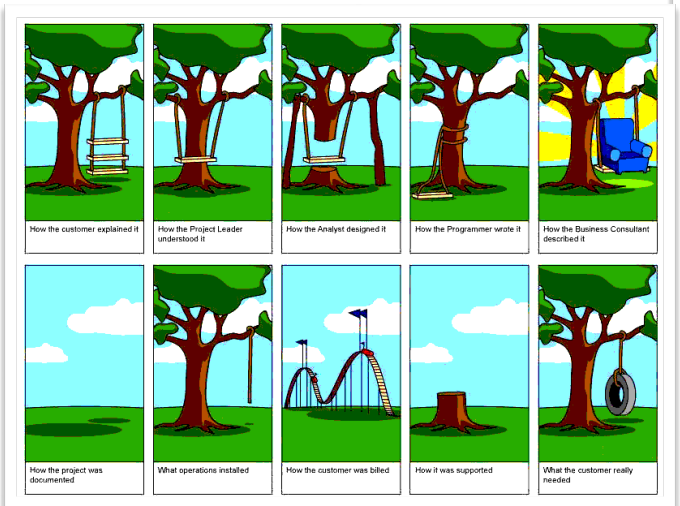


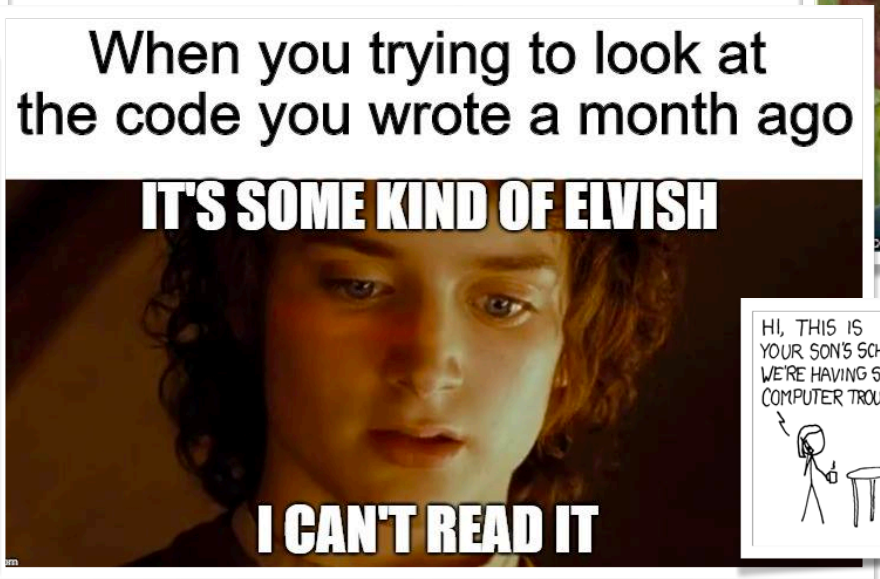


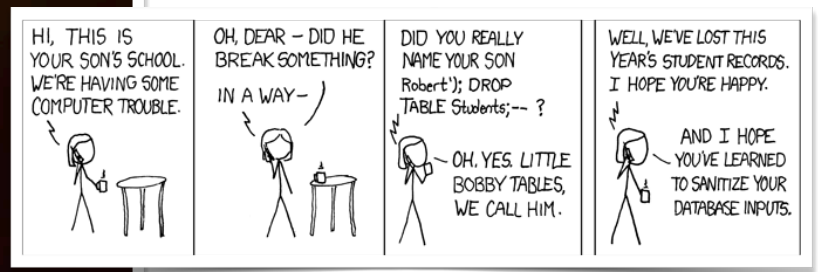




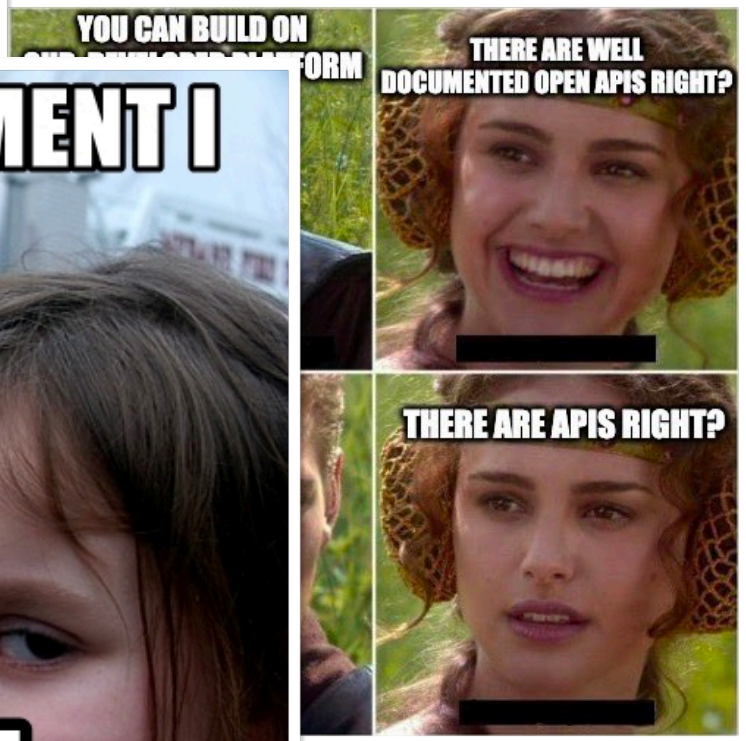
… pero … maravilla

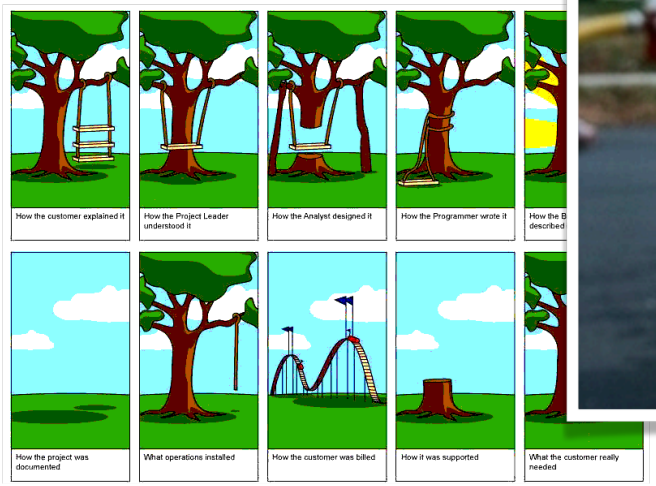


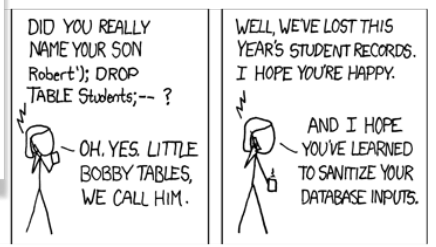




... pero maravilla







¿Qué produce todo estos problemas?

- Errores en el desarrollo del software
- Fallas externas al software pero que son parte del sistema
  (y son mal atendidas)
- Programación malintencionada

¿Qué produce todo estos problemas?

- Errores en el desarrollo del software

Bugs

- Fallas externas al software pero que son parte del sistema
  (y son mal atendidas)

- Programación malintencionada

¿Qué produce todo estos problemas?

- Errores en el desarrollo del software

- Fallas externas al software pero que son parte del sistema
  (y son mal atendidas)

- Programación malintencionada

¿Qué produce todo estos problemas?

- Errores en el desarrollo del software

- Fallas externas al software pero que son parte del sistema
  (y son mal atendidas)

- Programación malintencionada

¿Qué produce todo estos problemas?

- Errores en el desarrollo del software

- Fallas externas al software pero que son parte del sistema
  (y son mal atendidas)

- Programación malintencionada

¿Qué produce todo estos problemas?

- Errores en el desarrollo del software

- Fallas externas al software pero que son parte del sistema
  (y son mal atendidas)

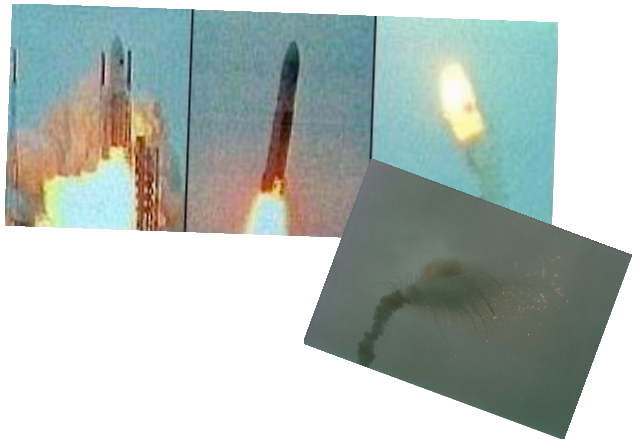
- Programación malintencionada

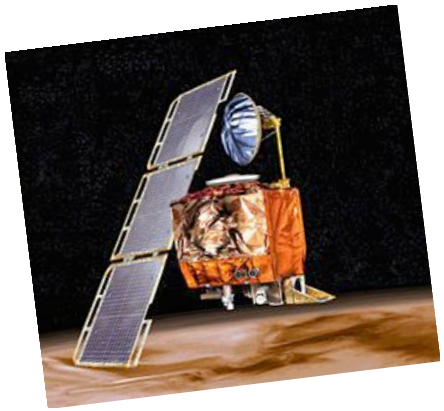
Errores

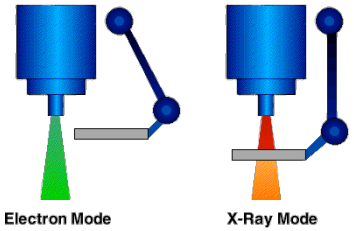
Errores famosos

Pentium:
FDIV

Ariane 5:
64 bits fp
vs 16 bits int

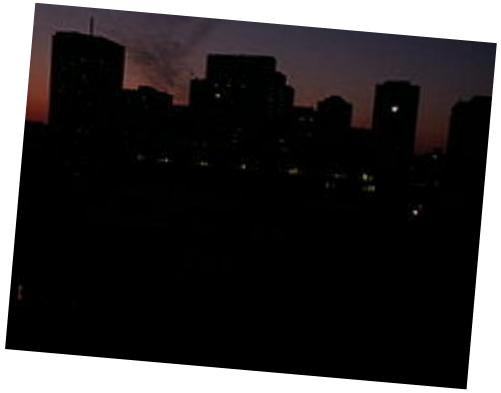






Therac-25:
Condición de
carrera

Mars Climate
Orbiter:
Métrico vs Imperial



Northeast blackout
en 2003:
Condición de carrera

Heartbleed:
Integridad/Confidencialidad

CONICET

UNC

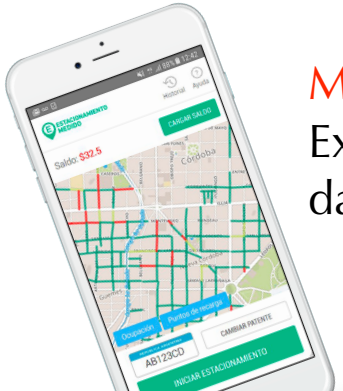
Más errores

911 blackout:
MAX value reached

Nest Thermostat:
Drenado de batería

Nissan airbag:
Sensado incorrecto

Boeing 737 MAX 8:
Sensado incorrecto



Movypark:
Exposición de datos personales

Tesla/Uber/Google self-driving car:
aprendizaje con limitaciones??

CONICET

UNC

El problema de la corrección

$$Sistema \models Propiedad$$

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)

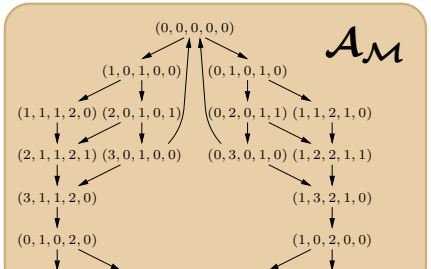
Model Checking

$\mathcal{M}$

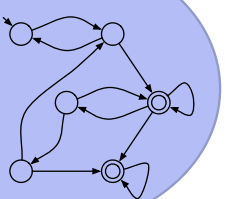
```
int y1 = 0;
int y2 = 0;
short in_critical = 0;

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
}
```

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



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

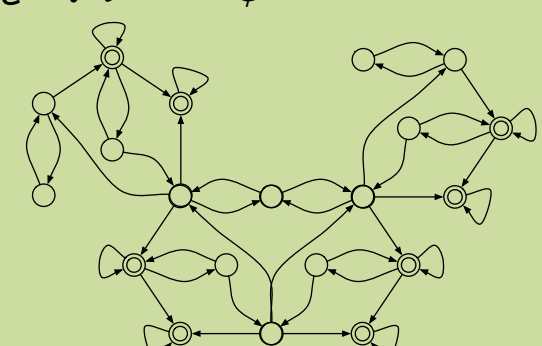


ϕ : $\Box\Diamond crit_1 \wedge \Box\Diamond crit_2$

¿ $\mathcal{M} \models \phi$?

El problema se reduce a análisis de grafos

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



CONICET

UNC

Limitaciones del Model Checking

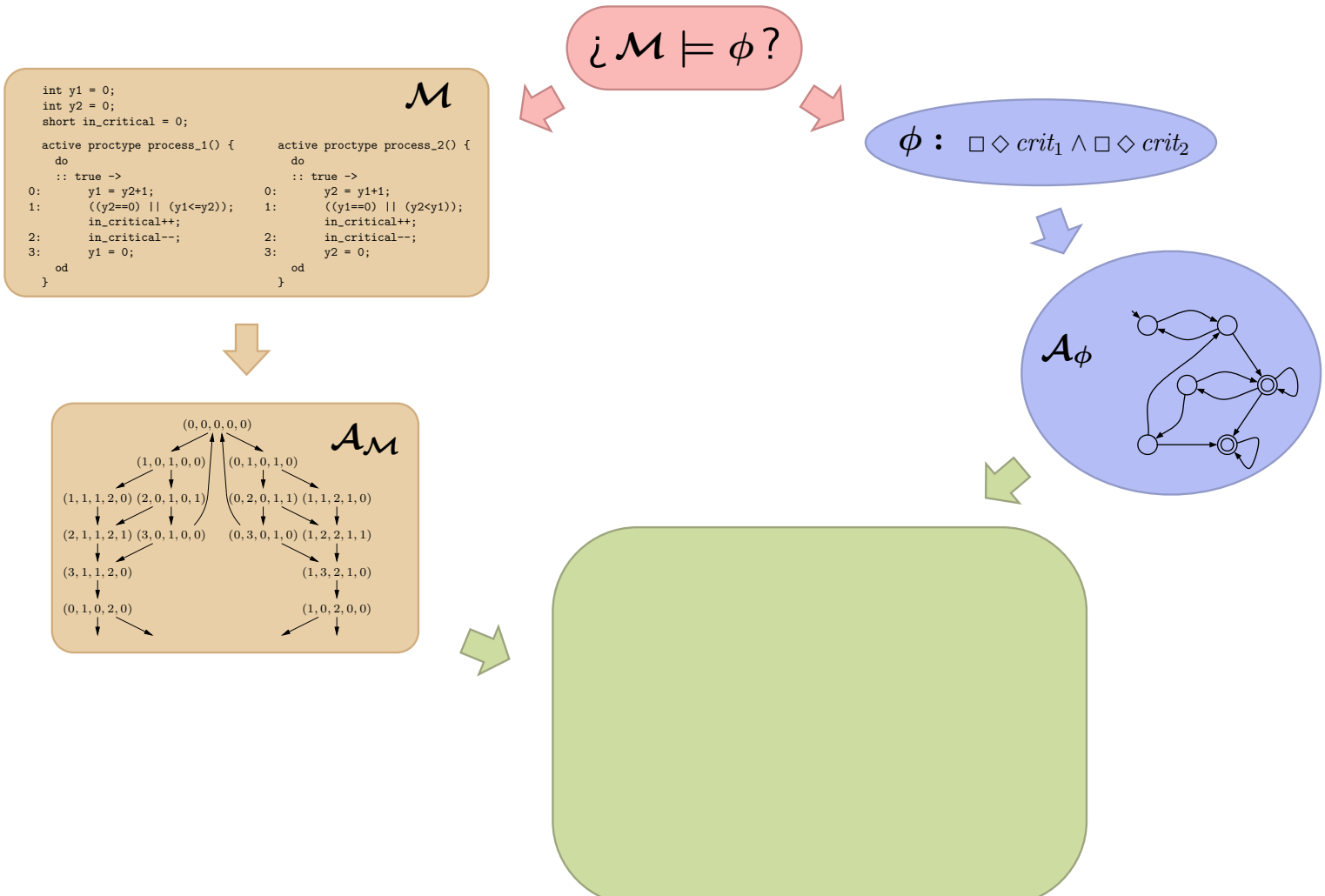
- Muchos algoritmos proponen (mejores) soluciones utilizando aleatoriedad.
  - Leader Election Protocol en IEEE 1394 “Firewire”
  - Binary Exponential Backoff en IEEE 802.3 “Ethernet”

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”

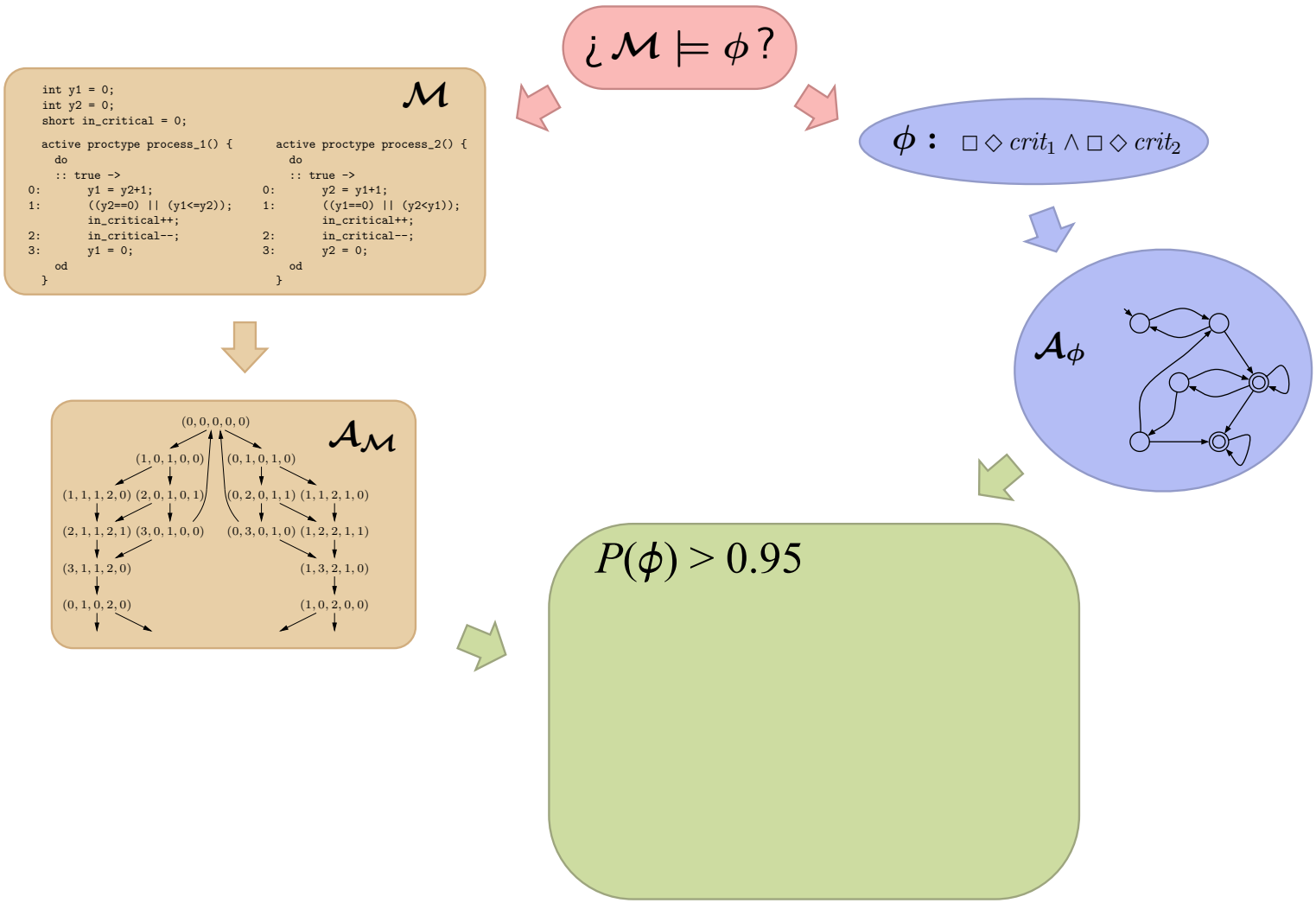
Model Checking Cuantitativo



CONICET

UNC

Model Checking Cuantitativo



CONICET

UNC

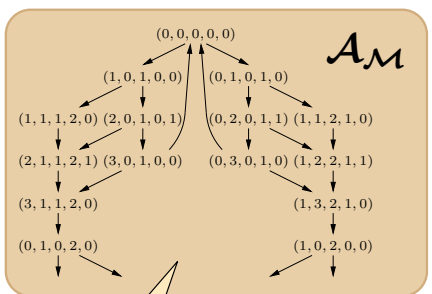
Model Checking Cuantitativo

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
}
```

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

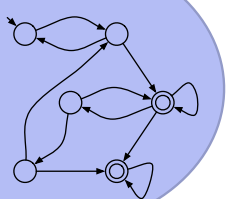


Proceso de decisión de Markov

$¿\mathcal{M} \models \phi?$

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

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



$P(\phi) > 0.95$

CONICET

UNC

Model Checking Cuantitativo

Incluye primitivas de aleatoriedad

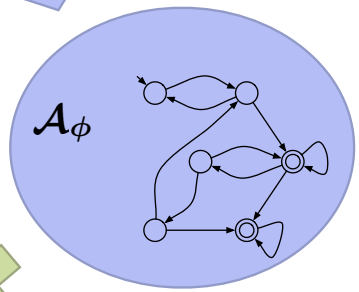
$\mathcal{M}$

```
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
}
```

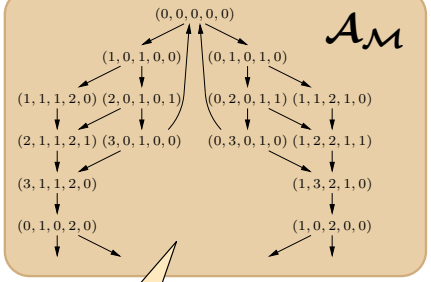
$¿\mathcal{M} \models \phi?$

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

$\mathcal{A}_\phi$

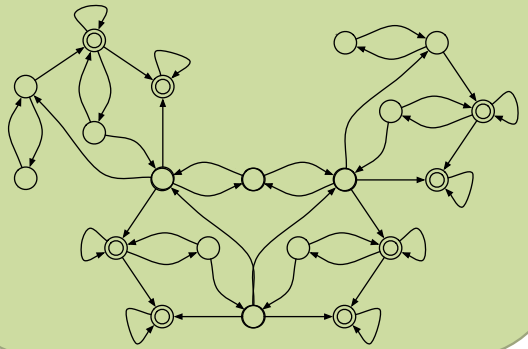


$\mathcal{A}_\mathcal{M}$



Proceso de decisión de Markov

$P(\phi) > 0.95$



Proceso de decisión de Markov

CONICET

UNC

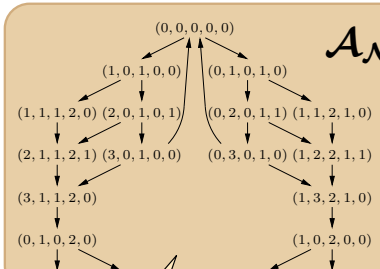
Model Checking Cuantitativo

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
}
```

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

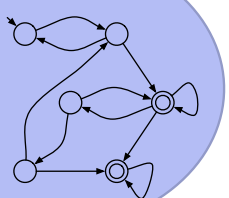


Proceso de decisión de Markov

$¿\mathcal{M} \models \phi?$

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

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



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

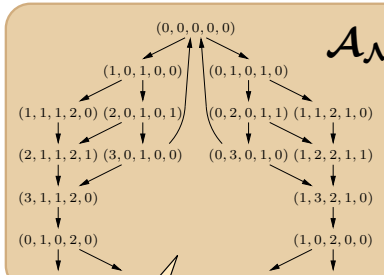
Model Checking Cuantitativo

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
}
```

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

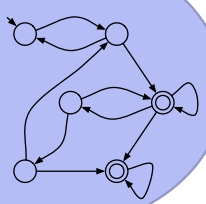


Proceso de decisión de Markov

¿ $\mathcal{M} \models \phi$?

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

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



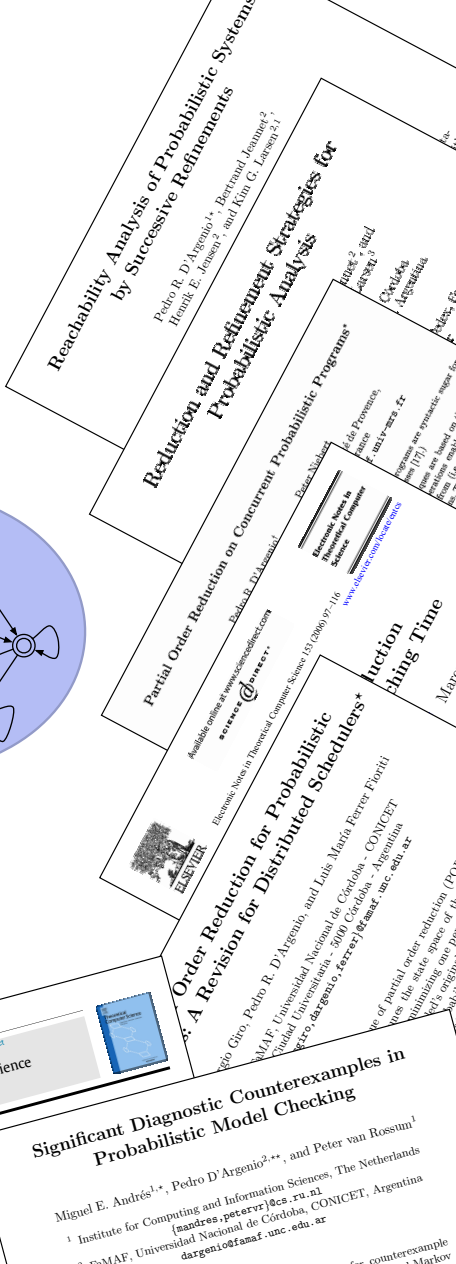
Se reduce a un problema de optimización lineal

$P(\phi) > 0.95$

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

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

$$x_s = 0 \quad \text{if } s \notin Pr$$



Fallas

Fallas

CONICET

UNC

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?

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

(Algunas) Técnicas de análisis

- Model checking cuantitativo

Ya lo vimos

(Algunas) Técnicas de análisis

- Model checking cuantitativo

Ya lo vimos

- Simulación por eventos discretos

(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

(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

(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

(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

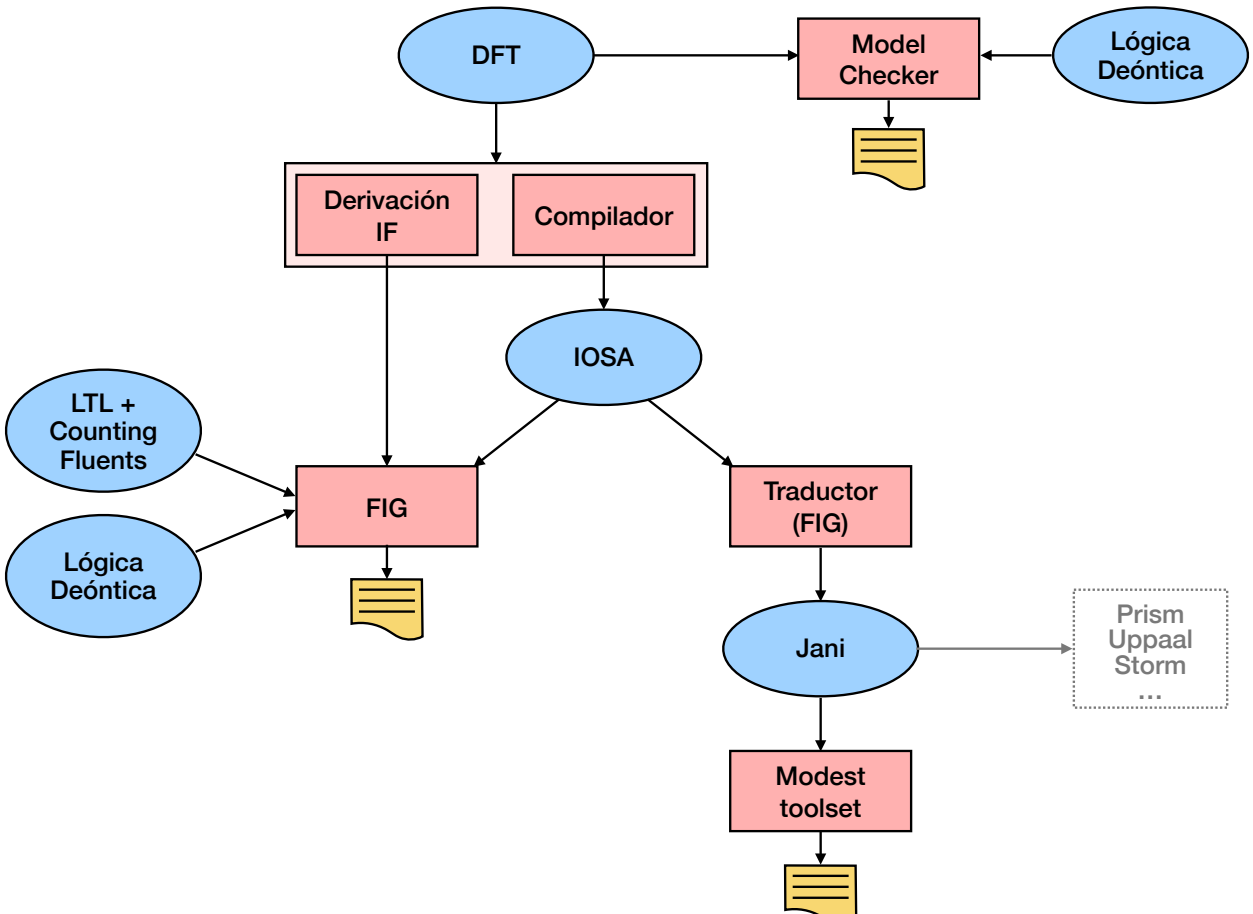
  Es una variante específica de la simulación

  + no-determinismo

Agencia I+D+i

UNC Universidad Nacional de Córdoba

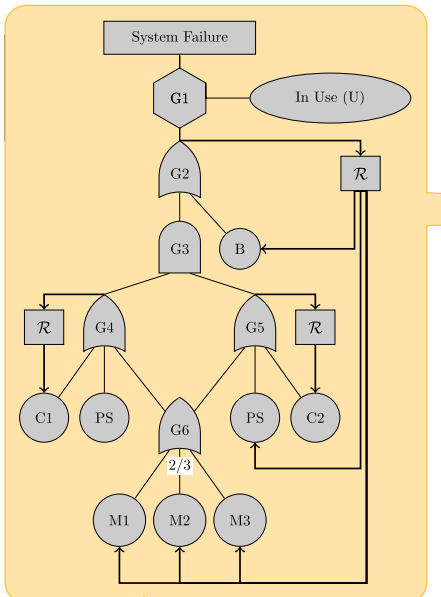
Proyectos RAFTSys y ARES



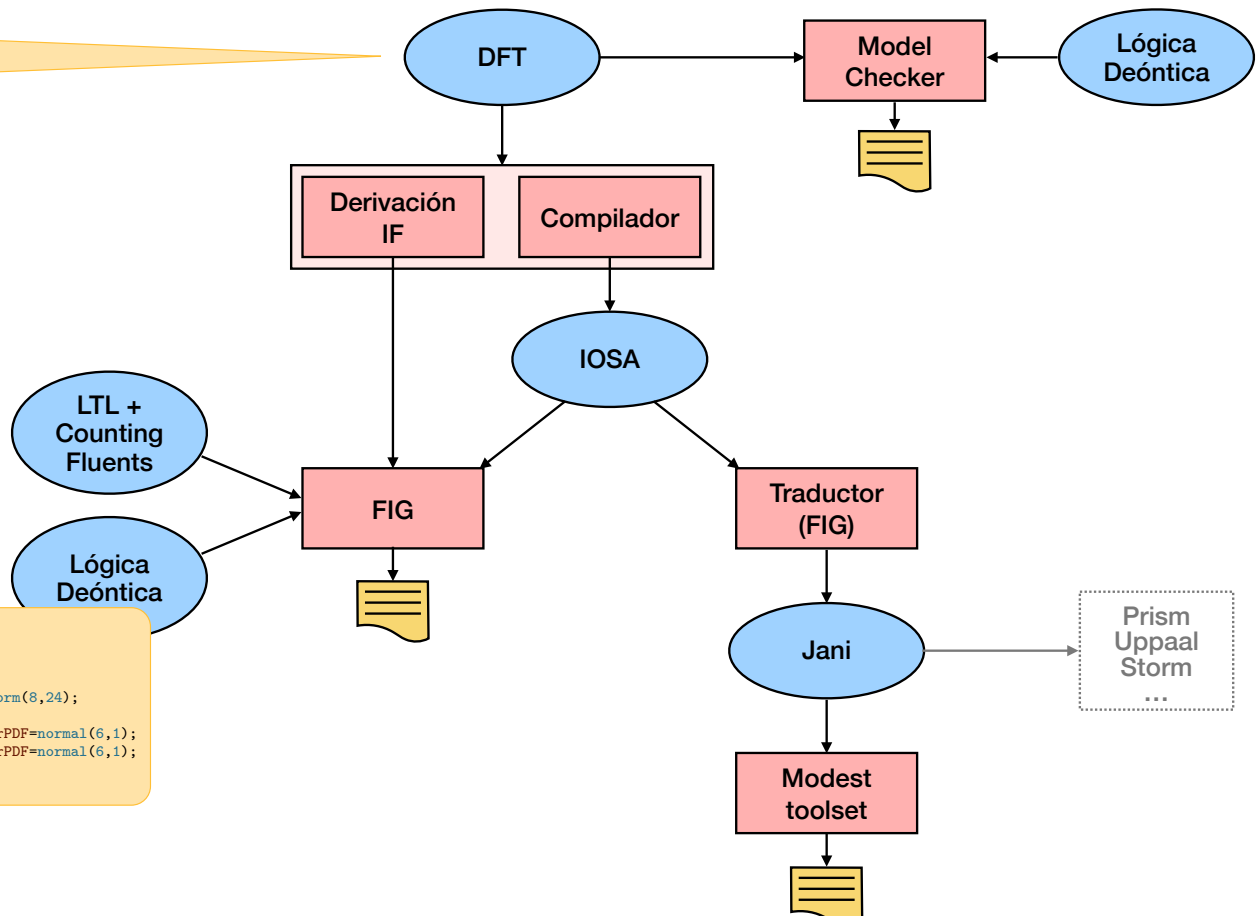
CONICET

UNC

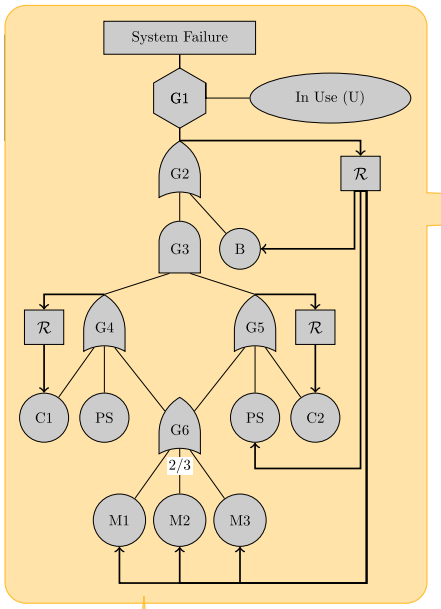
Proyectos RAFTSys y ARES




```
toplevel "G1";
"G1" and "G2" "B";
"G2" wsp "A" "S1" "S2";
"B" EXT_failPDF=rayleigh(6.0e-2) EXT_repairPDF=uniform(8,24);
"A" lambda=1.11e-3 EXT_repairPDF=normal(6,1);
"S1" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"S2" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"RB" repairbox_priority "B" "S2" "S1" "A";
```



Proyectos RAFTSys y ARES




```
toplevel "G1";
"G1" and "G2" "B";
"G2" wsp "A" "S1" "S2";
"B" EXT_failPDF=rayleigh(6.0e-2) EXT_repairPDF=uniform(8,24);
"A" lambda=1.11e-3 EXT_repairPDF=normal(6,1);
"S1" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"S2" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"RB" repairbox_priority "B" "S2" "S1" "A";
```

DFT

Model Checker

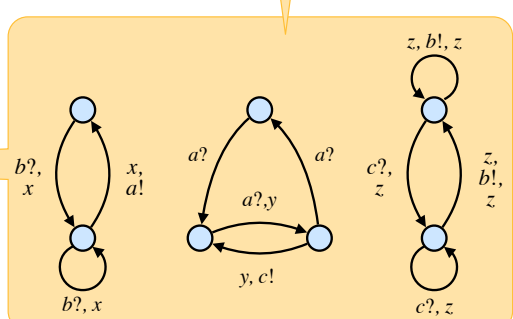
Lógica Deóntica

```
module BE
 fc, rc : clock;
 inform : [0..2] init 0;
 broken : [0..2] init 0;

 [fl!] broken=0 @ fc -> (inform'=1) & (broken'=1);
 [r??] broken=1 -> (broken'=2) & (rc'=γ);
 [up!] broken=2 @ rc -> (inform'=2) &
                        (broken'=0) & (fc'=μ);

 [f!!] inform=1 -> (inform'=0);
 [u!!] inform=2 -> (inform'=0);
endmodule
```

Derivación IF

Compilador

IOSA



LTL + Counting Fluents

Lógica Deóntica

FIG

Traductor (FIG)

Jani

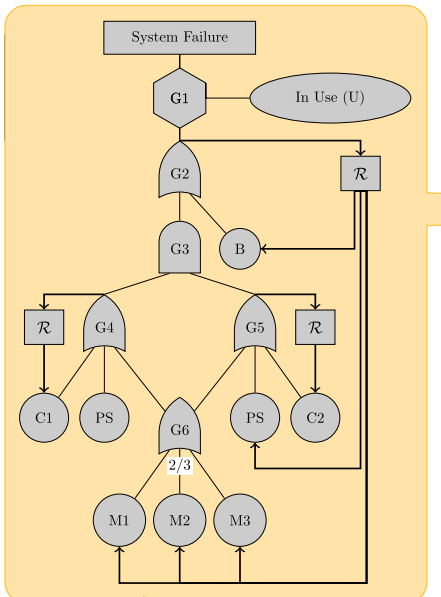
Prism
Uppaal
Storm
...

Modest toolset

CONICET

UNC

Proyectos RAFTSys y ARES




```
toplevel "G1";
"G1" and "G2" "B";
"G2" wsp "A" "S1" "S2";
"B" EXT_failPDF=rayleigh(6.0e-2) EXT_repairPDF=uniform(8,24);
"A" lambda=1.11e-3 EXT_repairPDF=normal(6,1);
"S1" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"S2" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"RB" repairbox_priority "B" "S2" "S1" "A";
```

DFT

Model Checker

Lógica Deóntica

Derivación IF

Compilador

IOSA

LTL + Counting Fluents

Lógica Deóntica

FIG

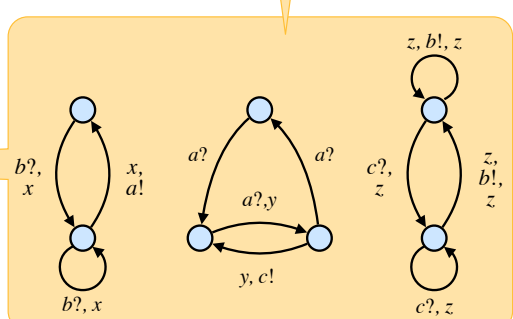
Traductor (FIG)

Jani

Prism
Uppaal
Storm
...

Modest toolset

```
module BE
 fc, rc : clock;
 inform : [0..2] init 0;
 broken : [0..2] init 0;

 [fl!] broken=0 @ fc -> (inform'=1) & (broken'=1);
 [r??] broken=1 -> (broken'=2) & (rc'=γ);
 [up!] broken=2 @ rc -> (inform'=2) &
                        (broken'=0) & (fc'=μ);

 [f!!] inform=1 -> (inform'=0);
 [u!!] inform=2 -> (inform'=0);
endmodule
```



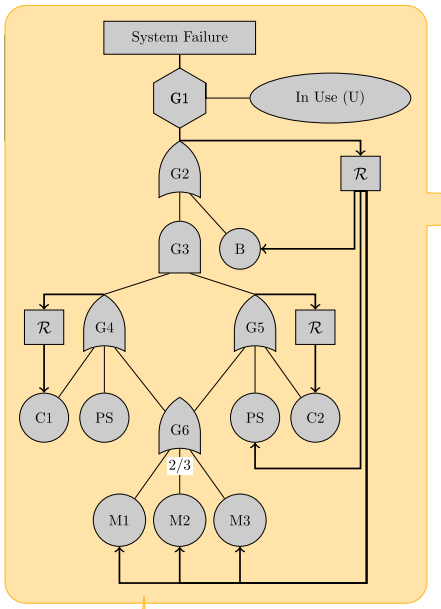
CONICET

UNC

Agencia I+D+i

Proyectos RAFTS

Medición de tolerancia a fallas
+ Juegos estocásticos




```
toplevel "G1";
"G1" and "G2" "B";
"G2" wsp "A" "S1" "S2";
"B" EXT_failPDF=rayleigh(6.0e-2) EXT_repairPDF=uniform(8,24);
"A" lambda=1.11e-3 EXT_repairPDF=normal(6,1);
"S1" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"S2" lambda=0.2 EXT_dormPDF=erlang(9,0.3) EXT_repairPDF=normal(6,1);
"RB" repairbox_priority "B" "S2" "S1" "A";
```

DFT

Model Checker

Lógica Deóntica

Derivación IF

Compilador

IOSA

LTL + Counting Fluents

Lógica Deóntica

FIG

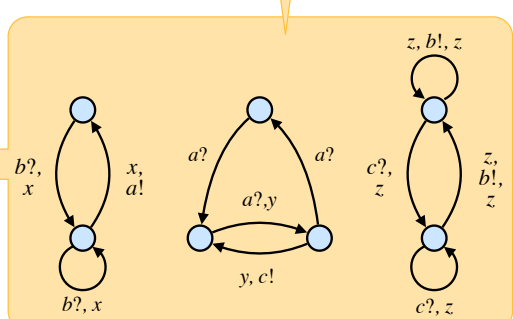
Traductor (FIG)

Jani

Prism
Uppaal
Storm
...

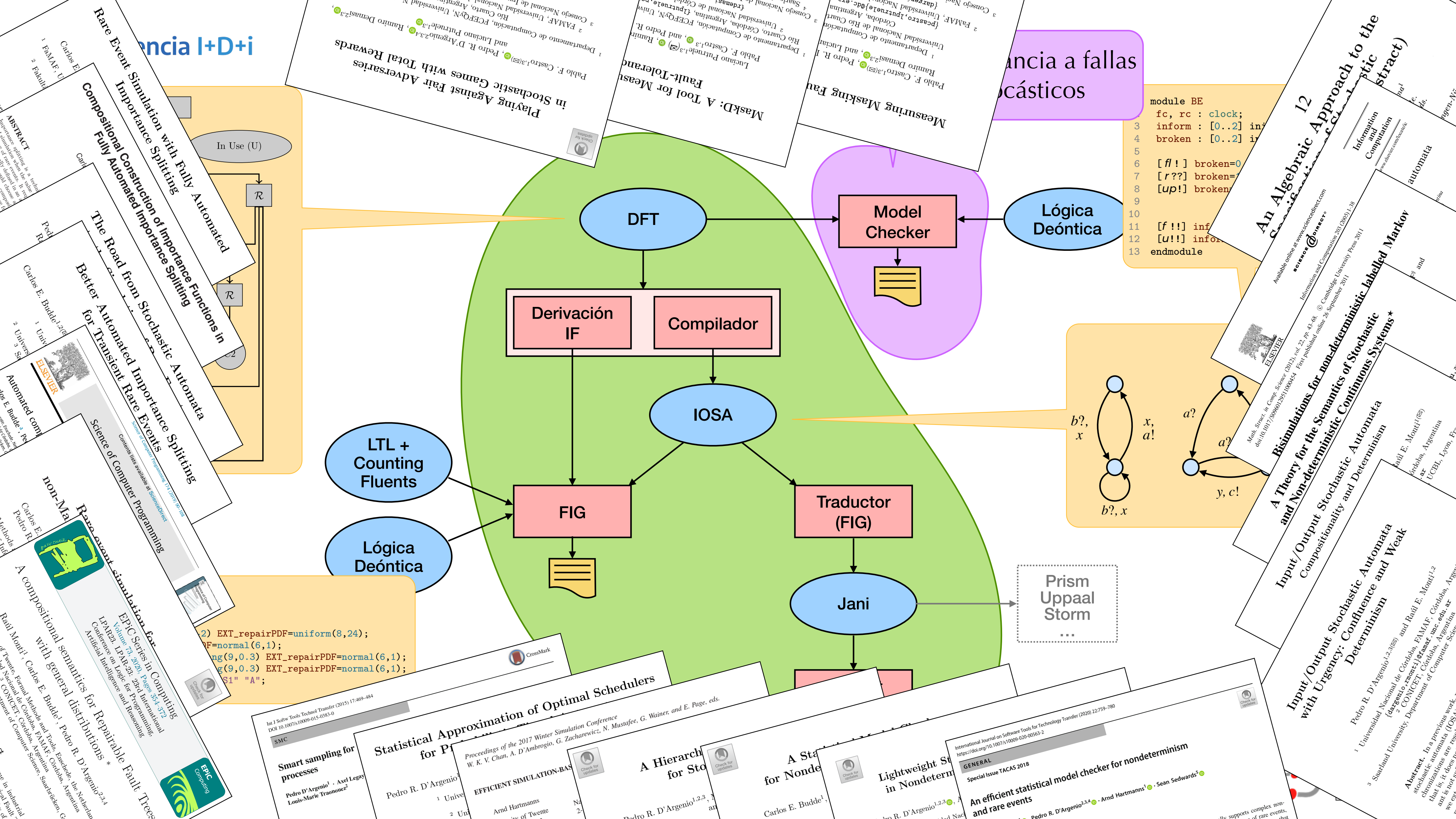
Modest toolset

```
module BE
 fc, rc : clock;
 inform : [0..2] init 0;
 broken : [0..2] init 0;

 [fl!] broken=0 @ fc -> (inform'=1) & (broken'=1);
 [r??] broken=1 -> (broken'=2) & (rc'=γ);
 [up!] broken=2 @ rc -> (inform'=2) &
                        (broken'=0) & (fc'=μ);

 [f!!] inform=1 -> (inform'=0);
 [u!!] inform=2 -> (inform'=0);
endmodule
```



CONICET

UNC

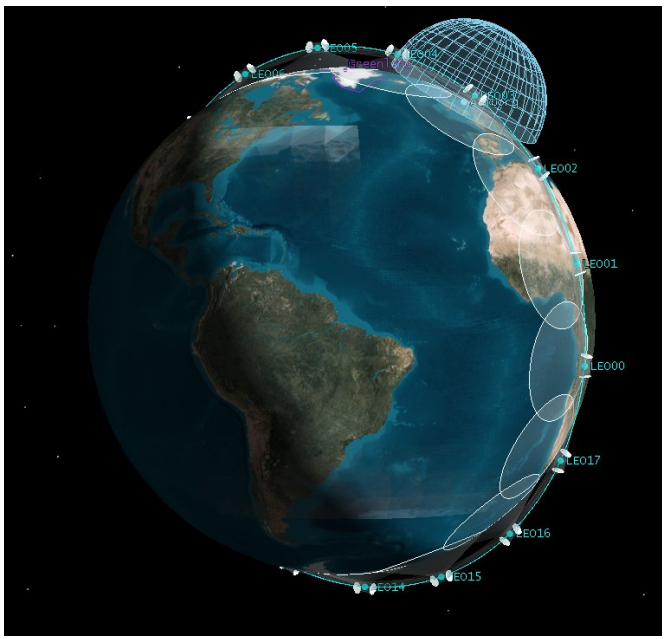

DFT
Derivación IF
Compilador
IOSA
FIG
Traductor (FIG)
Jani
LTL + Counting Fluents
Lógica Deóntica
Model Checker
Lógica Deóntica
Prism Uppaal Storm …
In Use (U)
module BE
fc, rc : clock;
inform : [0..2]
broken : [0..2]
endmodule
Playing Stochastic Games Against Fair Adversaries with Total Rewards
MaskD: A Tool for Measuring Fault-Tolerance
Measuring Masking Fault
Rare Event Simulation with Fully Automated Importance Splitting
Compositional Construction of Importance Functions in Fully Automated Importance Splitting
The Road from Stochastic Automata
Better Automated Importance Splitting for Transient Rare Events
Science of Computer Programming
A compositional semantics for Repairable Fault Trees with general distributions
An Algebraic Approach to the
Bisimulations for non-deterministic labelled Markov
A Theory for the Semantics of Stochastic and Non-deterministic Continuous Systems
Input/Output Stochastic Automata
Compositionality and Determinism
Input/Output Stochastic Automata with Urgency: Confluence and Weak Determinism
Statistical Approximation of Optimal Schedulers
Smart sampling for
An efficient statistical model checker for nondeterminism and rare events
Lightweight St
in Nondeterm

Redes de Satélites

GomSpace (DK)

Redes de Satélites

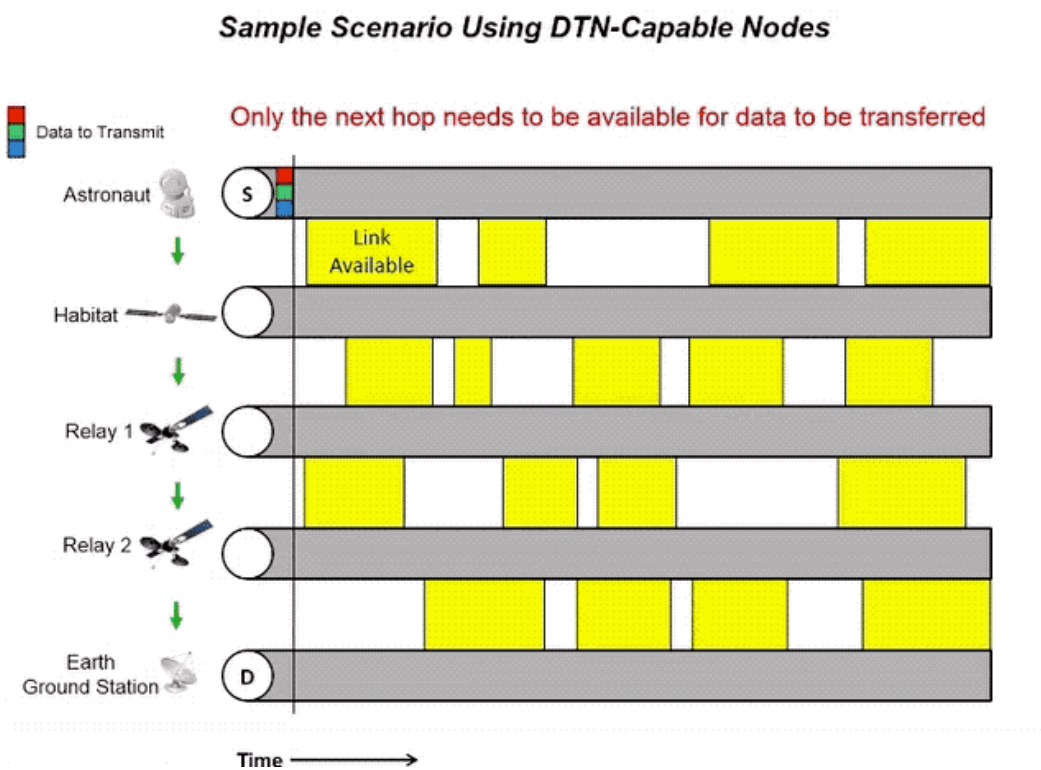
GomSpace (DK)



CONICET

UNC

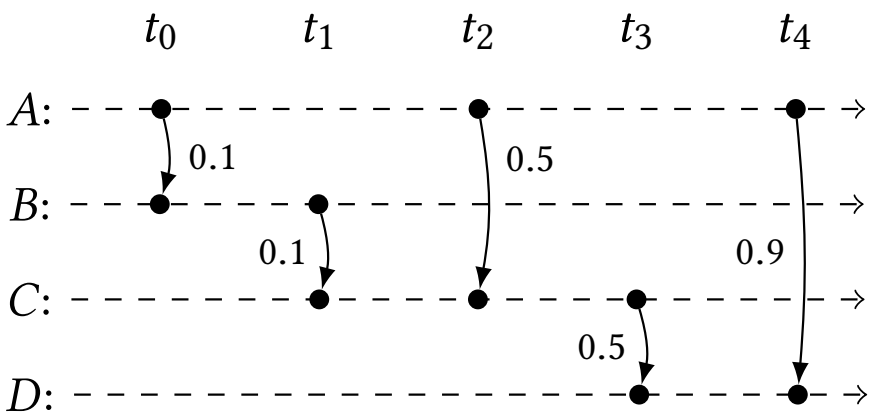
Redes Tolerantes a Demora



CONICET

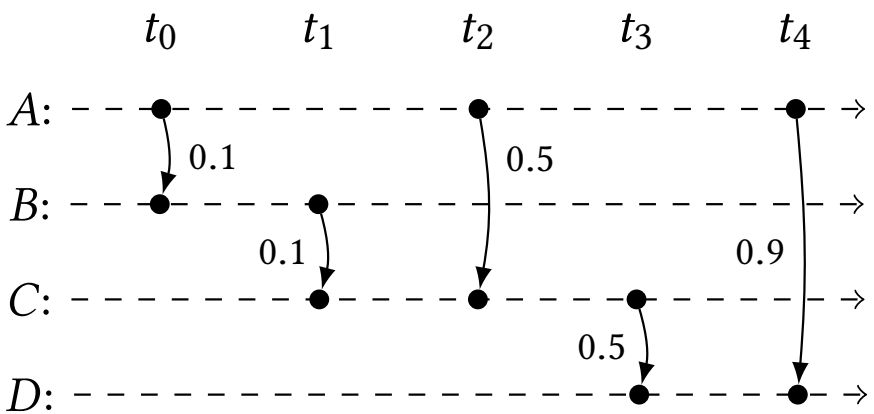
UNC

Redes Tolerantes a Demora
Ruteo basado en Model Checking Cuantitativo

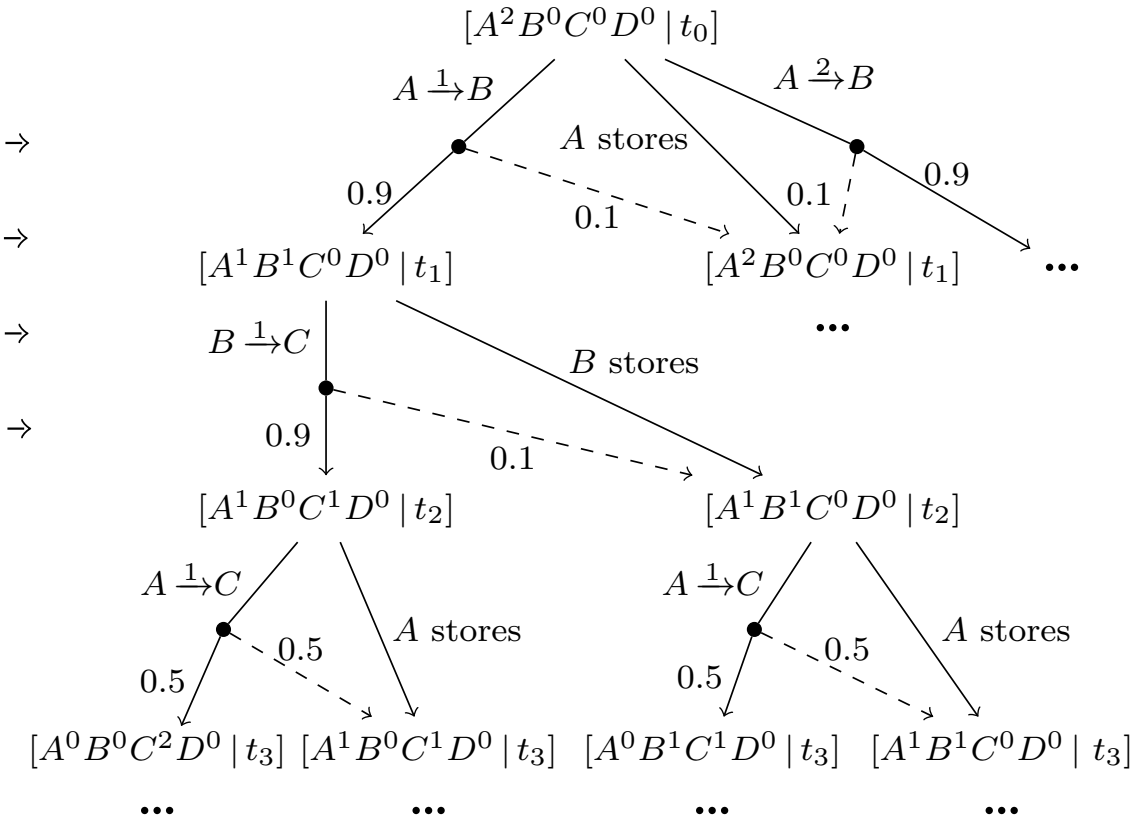


Plan de contacto

Redes Tolerantes a Demora
Ruteo basado en Model Checking Cuantitativo



Plan de contacto

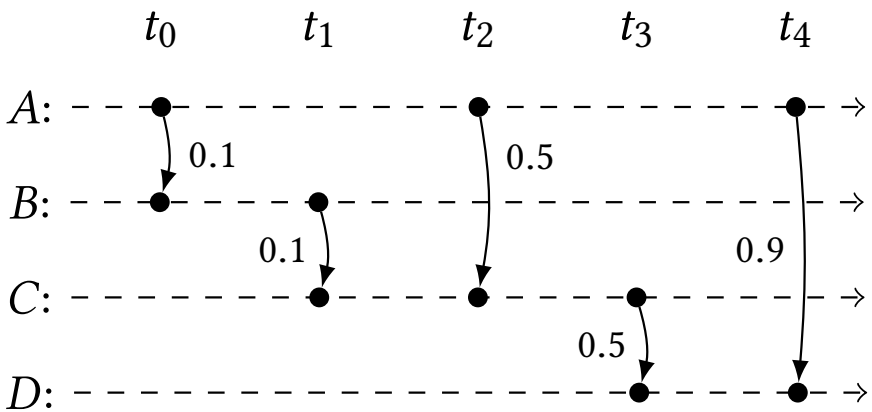


Proceso de decisión de Markov

CONICET

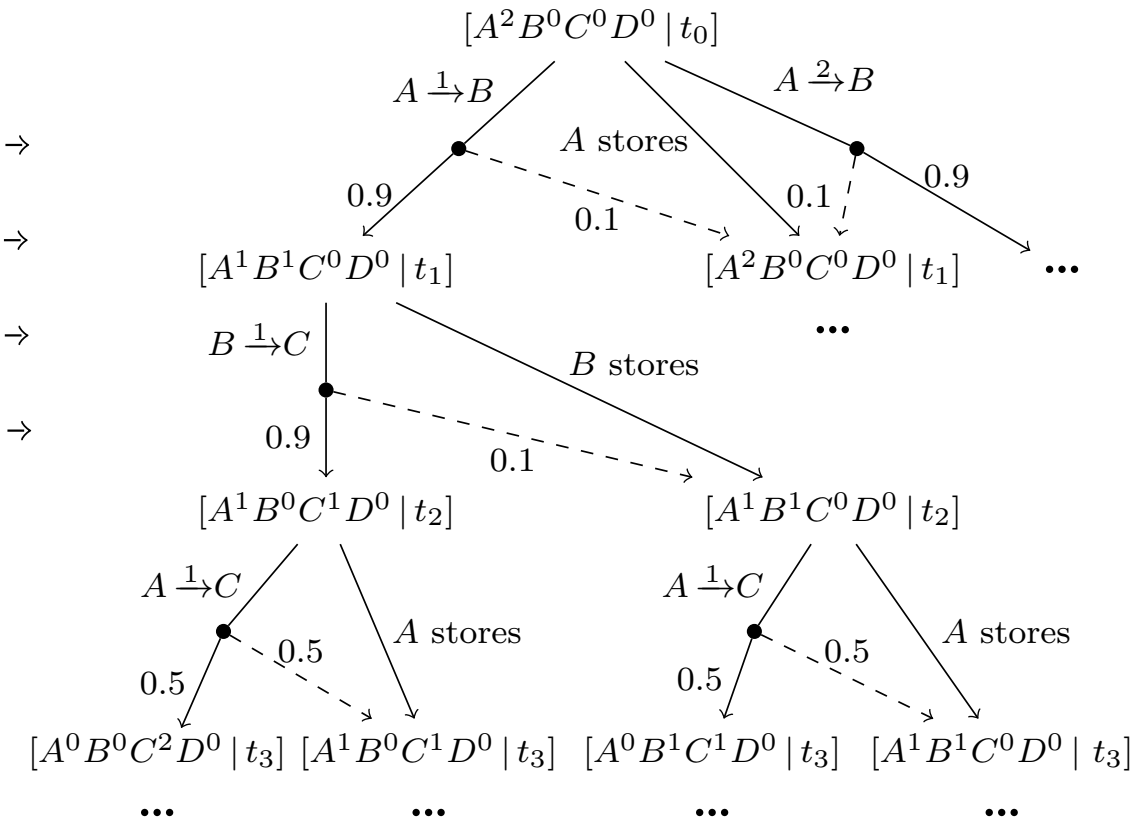
UNC

Redes Tolerantes a Demora
Ruteo basado en Model Checking Cuantitativo



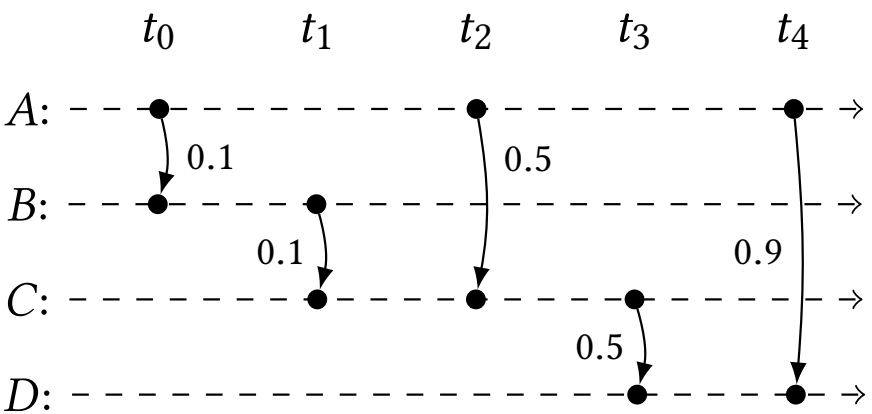
Plan de contacto

- RUCOP
- L-RUCOP
- CGR-UCOP



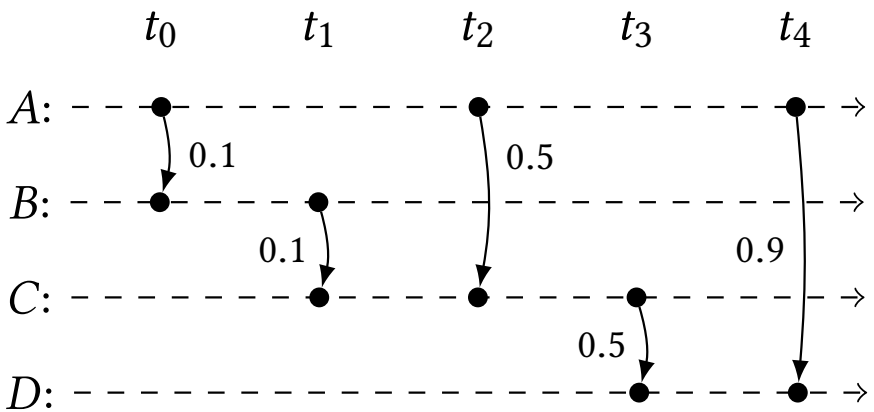
Proceso de decisión de Markov

Redes Tolerantes a Demora
Ruteo basado en Model Checking Estadístico con LSS

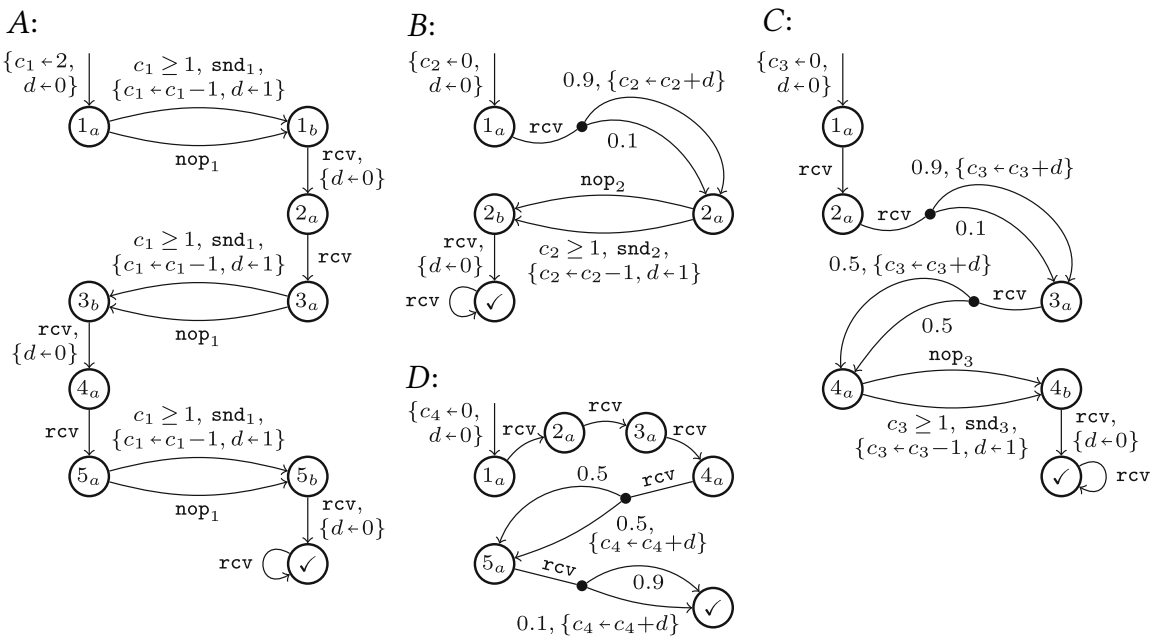


Plan de contacto

Redes Tolerantes a Demora
Ruteo basado en Model Checking Estadístico con LSS



Plan de contacto



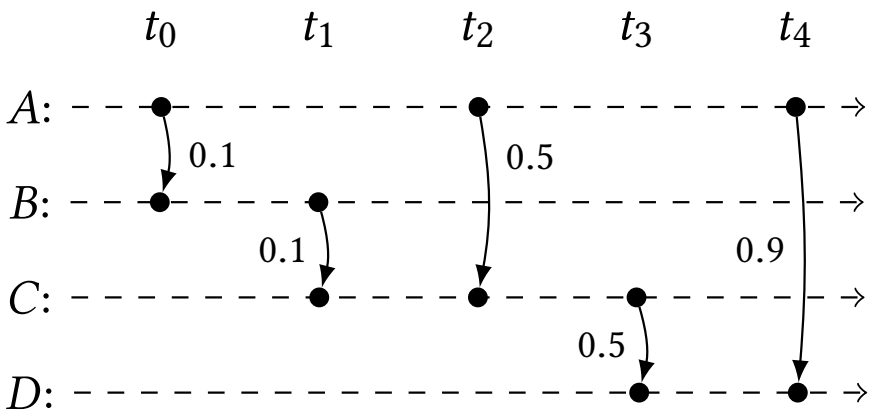
Red de procesos de decisión de Markov

Simulación por Eventos Discretos + LSS

CONICET

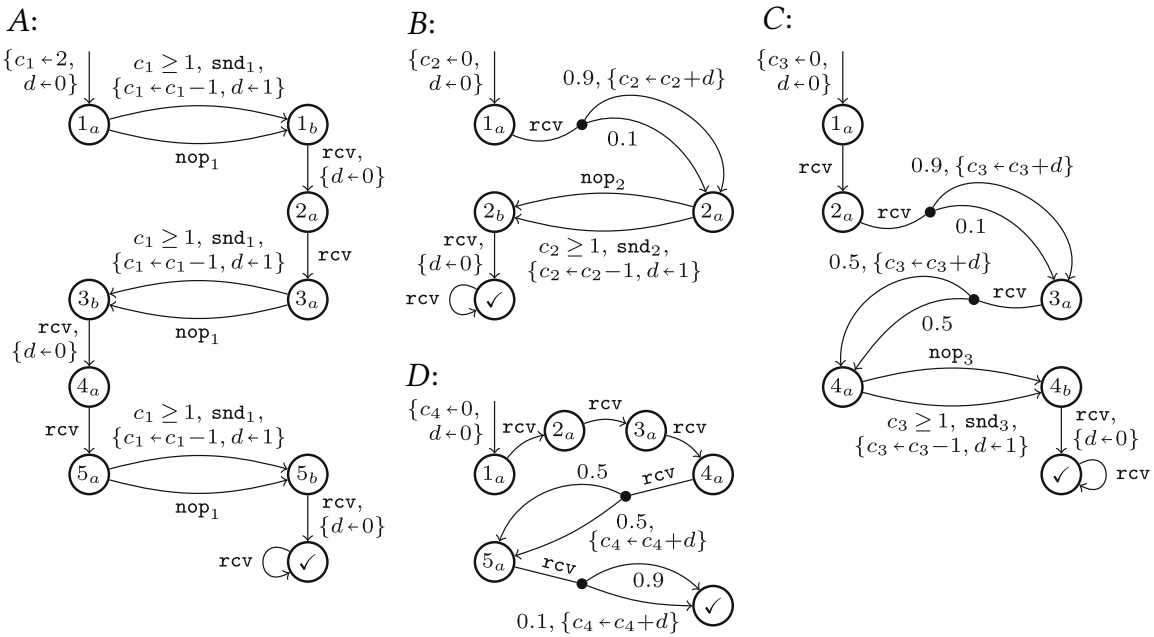
UNC

Redes Tolerantes a Demora
Ruteo basado en Model Checking Estadístico con LSS



Plan de contacto

- LSS
- L-LSS



Red de procesos de decisión de Markov

Simulación por Eventos Discretos + LSS

CONICET

UNC

Redes Tolerantes a Demora

(experimentos)

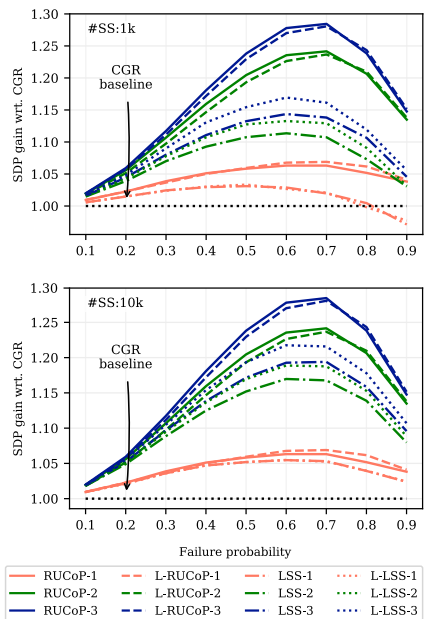


Figure 5: SDP gain over CGR in random networks.

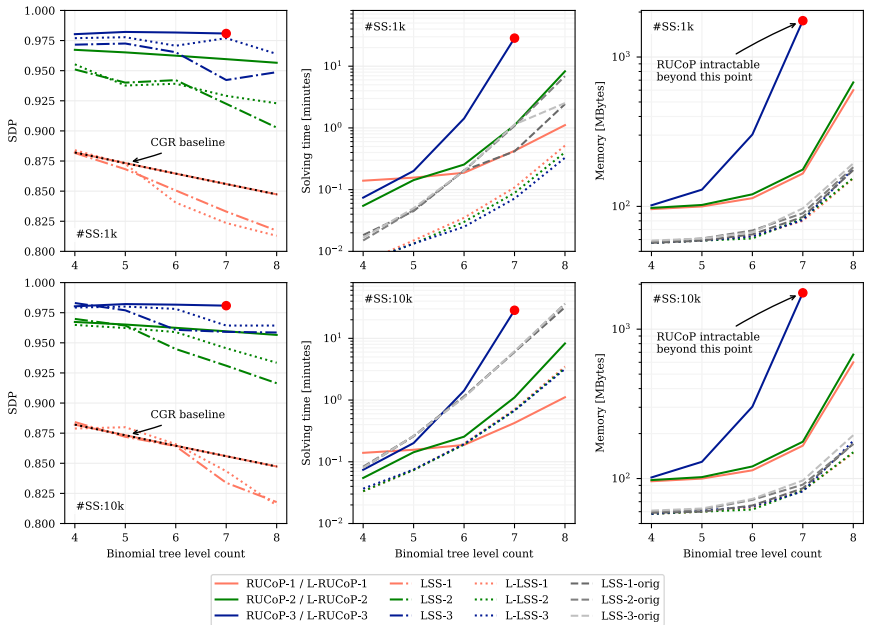


Figure 6: SDP, solving time, and memory for binomial networks with varying complexity (i.e., levels).

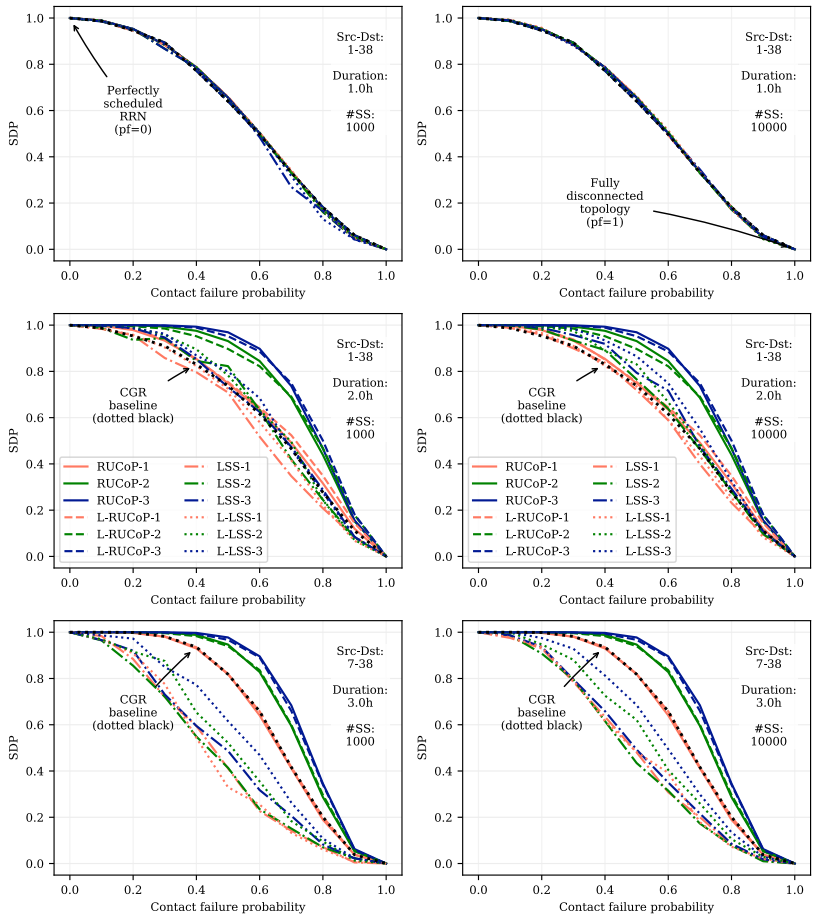


Figure 7: SDP for RRN for different source-target nodes, contact plan duration, and scheduler sampling.

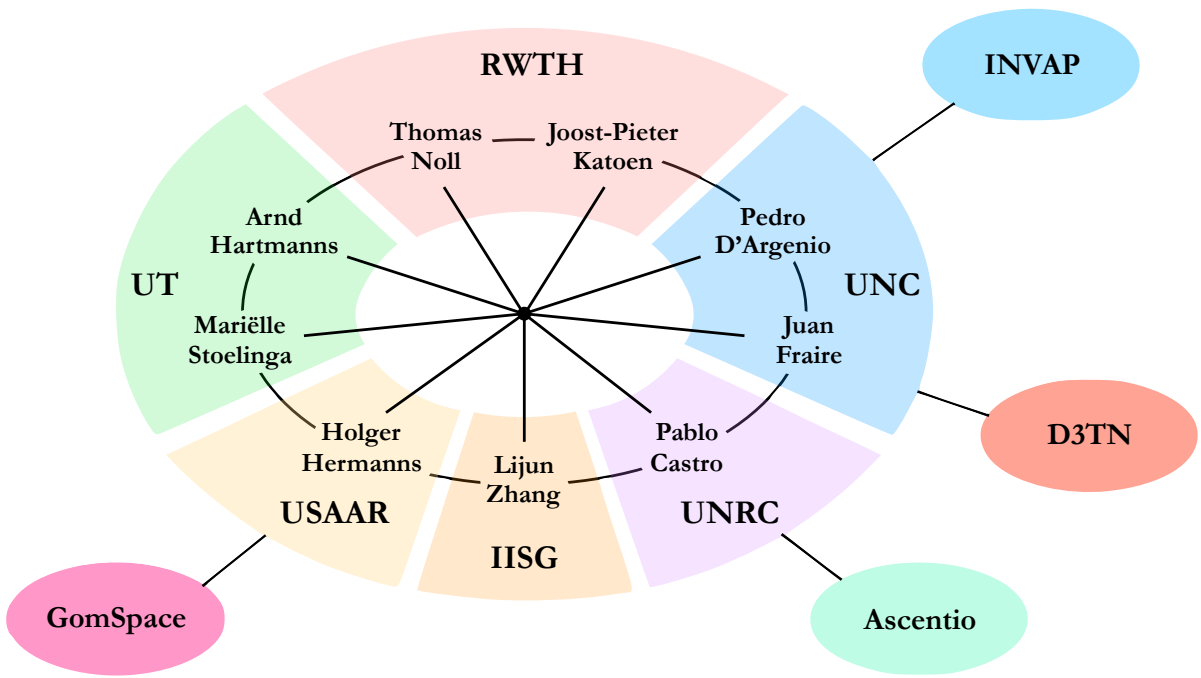
CONICET

UNC

MISSION

European Commission | Horizon 2020 European Union funding for Research & Innovation

Marie Skłodowska-Curie Actions



CONICET

UNC

MISSION

Traditional space paradigm

↓

Resilience & Reliability

New space paradigm

↓

Cost

CONICET

UNC

MISSION

Traditional space paradigm

↓

Resilience & Reliability

New space paradigm

↓

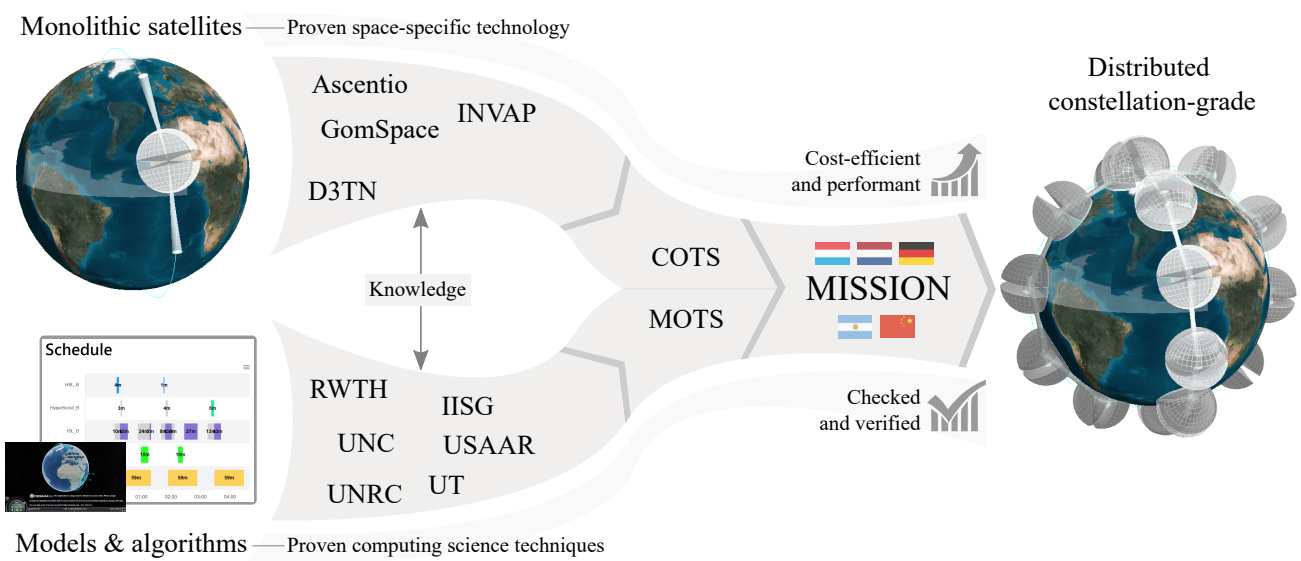
Cost

+

Resilience & Reliability

Links & Network

Energy & Resources



Chantadas

Chantadas abusivas

Estrategias de
"lock in"

CONICET

UNC

Chantadas famosas

CONICET

UNC

Chantadas famosas

Chantadas famosas

› Lifestyle › Cars › News

Diesel emissions scandal: Fiat under investigation

International Business Times

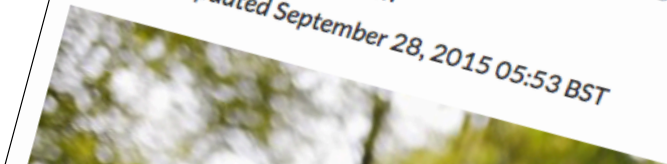
Economy | Companies | Markets | Finance | Regulation

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



CONICET

UNC

Chantadas famosas

› Lifestyle › Cars › News

Diesel emissions scandal: Fiat under investigation

Germany orders Porsche recall over diesel emissions cheating

May 18, 2018

Econ | Markets | Finance | Regulation

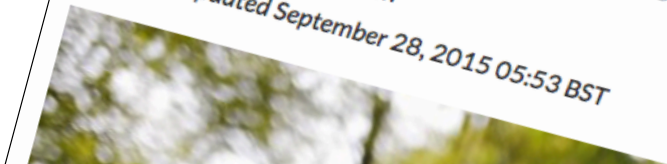
International Business Times

Business | Companies

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

By Karthick Arvinth

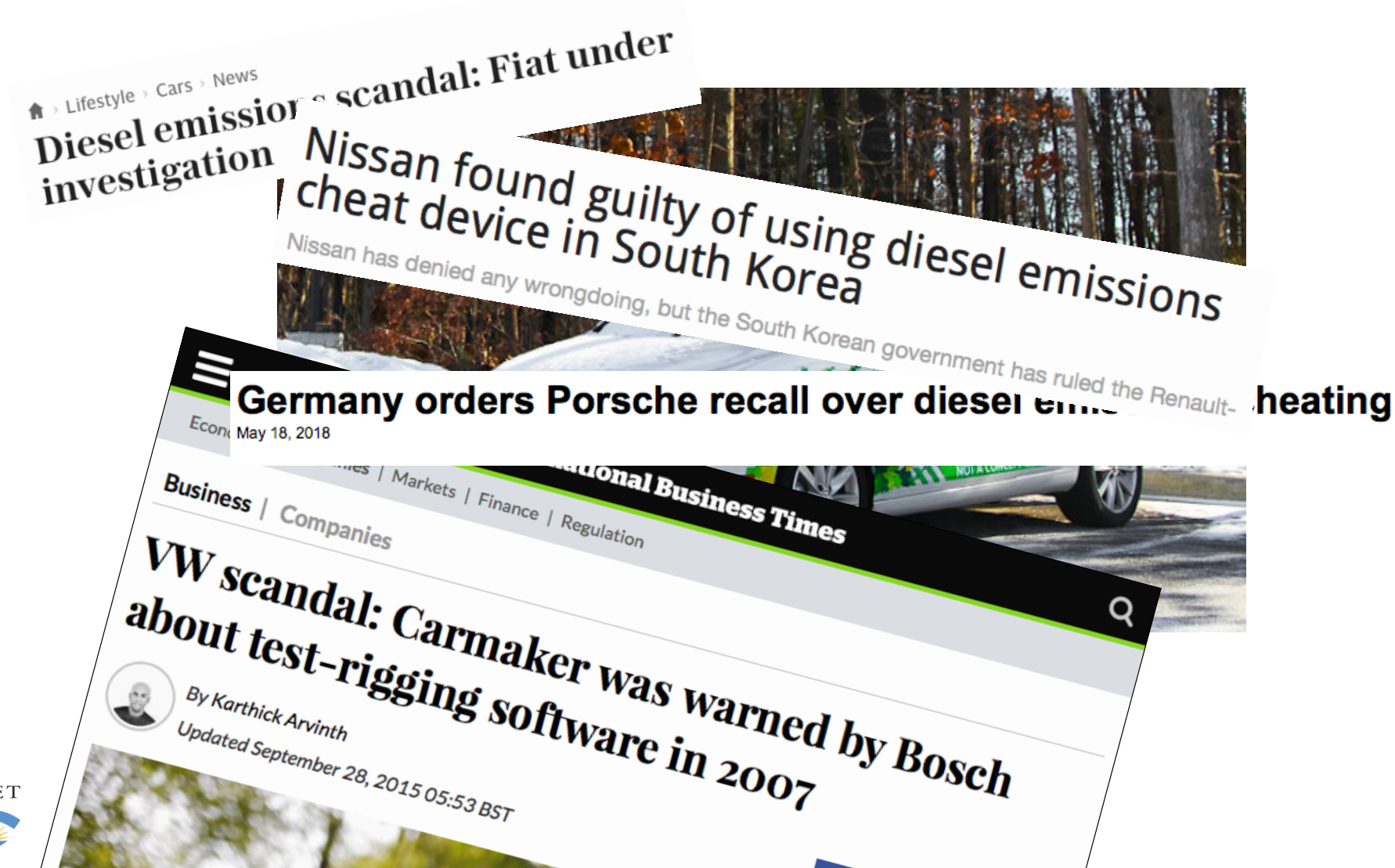
Updated September 28, 2015 05:53 BST



CONICET

UNC

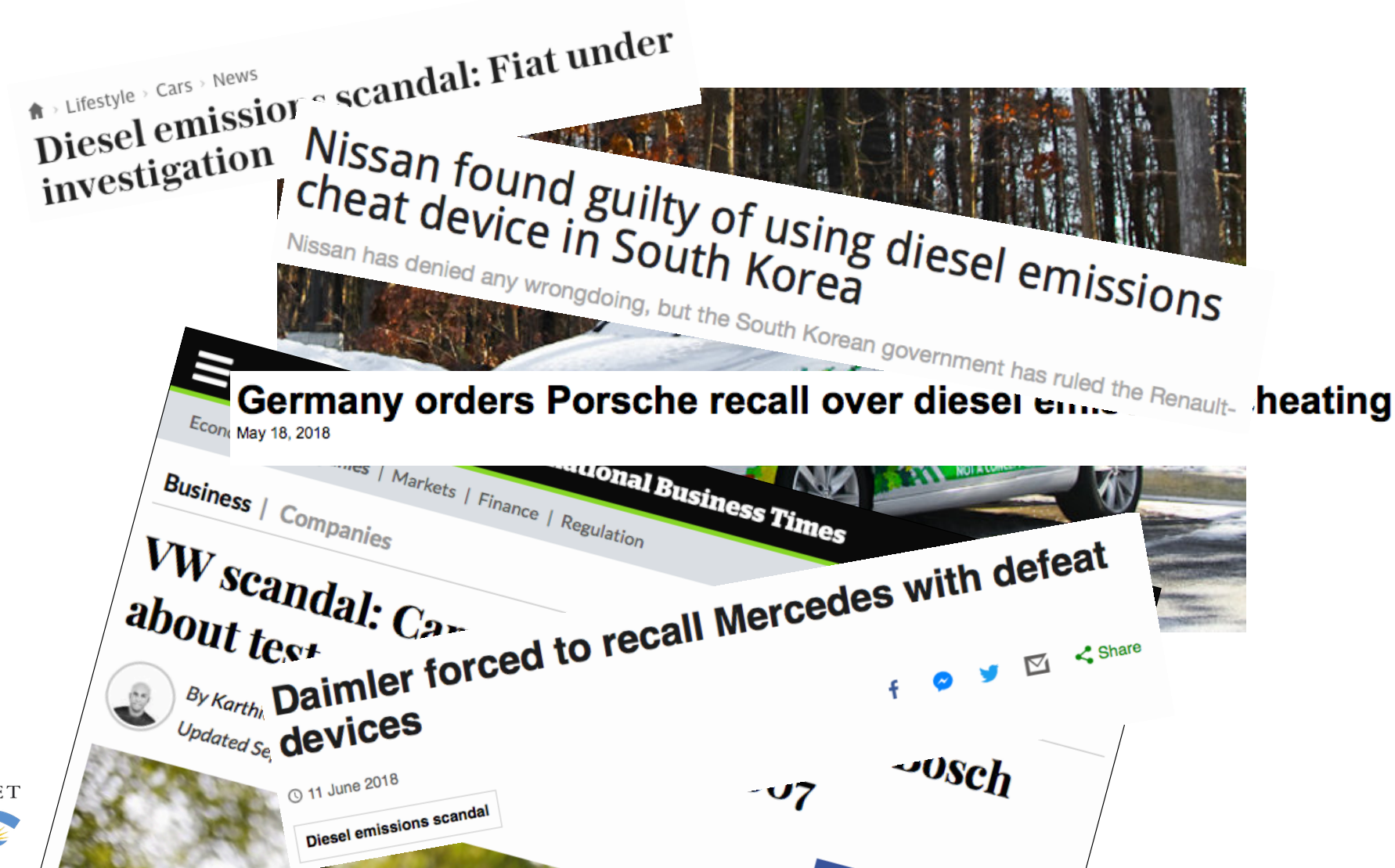
Chantadas famosas



CONICET

UNC

Chantadas famosas



CONICET

UNC

Chantadas f

CONICET

UNC

Chantadas f

CONICET

UNC

Chantadas f

CONICET

UNC

Chantadas f

CONICET

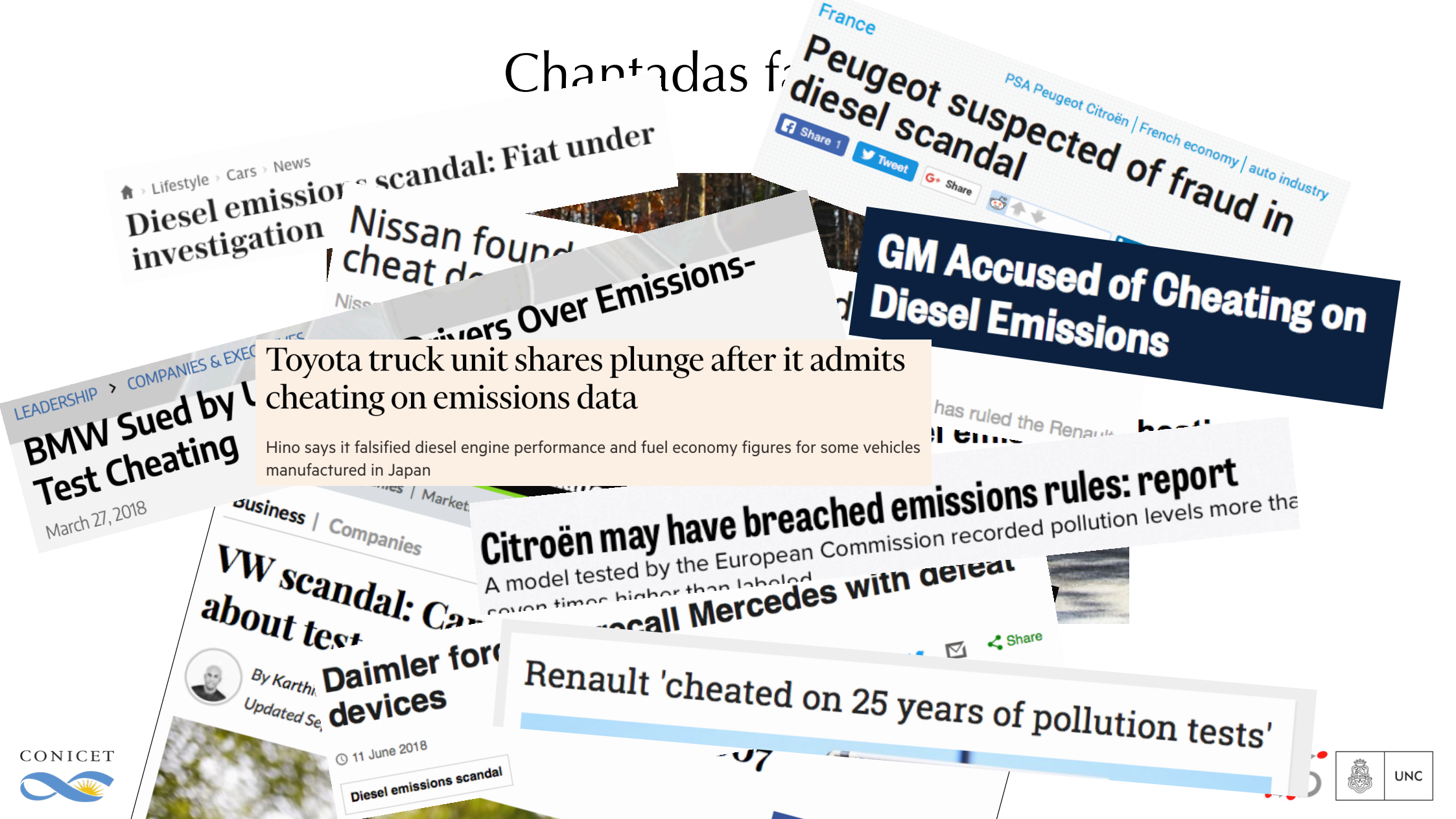
UNC

Chantadas f

CONICET

UNC

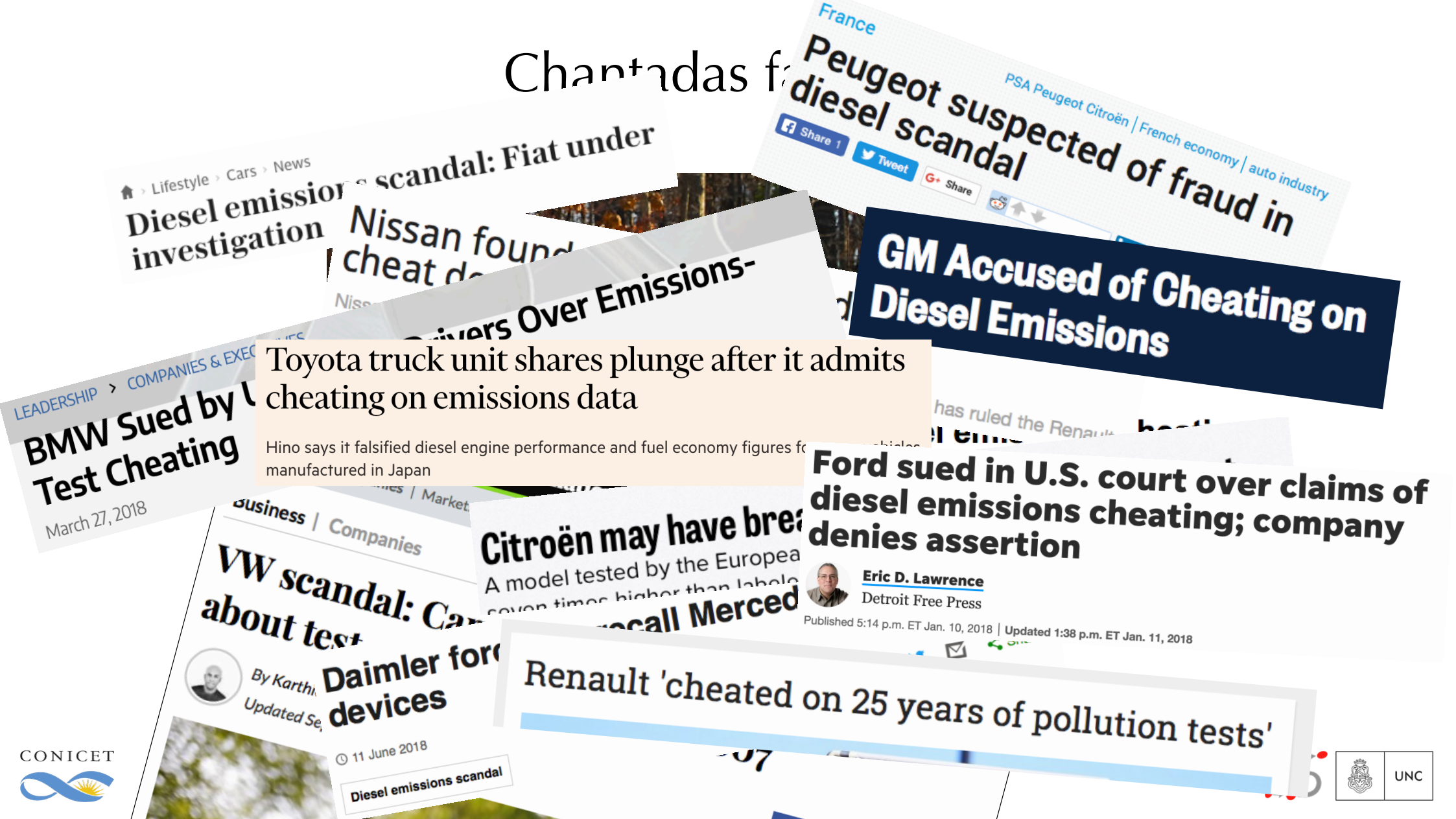
Chantadas f



CONICET

UNC

Chantadas f



CONICET

UNC

Chantadas imperdonables

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

Software doping

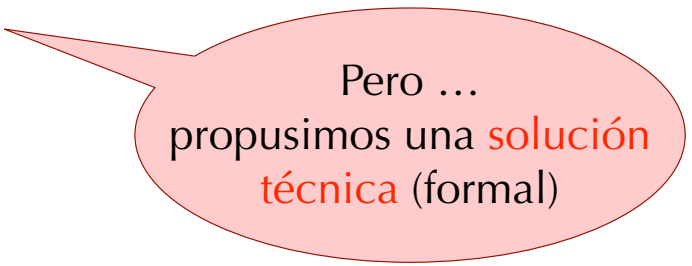
- Es un problema ético y hasta legal.

- 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.



- 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 …
propusimos 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 formalizar

Software doping

definición formal

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

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

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

"se parece"

CONICET

UNC

¿Qué conclusión sacan ustedes?

¿Qué conclusión sacan ustedes?

Las que saco yo:

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

¿Qué conclusión sacan ustedes?

Las que saco yo:

Las técnicas formales (matemáticas) son cruciales para todo esto

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

Objetivo del Grupo de Sistemas Confiables

$$Sistema \models Propiedad$$

Proveer lenguajes y lógicas para modelar sistemas y propiedades, y técnicas y herramientas que implementen la relación de satisfacción, todo dentro de un marco teórico matemático bien definido.

CONICET

UNC

Luchando contra Errores, Fallas y Chantadas para construir Sistemas Confiables

Pedro R. D'Argenio

Grupo de Sistemas Confiables
Universidad Nacional de Córdoba – CONICET (AR)

http://dsg.famaf.unc.edu.ar/

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

CONICET

UNC