100th birthday of Carl Adam Petri

FROM THE TOKEN GAME TO NET COMPLEXITY

RÜDIGER VALK

UNIVERSITY OF HAMBURG

PETRI NETS 2026

HAMBURG, JUNE 22-26 , 2026

CONTENT

- the beautiful and simple nets
- complexity of nets
- efforts to manage complexity
- a summary answer to Jörg Desel's question: "What make Petri nets so special?"

- In my talk, I would like to highlight the structural simplicity and beauty of Petri's nets. Klicken!
- This should not obscure the high level of complexity involved in larger applications. Klicken!
- However, significant efforts are being made to manage this complexity Klicken!
- Finally I will try to give a summary answer to Jörg Desel's question: what make Petri nets so special?
-

C.A. Petri, General Net Theory.
Computing System Design:
Proc. of the Joint IBM University
of Newcastle upon Tyne Seminar,
Sep. 1976, S. 131–169 (1977)

As a conceptual device, net theory should promote (...)
communication and provide means
for introducing new concepts,

in a precise, but nevertheless, easily visualiseable way, hence the importance of graph-theoretical methods, and of the idea of the 'token game' played by many independent actors.

Looking in Petri's paper from 1977 we repeat his own words: …..
lesen.
….
….
which fits in well with the view of nets today, and in the same paper we find …

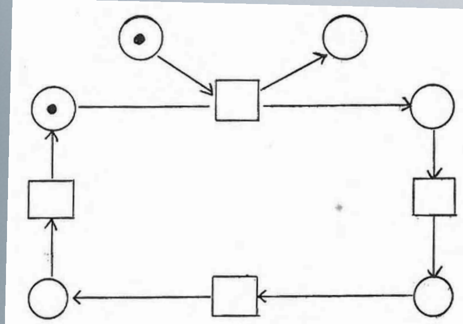


Figure 2b

C.A. Petri, General Net Theory.
Computing System Design:
Proc. of the Joint IBM University
of Newcastle upon Tyne
Seminar,
Sep. 1976, S. 131–169 (1977)

Special Net Theory

S: ◯ state elements (also called places if they can contain more than just one token);

… nets, as we are used to, but also an example of nets with object tokens, as they are formalized much later as Predicate/Transition-Nets or Coloured Nets. Petri is calling this the "**Special Net Theory**"

C.A. Petri, General Net Theory. *Computing System Design: Proc. of the Joint IBM University of Newcastle upon Tyne Seminar, Sep. 1976,* S. 131–169 (1977)

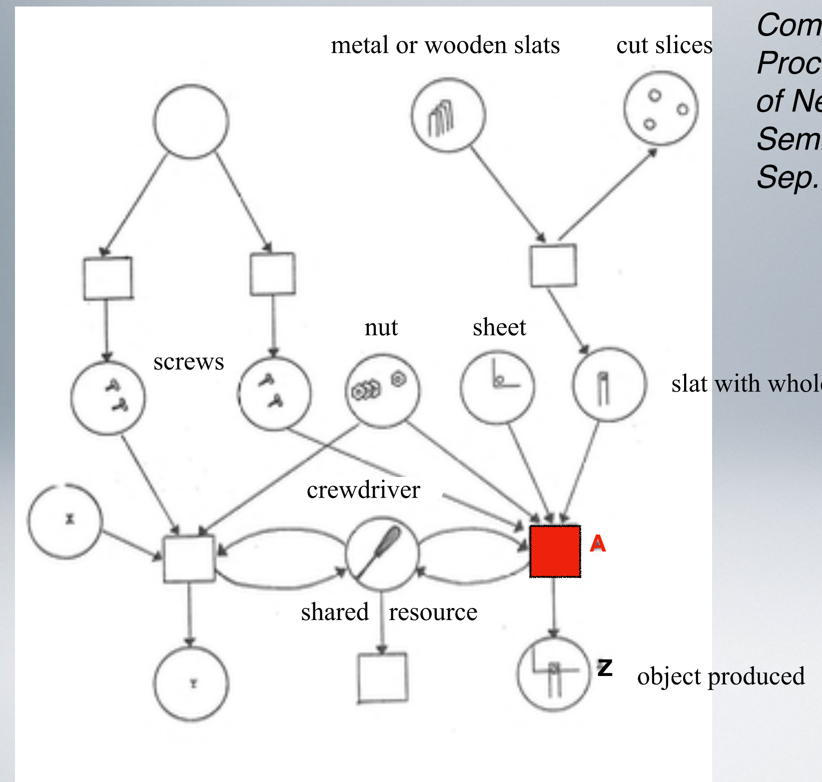


Figure 3

Petri’s description in the text is: Figure 3 shows several production activities, denoted by **boxes**, and several **places** in which resources can reside. We could imagine ourselves as being the worker in box A using one of the **screws** accessible to him, one nut and one sheet of each type, and, with the help of the screwdriver (which he returns after use), producing one object of the type shown in place Z. A very early model of a coloured net.

E.W. Dijkstra. Co-operating sequential processes. In F. Genuys, editor, Programming Languages, pages 43-112. Academic Press N.Y., 1968.

- The Banker's Algorithm -
This situation, when one process can only continue provided the other one is killed first, is called "The Deadly Embrace".

The problem to be solved is: how can we avoid the danger of the Deadly Embrace without being unnecessarily restrictive.

When it came to addressing the problems arising in concurrent processes in the 1960s, Edsger W. Dijkstra was just as - or even more - prominent than Carl Adam Petri. That is why, back then, **in my lectures** I enjoyed modelling the problems he raised using nets. Here, I am using one such example to demonstrate the great advantage of net modelling by their clarity and simplicity.
Remember: clarity and simplicity was **also** Dijkstra’s guiding principle.
The situation described by Dijkstra is as follows:
◉ This situation …read ◉ The problem to be solved …read

Dijkstra's solution:
Inspection, whether a situation is safe amounts to inspection, whether all customer transactions can be guaranteed to be able to finish.

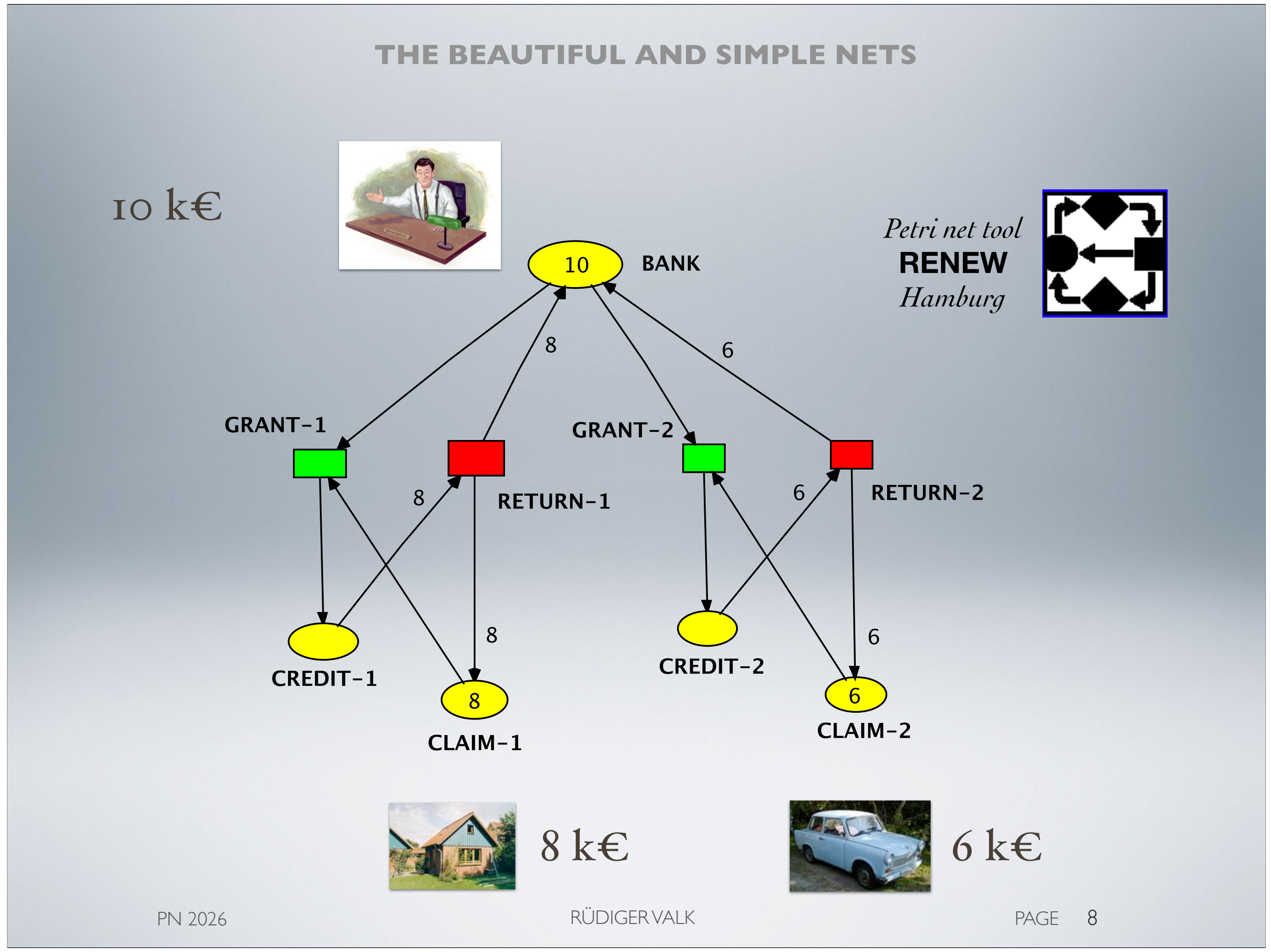
A formalization is given by some pseudo-code.

```
integer free money; Boolean safe; Boolean array
finish doubtful[1:N];
free money:= cash;
for i:= 1 step 1 until N do finish doubtful[i]:=
true;
L:for i:= 1 step 1 until N do
begin if finish doubtful[i] and claim[i] ≤ free
money then
      begin finish doubtful[i]:= false;
            free money:= free money + loan[i];
goto L
      end
end;
if free money = capital then safe:= true else
safe:= false.
```

Nets make the concept much more intuitive.

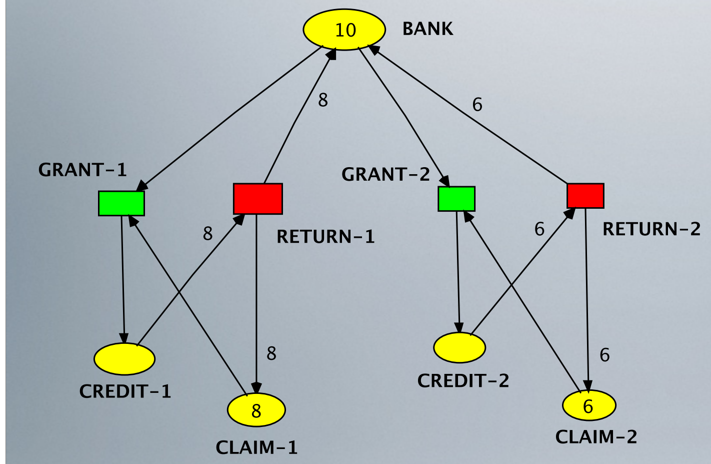
Dijkstra's solution is: Inspection, whether a situation is safe amounts to inspection, whether all customer transactions can be guaranteed to be able to finish.

A formalization is given by some pseudo-code. My opinion: Nets make the concept much more intuitive.



Here is an example involving **2** customers who wish to take out loans of **8** and **6** thousand euros, whilst the bank only has **10** thousand euros available.
In Dijkstra's problem, customers receive loans in increments of **1,000** euros and must repay them once they have been fully repaid.

THE BEAUTIFUL AND SIMPLE NETS

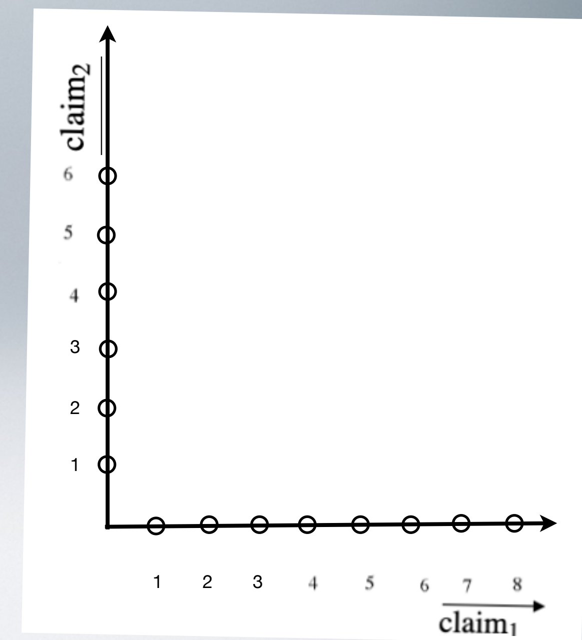


$\forall \mathbf{m} \in \mathbf{R}(\mathcal{N})$

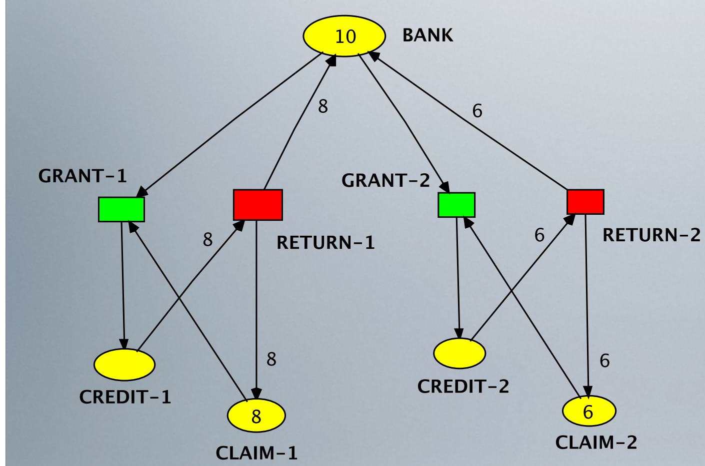
$\mathbf{m}(\mathrm{CREDIT}_1) = 8 - \mathbf{m}(\mathrm{CLAIM}_1)$

$\mathbf{m}(\mathrm{CREDIT}_2) = 6 - \mathbf{m}(\mathrm{CLAIM}_2)$

$\mathbf{m}(\mathrm{BANK}) =$
$10 - \mathbf{m}(\mathrm{CREDIT}_1) - \mathbf{m}(\mathrm{CREDIT}_2)$

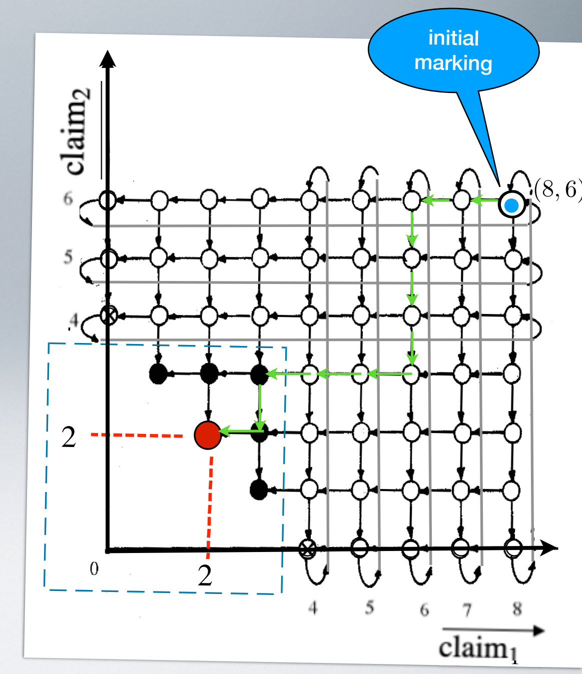


We have 5 places. But using three invariants, the reachability graph can be represented in two dimensions insted of 5

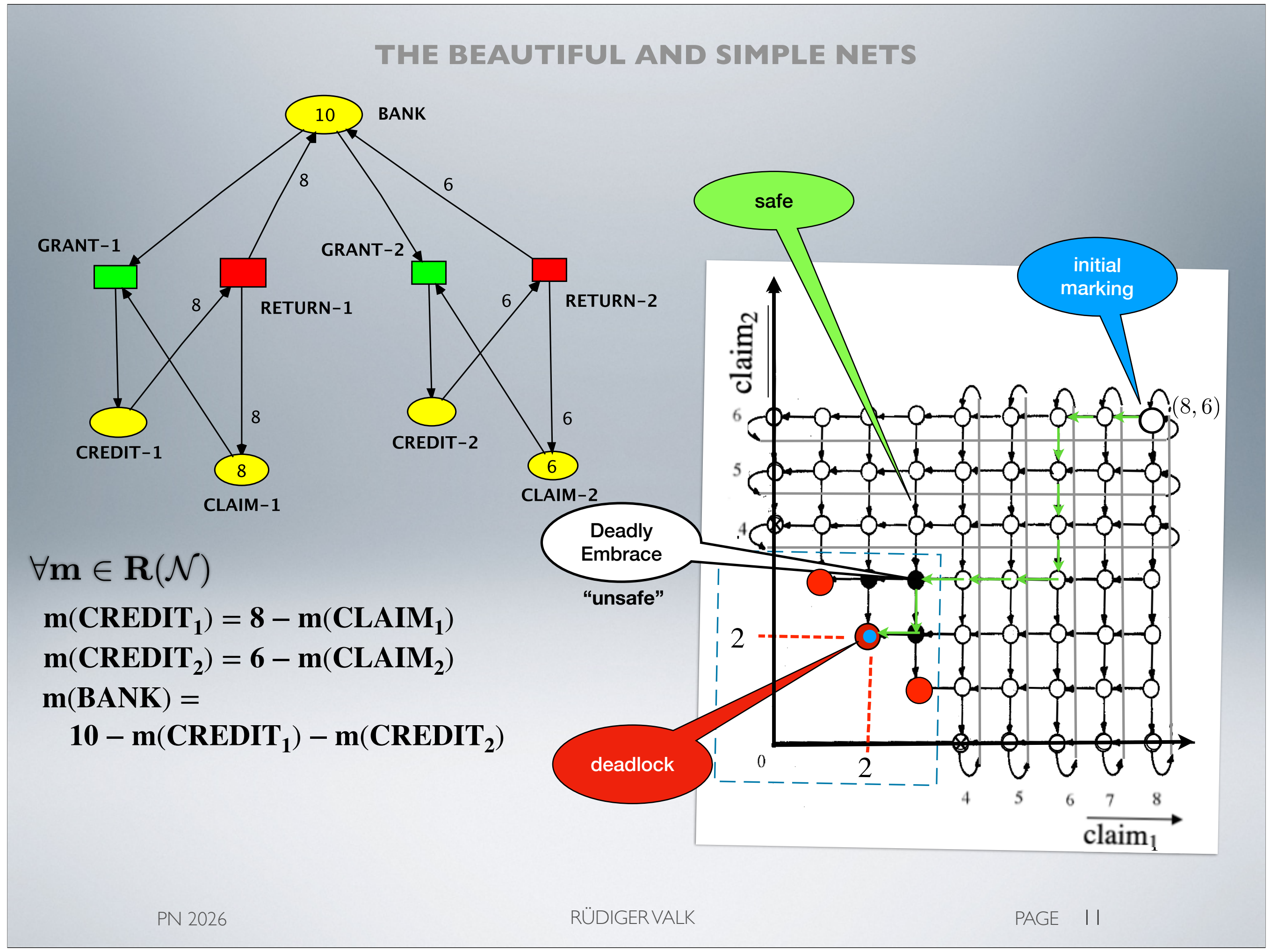


$\forall \mathbf{m} \in \mathbf{R}(\mathcal{N})$

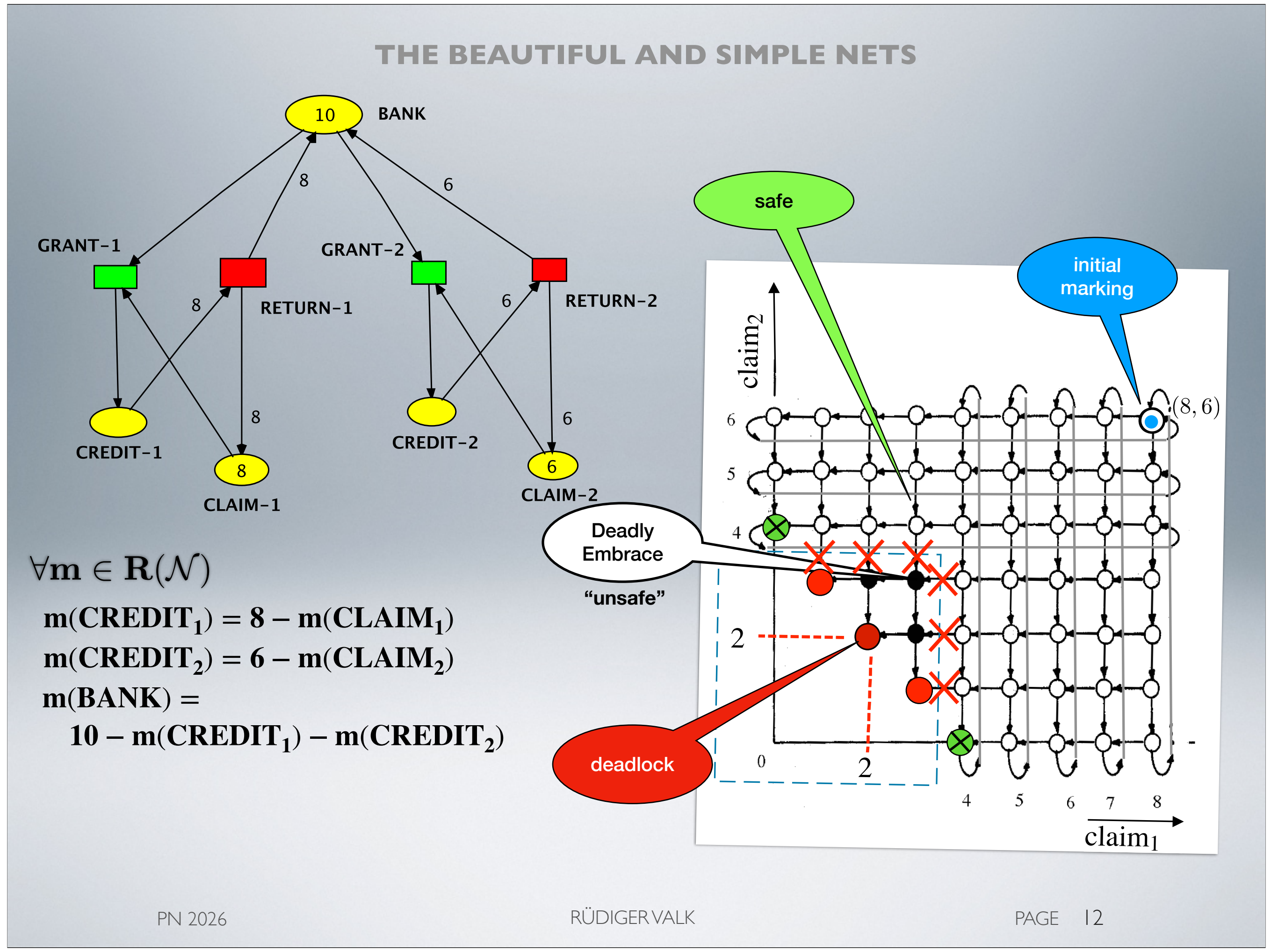
$\mathbf{m}(\mathbf{CREDIT}_1) = 8 - \mathbf{m}(\mathbf{CLAIM}_1)$
$\mathbf{m}(\mathbf{CREDIT}_2) = 6 - \mathbf{m}(\mathbf{CLAIM}_2)$
$\mathbf{m}(\mathbf{BANK}) =$
$10 - \mathbf{m}(\mathbf{CREDIT}_1) - \mathbf{m}(\mathbf{CREDIT}_2)$



Here is the reachability graph with the initial marking. ◉ **By an execution w**e can reach **deadlocks**. ——————- ◉

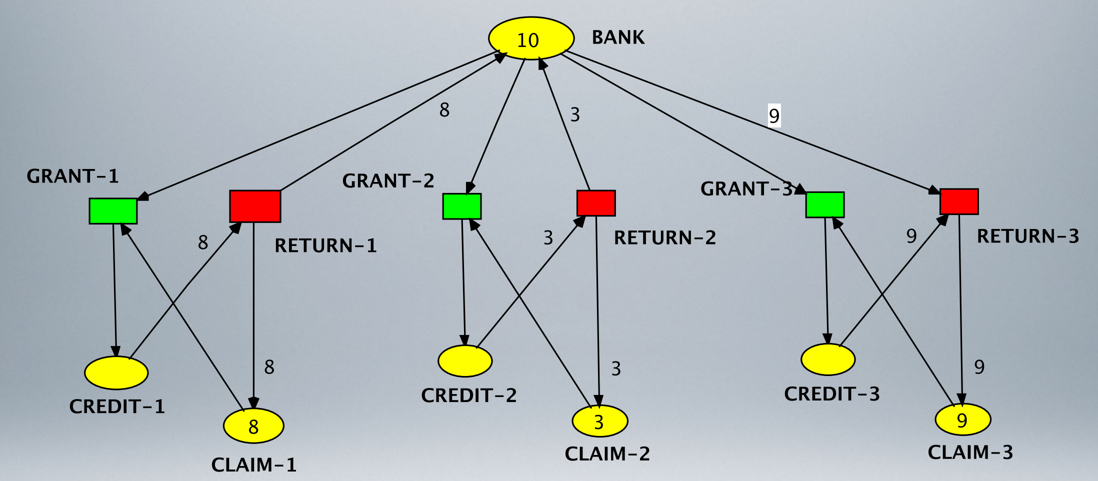


Here we have the safe marking, where a deadlock can be avoided and ◉ unsafe markings, where a deadlock is inevitable.

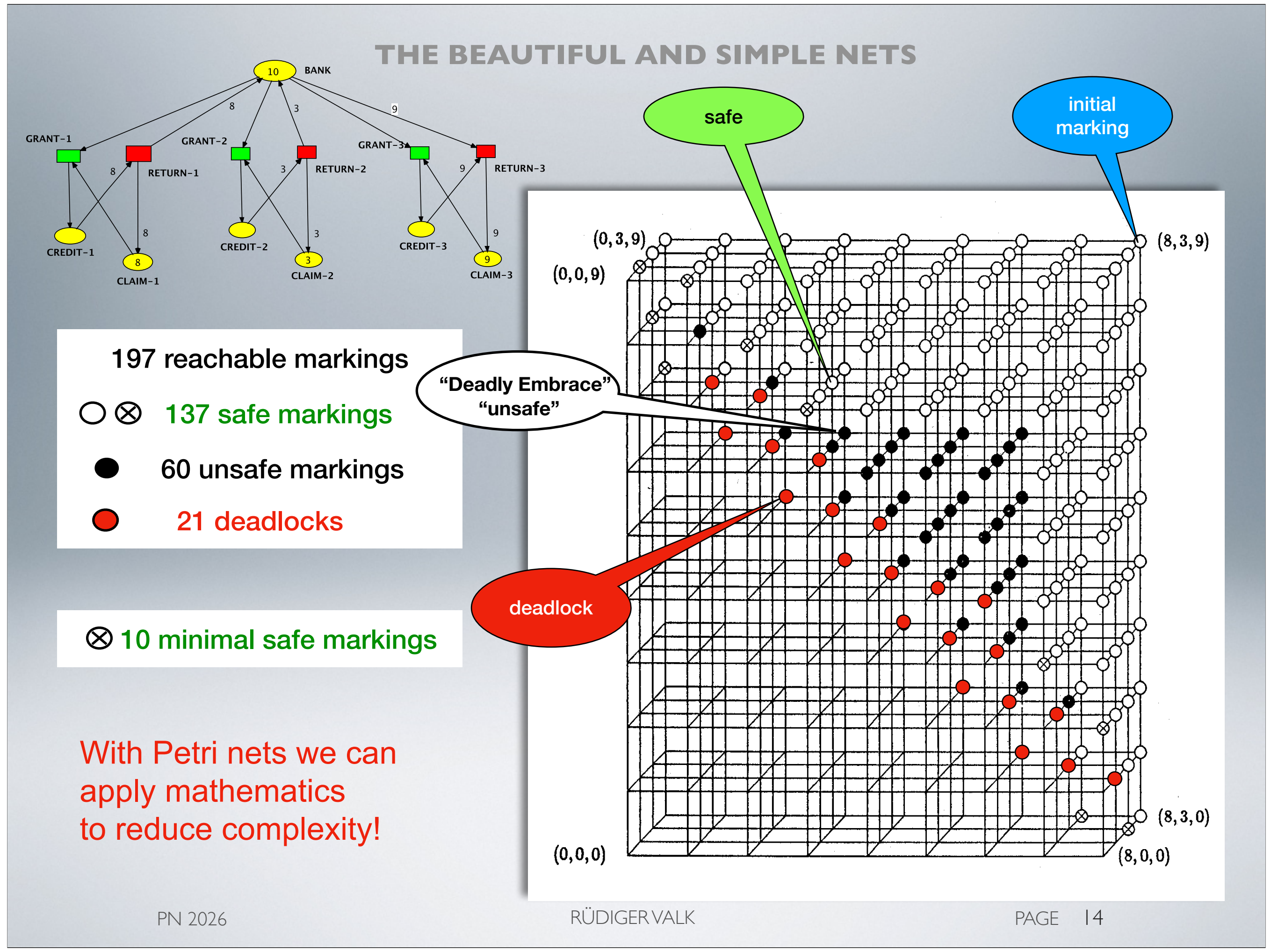


A safe control has to avoid these transitions. There are different possibilities, as Dijkstra's search procedure to test that the initial marking would be still reachable, ◉

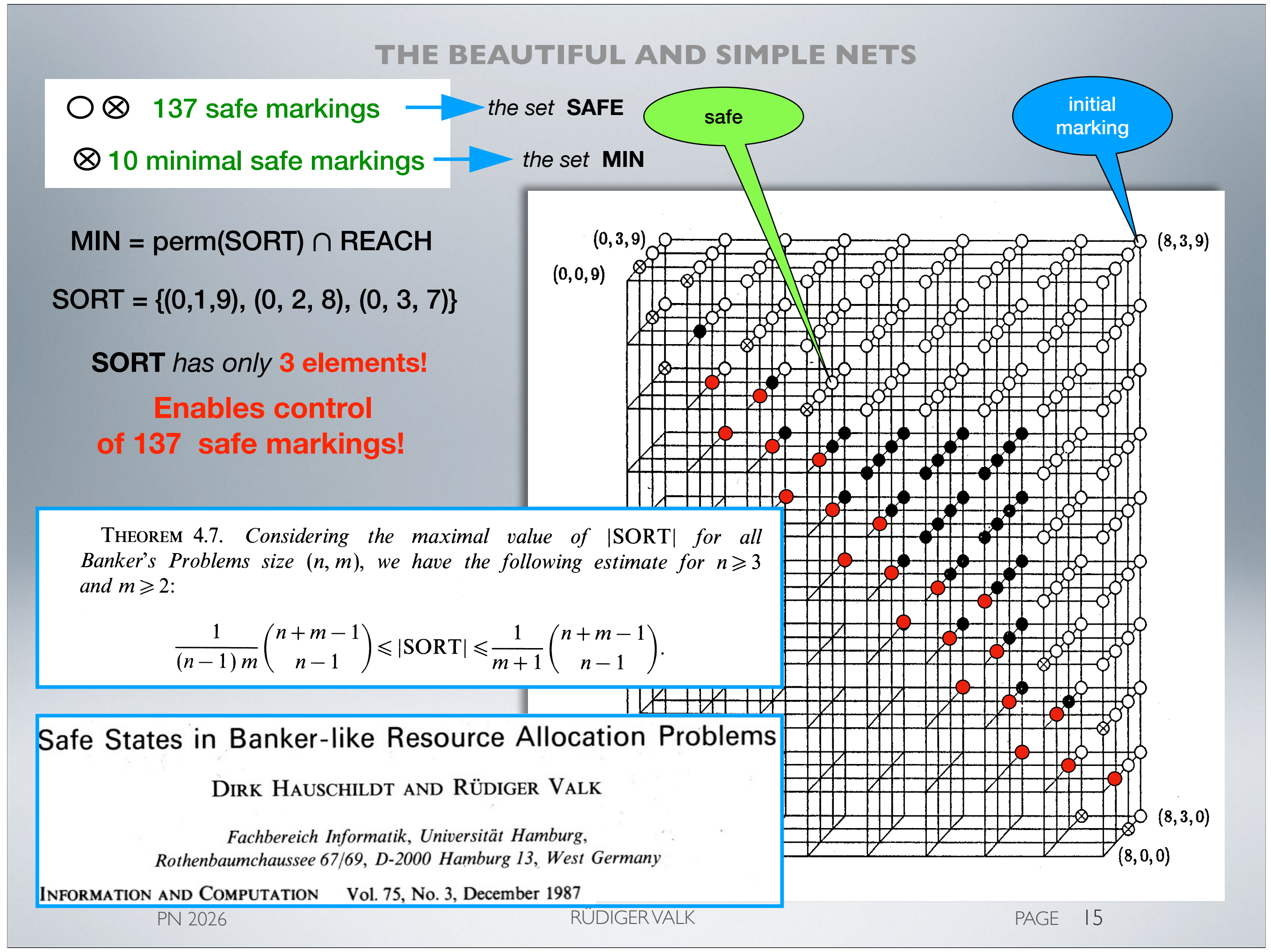
or to describe the set of safe markings by their set of minimal elements (2 in this case), shown here in green. So a safe control has to test whether a intendet next marking is still above of one of these minimal markings!



This instance with 3 clients comes from the **book on operating systems** by Per Brinch Hansen. I did like to show my students who by the switch from 2 clients to 3 simplicity of the reachability graph is lost.



In principle, we are in the same situation, but the number of markings is already considerably higher.
197 reachable markings, 21 deadlocks, 60 unsafe and 137 safe markings.
But the safe markings are described by **only** 10 minimal markings.
This shows: with Petri nets we can apply mathematics to reduce complexity. But we can prove even more!



In a joint paper, published in 1987, ◉ Dirk Hauschildt proved that the 10 minimal markings can be generated by just **3 such markings**, namely the set **SORT** ◉

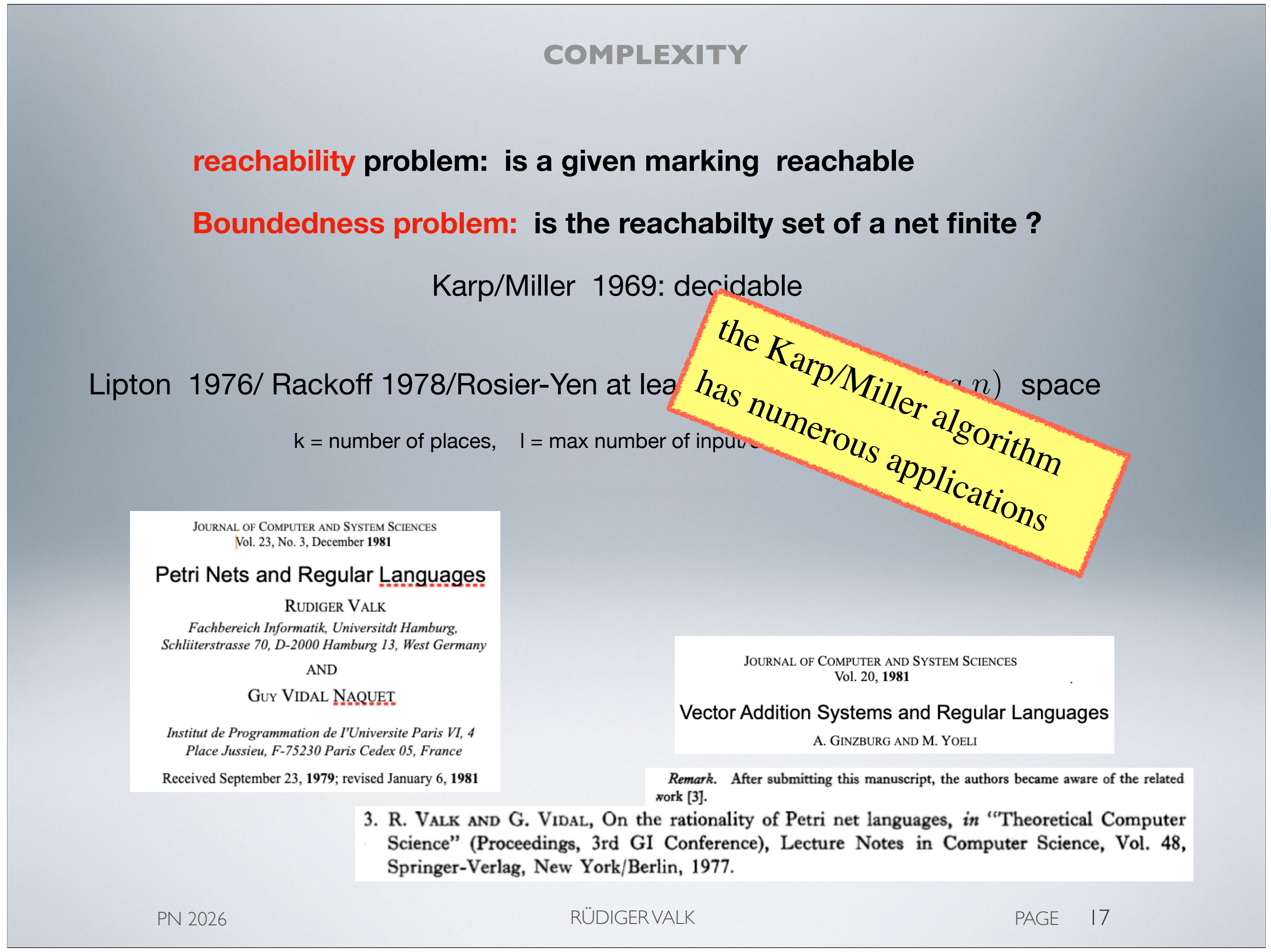
Enables control of 137 safe markings! ◉ He even provided an estimate for the size of SORT.

By this example, I wanted to show who the modelling by Petri nets, can lead to more effective solutions!!!
This leads to general statements about the complexity of nets.

Surprisingly, one of the oldest results is also the most extreme.
Marking set equality problem: are the reachabilty sets of two nets equal ?

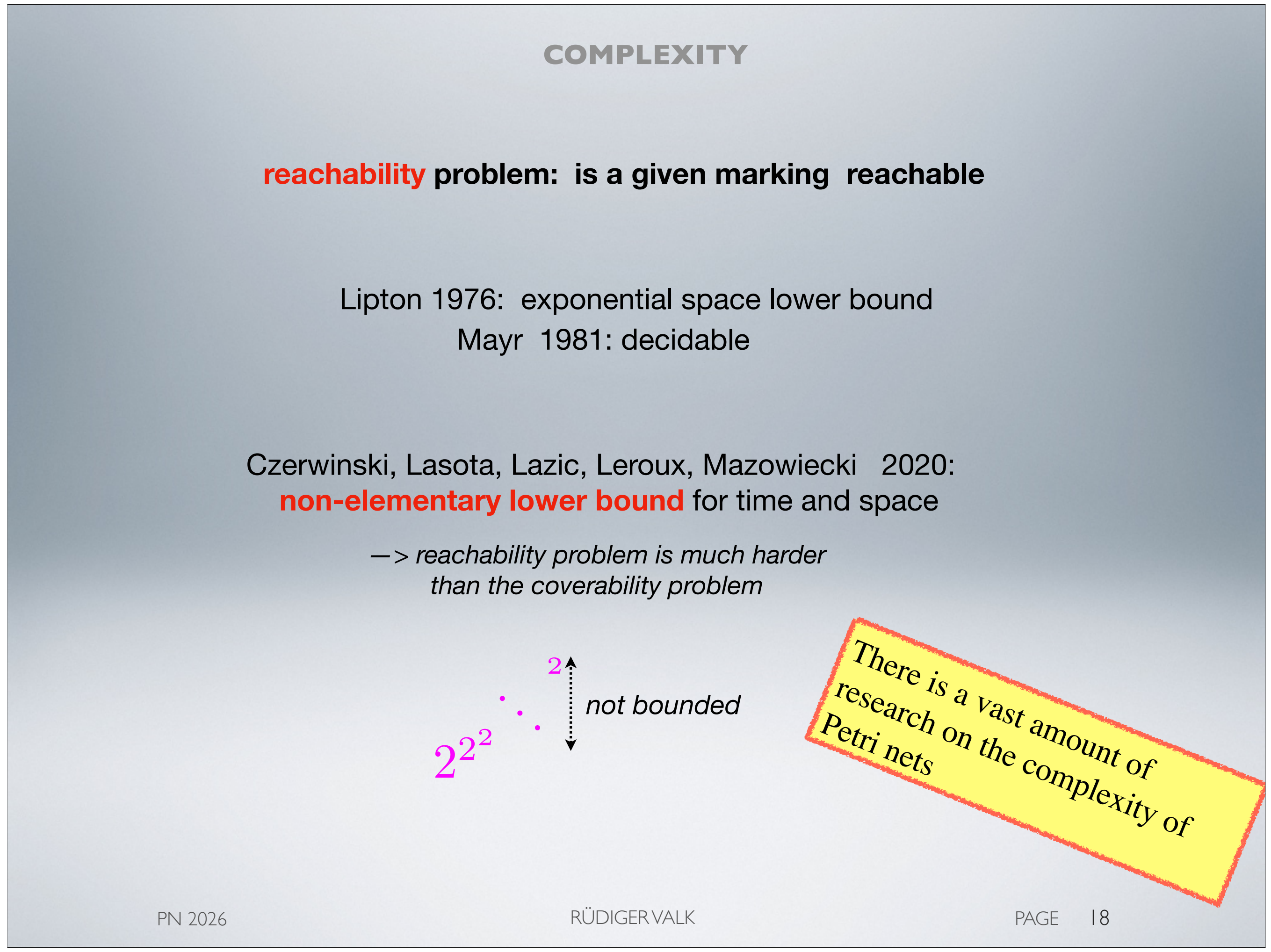
Rabin proved in 1965 that the marking inclusion problem is reducable
to Hilberts 10th problem (1900),

which is proved to be undecidable in 1972 . This used in 1976 that marking equality problem is undecidable



The main problems attacked at that time were the reachability- and the boundedness-problem. ◉ The Boundedness Problem was proven to be decidable by Karp and Miller's famous algorithm.

◉ The scientific community was surprised by just how complex it is – in fact, the memory requirements are exponential. The number of algorithms based on this is immense. ◉ In the I'll just give one example from my field., Petri nets often replaced the interest in finite automata and regular languages. In 1976 we proved that the regularity of the language of nets is decidable, which frequently cited up to now. ◉ At the same time, but independently, this was proven by Ginzburg and Yoeli. Both papers use the Karp–Miller algorithm and therefore have the same computational complexity.



Reachability problem: is a given marking reachable In 1976, Lipton proved an exponential lower bound for the reachability problem, whilst its decidability was not demonstrated until 1981 by Mayr.

It was not until 2020 that these authors were able to prove that it is not elementary and is therefore definitely more difficult than the boundedness problem.There is a vast amount of research on the complexity classes below these.

Javier Esparza has organised this research and summarised it as a set of rules of thumb.

Esparza, J. (1998). Decidability and complexity of Petri net problems — LNCS Vol 1491

Esparza´s rules of thumb:

> A linearly bounded automaton of size n can be simulated by a 1-safe Petri net of size $O(n^2)$. Moreover, there is a polynomial time procedure which constructs this net. Jones, Landweber, Lien, 1977

> Rule of thumb 1:
> All interesting questions about the behaviour of 1-safe Petri nets are PSPACE-hard.

1-bounded nets

14 examples

- Is the Petri net live?
- Is some reachable marking a deadlock?
- Is a given marking reachable from the initial marking?
- Is there a reachable marking that puts a token in a given place?
- Is there a reachable marking that does not put a token in a given place?
- Is there a reachable marking that enables a given transition?
- Is there a reachable marking that enables more than one transition?
- Is the initial marking reachable from every reachable marking?
- Is there an infinite run?
- Is there exactly one run?
- Is there a run containing a given transition?
- Is there a run that does not contain a given transition?
- Is there a run containing a given transition infinitely often?
- Is there a run which enables a transition infinitely often but contains it only finitely often?

A linearly bounded automaton of size n can be simulated by a 1- safe Petri (1-bounded) net of size $O(n^2)$ ◉

It follows Rule of thumb 1:
All interesting questions about the behaviour of 1-safe nets or 1-bounded nets are PSPACE-hard. ◉

“interesting” is obviously informal, but there are 14 examples.
Here is not the place to explain them all. You may have a quick look and study the slides later on.

Esparza, J. (1998). Decidability and complexity of Petri net problems — LNCS Vol 1491

Esparza´s rules of thumb:

> A linearly bounded automaton of size n can be simulated by a 1-safe Petri net of size $O(n^2)$. Moreover, there is a polynomial time procedure which constructs this net. Jones, Landweber, Lien, 1977

> Rule of thumb ~~1:~~ 2
> Nearly all interesting questions about the behaviour of 1-safe Petri nets are in PSPACE

1-bounded nets

14 examples

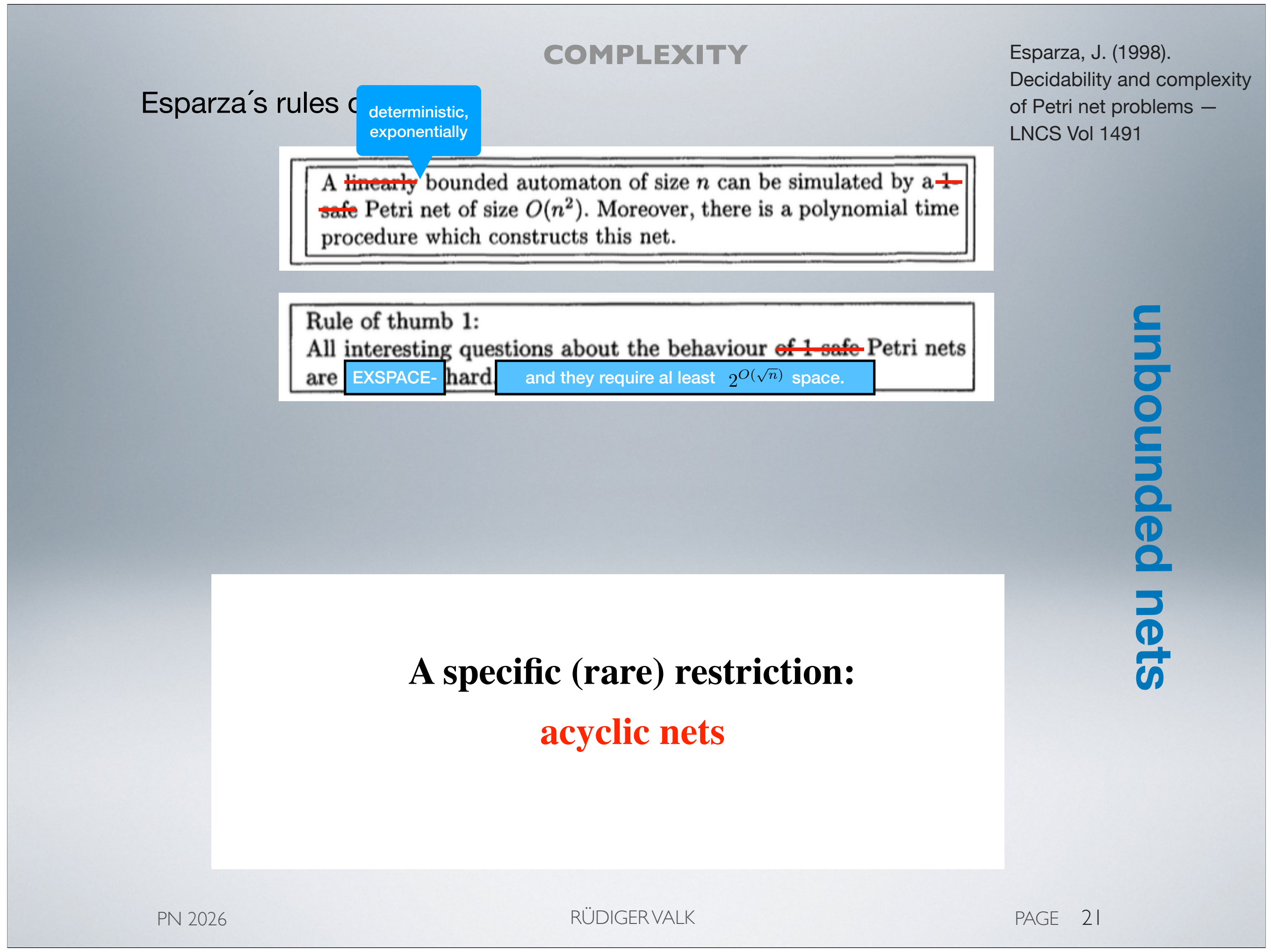
- Is the Petri net live?
- Is some reachable marking a deadlock?
- Is a given marking reachable from the initial marking?
- Is there a reachable marking that puts a token in a given place?
- Is there a reachable marking that does not put a token in a given place?
- Is there a reachable marking that enables a given transition?
- Is there a reachable marking that enables more than one transition?
- Is the initial marking reachable from every reachable marking?
- Is there an infinite run?
- Is there exactly one run?
- Is there a run containing a given transition?
- Is there a run that does not contain a given transition?
- Is there a run containing a given transition infinitely often?
- Is there a run which enables a transition infinitely often but contains it only finitely often?

exeption: controllability property is EXPSPACE-complete

Jones, Landweber, Lien, 1977

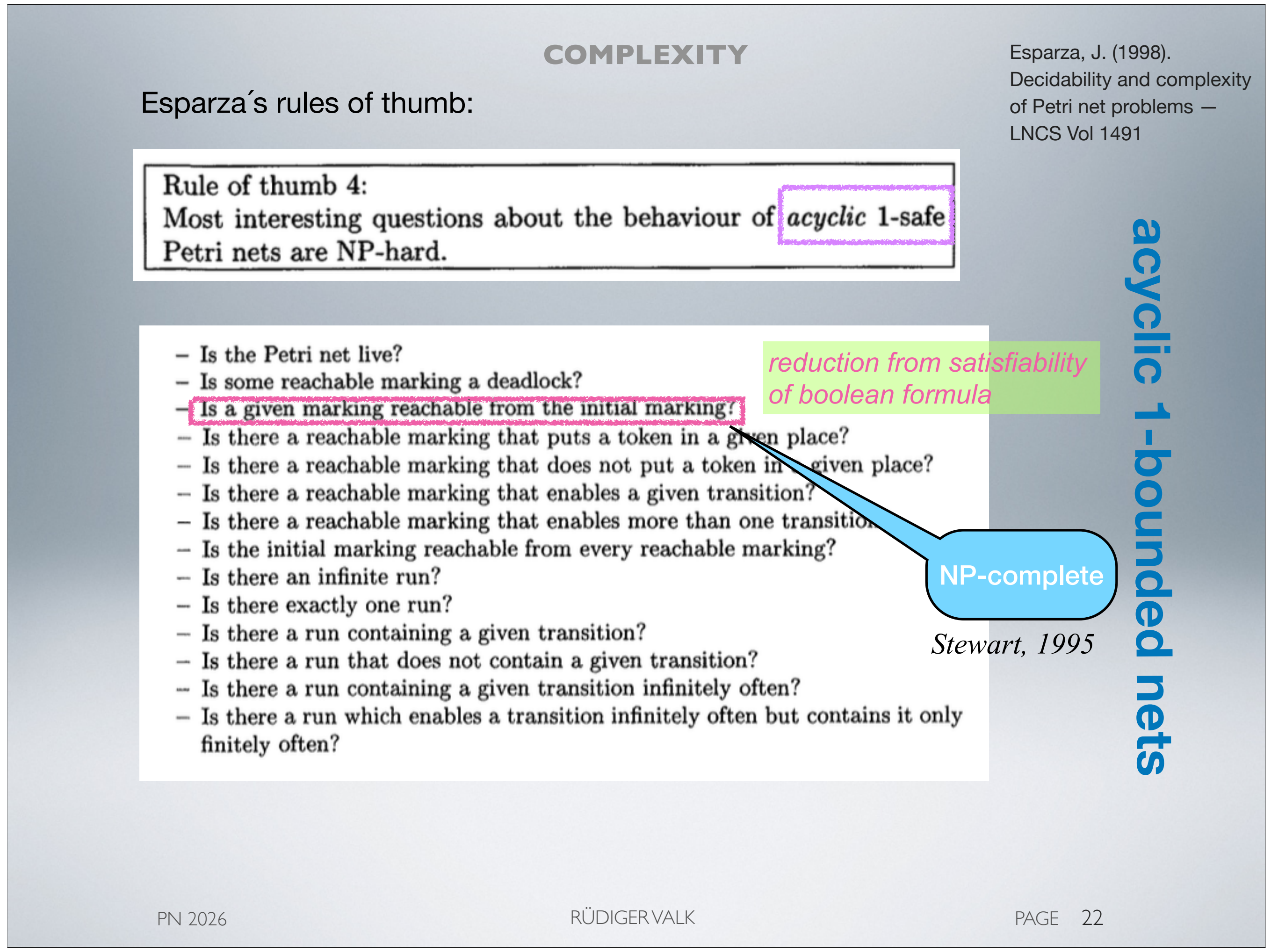
Most of them are in PSPACE? ◉ — exeption: controllability property is EXPSPACE-complete

Let me just add the corresponding result for unbounded nets.



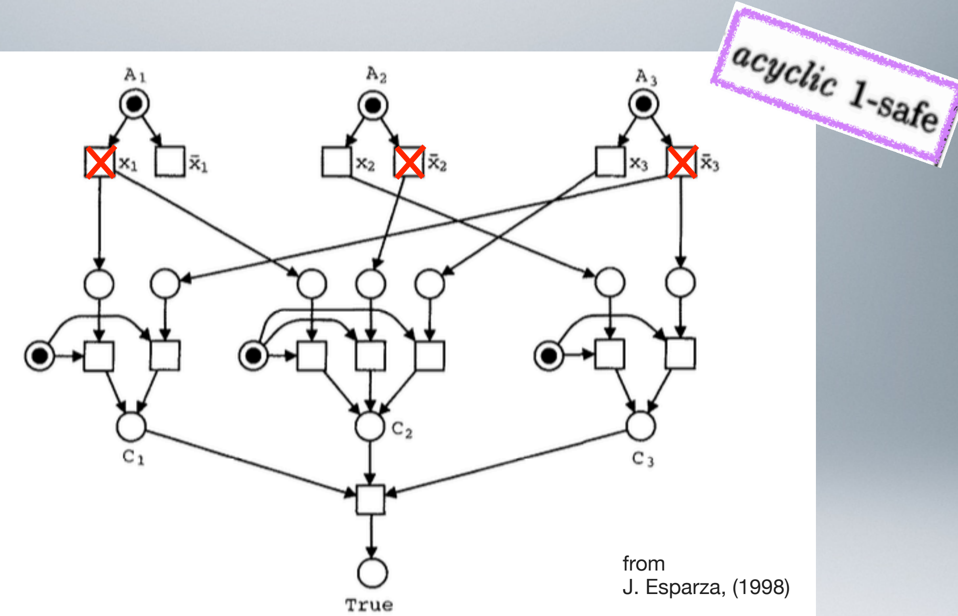
For (unbounded) PT-nets we have the following rule of thumb. lesen “2 to the O of root n”
Exeption: Marking equality. ◉

In order to see a class with low complexity we look for
a specific (rare) restriction: acyclic nets, ◉ where we have some time complexity results



Most interesting questions about the behaviour of acylic 1-safe/1-bounded nets are NP-hard.

🔴 A well-known NP-hard problem is the satisfiability problem of boolean formula and I will show you an example of reduction to the reachability problem, which is simular as for free-choice nets, as given in the book of Jörg Desel and Esparza.



from
J. Esparza, (1998)

Fig. 3. Acyclic net corresponding to the formula $(x_1 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_2 \vee \overline{x}_3)$

conjunctive normal form

`True` is reachable iff the formula is satisfiable

The place “True” is reachable if and only if the formula is satisfiable. ◉

In this case: the occurrences of these alternative transitions corresponds to binding satisfying the formula.

Reductions of nets by Gerard Berthelot

preservation of
- **liveness,**
- **boundedness,**
- **....**

Berthelot, Roucairol:
Reduction of Petri-Nets.
LNCS 45 (1976)

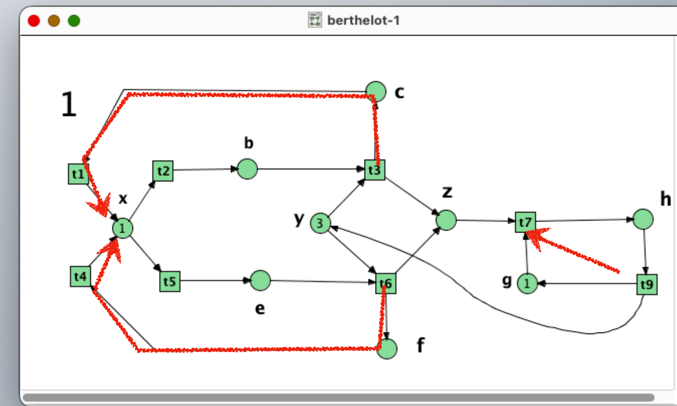
Berthelot, Roucairol, Valk:
Reductions of Nets and Parallel Prgrams.
LNCS 84 (1980)

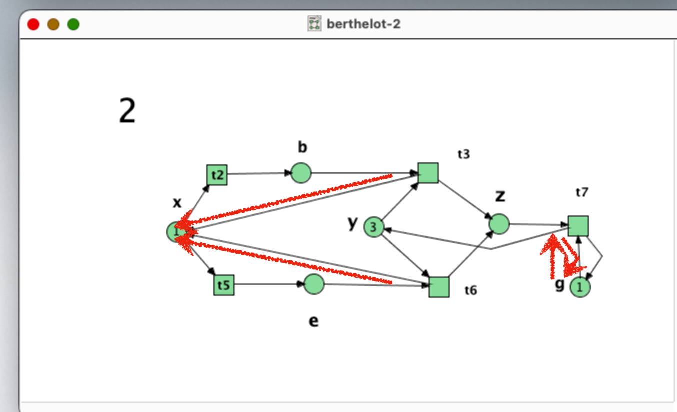
Next, I will look at some approaches to managing complexity. ◉ **In addition to the study of subclasses of nets,** like free-choice nets, **there were the Berthelot reductions.** ◉ **Here are the earliest references I found.** **Berthelot considered simple transformations that reduced the size of the nets whilst preserving important properties, such as liveness and boundedness. In the simplified net, this property should then be easy to decide.**

Reductions of nets by Gerard Berthelot

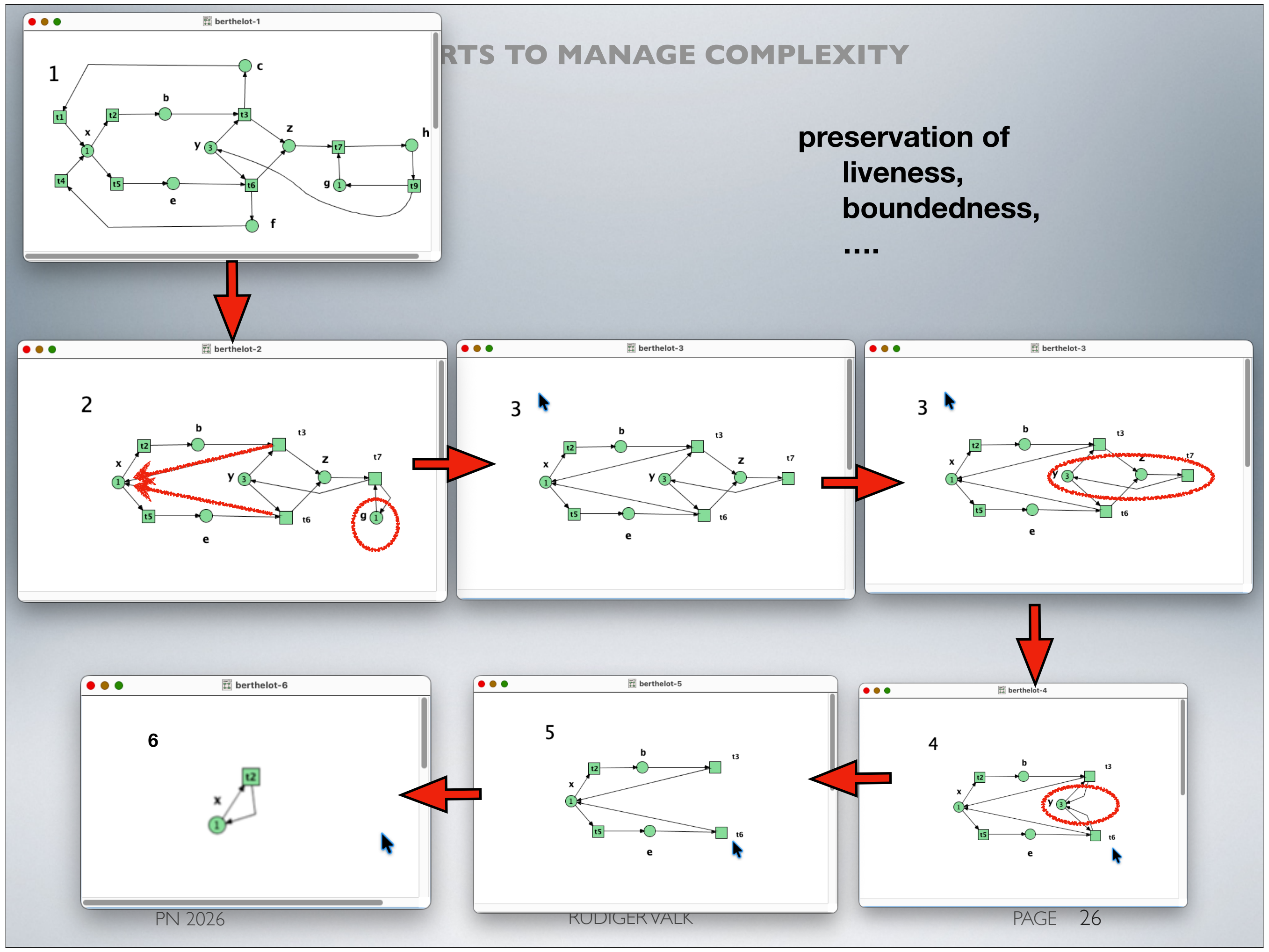
preservation of
liveness,
boundedness,
....

```
Berthelot, Roucairol
LNCS 45 (1976)
```





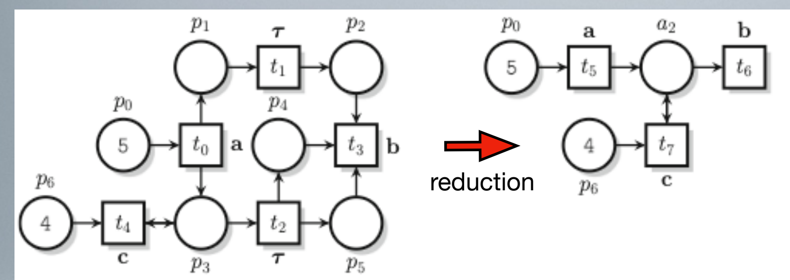
Here, Step 1 to step two from a paper from 1976 ◉ **The places c and f can be eliminated under preservation ◉ of liveness and boundedness. ◉ The same with place g.**



This continues like this. ◉ Step 6 is, of course, both live and bounded, and is therefore also the original net.

And 30 years after Berthelot's "Structural Reductions" [Berthelot, 1987]

"Structural Reductions with linear equations" [Berthomieu et al., 2019]



linear equations

$$E_M \triangleq (p_5 = p_4) \wedge$$

$$(a_1 = p_1 + p_2) \wedge$$

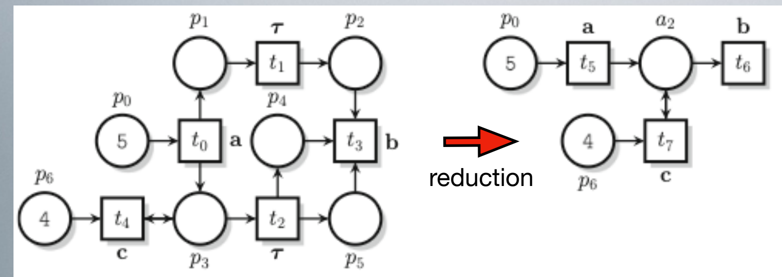
$$(a_2 = p_3 + p_4) \wedge$$

$$(a_1 = a_2)$$

On the Combination of Polyhedral Abstraction and SMT-Based Model Checking for Petri Nets — Nicolas Amat , Bernard Berthomieu , and Silvano Dal Zilio — PETRI NETS 2021, LNCS 12734, pp. 164–185, 2021.

On the Complexity of Proving Polyhedral Reductions — Nicolas Amat, Silvano Dal Zilio, Didier Le Botlan — Fundamenta Informaticae 192(3-4) : 363–394 (2024)

- **And 30 years after Berthelot's "Structural Reductions" ◉ the concept of "Structural Reductions with linear equations" is introduced by the Toulouse research group. ◉**
- **For this example the equations are like invariants, but introduce additional variables to reduce the net, but preserve all information about reachable markings. ◉**
  - **Recently there apeared also a complexity result on these reductions.**

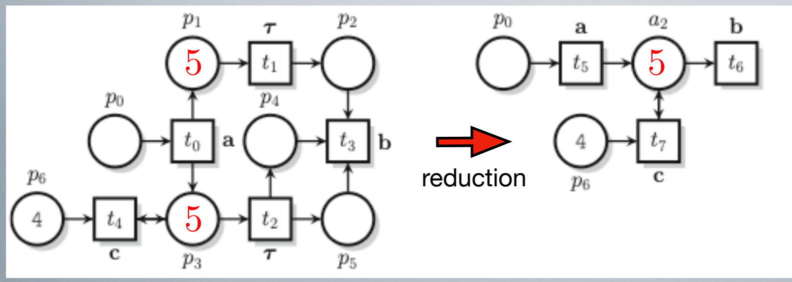


linear equations

$$E_M \triangleq (p_5 = p_4) \wedge$$
$$(a_1 = p_1 + p_2) \wedge$$
$$(a_2 = p_3 + p_4) \wedge$$
$$(a_1 = a_2)$$

p_0	5
p_1	
p_2	
p_3	
p_4	
p_5	
p_6	4
$a_1=p_1+p_2$	0
$a_2=p_3+p_4$	0

- **Here is an occurrence sequence together with some evaluations of the equations.**



linear equations

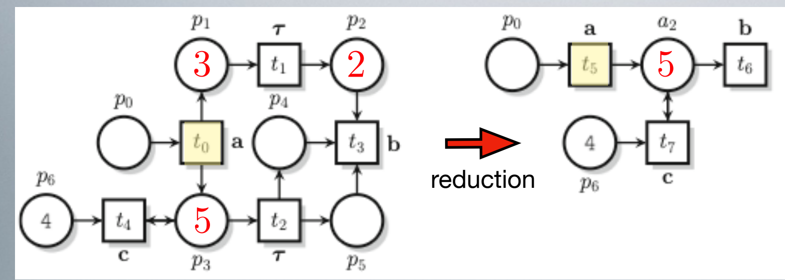
$$E_M \triangleq (p_5 = p_4) \wedge (a_1 = p_1 + p_2) \wedge (a_2 = p_3 + p_4) \wedge (a_1 = a_2)$$

$5 \times t_0$

p_0	5	
p_1		5
p_2		
p_3		5
p_4		
p_5		
p_6	4	
$a_1=p_1+p_2$	0	5
$a_2=p_3+p_4$	0	5

occurrence of t_0 5 times

linear equations

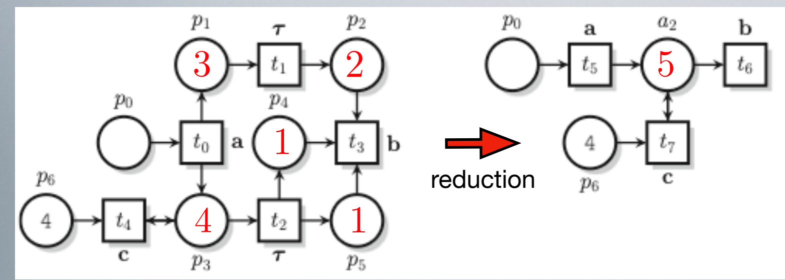


$$E_M \triangleq (p_5 = p_4) \wedge$$
$$(a_1 = p_1 + p_2) \wedge$$
$$(a_2 = p_3 + p_4) \wedge$$
$$(a_1 = a_2)$$

		$5 \times t_0$	$2 \times t_1$
p_0	5		
p_1		5	3
p_2			2
p_3		5	5
p_4			
p_5			
p_6	4		
$a_1=p_1+p_2$	0	5	5
$a_2=p_3+p_4$	0	5	5

occurrence of t_1 2 times

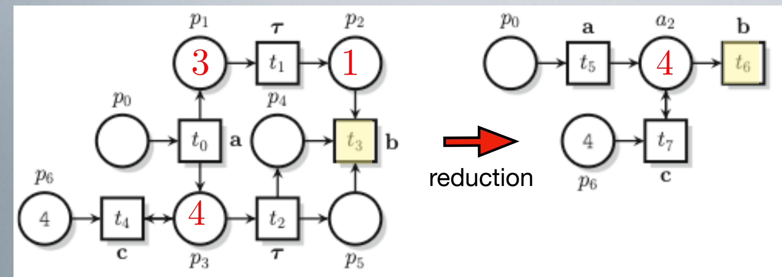
linear equations



$$E_M \triangleq (p_5 = p_4) \wedge (a_1 = p_1 + p_2) \wedge (a_2 = p_3 + p_4) \wedge (a_1 = a_2)$$

		$5 \times t_0$	$2 \times t_1$	$1 \times t_2$
p_0	5			
p_1		5	3	3
p_2			2	2
p_3		5	5	4
p_4				1
p_5				1
p_6	4			
$a_1=p_1+p_2$	0	5	5	5
$a_2=p_3+p_4$	0	5	5	5

occurrence of t_2

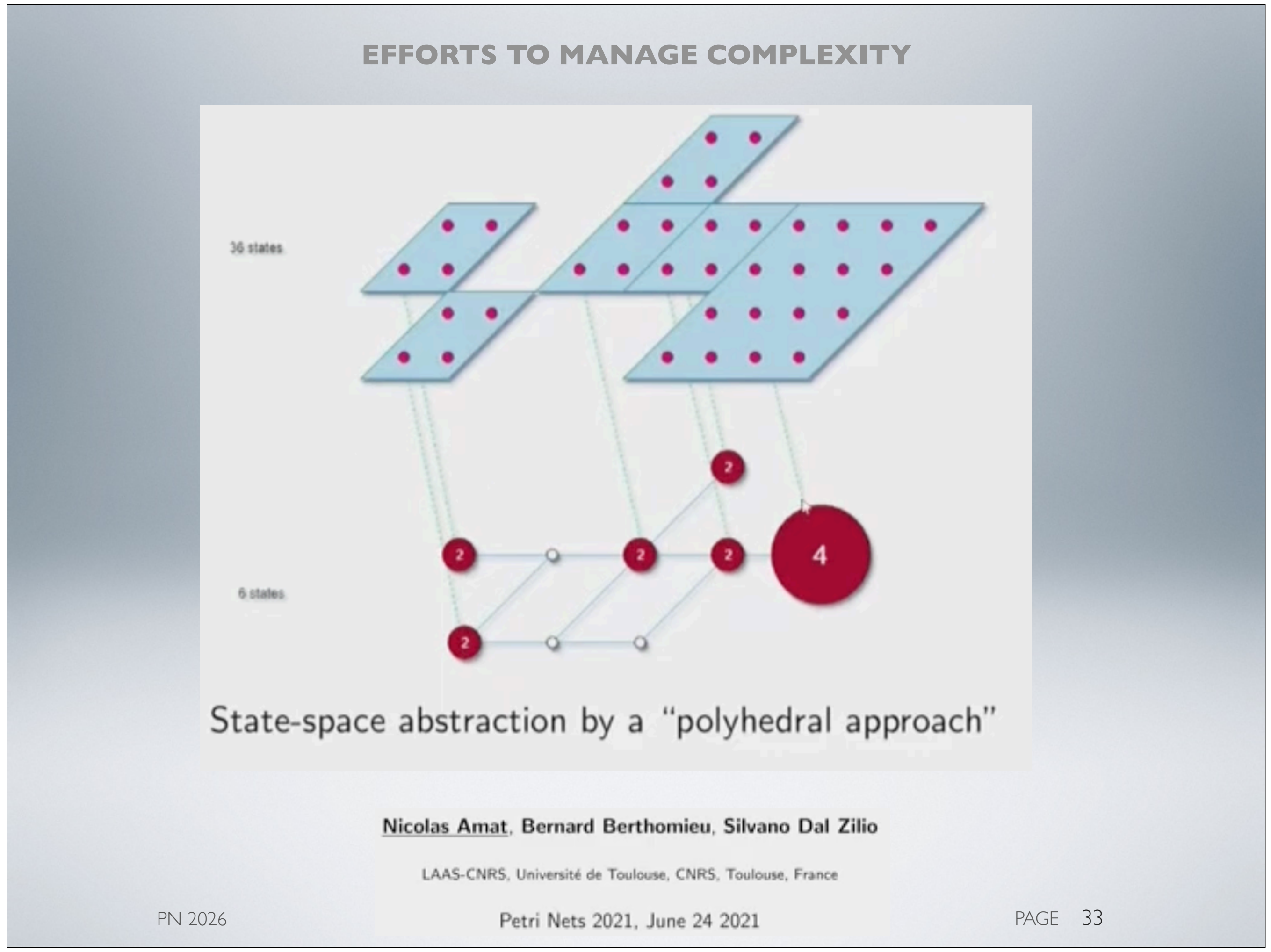


linear equations

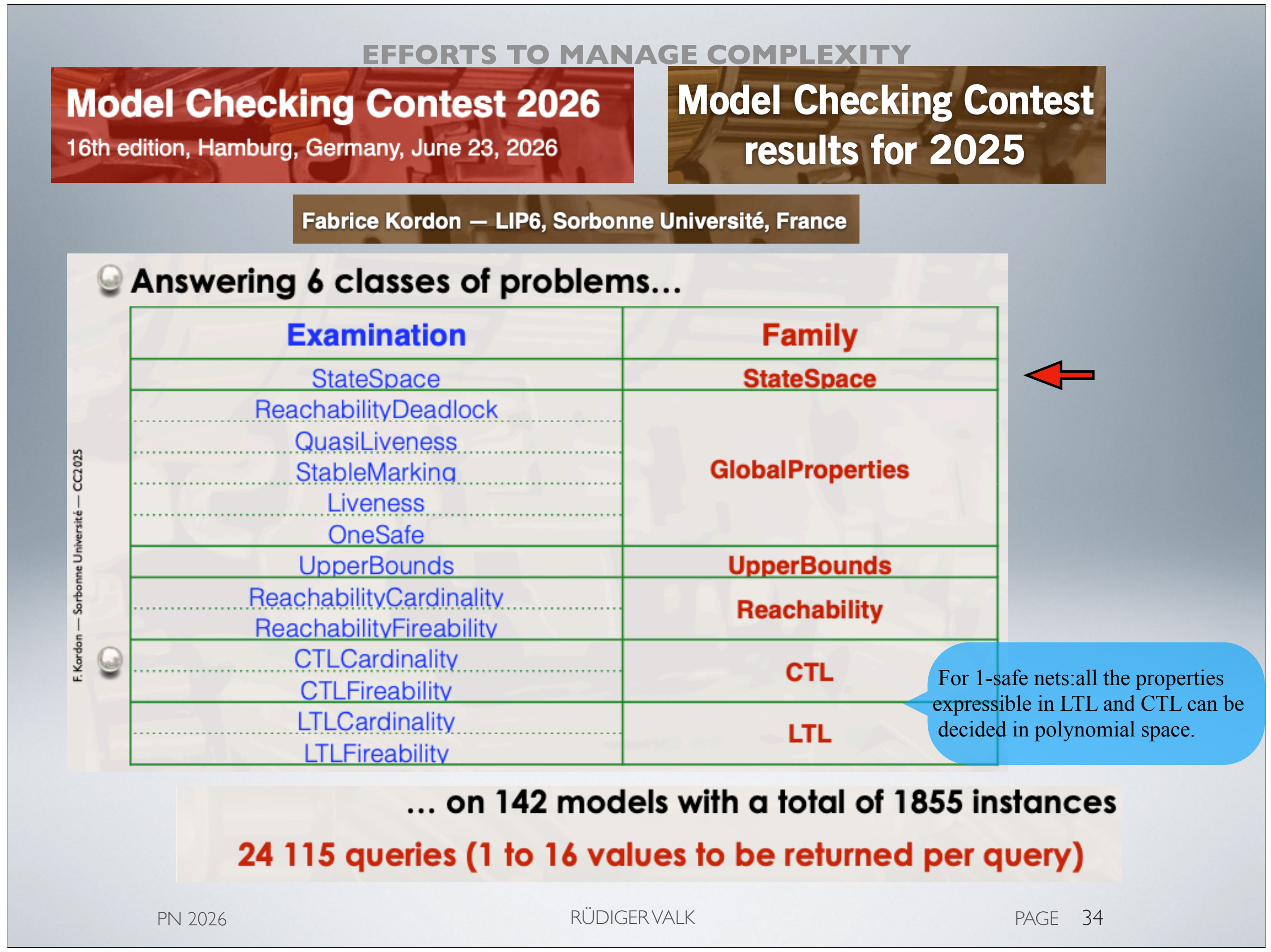
$$E_M \triangleq (p_5 = p_4) \wedge$$
$$(a_1 = p_1 + p_2) \wedge$$
$$(a_2 = p_3 + p_4) \wedge$$
$$(a_1 = a_2)$$

		$5 \times t_0$	$2 \times t_1$	$1 \times t_2$	$1 \times t_3$
p_0	5				
p_1		5	3	3	3
p_2			2	2	1
p_3		5	5	4	4
p_4			1	1	
p_5			1	1	
p_6	4				
$a_1=p_1+p_2$	0	5	5	5	4
$a_2=p_3+p_4$	0	5	5	5	4

occurrence of t_3 - always the equations are valid



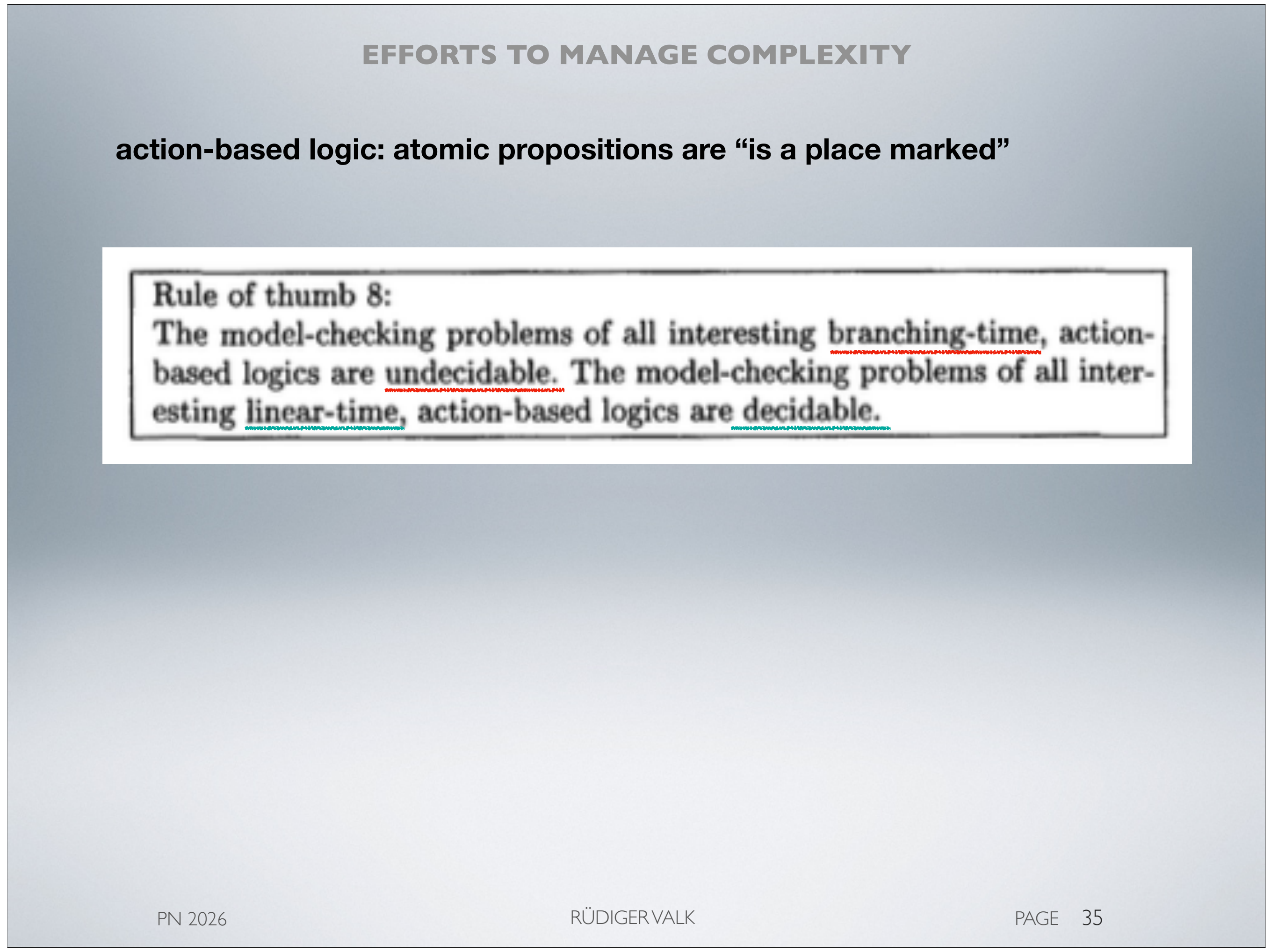
Of Nicolas Amat's slides, I have this one, which shows the abstraction of the 'Reachable' markings. Transition effects are not retained. As a result, they perform…



….. (they perform) particularly well in the 'Model Checking Contests' in the separated 'state space' category. The MCC is a striking example of how tools can help manage complexity today.◉ Maybe you attendes the MCC-meeting Thuesday afternoon. There are problem families for

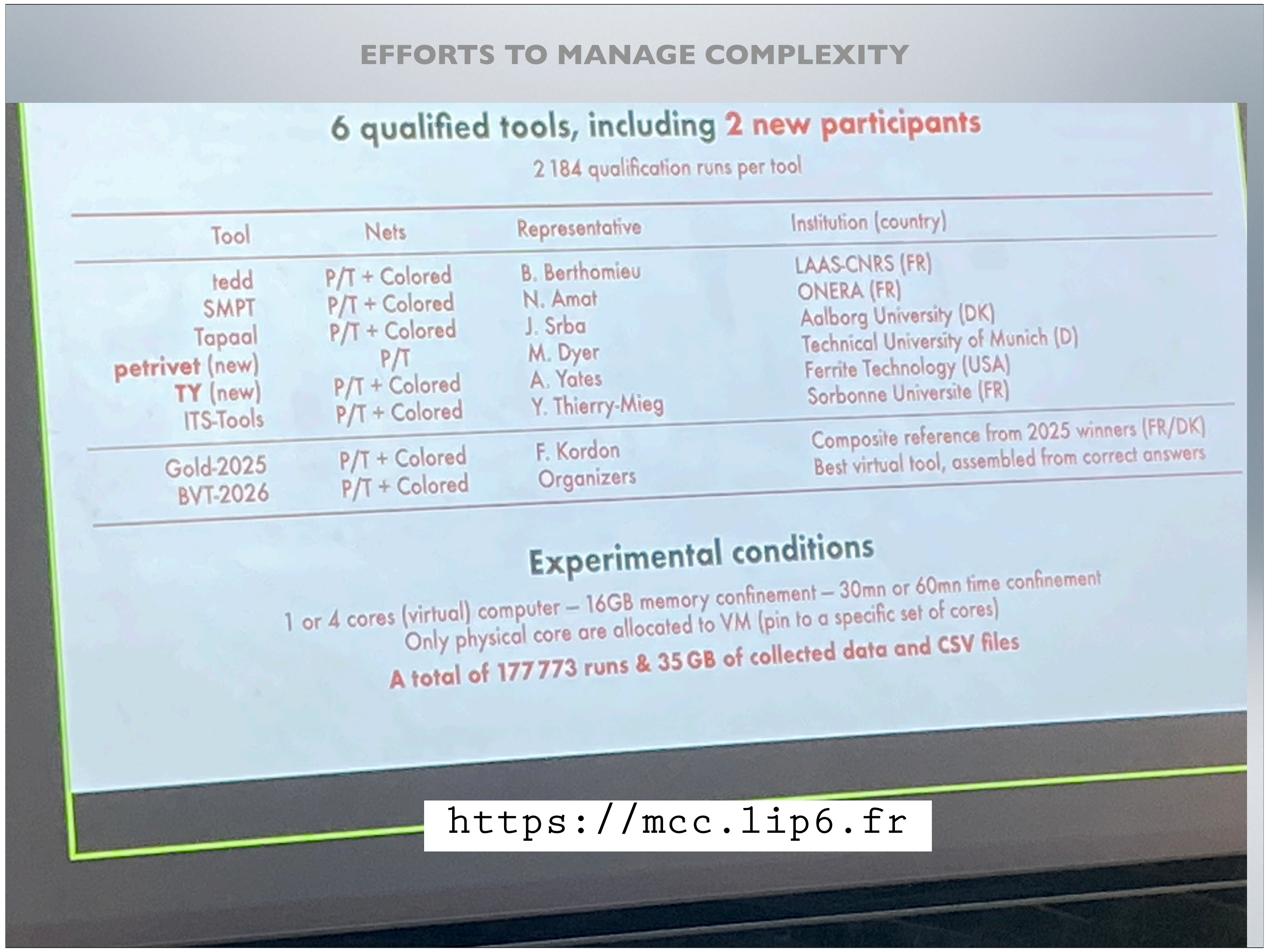
Global Properties, UpperBounds, Reachability, CTL, and LTL ◉ … on 142 models

- lesen- ◉ -Let me add a complexity result: For 1-safe nets:all the properties expressible in LTL and CTL can be decided in polynomial space.



There is a different complexity result if you are using action-based logic whrer the atomic propositions are : “is a place marked” then◉

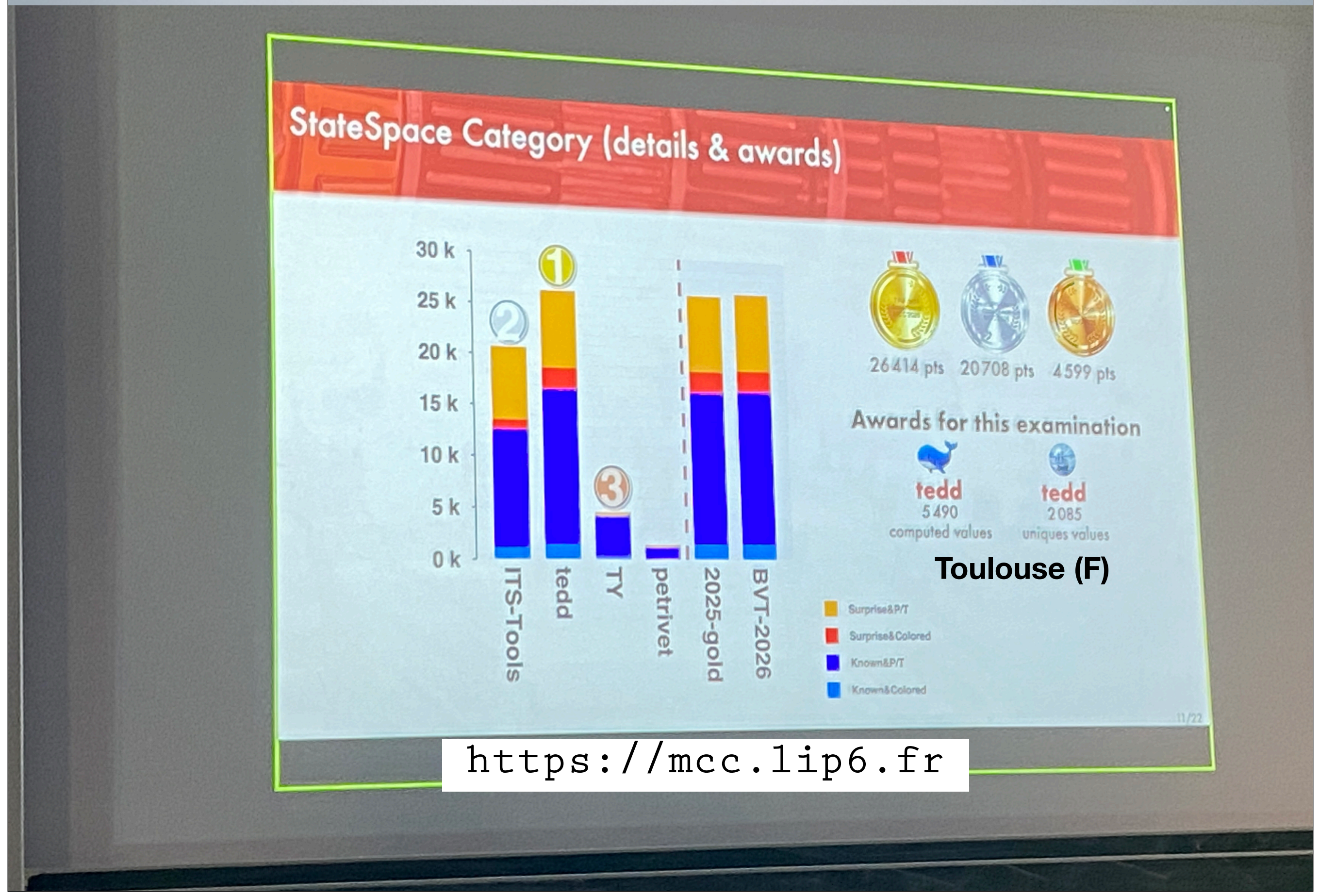
CTL: undecidable   while **LTL: decidable**



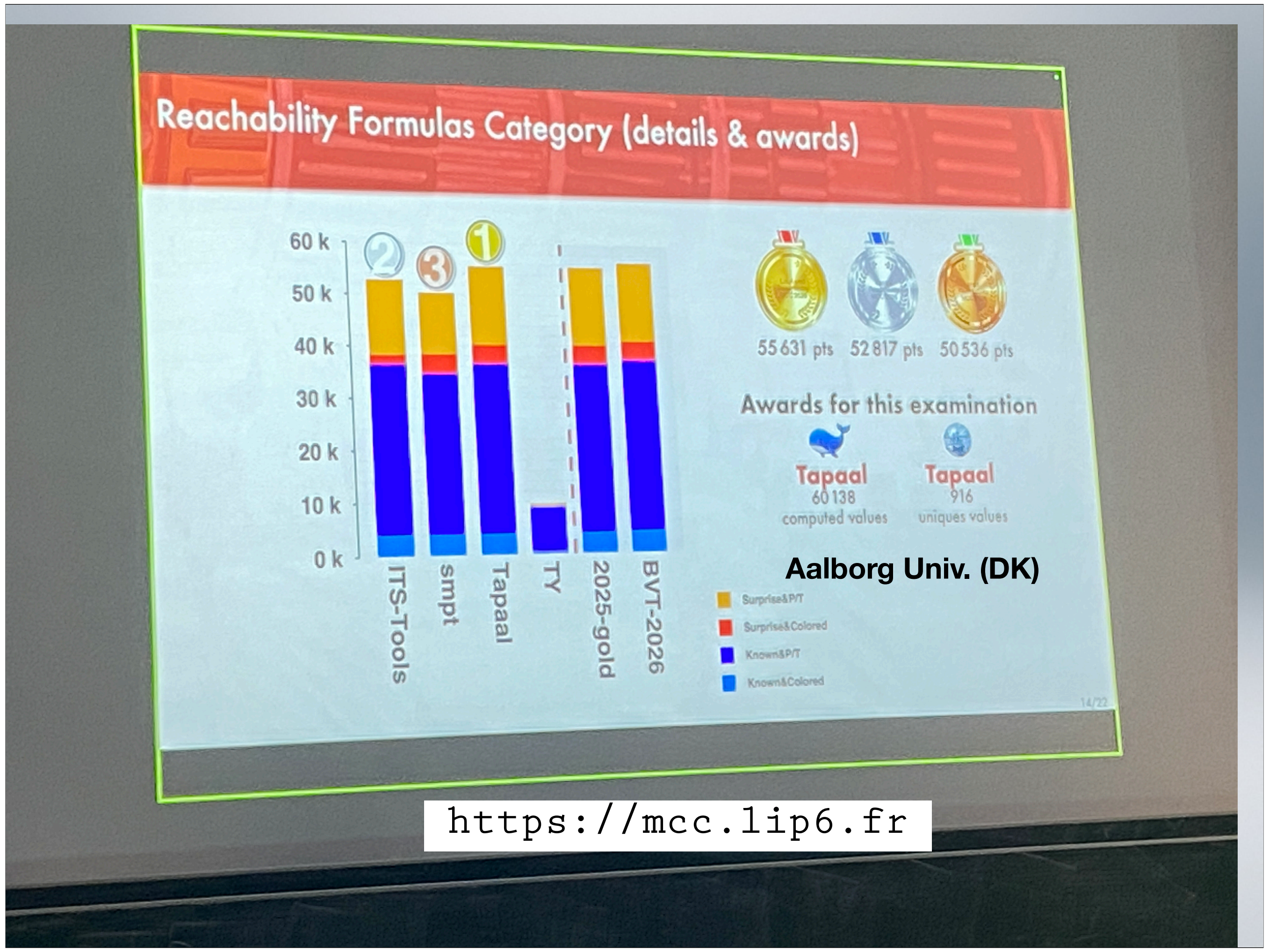
There is a different complexity result if you are using action-based logic whrer the atomic propositions are : “is a place marked” then◉

CTL: undecidable while **LTL: decidable**

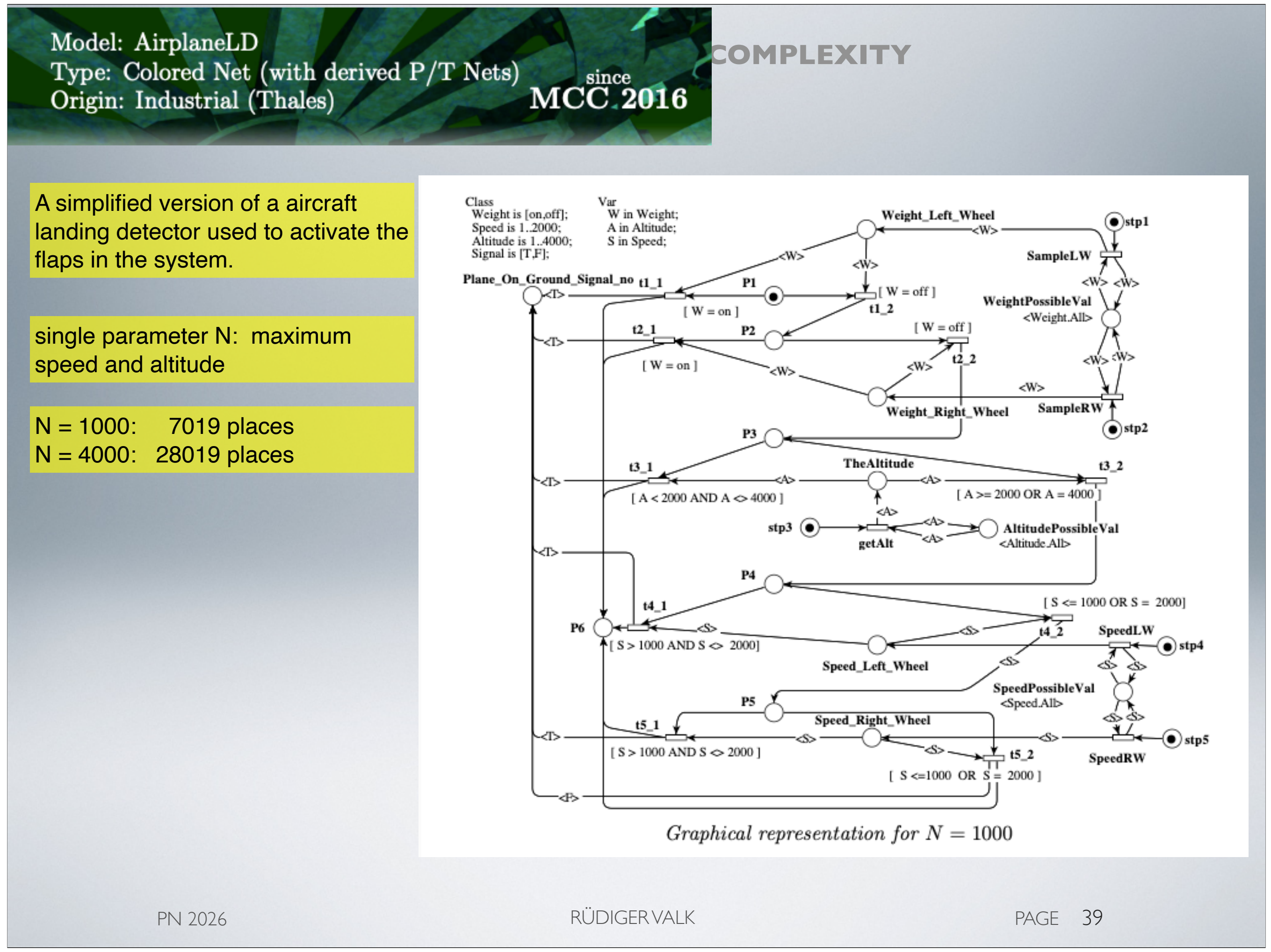
EFFORTS TO MANAGE COMPLEXITY



◉ **and a good result for tedd**



- and a good result for tedd



An example for a model used in the contest is the **AirplaneLandingDetector:** A simplified version of a aircraft landing detector used to activate the flaps in the system.

- There is a single parameter N: maximum speed and altitude
- N = 1000: 7019 places N = 4000: 28019 places

“Structural Reductions” with linear equations

Some models in the MCC benchmark are “fully reduced”

represent the whole state space

by the solutions to a system of linear integer expressions.

SwimmingPool or Kanban

Some models in the MCC benchmark are “massively reduced”

represent the whole state space

by a smaller net and linear integer expressions.

AirplaneLD

- Some models in the MCC benchmark are “fully reduced”
- They **represent the whole state space by the solutions to a system of linear integer expressions.**
- **Some models in the MCC benchmark are “massively reduced”:**

represent the whole state space by a smaller net and linear integer expressions.

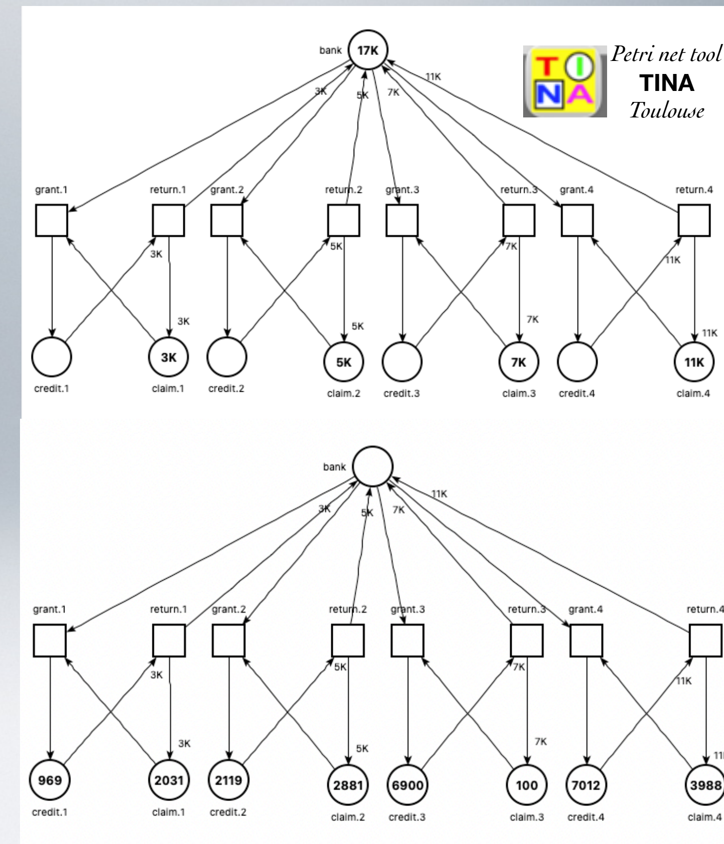
$$9.4764 \cdot 10^{14} \text{ markings}$$
$$3.7903 \cdot 10^{15} \text{ transitions}$$

THEOREM 4.7. *Considering the maximal value of* |SORT| *for all Banker's Problems size* (n, m), *we have the following estimate for* $n \geqslant 3$ *and* $m \geqslant 2$:

$$\frac{1}{(n-1)\,m}\binom{n+m-1}{n-1} \leqslant |\mathrm{SORT}| \leqslant \frac{1}{m+1}\binom{n+m-1}{n-1}.$$

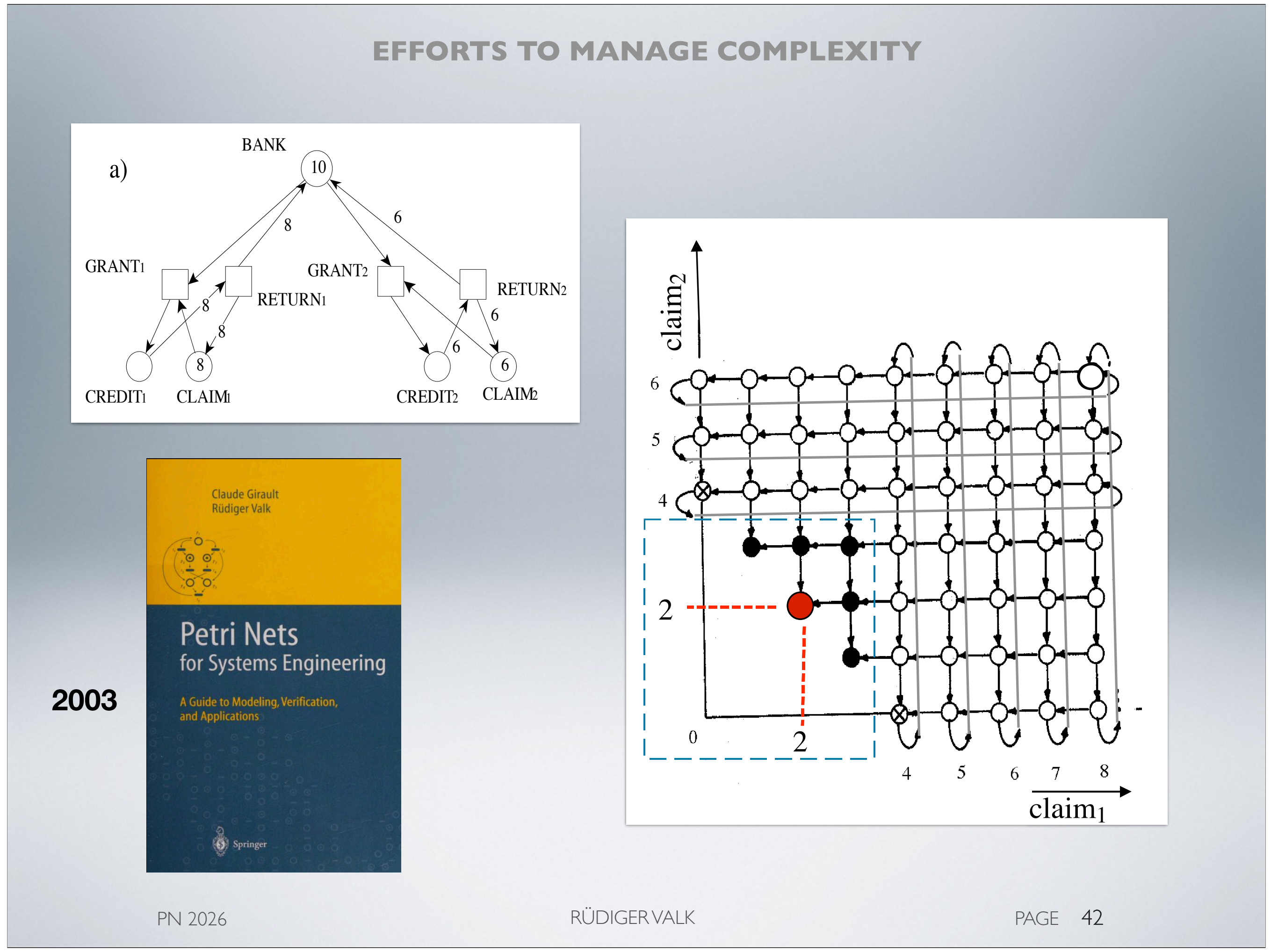
$$n = 4, m = 11000$$

$$20 \cdot 10^6 < |SORT|$$



- Whilst preparing for this talk, I showed the Bankiers net to some of the authors of the TINA tool: Silvano dal Zilio and Bernard Berthomieu. They found it interesting because, on the one hand, the reachability set is exactly the solution set of the net's state equation, but on the other hand, it is not free of deadlocks.

- Here is the number of markings and transitions for this instance. According to Hauschildt's estimate, SORT is of considerable size. One automatic execution results in this deadlock. Perhaps it will feature among next year's surprise models in the next year Model Check Contest. New models are named “surprise models”.



◉ It was interesting for me to see how the simplification I used 30 years ago for teaching purposes has now become central to an important tool.The bankers net is also dicussed in our book from 2003.

WHAT MAKE PETRI NETS SO SPECIAL?

The clear and direct representation of systems.

banker's problem:

concurrency (of clients)

conflict (decision of banker, scarce resources)

operational semantics (token game & reachabilty graph)

verfication by invariants

verfication by model checking

formal analysis by mathematical methods

... and another reason ...

WHAT MAKE PETRI NETS SO SPECIAL?

A wide variety of formalisms with related properties

	concurrency	conflict	operational semantics	invariants	model checking	formal analysis
P/T-nets	x	x	x	x	x	x
self-modifying-nets *(Hamburg)*	x	x	x	x	x	x
coloured nets	x	x	x	x	x	x
nets within nets *(Hamburg)*	x	x	x	x	x	x
timed nets	x	x	x	x	x	x
continous nets	x	x	x	x	x	x
stochastic nets	x	x	x	x	x	x
net agents	x	x	x	x	x	x
PAOSE (Hamburg) *(**P**etri Net-Based **A**gent-and-**O**rganization-Oriented **S**oftware **E**ngineering)*	x	x	x	x	x	x

◉ It was interesting for me to see how the simplification I used 30 years ago for teaching purposes has now become central to an important tool.The bankers net is also dicussed in our book from 2003.

These slides can be found on the web side

https://cycloids.de/

Thank you for your attention !

RÜDIGER VALK
UNIVERSITY OF HAMBURG

PN 2026
HAMBURG