Industrial Automation

(Automação de Processos Industriais)

Analysis of Discrete Event Systems
Complexity and Decidability

http://users.isr.ist.utl.pt/~jag/courses/api19b/api1920.html

Prof. Paulo Jorge Oliveira, original slides
Prof. José Gaspar, rev. 2019/2020

Syllabus:

Chap. 6 – Discrete Event Systems [2 weeks]

…

Chap. 7 – Analysis of Discrete Event Systems [2 weeks]

Properties of DESs.

Methodologies to analyze DESs:

* The Reachability tree.
* The Method of Matrix Equations.

…

Chap. 8 – DESs and Industrial Automation [1 week]

Some pointers to Discrete Event Systems

History: http://prosys.changwon.ac.kr/docs/petrinet/1.htm

Tutorial: http://vita.bu.edu/cgc/MIDEDS/
http://www.daimi.au.dk/PetriNets/

Analyzers, and Simulators: http://www.ppgia.pucpr.br/~maziero/petri/arp.html (in Portuguese)
http://wiki.daimi.au.dk:8000/cpntools/cpntools.wiki
http://www.informatik.hu-berlin.de/top/pnk/download.html

Bibliography:
* Cassandras, Christos G., **"Discrete Event Systems - Modeling and PerformanceAnalysis"**, Aksen Associates, 1993.
* Peterson, James L., **"Petri Net Theory and the Modeling of Systems"**, Prentice-Hall,1981
* **Petri Nets and GRAFCET: Tools for Modelling Discrete Event Systems** R. DAVID, H. ALLA, New York : PRENTICE HALL Editions, 1992

Complexity and Decidibility

*The **reachability tree** and **matrix equation** techniques allow properties of **safeness**, **boundedness**, **conservation**, and **coverability** to be determined for Petri nets. In particular, a necessary condition for reachability is established.*

*However, these techniques are not sufficient to solve several other problems, especially **liveness**, **reachability (sufficient condition)**, and **equivalence**.*

[Peterson 81, ch5]

In the following: we will discuss the complexity and decidability of the problems listed in the later group of the previous paragraph.

Complexity and Decidibility

• Till the end of this chapter, *problem* is intended as a question with yes/no answer,
e.g. Does $\mu' \in R(C,\mu)$ $\forall C, \mu, \mu'$?

• A *problem* is *undecidable* if it is proven that no algorithm to solve it exists.

An example of an undecidable problem is the halting of a Turing machine (TM):

"Will the TM stop for the program n while using the tape m?".

• For *decidable problems*, the *complexity* of the solutions has to be taken into account, that is, the computational cost in terms of memory and time.

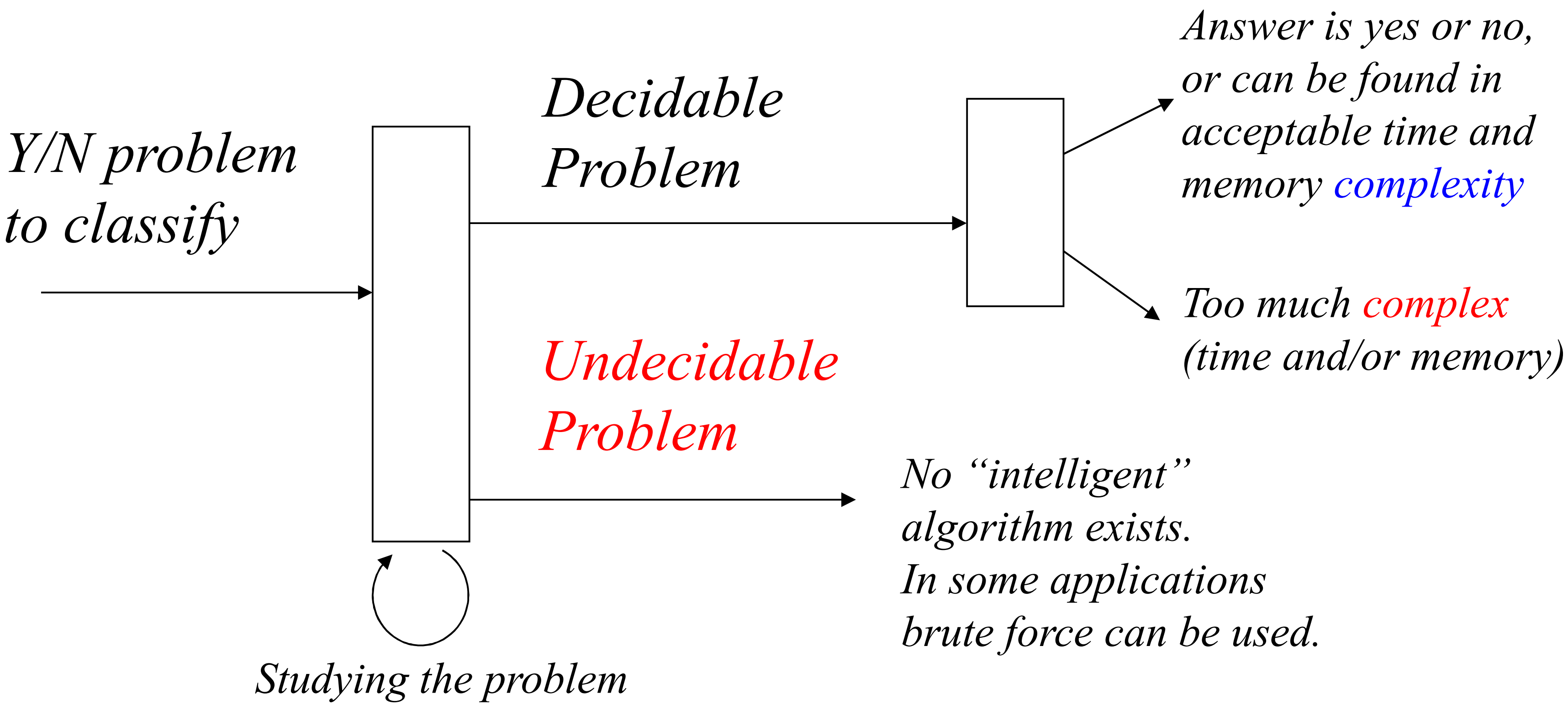
Basic example: a multiplication of numbers has solution (algorithm taught in the school),

but the complexity was different in the arabic and latin civilizations

(how to do a multiplication using roman numbers?)

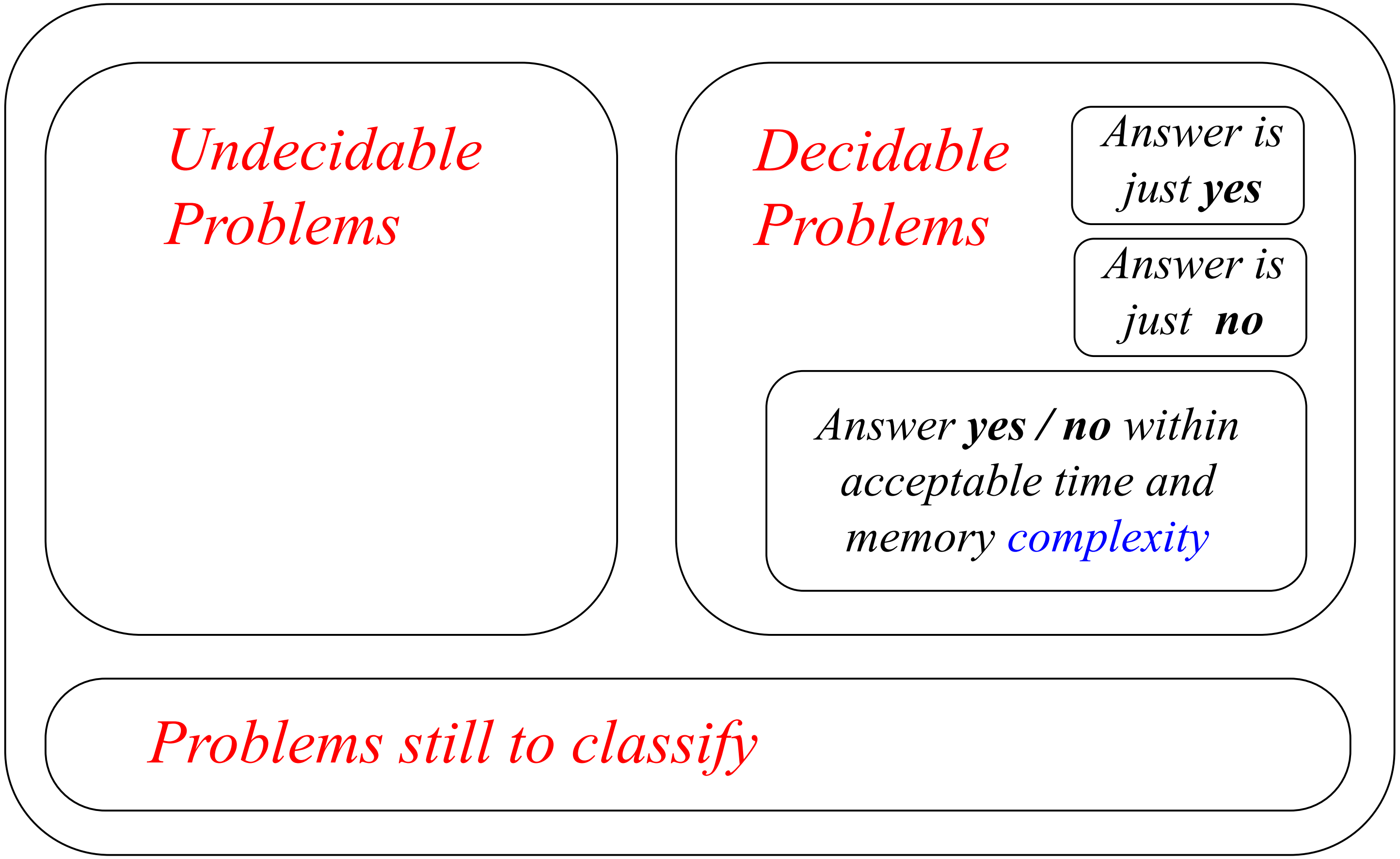
Complexity and Decidibility

Problems with yes or no answers



Complexity and Decidibility

Problems with yes or no answers



Decidibility

If a problem is ≈ undecidable does it mean that it is not solvable?

No, while not proved to be undecidable there is hope it can be solved!

Classical example, Fermat Last Theorem:

Does $x^n + y^n = z^n$ have a solution for n>2 and nontrivial integers x, y e z?

(note that n=2 has infinite solutions, e.g. $3^2+4^2=5^2$ and then $(3m)^2+(4m)^2=(5m)^2$)

Now, it is known that the problem is impossible, i.e. is decidable and needs no algorithm, the answer is *No*. The problem remained ≈ undecidable for more than 300 years (solution proven in 1998).

Turing Machines:

The *Turing Machine (TM) Halting problem* is undecidable.

If it were decidable, for instance the Fermat last theorem would have been proven long time ago, i.e. there would be an algorithm (*TM* with code *n*) that computing all combinations of x,y,z and n>2 (number m) to find a solution verifying $x^n + y^n = z^n$.

Reducibility

One *benefits of reducibility* when to solve a given problem it is *possible to* ***reduce*** *it to another problem with known solution.*

Theorem: Assume that the problem A is reducible to problem B,

then an instance of A can be transformed in an instance of B and:

- If B is decidable then A is decidable.
- If A is undecidable then B is undecidable.

Reducibility

Equality Problem: Given two marked Petri nets

$C_1=(P_1, T_1, I_1, O_1)$ and $C_2=(P_2, T_2, I_2, O_2)$, with markings μ_1 e μ_2, respectively,

is $R(C_1, \mu_1) = R(C_2, \mu_2)$?

Subset Problem: Given two marked Petri nets

$C_1=(P_1,T_1,I_1,O_1)$ and $C_2=(P_2,T_2,I_2,O_2)$, with markings μ_1 e μ_2, respectively,

is $R(C_1,\mu_1) \subseteq R(C_2,\mu 2)$?

The **equality** problem is **reducible** to the **subset** problem
(equality is obtained by proving that each set is a subset of the other)

Reachability Problems

Given a Petri net $C=(P,T,I,O)$ with initial marking μ

Reachability Problem:

Considering a marking μ', does $\mu' \in R(C, \mu)$?

Sub-marking Reachability Problem:

Given the marking μ' and a subset $P' \subseteq P$, exists $\mu'' \in R(C,\mu)$ such that $\mu''(p_i) = \mu' \ \forall p_i \in P'$?

Zero Reachability Problem:

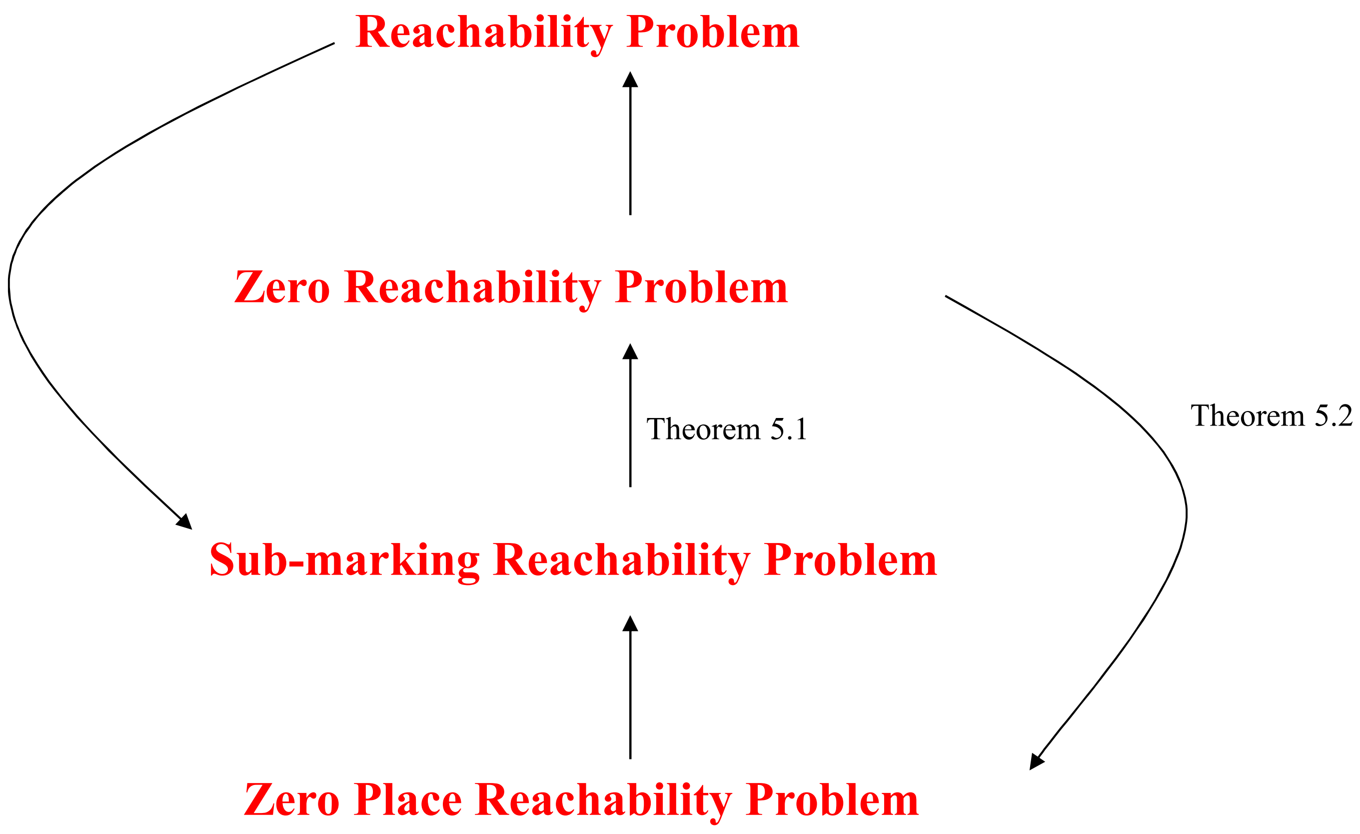
Given the marking $\mu'=(0\ 0\ \ldots\ 0)$, does $\mu' \in R(C,\mu)$?

Zero Place Reachability Problem:

Given the place $p_i \in P$, does $\mu' \in R(C,\mu)$ with $\mu'(p_i) = 0$?

Reachability Problems

Legend:
A→B means A is reducible to B



Reachability Problems

Theorem 5.3: The following reachability problems are equivalent:

- Reachability Problem;
- Zero Reachability Problem;
- Sub-marking Reachability Problem;
- Zero Place Reachability Problem.

[Peterson81]

Liveness and Reachability

(Given a Petri net C=(P,T,I,O) with initial marking μ)

Liveness Problem

Are all transitions t_j of T live?

Transition Liveness Problem

For the transition t_j of T, is t_j live?

The **liveness** problem is **reducible** to the **transition** liveness problem. To solve the first it remains only to solve the second for the *m* Petri net transitions (#T = *m*).

Liveness and Reachability

(Given a Petri net C=(P,T,I,O) with initial marking μ)

Theorem 5.5: The problem of reachability is reducible to the liveness problem.

Theorem 5.6: The problem of liveness is reducible to the reachability problem.

Theorem 5.7: The following problems are **equivalent**:

- **Reachability problem**
- **Liveness problem**

From Esparza and Nielsen [Esparza94]:

Reachability:

*Sacerdote and Tenney claimed in [71] that reachability was **decidable**, but did not give a complete proof. This was not done until 1981 by Mayr [56]; later on, Kosaraju simplified the proof [50], basing on the ideas of [71] and [56]. The proof is very complicated. A detailed and self-contained description can be found in Reutenauer's book [69], which is devoted to it. In [51], Lambert has simplified the proof further.*

(...)

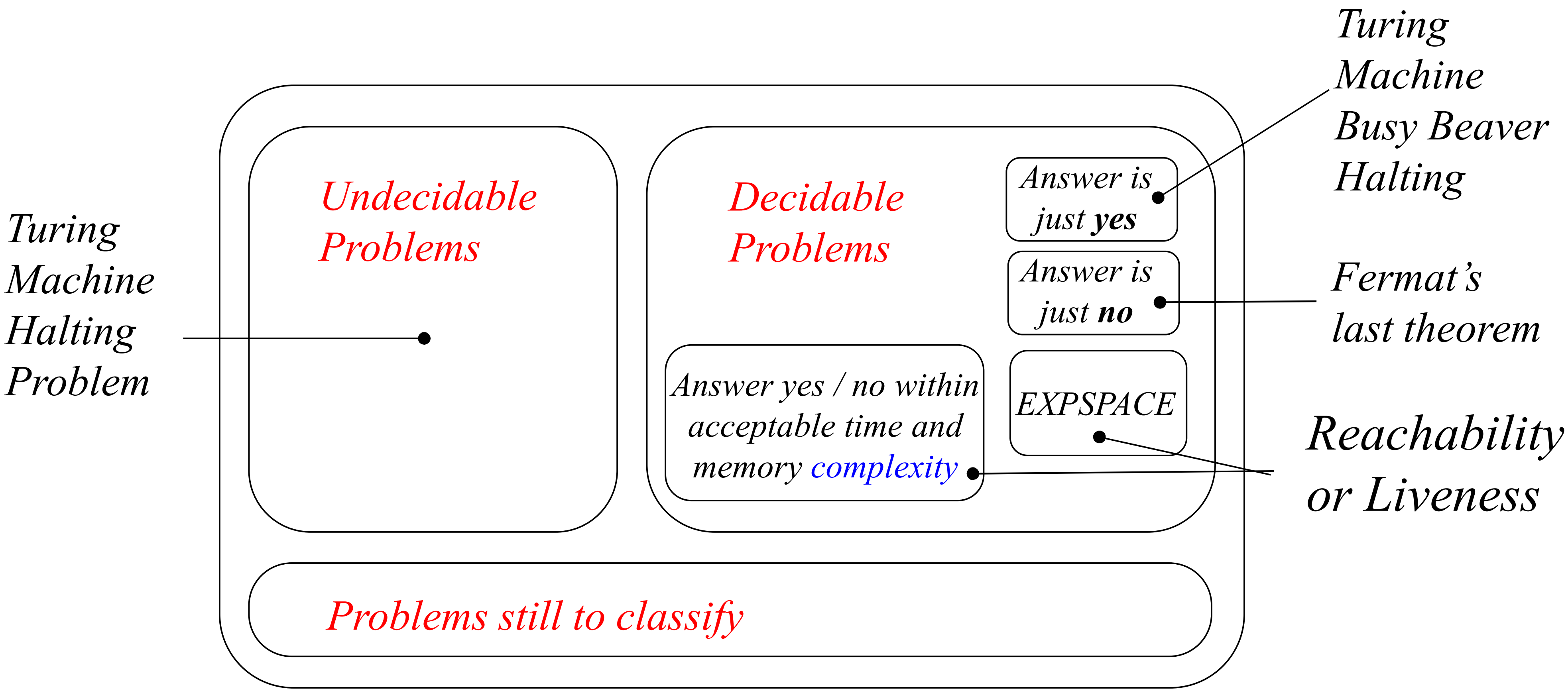
*The **complexity** of the reachability problem has been open for many years. Lipton proved an exponential space lower bound [55], while the known algorithms require non-primitive recursive space.*

Liveness:

*Hack showed in [27] that the liveness problem is recursively equivalent to the reachability problem (see also [1]), and thus **decidable**.*

[Esparza94] Esparza, Javier, and Mogens Nielsen. "Decidability issues for Petri nets." Petri nets newsletter 94 (1994): 5-23.

Complexity and Decidibility



Decidibility

"... most decision problems involving finite-state automata can be solved algorithmically in finite time, i.e., they are ***decidable****. Unfortunately, many problems that are decidable for* ***finite state automata*** *are no longer decidable for* ***Petri nets****, reflecting a natural trade off between decidability and model-richness. (...) Overall, it is probably most helpful to think of Petri nets and automata as* ***complementary modeling approaches****, rather than competing ones.“*

[Cassandras 2008]