Exploring other modal logics

The condensed recapitulative version

Ethan Arm Dany A. Darghouth – May 2024

MODAL LOGIC

What is it ?

Definition:

A kind of logic used to represents statements about necessity and possibility.

Formally:

It is a formal system which include unary operators such as ◊ and □, which represents the possibility and obligation respectfully.

Example:

$\Diamond P$ *can be read as possibly P*
$\Box P$ *can be read as necessarily P*

Logics:

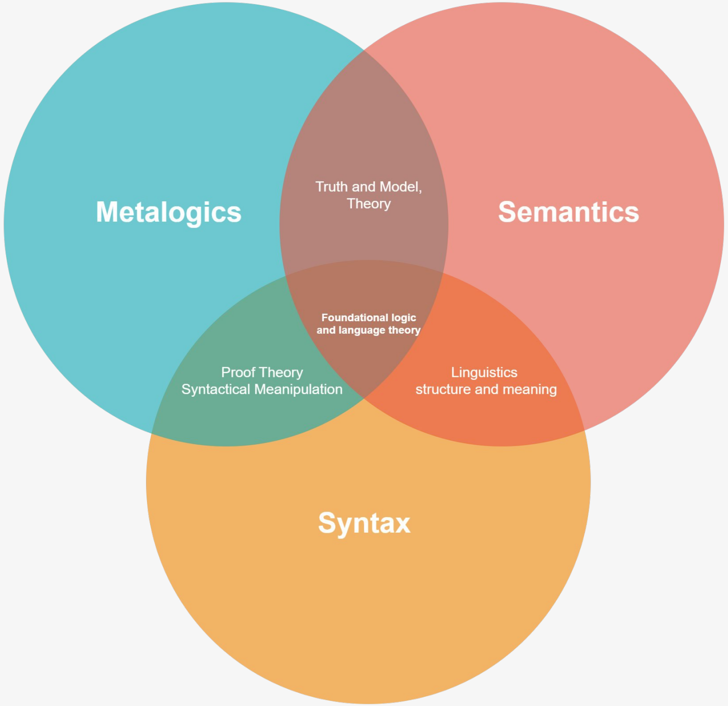
In this category we have epistemic logic, deontic logic, but also CTL, PAST + CTL or PCTL, etc…

META-LOGICS

The logic of logics

Definition:

"The study of the properties, structure, and limitations of logical systems."



Soundness:

A logical system is sound if all statements that can be proven within the system are true in its interpretation.

Example:

In a sound system, if we can prove P→Q and we know P is true, then Q must also be true.

Completeness:

A logical system is complete if every statement that is true in its interpretation can be proven within the system.

Example:

In a complete system, if P∧Q is true, then there exists a proof for P∧Q within the system.

1st Theorem:

There are true statements in the system that cannot be proven within the system.

Example:

"This statement cannot be proven."

2nd Theorem:

No consistent formal system capable of expressing elementary arithmetic can prove its own consistency.

Example:

If a system could prove its own consistency, it would lead to paradoxical situations, undermining the system's basis.

Orders of logics

What are the classes of logics ?

Orders of logics

0th order

1st order

Higher order

Properties over orders

Specifically soundness and completeness

Properties over orders

0th order

1st order

Higher order

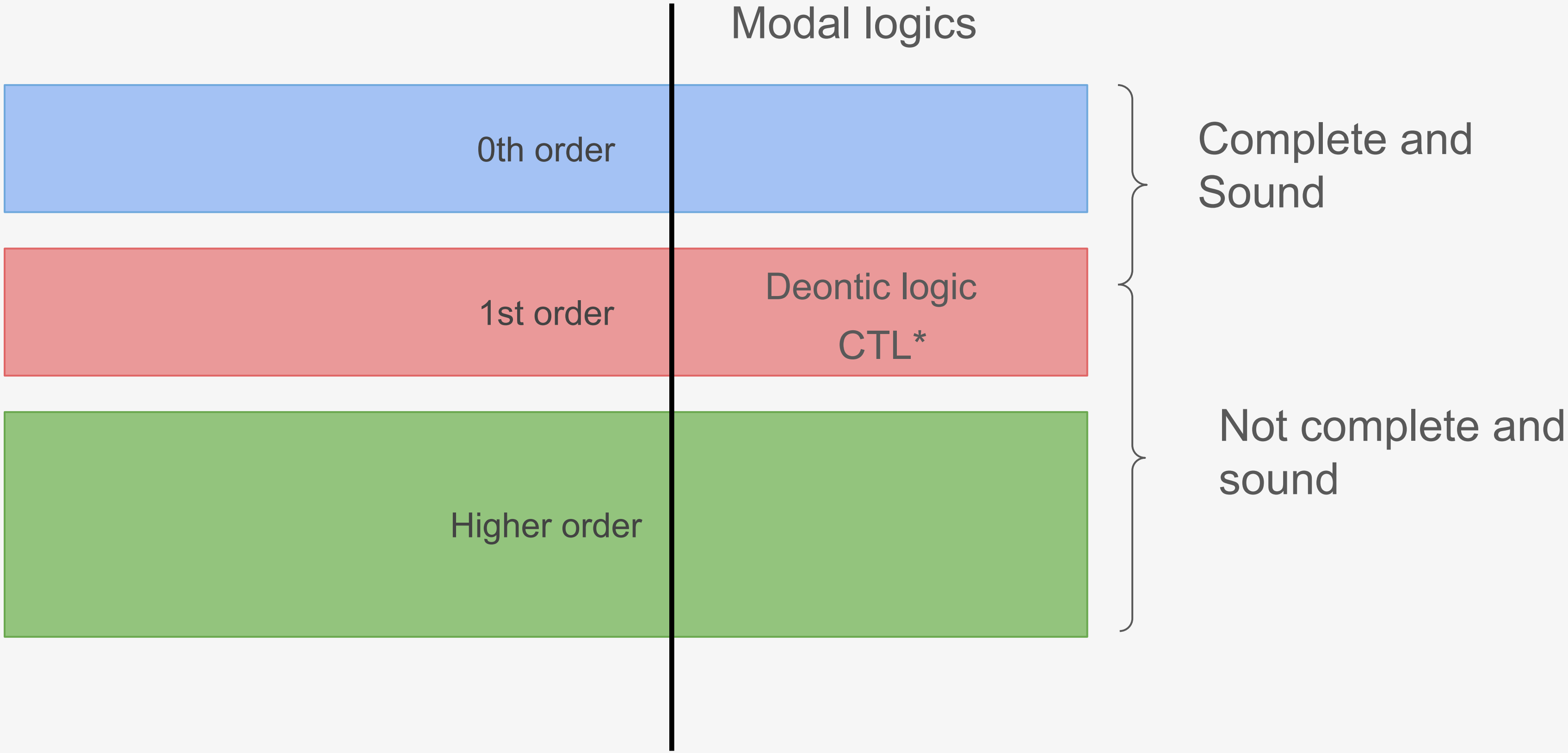
Complete and Sound

Not, complete and sound

Applications to modal logics

How does that affects our work?

Properties over orders



Translating Specifications

Translating Specification

Expressing properties using either deontic logic and or ctl(*), and translating them to the other:

- **P** is a property of a given system
- A Kripke structure is implicitly defined

Translating Specification

DEONTIC ➨ CTL(*)

- **O(p)** → **AX p** is enough
- **P(p)** → **EX p** is enough
- **F(p)** → **AG ¬p**

Deontic logic is less expressive so everything is fine…

Translating Specification

CTL(*) ➨ DEONTIC LOGIC

Target logic is less expressive

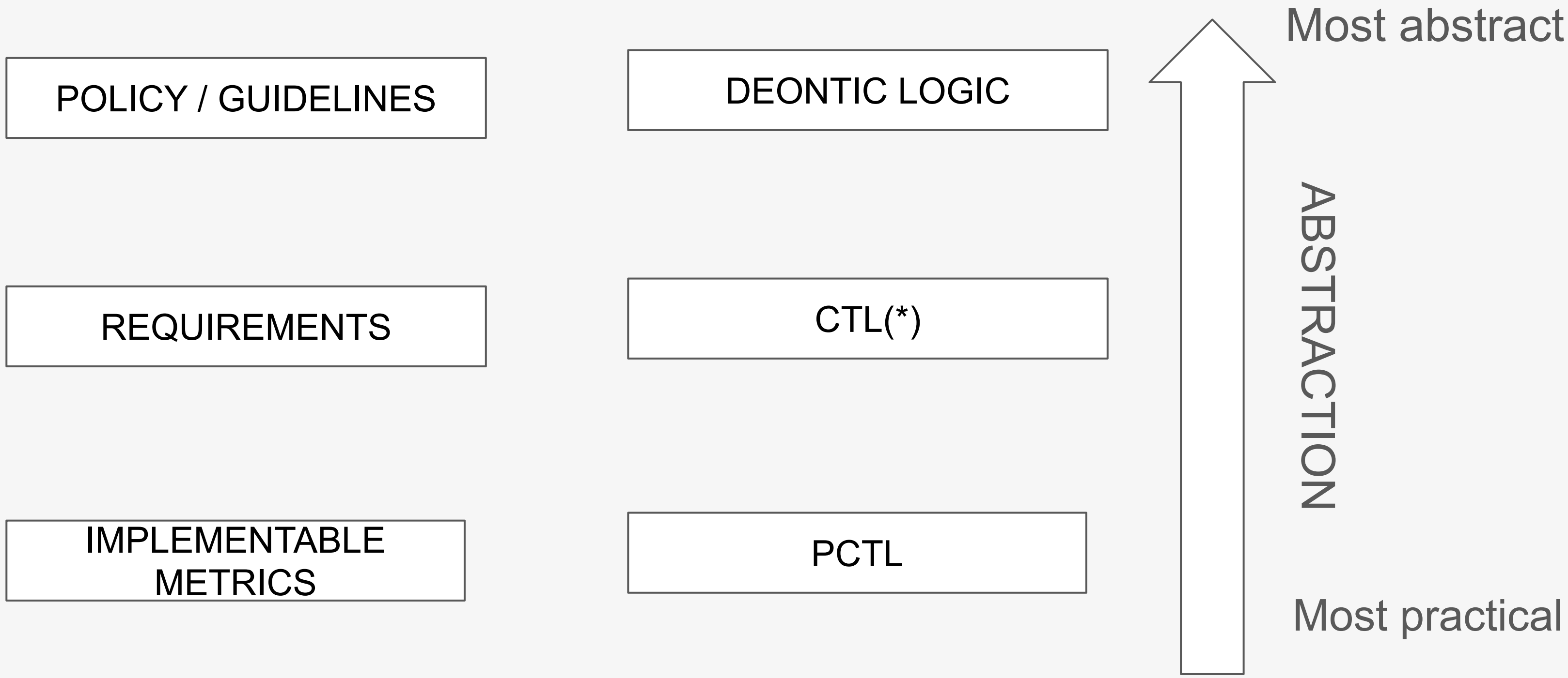
➨ INFORMATION LOSS

Defining a property p’ that works well is a solution..

Abstraction pipeline

How all of this could work in a real setting.

Abstraction pipeline



Conclusion

Conclusion

- Each logic has its use cases
- All three can intervene (even if informally) in the life of a project

THANK YOU

Any questions? Remarks ?