Linking Proof Theory to Game Design

Paola Bruscoli

University of Bath

Univ of Strathclyde - BCTCS (Educational Track) – April 14–16, 2025

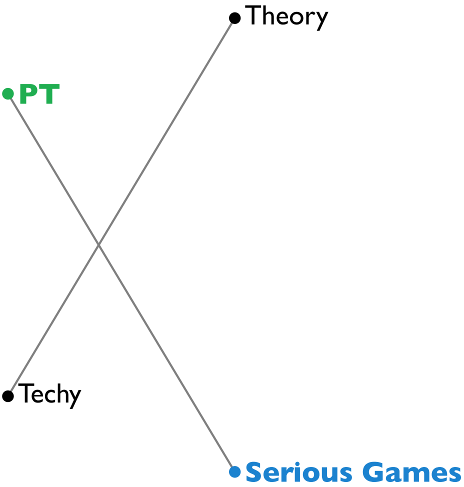
Context

- UG/PGT in CompSci (+ Maths), campus-based
- Environment influenced by AI/ML/HCI
- Individual dissertation projects opportunity to promote TCS
- Feasibility and affordability - 24 credits, busy students

My profile: substructural logics, proof theory, deep inference

Students have seen natural deduction

Context



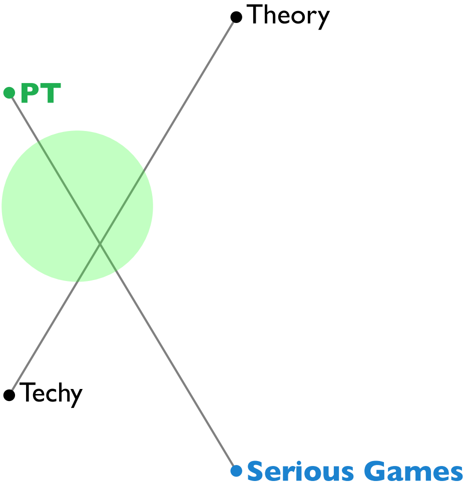
Theory

PT

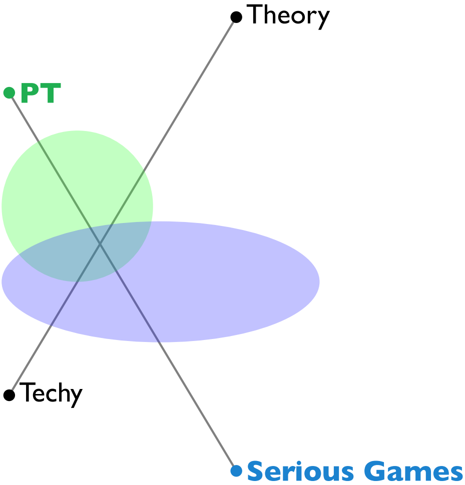
Techy

Serious Games

Context



Context



Deep Inference[1]

- No main connective (trees) (no branching rules)
- rules applied ’deep’ inside formulae (contextual closure preserves logical implication)

Compose $\phi = \begin{array}{c} A \\ \| \\ B \end{array}$ and $\psi = \begin{array}{c} C \\ \| \\ D \end{array}$

with $\wedge$ / $\vee$ to obtain

$$(\phi \wedge \psi) = \begin{array}{c} (A \wedge C) \\ \| \\ (B \wedge D) \end{array} \quad \text{and} \quad [\phi \vee \psi] = \begin{array}{c} [A \vee C] \\ \| \\ [B \vee D] \end{array}$$

[1] *Deep inference web site:* http://alessio.guglielmi.name/res/cos/

Deep Inference[2]

- A language of ’formulae’/structures..
- An equality theory on the language of structures..
- The (carefully designed) proof system (set of inference rules)..
- The proof theory (indeed!)

$$\begin{array}{c}(a \vee b) \wedge a \\ \| \\ ((a \vee b) \wedge a) \wedge ((a \vee b) \wedge a)\end{array} \equiv \boxed{\mathsf{m}\frac{\boxed{\mathsf{c}{\uparrow}\frac{a}{a \wedge a}} \vee \boxed{\mathsf{c}{\uparrow}\frac{b}{b \wedge b}}}{(a \vee b) \wedge (a \vee b)}} \wedge \boxed{\mathsf{c}{\uparrow}\frac{a}{a \wedge a}}$$

[2] *Deep inference web site:* http://alessio.guglielmi.name/res/cos/

I) Deep Inference projects

Brief - Construct an interpreter for some proof system and logic

Learning lens – knowledge acquisition in structural proof theory; skills in implementation (Haskell)

Typical Challenges in Implementation

- Nested contexts/deep application of rules
- "locality" (atomic axioms + depth) increments non-determinism in proof search space
- "strategies" in rules' application needed (informed by theory)

... can be more demanding in some logics/proof systems

I) BV – Sequentiality and Depth

CCS: $a.b \mid \bar{a}.\bar{b} \xrightarrow{\tau} b \mid \bar{b} \xrightarrow{\tau} 0.$

BV:

$$\circ\!\downarrow \frac{}{\circ} \qquad \mathsf{ai}\!\downarrow \frac{S\{\circ\}}{S[a,\bar{a}]} \qquad \mathsf{q}\!\downarrow \frac{S\langle [R,T];[R',T']\rangle}{S[\langle R;R'\rangle,\langle T;T'\rangle]} \qquad \mathsf{s}\,\frac{S([R,T],R')}{S[(R,R'),T]}$$

BVL: Enforces prefixing, with a proviso in the choice of context

$$\mathsf{ai}\!\downarrow_{\llcorner} \frac{S\{\circ\}^{\llcorner}}{S[a,\bar{a}]^{\llcorner}} \,:$$

I) BVL: Prefixing + Par

$$\mathsf{q}{\downarrow}\,\dfrac{\mathsf{q}{\downarrow}\,\dfrac{\mathsf{q}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}_{\mathsf{L}}\,\dfrac{\mathsf{ai}{\downarrow}_{\mathsf{L}}\,\dfrac{\mathsf{ai}{\downarrow}_{\mathsf{L}}\,\dfrac{\circ{\downarrow}\,\dfrac{}{\circ}}{[a,\bar{a}]}}{\langle[c,\bar{c}];[a,\bar{a}]\rangle}}{\langle[a^\bullet,\bar{a}^\bullet];[c,\bar{c}];[a,\bar{a}]\rangle}}{\langle[a^\bullet,\bar{a}^\bullet];[a,\bar{a},c,\bar{c}]\rangle}}{[a,\langle[a^\bullet,\bar{a}^\bullet];[\bar{a},c,\bar{c}]\rangle]}}{[a,\langle a^\bullet;[\bar{a},c]\rangle,\langle\bar{a}^\bullet;\bar{c}\rangle]} \;\longrightarrow\; \mathsf{q}{\downarrow}\,\dfrac{\mathsf{ai}{\downarrow}_{\mathsf{L}}\,\dfrac{\mathsf{ai}{\downarrow}_{\mathsf{L}}\,\dfrac{\circ{\downarrow}\,\dfrac{}{\circ}}{[a,\bar{a}]}}{\langle[c^\bullet,\bar{c}^\bullet];[a,\bar{a}]\rangle}}{[a,\bar{a},c^\bullet,\bar{c}^\bullet]} \;\longrightarrow\; \mathsf{ai}{\downarrow}_{\mathsf{L}}\,\dfrac{\circ{\downarrow}\,\dfrac{}{\circ}}{[a^\bullet,\bar{a}^\bullet]} \;\longrightarrow\; \circ{\downarrow}\,\dfrac{}{\circ} \quad ;$$

$$a.\circ \mid a.(\bar{a}.\circ \mid c.\circ) \xrightarrow{a} a.\circ \mid \bar{a}.\circ \mid c.\circ \xrightarrow{c} a.\circ \mid \bar{a}.\circ \xrightarrow{\circ} \circ \quad .$$

2) Serious Games Development

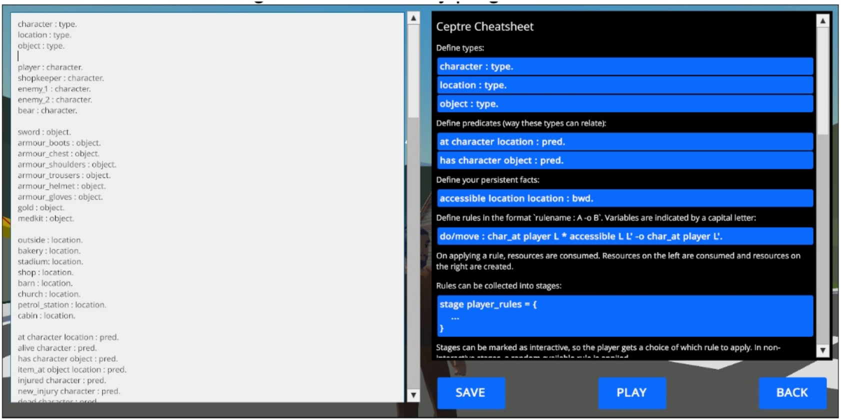
Brief - Use a logical engine to specify the rules of a game, control the game mechanics in game prototyping.
Proposed system: CEPTRE (Chris Martens) fragment of ILL

Learning lens – logics for action and change; use of rule based systems; creation of the game; problem solving to create the pipeline for a GUI

Typical Challenges

- Understanding the logical background starting from the implemented logical framework
- Use CEPTRE output (StandardML) to Unity (C#) may require some choices (scripting)

2) Serious Games Development



```
character : type.
location : type.
object : type.

player : character.
shopkeeper : character.
enemy_1 : character.
enemy_2 : character.
bear : character.

sword : object.
armour_boots : object.
armour_chest : object.
armour_shoulders : object.
armour_trousers : object.
armour_helmet : object.
armour_gloves : object.
gold : object.
medkit : object.

outside : location.
bakery : location.
stadium: location.
shop : location.
barn : location.
church : location.
petrol_station : location.
cabin : location.

at character location : pred.
alive character : pred.
has character object : pred.
item_at object location : pred.
injured character : pred.
new_injury character : pred.
dead character : pred.
```

Ceptre Cheatsheet

Define types:

```
character : type.
location : type.
object : type.
```

Define predicates (way these types can relate):

```
at character location : pred.
has character object : pred.
```

Define your persistent facts:

```
accessible location location : bwd.
```

Define rules in the format `rulename : A -o B`. Variables are indicated by a capital letter:

```
do/move : char_at player L * accessible L L' -o char_at player L'.
```

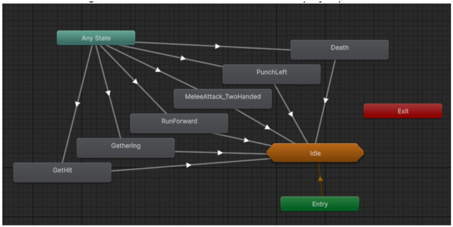
On applying a rule, resources are consumed. Resources on the left are consumed and resources on the right are created.

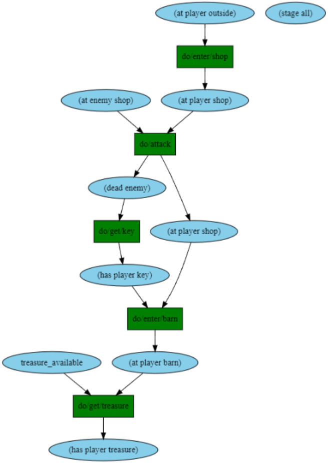
Rules can be collected into stages:

```
stage player_rules = {
    ...
}
```

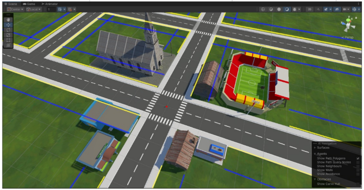
Stages can be marked as interactive, so the player gets a choice of which rule to apply. In non-interactive stages, a random available rule is applied.

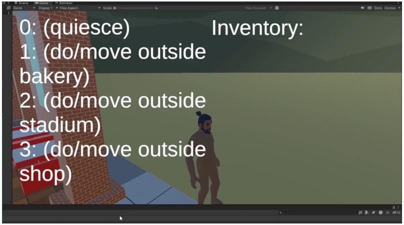
2) Serious Games Development





2) Serious Games Development





0: (quiesce)
1: (do/move outside bakery)
2: (do/move outside stadium)
3: (do/move outside shop)

Inventory:

Some thoughts on these projects

- CEPTRE has been positively valued by our students/game-designers
- The more theoretical projects are an **excellent and impressive presentation** for UG candidates in jobs market!
- Student's interests/priorities change
- Now: the generation of post-Covid UG-students
- Teaching specific TCS courses in final years helps attracting students - unbalance towards ML
- Would reading seminars for UG/MComp help in constructing a dynamic chain towards more theoretically solid projects?

References

[1] Paola Bruscoli (2002): *A Purely Logical Account of Sequentiality in Proof Search.* In Peter J. Stuckey, editor: *Logic Programming, 18th International Conference (ICLP)*, *Lecture Notes in Computer Science* 2401, Springer-Verlag, pp. 302–316, doi:10.1007/3-540-45619-8_21. Available at http://cs.bath.ac.uk/pb/bvl/bvl.pdf.

[2] Alessio Guglielmi (2007): *A System of Interaction and Structure.* *ACM Transactions on Computational Logic* 8(1), pp. 1:1–64, doi:10.1145/1182613.1182614. Available at http://cs.bath.ac.uk/ag/p/SystIntStr.pdf.

[3] Alessio Guglielmi & Lutz Straßburger (2011): *A System of Interaction and Structure V: The Exponentials and Splitting.* *Mathematical Structures in Computer Science* 21(3), pp. 563–584, doi:10.1017/S096012951100003X. Available at http://www.lix.polytechnique.fr/~lutz/papers/NEL-splitting.pdf.

[4] Ozan Kahramanoğulları (2008): *On Linear Logic Planning and Concurrency.* In Carlos Martín-Vide, Friedrich Otto & Henning Fernau, editors: *Language and Automata Theory and Applications*, *Lecture Notes in Computer Science* 5196, Springer-Verlag, pp. 250–262. Available at http://www.wv.inf.tu-dresden.de/~guglielm/ok/lata.pdf.

[5] Chris Martens (2021): *CEPTRE: A language for modelling generative interactive systems.* In Arnav Jhala & Nathan Sturtevant, editors: *Proceedings of the AAAI Conference on Artificial Intelligence and Digital Entertainment*, PKP | PS, pp. 51–57.

References (cont.)

[6] Luca Roversi (2011): *Linear Lambda Calculus and Deep Inference.* In Luke Ong, editor: *Typed Lambda Calculi and Applications*, *Lecture Notes in Computer Science* 6690, Springer-Verlag, pp. 184–197, doi:10.1007/978-3-642-21691-6_16. Available at `http://www.di.unito.it/~rover/RESEARCH/PUBLICATIONS/2011-TLCA/Roversi2011TLCA.pdf`.

[7] Lutz Straßburger & Alessio Guglielmi (2011): *A System of Interaction and Structure IV: The Exponentials and Decomposition.* *ACM Transactions on Computational Logic* 12(4), pp. 23:1–39, doi:10.1145/1970398.1970399. Available at `http://arxiv.org/pdf/0903.5259`.