CUTECat: Concolic Execution for Computational Law

Pierre Goutagny*, Aymeric Fromherz, Raphaël Monat

pierre.goutagny@inria.fr Inria

The Catala programming language

Catala [1, 5] is a domain-specific language designed to implement computational laws.

Article 1 **Formula**

The income tax for an individual is defined as a fixed percentage of the individual's income over a year.

```
definition income_tax equals
  income * tax_rate
```

Article 2 **Default case**

The fixed percentage mentioned at article 1 is equal to 20%.

```
definition tax_rate equals 20%
```

Article 3 **Exception**

If the individual's income is less than $10,000, the fixed percentage mentioned at article 1 is equal to 10%.

```
exception definition tax_rate
  under condition income <= $10,000
  consequence equals 10%
```

Article 4 **Exception**

If the individual is in charge of 3 or more children, then the fixed percentage mentioned at article 1 is equal to 15%.

```
exception definition tax_rate
  under condition nb_children >= 3
  consequence equals 15%
```

Catala programs errors

Interpretation conflicts

To prevent implicit assumptions, Catala is designed to crash if two provisions apply at the same time. Such ambiguities must be resolved with lawyers.

Example: articles 3 and 4 will conflict when `income` = $10,000 and `nb_children` = 4.

Unhandled cases

Catala also crashes when no rule applies.

Example: the law stops applying after a date, and that date has passed.

Other bugs

Usual programming errors happen too.

Example: division by zero, assertion errors...

Default term

$$e ::= \langle \underbrace{e_1, \ldots, e_n}_{\text{exceptions}} \mid \underbrace{b_{default}}_{\text{guard}} \texttt{ :- } \underbrace{e_{default}}_{\text{base case}} \rangle$$

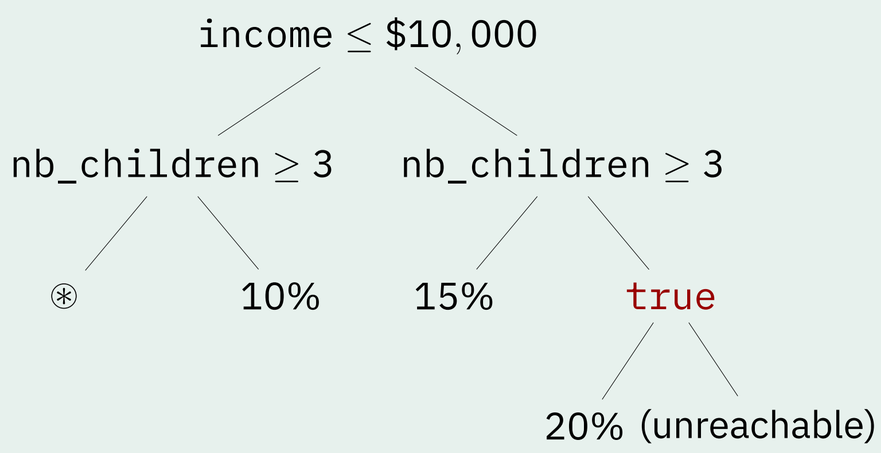
```
let tax_rate =
  ⟨
    ⟨ | income ≤ $10,000 :- 10%⟩,
    ⟨ | nb_children ≥ 3 :- 15%⟩
    | true :- 20%
  ⟩
in income * tax_rate
```

Default term semantics

Evaluate all exceptions, then count how many are *raised*:

- If exactly 1 exception is raised, then return its value
- Else if at least 2 exceptions are raised, then return ⊛ (conflict)
- Else if 0 exceptions are raised, evaluate $b_{default}$ and
  - If $b_{default} =$ `true`, then evaluate $e_{default}$
  - Else if $b_{default} =$ `false`, then return ∅ (empty)

Concolic execution of default terms



Step	income	nb_children	Output
1	9,000	4	⊛
2	9,000	2	10%
3	11,000	4	15%
4	11,000	2	20%
5	*no SAT execution paths remaining*		

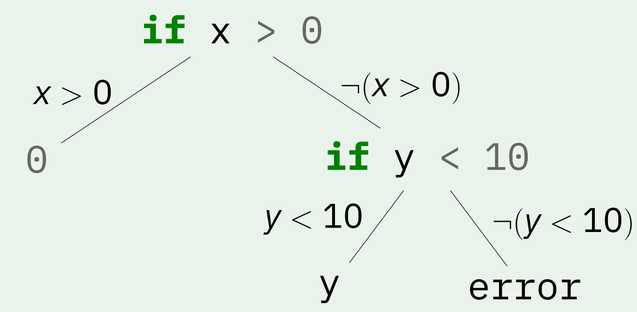
Concolic execution

Concolic execution [3] combines *conc*rete and symb*olic* execution.

Program

```
if x > 0
    then 0
    else if y < 10
        then y
        else error
```

Execution tree



Execution trace

Step	x	y	Output	Collected constraints	Next path to try	
1	1	20	0	$x > 0$	$\neg(x > 0)$	↻ SMT
2	0	20	error	$\neg(x > 0) \wedge \neg(y < 10)$	$\neg(x > 0) \wedge y < 10$	↻ SMT
3	0	9	9	$\neg(x > 0) \wedge y < 10$	-	

Optimizations

For performance

- SMT solver incremental mode
- redundant constraint elimination
- reordering exceptions

For usability

Simplify the format of generated values using *soft*, optional constraints.

Example: round money to the unit

Results

- CUTECat tool [2, 4] integrated in Catala ecosystem
- On medium-sized implementation of French housing benefits law:
  - Generates ∼200k test cases in 6h
  - Finds an interpretation conflict
  - Optimizations cut analysis time in half
  - 4.5x overhead w.r.t. Catala concrete interpreter, comparable to other symbolic analysis tools

[1] Catala website (2025), URL https://catala-lang.org/

[2] CUTECat peer-reviewed artifact (2025), DOI: 10.5281/zenodo.14677860

[3] Godefroid, P., Klarlund, N., Sen, K.: DART: Directed automated random testing. In: PLDI (2005), DOI: 10.1145/1065010.1065036

[4] Goutagny, P., Fromherz, A., Monat, R.: CUTECat: Concolic Execution for Computational Law. In: ESOP (2025), DOI: 10.1007/978-3-031-91121-7_2

[5] Merigoux, D., Chataing, N., Protzenko, J.: Catala: A programming language for the law. ICFP (2021), DOI: 10.1145/3473582

*Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, 59000 Lille, France