# From release 0.91 to release 0.95.1

The Alt-Ergo developers team

# Overview of the main changes

- arithmetic enhancements
- AC symbols
- new built-in theories : arrays, enumerated data types, records
- models / unsat cores extraction
- a graphical interface (AltGr-Ergo)
- Alt-Ergo-Zero library

# Arithmetic reasoning

Linear arithmetic on $\mathbb{Z}$

- a new decision procedure FM-Simplex
- good results on QF-LIA categoy of SMT benchmarks
- published at [IJCAR 2012]

Non-linear arithmetic

- Euclidean division and modulo operators
- interval calculus
- non-linear multiplication
- good results on ANR Decert benchmark

# AC Symbols

AC(X), new algorithm for combining a Shostak theory X with a decision procedure for AC symbols

- published at [TACAS 2011, LMCS 2012]
- EATCS award for Best Theoretical Paper at ETAPS 2011

```
logic ac u : int, int -> int
goal g :
 forall x,y,z,a,b:int.
  u(a,b)-b = x and u(a+b,c) = y and b = 0 ->
  u(0,y) = u(c,x)
```

# New built-in theories

## Functional arrays

```
logic a : (int, int) farray
goal g1 : forall i:int. i=6 -> a[i<-4][5] = a[i-1]
```

## Records

```
type 'a t = { a : int; b : 'a }
goal g2 : forall v,w:int t.
  2 * v.a  = 10 -> { v with b = 5} = w -> w.a = 5
```

## Enumerated data types

```
type t = A | B | C
logic P : t -> prop
goal g3 : forall x:t. P(C) -> x<>A and x<>B -> P(x)
```

# Models extraction

```
logic x "model:0", y "model:0" : int
goal g: x >= 42 -> x <> y -> y = 45 -> ((x + 1)) <= 40
```

```
alt-ergo -model <file>
```

```
Propositional:
42 <= x
x <> y
y = 45
```

```
Theory:
y = X1(arith):[45 [int]]
x <> y
```

```
Relation:
```

$x \in [42; 44] \cup [46; +\infty[$

# Unsat cores extraction

```
logic x, y : int
goal g: x >= 4 -> x <> y -> y = 2 -> y - x <= 0
```

```
alt-ergo -proof <file>
```

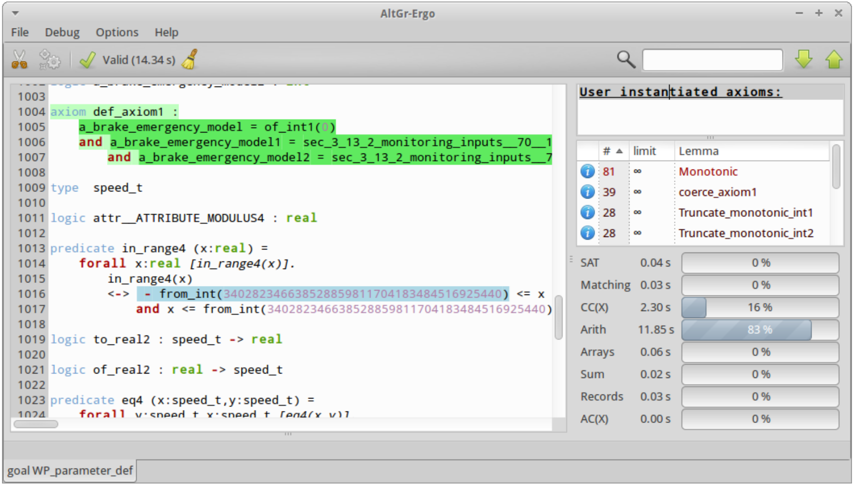
```
Proof:
   4 <= x
   y = 2
   (y - x) > 0
```

x <> y is not used to derive the unsatisfiability

# AltGr-Ergo : capabilities

- selection/deletion of axioms and hypotheses
- deletion/modification of triggers
- manual (and possibly partial) axioms instantiation
- highlight which axioms/hypotheses were useful to prove a goal
- axioms instantiation and decision procedures profiling
- save/replay modifications in/from a session file

# AltGr-Ergo : example



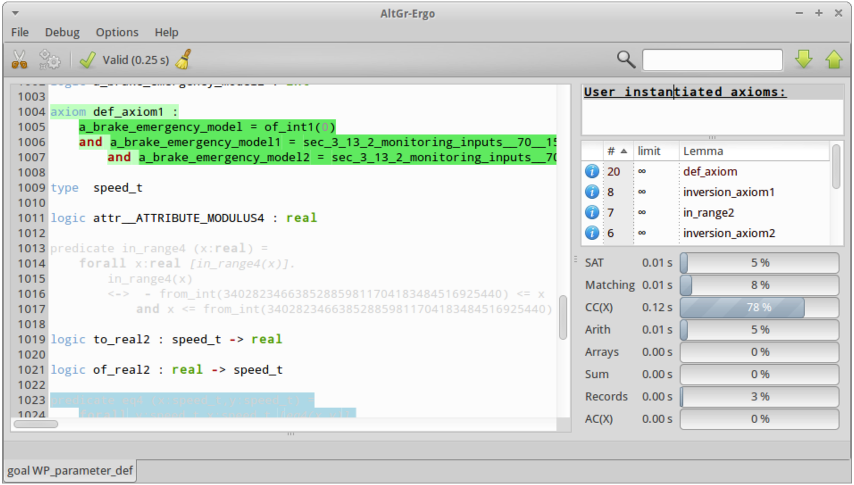
```
axiom def_axiom1 :
    a_brake_emergency_model = of_int1(0)
    and a_brake_emergency_model1 = sec_3_13_2_monitoring_inputs__70__1
        and a_brake_emergency_model2 = sec_3_13_2_monitoring_inputs__7

type  speed_t

logic attr__ATTRIBUTE_MODULUS4 : real

predicate in_range4 (x:real) =
    forall x:real [in_range4(x)].
        in_range4(x)
        <->  - from_int(340282346638528859811704183484516925440) <= x
            and x <= from_int(340282346638528859811704183484516925440)

logic to_real2 : speed_t -> real

logic of_real2 : real -> speed_t

predicate eq4 (x:speed_t,y:speed_t) =
    forall y:speed_t,x:speed_t [eq4(x,y)].
```

User instantiated axioms:

|  | # ▲ | limit | Lemma |
|---|---|---|---|
| ⓘ | 81 | ∞ | Monotonic |
| ⓘ | 39 | ∞ | coerce_axiom1 |
| ⓘ | 28 | ∞ | Truncate_monotonic_int1 |
| ⓘ | 28 | ∞ | Truncate_monotonic_int2 |

|  |  |  |
|---|---|---|
| SAT | 0.04 s | 0 % |
| Matching | 0.03 s | 0 % |
| CC(X) | 2.30 s | 16 % |
| Arith | 11.85 s | 83 % |
| Arrays | 0.06 s | 0 % |
| Sum | 0.02 s | 0 % |
| Records | 0.03 s | 0 % |
| AC(X) | 0.00 s | 0 % |

# AltGr-Ergo : example



```
axiom def_axiom1 :
    a_brake_emergency_model = of_int1(0)
    and a_brake_emergency_model1 = sec_3_13_2_monitoring_inputs__70__15
       and a_brake_emergency_model2 = sec_3_13_2_monitoring_inputs__70

type  speed_t

logic attr__ATTRIBUTE_MODULUS4 : real

predicate in_range4 (x:real) =
    forall x:real [in_range4(x)].
       in_range4(x)
       <->  - from_int(340282346638528859811704183484516925440) <= x
          and x <= from_int(340282346638528859811704183484516925440)

logic to_real2 : speed_t -> real

logic of_real2 : real -> speed_t

predicate eq4 (x:speed_t,y:speed_t) =
    forall y:speed_t,x:speed_t [eq4(x,y)].
```

**User instantiated axioms:**

| # ▲ | limit | Lemma |
|---|---|---|
| 20 | ∞ | def_axiom |
| 8 | ∞ | inversion_axiom1 |
| 7 | ∞ | in_range2 |
| 6 | ∞ | inversion_axiom2 |

| | | |
|---|---|---|
| SAT | 0.01 s | 5 % |
| Matching | 0.01 s | 8 % |
| CC(X) | 0.12 s | 78 % |
| Arith | 0.01 s | 5 % |
| Arrays | 0.00 s | 0 % |
| Sum | 0.00 s | 0 % |
| Records | 0.00 s | 3 % |
| AC(X) | 0.00 s | 0 % |

goal WP_parameter_def

# A new library

an OCaml SMT library

Enhanced and light version of Alt-Ergo :

- a new SAT solver based on a re-implementation of `minisat`
- incremental
- support several instances
- no quantifiers
- used in `model-checking` and `k-induction`

# Additional explored topics

- a lightweight proofs certification mechanism using COQ
- built-in support of floating point numbers
    - integration of Gappa in Alt-Ergo [SMT-Workshop 2012]

# Why3 benchmark

**1920 formulas** **timeout : 30 seconds**

| | trunk | 0.95.1 | 0.94 | 0.93 | 0.92.2 | 0.91 |
|---|---|---|---|---|---|---|
| valid | **1841** | **1841** | 1811 | 1773 | 1737 | 1685 |
| time | 353 | 362 | 465 | 411 | 527 | 555 |
| unknown | 21 | 20 | 25 | 24 | 22 | 21 |
| time | 47 | 13 | 31 | 25 | 33 | 45 |
| timeout | 58 | 59 | 83 | 99 | 128 | 175 |
| errors | 0 | 0 | 1 | 24 | 33 | 39 |

# Hi-lite public benchmark

**3583 formulas** **timeout : 30 seconds**

| | trunk | 0.95.1 | 0.94 | 0.93 | 0.92.2 | 0.91 |
|---|---|---|---|---|---|---|
| valid | **2418** | 2397 | 2352 | 1526 | 2286 | 2374 |
| time | 847 | 723 | 988 | 158 | 869 | 1110 |
| unknown | **649** | 487 | 138 | 76 | 378 | 387 |
| time | 2931 | 1438 | 975 | 196 | 1217 | 1353 |
| timeout | **518** | 691 | 995 | 283 | 910 | 815 |
| errors | 0 | 10 | 100 | 1700 | 11 | 7 |

# What is next ?

- Floating point numbers, COQ certification, lemmas instantiation, models generation, non-linear arithmetic, . . .
- ANR bware project
  - improving Alt-Ergo for POs coming from Atelier-B
- commercial support for Alt-Ergo by OCamlPro