

Verificación de Programas con F^*

Clase 10 – 28/05/2024

Anuncios parroquiales

- La próxima semana (martes 4/6) no hay clase
- **Empezar a pensar proyectos**
 - Una estructura de datos
 - Algún algoritmo relativamente simple
 - Algún programa real (con IO, etc) y algunas pruebas ligeras sobre el mismo
 - Formalizar alguna estructura matemática (grupo, anillo, etc)
 - Formalizar algún lenguaje de programación
- 11/6: Gabriel Ebner sobre Lean4/mathlib
- Otra clase invitada: Ideas?


Pruebas calculacionales

```
let lem1 (a : pos) : Lemma (2 * a > a) = ()
```

```
let calc0 (a : pos) : Lemma (a + a > a) =
```

```
  calc (>) {  Relación final
```

```
    a + a;
```

```
    == {} 
```

```
    2 * a;
```

```
    > { lem1 a } 
```

```
    a;
```

```
  }
```

Relaciones intermedias
(deben ser *compatibles*)

Pruebas calculacionales, elaboradas

```
let lem1 (a : pos) : Lemma (2 * a > a) = ()
let calc0_desugared (a : pos) : Lemma (a + a > a) =
  calc_finish (fun x y -> (>) x y <: Type) (fun () ->
    calc_step (fun x y -> (>) x y <: Type) a (fun () ->
      calc_step (fun x y -> (==) x y <: Type) (2 * a) (fun () ->
        calc_init (a + a)
      ) (fun () -> ()))
    ) (fun () -> lem1 a)
  )
```

- Cada paso se chequea *independientemente*: la prueba de uno (o **admit()**) no puede afectar al otro.

Tácticas

(otra presentación y archivo)

Poly1305: performance con y sin canon

- Sin canon: lento e impredecible, incluso con 100x de límite.
- Con canon: rápido y robusto.
 - (Rate: % de éxito, otras columnas en segundos)
- Ver sección 6.1 del [Paper de Meta-F*](#) para más detalles.

	Rate	Queries	Tactic	Total
smt1x	0.5%	0.216 ± 0.001	–	2.937
smt2x	2%	0.265 ± 0.003	–	2.958
smt3x	4%	0.304 ± 0.004	–	3.022
smt6x	10%	0.401 ± 0.008	–	3.155
smt12x	12.5%	0.596 ± 0.031	–	3.321
smt25x	16.5%	1.063 ± 0.079	–	3.790
smt50x	22%	2.319 ± 0.230	–	5.030
smt100x	24%	5.831 ± 0.776	–	8.550
interp	100%	0.141 ± 0.001	1.156	4.003
native	100%	0.139 ± 0.001	0.212	3.071

Tareas

- **Proyectos.** Entramos en modo consulta / a pedido.
- No hay mucha práctica de lo de hoy.
 - Pueden completar el archivo y/o hacer los archivos del curso de la ECI
- Leer el capítulo 29 (tácticas, es corto)
 - Más material de F* y tácticas: [Formal Verification with F* and Meta-F* \(fstar-lang.org\)](http://fstar-lang.org)
 - Si les interesa lógica de separación (verificación de programas con estado, punteros, etc), ver “Parte VII” del tutorial (Pulse).