Ejercicio Semanal 1

Oscar Andrés Rosas Hernández *Facultad de Ciencias, Universidad Nacional Autonoma de Mexico, CDMX*

I. On the unusual effectiveness of logic in computer science: Type Theory

Definitivamente la sección que más me interesa está relacionada con la teoría de tipos y su relación con la lógica computacional.

Como gran amante de lenguajes como c++, rust, haskell y typescript conozco de la gran variedad de sistemas de tipados que existen.

Me gustaría tener una gran texto para escribir para lograr el objetivo de llegar a una camarilla, pero me temo que este no será el caso, es interesante hablar de lla lógica y cómo esta influye en el sistema de tipos.

Por ejemplo en c++ podemos usar la información que nos da el sistema de tipos con ayuda del compilar para demostrar que hay ciertos estados que no son posible o estados a los que siempre se llega, con esto se puede lograr optimizaciones muy importantes, como por ejemplo, se pueden eliminar grandes fragmentos de codigo, pues mediante la negocia se demostró que dicha sección nunca será accesada, igualmente se puede hacer lo inverso, se puede demostrar que cierto procedimiento siempre tendrá un resultado, por lo tanto se puede realizar dicho procedimiento en tiempo de compilación y no de ejecución, obtienen un archivo binario mucho más eficiente.

En Rust por ejemplo podemos usar ciertas reglas que le permiten al compilador saber en todo momento quién es el dueño de cierto recurso del sistema, gracias a este sistema el compilador no nos dejará llegar a ambigüedades.

En Typescript por ejemplo, al ser un lenguaje que nace de javascript tenemos muchos problemas heredados, como que el sistema de tipos que construye por encima no tiene "soundness", eso es muy interesante de investigar y de entender las implicaciones que dicha afirmación tiene y como impactan en la seguridad que tenemos con nuestro codigo.

De TS también podemos empezar a hablar de tipos de datos algebraicos y como estos se pueden ver coom proposiciones, buscar un equivalente a la conjunción y a la disfunción, y ver como las herramientas de lógica se trasladan a este nuevo terreno

Dejamos aqui varios links que considero interesantes:

- https://codewords.recurse.com/issues/one/type-systems-and-logic
- https://youtu.be/gwlyrj1JtrE

II. Formalización

Si el programa es eficiente, se ejecuta rapidamente. El programa es eficiente, o tiene un error. Sin embargo, el programa no se ejecuta rapidamente. Por lo tanto tiene un error.

- Sea p = el programa es eficiente
- Sea q = el programa se ejecuta rapidamente
- Sea r = el programa tiene un error

Entonces esto se puede ver mejor como:

$((p \therefore q) \land (p \lor r) \land (\neg q)) \therefore r.$

Podemos hacer la tabla de verdad y ver que nunca llegamos a un caso en la tengamos que $(p \therefore q) \land (p \lor r) \land (\neg q)$ sea verdadera y r sea falsa.

Por lo tanto el argumento es correcto.