

## Anexo: Regex

### **Expresión completa:**

```
(?P<S0>T0)(?P<PT0>(?:\d+)*?)(?: (?P<S1>T1)(?P<PT1>(?:\d+)*?)(?P<S3>T3)|(?P<S2>T2)
(?P<PT2>(?:\d+)*?)(?P<S4>T4))(?P<PT3T4>(?:\d+)*?)(?: (?P<S5>T5)(?P<PT5>(?:\d+)*
?)(?P<S7>T7)|(?P<S6>T6)(?P<PT6>(?:\d+)*?)(?P<S8>T8))(?P<PT7T8>(?:\d+)*?)(?: (?P<S
10>T10)(?P<PT10>(?:\d+)*?)(?P<S12>T12)|(?P<S9>T9)(?P<PT9>(?:\d+)*?)(?P<S11>T11
))(?P<PT11T12>(?:\d+)*?)(?P<S13>T13)(?P<PT13>(?:\d+)*?)(?P<S14>T14)(?P<PT14>(?:
\d+)*?)
```

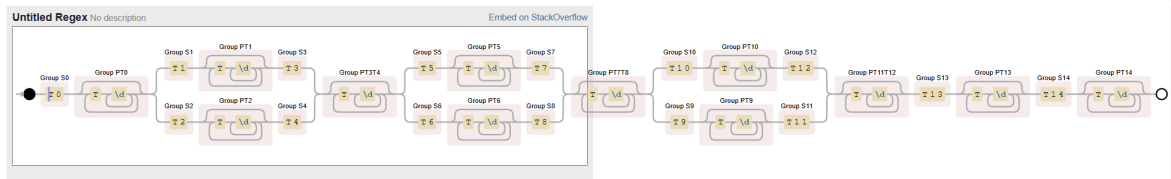


Fig. Vista en Debuggex.

### **(?P<S0>T0):**

- ?P<S0>: Esto le da un nombre al grupo de captura. En este caso, el grupo se llama S0.
- T0: Busca exactamente la cadena T0.
- Resumen: Busca T0 y lo captura en un grupo llamado S0.

### **(?P<PT0>(?:\d+)\*?):**

- ?P<PT0>: Este grupo se llama PT0.
- (?: ... ): Esto es un grupo no capturador. Significa que el contenido dentro de este grupo no se guardará como un grupo separado, pero se usa para agrupar patrones.
- \d+: Busca una T seguida de uno o más dígitos (\d+). Por ejemplo, T1, T2, T12, etc.
- \*?: Esto significa "cero o más veces, pero de manera no codiciosa. Es decir, intentará coincidir con la menor cantidad de texto posible.
- Resumen: Busca cualquier secuencia de transiciones (como T1, T2, etc.) después de T0, pero no las captura en un grupo separado. Las transiciones se guardan en el grupo PT0.

### **(?: (?P<S1>T1)(?P<PT1>(?:\d+)\*?)(?P<S3>T3)|(?P<S2>T2)(?P<PT2>(?:\d+)\*?)(?P<S4>T4)):**

- (?: ... ): Otro grupo no capturador.
- |: Es una OR. La regex intentará coincidir con lo que está a la izquierda del | o con lo que está a la derecha.
- (?P<S1>T1): Busca T1 y lo captura en el grupo S1.
- (?P<PT1>(?:\d+)\*?): Busca transiciones opcionales después de T1 y las captura en el grupo PT1.
- (?P<S3>T3): Busca T3 y lo captura en el grupo S3.
- (?P<S2>T2): Busca T2 y lo captura en el grupo S2.

- (?P<PT2>(?:T\d+)\*?): Busca transiciones opcionales después de T2 y las captura en el grupo PT2.
- (?P<S4>T4): Busca T4 y lo captura en el grupo S4.
- Resumen: Busca una secuencia que comience con T1 y termine con T3, o que comience con T2 y termine con T4. Además, captura cualquier transición intermedia (con los grupos PT1 y PT2).

**(?P<PT3T4>(?:T\d+)\*?):**

- ?P<PT3T4>: Este grupo se llama PT3T4.
- (?:T\d+)\*?: Busca cualquier secuencia de transiciones (como T5, T6, etc.) después de T3 o T4.
- Resumen: Captura transiciones opcionales después del bloque T1-T3 o T2-T4.

**(?:(?P<S5>T5)(?P<PT5>(?:T\d+)\*?)(?P<S7>T7))|(?P<S6>T6)(?P<PT6>(?:T\d+)\*?)(?P<S8>T8)):**

- Similar al bloque anterior, pero busca secuencias que arranquen con T5 y terminen con T7, o que comiencen con T6 y terminen con T8.
- Resumen: Captura transiciones entre T5-T7 o T6-T8.

**?P<PT7T8>(?:T\d+)\*?:**

- Captura transiciones opcionales después del bloque T5-T7 o T6-T8.

**(?:(?P<S10>T10)(?P<PT10>(?:T\d+)\*?)(?P<S12>T12))|(?P<S9>T9)(?P<PT9>(?:T\d+)\*?)(?P<S11>T11)):**

- Similar a los bloques anteriores, pero busca secuencias que arrancan con T10 y terminan con T12, o que comiencen con T9 y terminen con T11.
- Resumen: Captura transiciones entre T10-T12 o T9-T11.

**(?P<PT11T12>(?:T\d+)\*?):**

- Captura transiciones opcionales después del bloque T10-T12 o T9-T11.

**(?P<S13>T13)(?P<PT13>(?:T\d+)\*?)(?P<S14>T14)(?P<PT14>(?:T\d+)\*?):**

- (?P<S13>T13): Busca T13 y lo captura en el grupo S13.
- (?P<PT13>(?:T\d+)\*?): Captura transiciones opcionales después de T13.
- (?P<S14>T14): Busca T14 y lo captura en el grupo S14.
- (?P<PT14>(?:T\d+)\*?): Captura transiciones opcionales después de T14.
- Resumen: Captura las transiciones finales, incluyendo T13 y T14, junto con cualquier transición intermedia.

**Resumen de grupos de captura:**

- Grupo S0: Coincide con T0.
- Grupo PT0: Transiciones opcionales después de T0.
- Grupo S1 y S3: Coincide con T1 y T3, respectivamente.
- Grupo PT1: Transiciones opcionales entre T1 y T3.
- Grupo S2 y S4: Coincide con T2 y T4, respectivamente.
- Grupo PT2: Transiciones opcionales entre T2 y T4.
- Grupo PT3T4: Transiciones opcionales después del bloque T1-T3 o T2-T4.
- Grupo S5 y S7: Coincide con T5 y T7, respectivamente.

- Grupo PT5: Transiciones opcionales entre T5 y T7.
- Grupo S6 y S8: Coincide con T6 y T8, respectivamente.
- Grupo PT6: Transiciones opcionales entre T6 y T8.
- Grupo PT7T8: Transiciones opcionales después del bloque T5-T7 o T6-T8.
- Grupo S10 y S12: Coincide con T10 y T12, respectivamente.
- Grupo PT10: Transiciones opcionales entre T10 y T12.
- Grupo S9 y S11: Coincide con T9 y T11, respectivamente.
- Grupo PT9: Transiciones opcionales entre T9 y T11.
- Grupo PT11T12: Transiciones opcionales después del bloque T9-T11 o T10-T12.
- Grupo S13 y S14: Coincide con T13 y T14, respectivamente.
- Grupo PT13 y PT14: Transiciones opcionales después de T13 y T14.

### Script en Python

El script analiza un .txt que tiene las transiciones en una ejecución completa de nuestra red con 200 invariantes completados, estas transiciones están agrupadas en el archivo en el orden que se van ejecutando, es decir no siguen el orden del invariante.

El programa ha tenido cambios en cuanto la versión anterior, ya que antes solamente detectaba si para el total de transiciones disparadas se cumplía exactamente la regex de tal manera que no sobraban transiciones. Ahora, debía detectar cuáles y cuántos invariantes existían en la secuencia de disparo

Funciones principales:

- procesar\_archivo: Lee el archivo línea por línea usando la función re.subn() para buscar las transiciones según la regex. Si tiene conexión con una secuencia válida la guarda en transiciones.
- contar\_transiciones: Procesa el resultado de la regex y aumenta el contador del invariante.

El main recibe el archivo de texto que se quiere analizar

```
# Ejecutar el procesamiento del archivo
if __name__ == "__main__":
    nombre_archivo = "log_regex5.txt" # Archivo de registro
    with open(nombre_archivo, "r") as archivo_log:
        procesar_archivo(archivo_log)
```

y la salidas pueden ser

```
if resultado[0]:
    print("ERROR EN LA VALIDACIÓN DE INVARIANTES PARA: ", archivo.name)
else:
    print("LAS TRANSICIONES DE LOS INVARIANTES CORRECTAS PARA: ", archivo.name)
```

En caso de tener las transiciones correctas, expresa el contador de invariantes.