Cose da fare:

* Abbiamo inserito troppi TOKEN, alcuni non vanno bene. A parte quelli che servono per iniziare le regole sintattiche (come per esempio FLAG, WIRE, SYMATTR) gli altri devono esere declassati ad ID. Il controllo viene poi eseguito semanticamente: si prende l’ID e si verifica che faccia parte di una lista predefinita. Questo perché così facendo lo scanner diventa più leggero ed è più semplice fare modifiche a livello semantico piuttosto che mettere mano allo scanner.
* Dove può essere accettato un ID bisogna mettere in OR una *reservedWordRule*, dove si raggruppano le parole chiave (poiché la parola chiave può essere inserita come id); anche ASSIGN(=) può essere utilizzato negli ID.
* Inserire le lettere greche (almeno). Vanno inserite come un nuovo fragment e poi messe insieme alle altre lettere.
* Inserire tutti i caratteri speciali. Vanno messi uno a uno in un fragment SPECIAL\_LETTERS: alcuni caratteri da errore nel name (es. parentesi ma invece nel value si può. Direi di non farci problemi per ora)
  + Per fare riconoscere ID separati da spazio posti alla fine della regola mettere ID\*
  + Inserire EOF per risolvere il problema che non segnalava l’errore

Per l’analisi semantica:

* Se ci fosse un lato di un componente che non è collegato a nulla potremmo dare un warning

Per la traduzione:

* Forniamo in uscita anche il file formattato in modo corretto (qualora non lo fosse). Questo dovrebbe essere abbastanza semplice. Prendiamo un buffer e ogni volta che leggo un token ci aggiungo uno spazio. Nel momento in cui identifichiamo una regola inseriamo un a capo e poi scriviamo il tutto in un file.