

Gesucht: LL(1)-Parsetabelle

Kopfspalte: alle Nichtterminale

Kopfzeile: alle Terminale und \$

Alle anderen Zellen zeigen auf eine Regelnummer, wenn es für das Paar aus Nichtterminal und Terminal eine Regel gibt

Für mehr Vereinheitlichung in v3.5 noch Nichtterminale umbenannt, damit alle auf Englisch sind

Export aus FLACI:

$G = (N, T, P, s)$

$N = \{\text{ConditionsFollowing, DefinitionTrailing, ArgumentsFollowing, DefinitionLeading, AbleitungTrailing, LiteralsFollowing, AbleitungLeading, DefinitionMiddle, BerechneTrailing, BerechneLeading, AbleitungMiddle, SingleCondition, WendeTrailing, WendeLeading, PlotTrailing, PlotLeading, MinTrailing, MaxTrailing, MinLeading, MaxLeading, PlotMiddle, Conditions, Arguments, Literals, Start}\}$

$T = \{\text{befehlStokenberechne, relationalOperator, befehlStokenwende, befehlStokenplot, befehlStokenmin, befehlStokenmax, b_ableitung, temptoken, unknown_Id, function, literal, :=, (,), |, ,, ;}\}$

$P = \{$

Start $\rightarrow \epsilon$

| PlotLeading

| MinLeading

| MaxLeading

| WendeLeading

| AbleitungLeading

| BerechneLeading

| DefinitionLeading

PlotLeading \rightarrow befehlStokenplot (PlotMiddle

PlotMiddle \rightarrow temptoken PlotTrailing | function PlotTrailing

PlotTrailing \rightarrow) | Conditions)

MinLeading \rightarrow befehlStokenmin (MinTrailing

MinTrailing \rightarrow temptoken) | function)

MaxLeading \rightarrow befehlStokenmax (MaxTrailing

MaxTrailing \rightarrow temptoken) | function)

WendeLeading \rightarrow befehlStokenwende (WendeTrailing

WendeTrailing \rightarrow temptoken) | function)

AbleitungLeading \rightarrow b_ableitung (AbleitungMiddle

AbleitungMiddle \rightarrow temptoken | unknown_Id AbleitungTrailing | function | unknown_Id AbleitungTrailing

AbleitungTrailing \rightarrow | unknown_Id) |)

DefinitionLeading \rightarrow unknown_Id DefinitionMiddle

DefinitionMiddle \rightarrow := temptoken DefinitionTrailing | (Arguments) := temptoken DefinitionTrailing

Arguments \rightarrow unknown_Id ArgumentsFollowing

ArgumentsFollowing \rightarrow , unknown_Id ArgumentsFollowing | ϵ

DefinitionTrailing \rightarrow | Conditions | ϵ

Conditions \rightarrow SingleCondition ConditionsFollowing

ConditionsFollowing \rightarrow ; SingleCondition ConditionsFollowing | ϵ

SingleCondition \rightarrow unknown_Id relationalOperator literal

BerechneLeading \rightarrow befehlStokenberechne (BerechneTrailing

BerechneTrailing \rightarrow temptoken) | function (Literals))

Literals \rightarrow literal LiteralsFollowing

LiteralsFollowing \rightarrow , literal LiteralsFollowing | ϵ

}

$s = \text{Start}$

Da eine Regelnummer in der Tabelle eindeutig sein muss, habe ich die Regeln nochmal umgeschrieben, sodass jede Oder-Bedingung eine neue Nummer hat → erspart den Aufwand, während der Laufzeit für jede Regel nochmal prüfen zu müssen, welche Regel nun eigentlich genau eingesetzt werden muss, dabei noch Fehler entfernt (fehlendes Komma in Arguments[]; Conditions in Definition optional gemacht; weiteres, an das ich mich nicht erinnere) → v4

-> in diesem Zuge hätten die Substituierungen in Start und Instruction zwar entfernt werden können, zur besseren Übersicht aber stehen gelassen, da die Regeln in der Tabelle dann (?) ~~entweder übersprungen werden~~ (nicht gemacht, damit die implementierten Regeln zur Grammatik passen) oder einfach auf den substituierten Folgezustand verweisen, was zwar wenige Schritte hinzufügt, da die Parserlaufzeit aber nur einmal ausgeführt wird, ist das nicht schlimm

~~→ FLACI setzt diese Änderungen wieder zurück, wenn zwischen der grafischen und der Textansicht gewechselt wird -> doch so gelassen, doch kein v4~~

~~- Das oben beschriebene Prüfen nach der Unter Oder-Regel kann auf einfache Art und Weise mit einem Dict erledigt werden, welches die möglichen Termine als Schlüssel besitzt und dann auf die Unterregel verweist~~

Richtige Formatierung der Regeln in v4.txt gespeichert

In v5 noch den Berechne-Befehl hinzugefügt

Somit Parsingtable aus v5 erstellt, unter Zuhilfenahme von

<https://www.geeksforgeeks.org/compiler-design/construction-of-ll1-parsing-table/>

Bedingungen per Design und Redesigns erfüllt

Schritt 2: FIRST und bei Bedarf auch die FOLLOW-Mengen berechnet FLACI mit der Option >>>Prüfen/LL(1) Grammatik?<<<

Schritt 3: Tabelle mit den Terminalen und nichtterminalen: siehe Excel: Parsetabelle.xlsx

- Für die Terminale die TokenTypen eingesetzt