

Odstranjivanje beskorisnih znakova

Odstranjivanje mrtvih znakova

1. Tražimo žive znakove (prema algoritmu).
2. Svi koji nisu živi su mrtvi znakovi, brišemo produkcije s njima.

Odstranjivanje nedohvatljivih znakova

1. Krenemo od prve produkcije, gledamo samo nezavršne znakove s desne strane produkcije, popisujemo dohvatljive.
2. Svi koji nisu dohvatljivi su nedohvatljivi, produkcije s njima brišemo.

Odstranjivanje jediničnih i ϵ produkcija

Odstranjivanje jediničnih produkcija

1. Jedinične su sve koje imaju samo jedan nezavršni znak s desne strane.
2. Zamijenimo produkcije prema algoritmu.
3. Provjerimo za nedohvatljive znakove.

Odstranjivanje epsilon produkcija

1. Brišemo epsilon produkciju znaka npr. A.
2. Na svako pojavljivanje znaka A s desne strane u drugim produkcijama, napravimo kopiju te produkcije, ali bez znaka A, u svim mogućim kombinacijama (prema algoritmu).

Chomskijev oblik

Chomskijev oblik je $A \rightarrow BC$ ili $A \rightarrow a$. (produkcije prema dva nezavršna ili jednim završnim znakom)

1. Odstranimo beskorisne znakove, 1-prod. i ϵ -prod.
2. Provjerimo koje su produkcije već u Chomskijevom obliku i njih ne diramo.
3. Svaki završni znak u produkcijama zamijenimo s nekim nezavršnim (prim. 0 u E, 1 u F).
4. Razbijemo velike produkcije (prema više od 2 nezavršna znaka) na više produkcija, prema algoritmu.

Utvrđivanje (ne)jednoznačnosti gramatike

Gramatika je jednoznačna ako i samo ako postoji najviše jedan način zamjene krajnjih lijevih ili samo jedan način zamjene krajnjih desnih znakova za konstruiranje istog niza.

1. Zadanom gramatikom konstruiramo jedan niz zamjenom krajnje lijevog znaka i dobijemo niz.
2. Jednako tako, ali zamjenom krajnje desnog znaka, konstruiramo isti taj niz.
3. Ako smo uspjeli konstruirati taj niz na dva načina, gramatika je nejednoznačna.

Konstruiranje gramatike za prihvatanje nizova

Zadan je oblik niza, potrebno je napraviti produkcije. Moramo paziti na količinu i poziciju znakova u nizu, te da produkcija jednom mora i završiti (dodavanje epsilon-produkcije bi moralo biti).

Primjer: tutorial za 2.MI, zadatak 21 by maXimus

Iz PA prihv. stanjem u PA praznim stogom

Definiramo izgled automata (uređenu sedmorku za PA M_2).

1. Napravimo početni prijelaz.
2. Preuzmu se svi prijelazi (samo δ pretvorimo u δ').
3. Dodaju se za sva stanja stoga ϵ prijelazi.
4. Za sve ϵ prijelaze dodaju se ϵ prijelazi koji prazne stog.

Iz PA praznim stogom u PA prihv. stanjem

Definiramo izgled automata (uređenu sedmorku za PA M_2).

5. Napravimo početni prijelaz.
6. Preuzmu se svi prijelazi (samo δ pretvorimo u δ').
7. Dodaju se ϵ prijelazi za prihvatljivo stanje q_F .

Konstruiranje gramatike koja generira nizove koje prihvaća PA

1. Gramatiku počinjemo graditi s produkcijom S. Iz početnog stanja automata moguće je otići u N stanja gdje je N broj konačnih skupova stanja, zato trebamo N komada S produkcija.
2. Gradimo produkcije iz navedenih prijelaza automata. Početno stanje prve buduće produkcije mora uvijek biti isto kao početno stanje produkcije koju opisujemo, dok na svakoj sljedećoj produkciji završno stanje jedne mora biti jednako početnom stanju iduće.
3. Kad napravimo neku produkciju, dogodi se da dobijemo nove produkcije. Njih isto treba riješiti, pomoću nekog drugog prijelaza PA. Po prijelazima idemo uvijek po redu.
4. Izbacujemo mrtve produkcije – one za koje se prijelaz ne može dogoditi. Sve produkcije koje mogu generirati mrtve produkcije izbacujemo.
5. Preimenujemo produkcije, uvrstimo imena u popis i dobijemo traženu gramatiku.

Primjer: tutorial za 2.MI, zadatak 26 by P(M)ervan

LR parser

/