

Atribuutgrammatikad

Vahur Vaiksaar

Ajalugu

- D. E. Knuth: "Semantics of Context-Free Languages", 1968
- Eesmärk kirjeldada programmeerimiskeelte staatilist semantikat

Atribuutgrammatika definitsioon

- Atribuutgrammatika (AG) aluseks on kontekstivaba grammatika (KVG), mis kirjeldab keele süntaksit.
- KVG on nelik $G = (\Sigma, N, P, S)$, kus
 - Σ - lõplik terminaalide hulk
 - N - lõplik mitteterminaalide hulk
 - P - lõplik produktsioonide hulk
 - S - algsümbol
 - Produktsioonid on kujul

$$X \rightarrow \alpha, \quad X \in N \text{ \& } \alpha \in (\Sigma \cup N)^*$$

AG definitsioon

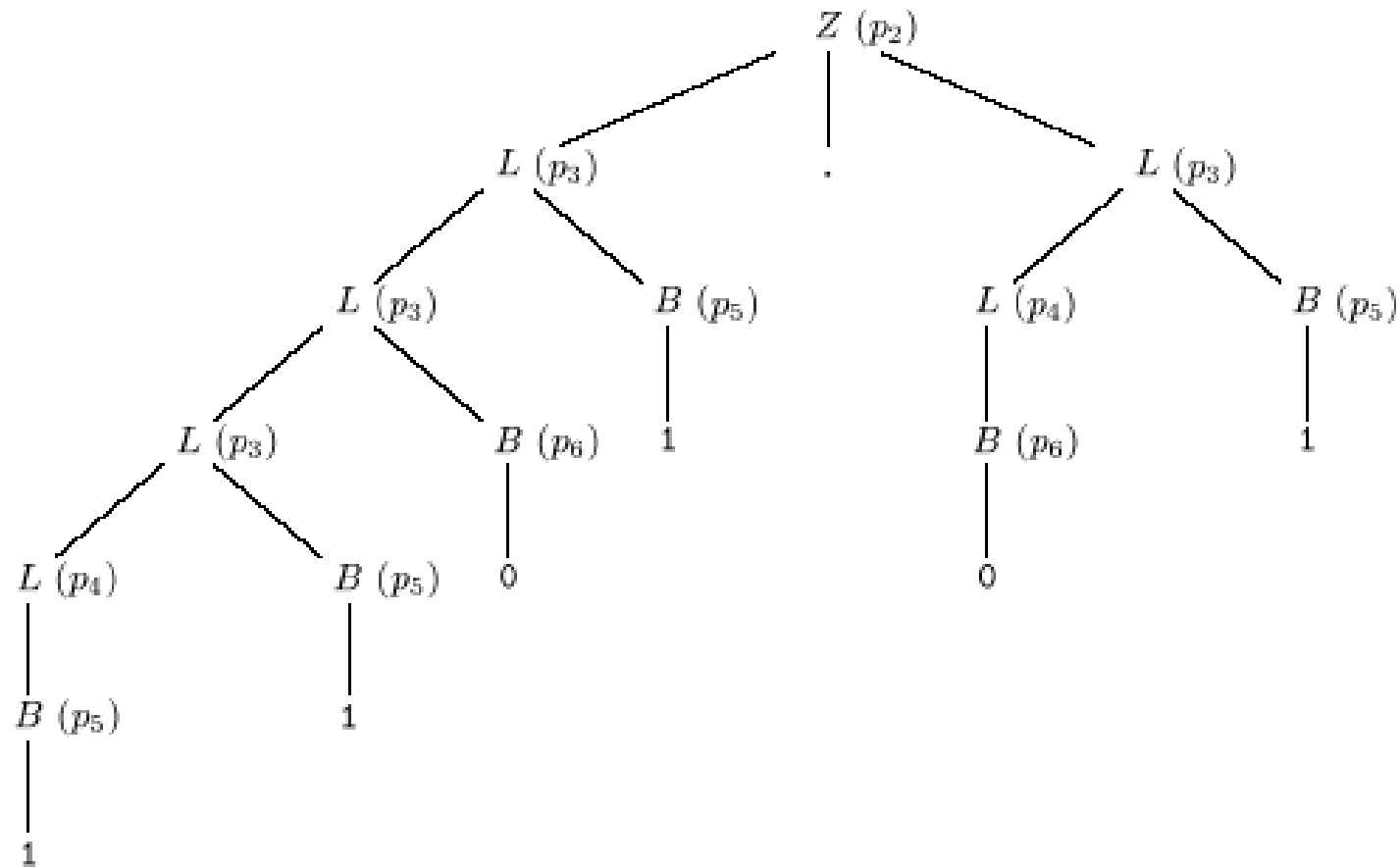
- AG laiendab KVG-d, seostades iga sümboliga $X \in \Sigma \cup N$ atribuutide hulga $A(X)$
- Iga atribuut esitab sümboli mingit spetsiifilist omadust
- Iga KVG produktsiooniga $X \rightarrow \alpha \in P$ on seotud semantiliste reeglite (võrrandite) hulk, mille põhjal arvutatakse atribuutide väärtused.

Näide: kahendarvud

Olgu meil KVG $G = (\Sigma, N, P, Z)$, kus

- $N = \{Z, L, B\}$
- $\Sigma = \{1, 0, .\}$
- P koosneb produktsioonidest
 - $p_1 : Z \rightarrow L$
 - $p_2 : Z \rightarrow L.L$
 - $p_3 : L \rightarrow LB$
 - $p_4 : L \rightarrow B$
 - $p_5 : B \rightarrow 1$
 - $p_6 : B \rightarrow 0$

Näide: kahendarvud



Näide: kahendarvud

- Eesmärk: ehitada grammatika G peale AG, mis arvutaks kahendsüsteemis kirja pandud arvu väärtuse kümnendsüsteemis.
- Defineerime mitteterminaalide Z , L ja B jaoks atribuudi v (*value*)
- Atribuudi v arvutamiseks tuleb produktsioonidega siduda semantilised reeglid

Näide: kahendarvud

● $p_1 : Z \rightarrow L$

$$Z.v = L.v$$

Näide: kahendarvud

● $p_1 : Z \rightarrow L$

$$Z.v = L.v$$

● $p_2 : Z \rightarrow L_1.L_2$

$$Z.v = L_1.v + L_2.v$$

Näide: kahendarvud

● $p_1 : Z \rightarrow L$

$$Z.v = L.v$$

● $p_2 : Z \rightarrow L_1.L_2$

$$Z.v = L_1.v + L_2.v$$

● $p_3 : L_0 \rightarrow L_1B$

$$L_0.v = L_1.v + B.v$$

Näide: kahendarvud

● $p_1 : Z \rightarrow L$

$$Z.v = L.v$$

● $p_2 : Z \rightarrow L_1.L_2$

$$Z.v = L_1.v + L_2.v$$

● $p_3 : L_0 \rightarrow L_1B$

$$L_0.v = L_1.v + B.v$$

● $p_4 : L \rightarrow B$

$$L.v = B.v$$

Näide: kahendarvud

● $p_1 : Z \rightarrow L$

$$Z.v = L.v$$

● $p_2 : Z \rightarrow L_1.L_2$

$$Z.v = L_1.v + L_2.v$$

● $p_3 : L_0 \rightarrow L_1B$

$$L_0.v = L_1.v + B.v$$

● $p_4 : L \rightarrow B$

$$L.v = B.v$$

● $p_6 : B \rightarrow 0$

$$B.v = 0$$

Näide: kahendarvud

● $p_1 : Z \rightarrow L$

$$Z.v = L.v$$

● $p_2 : Z \rightarrow L_1.L_2$

$$Z.v = L_1.v + L_2.v$$

● $p_3 : L_0 \rightarrow L_1B$

$$L_0.v = L_1.v + B.v$$

● $p_4 : L \rightarrow B$

$$L.v = B.v$$

● $p_6 : B \rightarrow 0$

$$B.v = 0$$

● $p_5 : B \rightarrow 1$

$$B.v = ?$$

Näide: kahendarvud

- Probleemi lahendamiseks mitteterminaalidele L ja B uus atribuut s (*scale*), siis $p_5 : B \rightarrow 1$ jaoks

$$B.v = 2^{B.s}$$

- Nüüd omakorda vaja semantilisi reegleid s arvutamiseks

Näide: kahendarvud

● $p_3 : L_0 \rightarrow L_1 B$

$$B.s = L_0.s$$

$$L_1.s = L_0.s + 1$$

Näide: kahendarvud

● $p_3 : L_0 \rightarrow L_1 B$

$$B.s = L_0.s$$

$$L_1.s = L_0.s + 1$$

● $p_4 : L \rightarrow B$

$$B.s = L.s$$

Näide: kahendarvud

● $p_3 : L_0 \rightarrow L_1 B$

$$B.s = L_0.s$$

$$L_1.s = L_0.s + 1$$

● $p_4 : L \rightarrow B$

$$B.s = L.s$$

● $p_1 : Z \rightarrow L$

$$L.s = 0$$

Näide: kahendarvud

● $p_3 : L_0 \rightarrow L_1 B$

$$B.s = L_0.s$$

$$L_1.s = L_0.s + 1$$

● $p_4 : L \rightarrow B$

$$B.s = L.s$$

● $p_1 : Z \rightarrow L$

$$L.s = 0$$

● $p_2 : Z \rightarrow L_1.L_2$

$$L_1.s = 0$$

$$L_2.s = ?$$

Näide: kahendarvud

- Seega tuleb defineerida veel üks atribuut, l (*length*) ja seda läheb vaja ainult mitteterminaalile L
- Siis $p_2 : Z \rightarrow L_1.L_2$ jaoks

$$L_2.s = -L_2.l$$

- Ja lõpuks, semantilised reeglid atribuudi l arvutamiseks

- $p_3 : L_0 \rightarrow L_1B$

$$L_0.l = L_1.l + 1$$

- $p_4 : L \rightarrow B$

$$L.l = 1$$

Näide: kahendarvud

$$A = \{v, s, l\}, A(Z) = \{v\}, A(L) = \{v, s, l\}, A(B) = \{v, s\}$$

$$\text{type}(v) = \text{real}, \text{type}(s) = \text{type}(l) = \text{integer}$$

$$p_1: Z ::= L$$

$$Z.v = L.v$$

$$L.s = 0$$

$$p_2: Z ::= L_1 . L_2$$

$$Z.v = L_1.v + L_2.v$$

$$L_1.s = 0$$

$$L_2.s = -L_2.l$$

$$p_3: L_0 ::= L_1 B$$

$$L_0.v = L_1.v + B.v$$

$$B.s = L_0.s$$

$$L_1.s = L_0.s + 1$$

$$L_0.l = L_1.l + 1$$

$$p_4: L ::= B$$

$$L.v = B.v$$

$$B.s = L.s$$

$$L.l = 1$$

$$p_5: B ::= 1$$

$$B.v = 2^{B.s}$$

$$p_6: B ::= 0$$

$$B.v = 0$$

Sünteesitud ja päritud atribuudid

- Sõltuvalt sellest, kus atribuudi väärtus arvutatakse, jagunevad atribuudid sünteesitud ja päritud atribuutideks
- Kui atribuut arvutatakse "ülemises" produktsioonis, siis on tegu sünteesitud atribuudiga, vastasel juhul päritud atribuudiga.
- Atribuut ei saa olla samaaegselt sünteesitud ja päritud
- Grammatika algsümbolil ei saa olla päritud atribuute

Semantilised reeglid

- Iga produktsiooniga $X \rightarrow \alpha \in P$ on seotud mingi hulk semantilisi reegleid kujul $b = f(c_1, c_2, \dots, c_k)$
 - Kui b on X sünteesitud atribuut, siis $c_1, c_2, \dots, c_k$ võivad olla produktsiooni parema poole sümboolite atribuudid
 - Kui b on produktsiooni parema poole mõne sümboli päritud atribuut, siis $c_1, c_2, \dots, c_k$ võivad olla produktsiooni mõlema poole sümboolite atribuudid

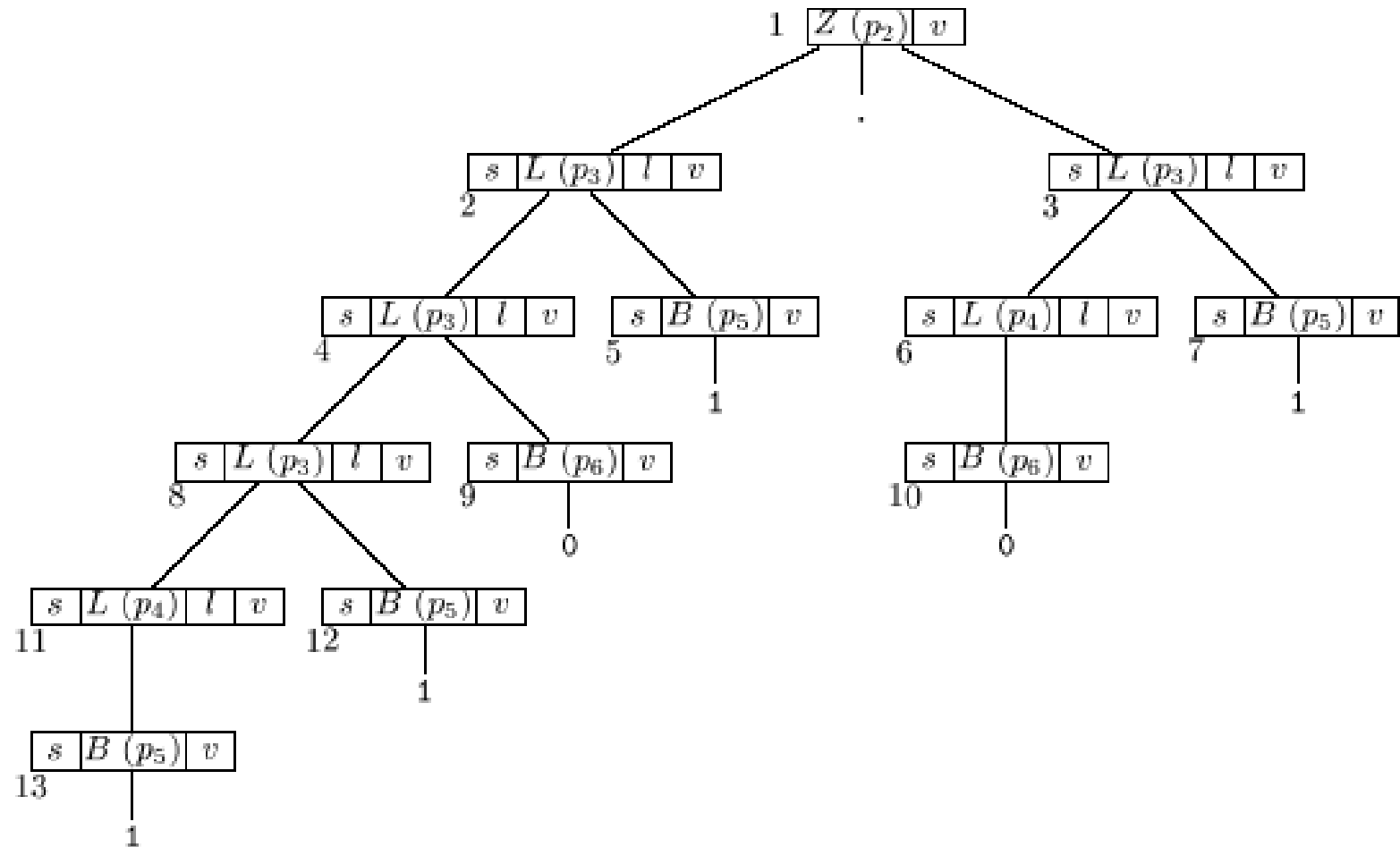
AG normaalkuju

- Produktsiooni väljundatribuudid on need, mis defineeritakse produktsiooni semantiliste reeglite poolt
- Produktsiooni sisendatribuudid on need, mille kaudu väljundatribuudid defineeritakse
- AG on (Bochmann'i) normaalkujul, kui iga produktsiooniga seotud semantilise reegli sisendatribuudid koosnevad produktsiooni vasaku poole päritud ja parema poole sünteesitud atribuutidest.
- Iga AG saab teisendada normaalkujule

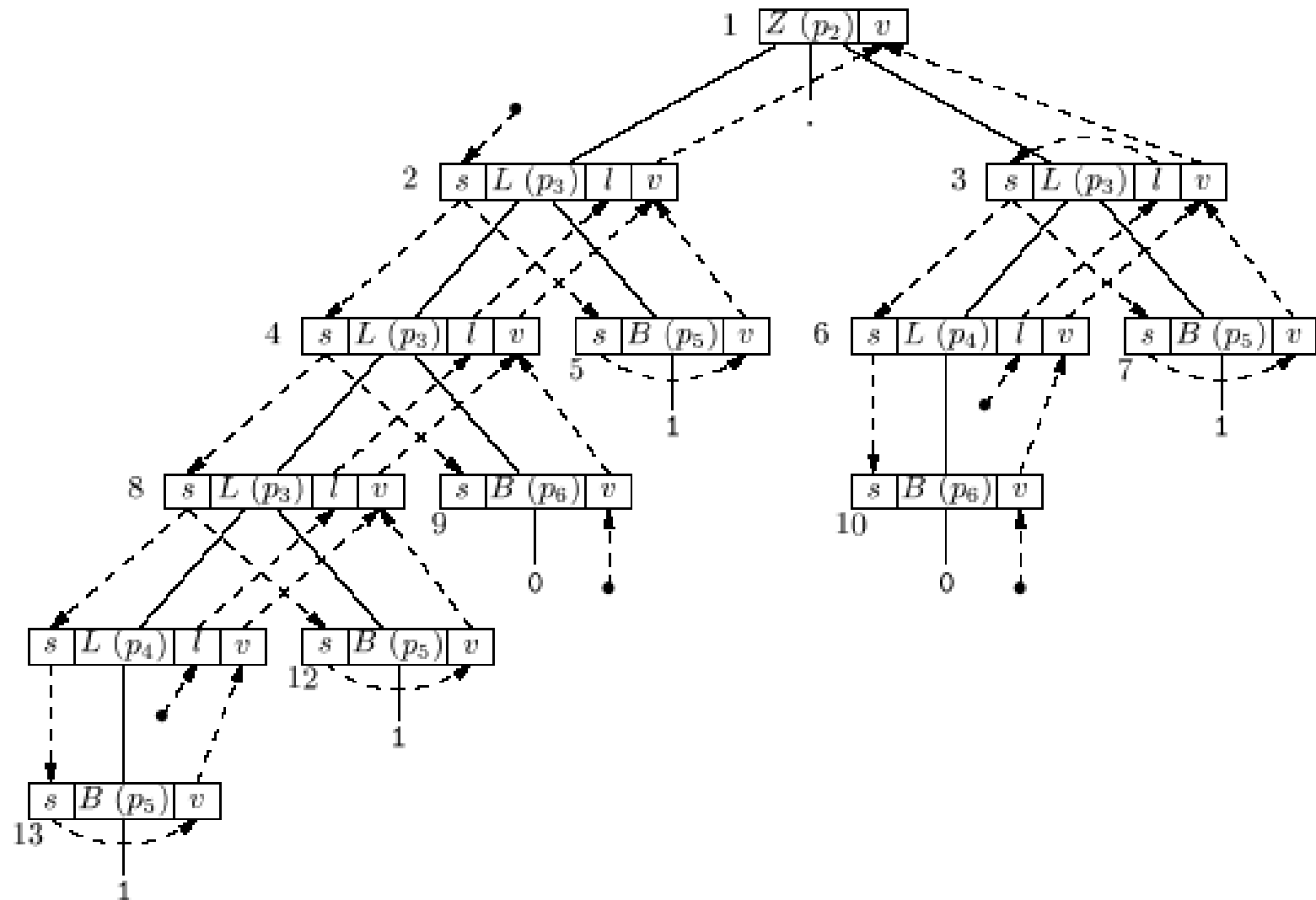
Semantikapuu ja sõltuvuste graaf

- Semantikapuu on süntaksipuu, mille iga tipu juures on märgitud ka atribuudid
- Semantikapuu T sõltuvuste graaf $D(T)$ on suunatud graaf (V, E) , kus
 - $V = \{T \text{ atribuutite isendid} \}$
 - $E = \{(a, b) \mid a, b \in V \ \& \ b \text{ sõltub } a\text{-st} \}$

Näite semantikapuu



Näite sõltuvuste graaf



Analüüs ja transleerimine

- Programmi staatilise semantika analüüs ja transleerimine toimub semantikapuu atribuute väärtustades
- Atribuutgrammatikate poolt genereeritud kompilaatorites salvestatakse veateated ja/või genereeritud kood semantikapuu juurtippu sünteesitud atribuudina.

Atribuutide väärtustamine

- Atribuutide väärtused leitakse atribuuti defineerivast semantilisest võrrandist
- Enne kui atribuuti saab väärtustada, peavad kõik teda defineeriva võrrandi paremas pooles leiduvad atribuudid olema eelnevalt väärtustatud
- AG atribuutide vahelised funktsionaalsed sõltuvused defineerivad semantikapuus leiduvate atribuutide isendite vahel osalise järjestuse

Atribuutide väärtustamine

Knuth pakkus välja lihtsa algoritmi:

- Sorteerida sõltuvuste graafi tipud topoloogiliselt
- Väärtustada atribuutide isendid topoloogilises järjekorras

Kahjuks...

- Töötab ainult siis, kui AG pole tsirkulaarne
- Vähene jõudlus

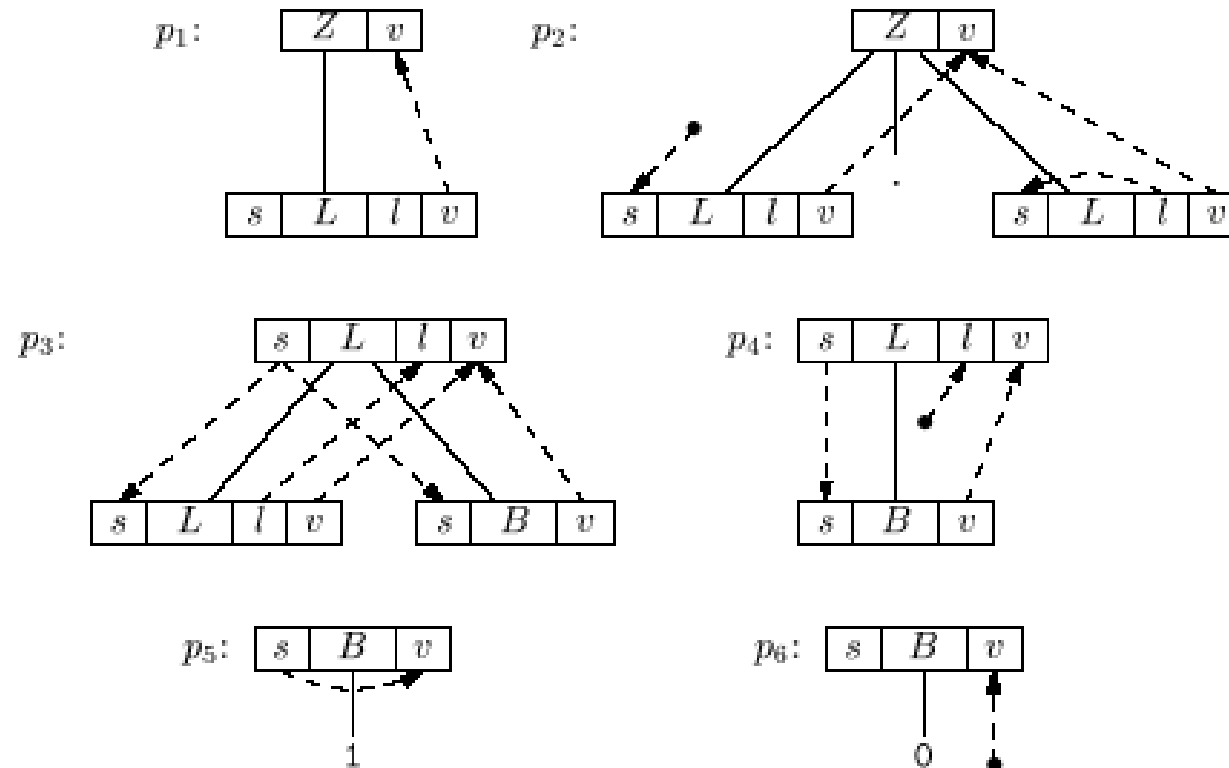
Ka efektiivsemaid algoritme on välja mõeldud

- Enamus tööst tehakse ära grammatikat analüüsides
- Töötavad ainult konkreetsetel tsirkulaarsete AG-de alamklassidel

Produktsioonide sõltuvuste graafid

- Produktsioonide semantilised reeglid on teada
- Atribuudid on teada
- Iga produktsioon $A \rightarrow X_1 \dots X_n$ on kirja pandav semantilise puu kujul
 - Puu juureks A koos atribuutidega
 - Lehtedeks $X_1, \dots, X_n$ koos atribuutidega
- Selliselt saadud graafe nimetatakse produktsioonide sõltuvuste graafideks

Produktsioonide sõltuvuste graafid



Produktsioonide sõltuvuste graafid

- Produktsioonide sõltuvuste graafidest saab koostada kõikvõimalikud sõltuvuste graafid
- See asjaolu võimaldab AG omadusi staatiliselt uurida
 - AG tsirkulaarsus

Mõned AG-de klassid

Atribuutgrammatika on

- S-attributed - kui ei sisalda päritud atribuute
- L-attributed - kui grammatika iga produktsiooni $A \rightarrow X_1 \dots X_n \in P$ korral iga päritud atribuut $X_j, 1 \leq j \leq n$ sõltub ainult
 - $X_1, \dots, X_{j-1}$ atribuutidest
 - A päritud atribuutidest
- tugevalt mittetsirkulaarne - iga mitteterminaali atribuudid saab igas puu tipus väärtustada samas järjekorras
- l-ordered - atribuutide väärtustamise järjekord staatiliselt määratav