Rekursiiviset tyypit – algoritmi

Panu Liukkonen

Helsinki 10. maaliskuuta 2003

Tyyppiteoria ja ohjelmointikielet – seminaari

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto

Rekursiivisten tyyppien alityypitys on tärkeä osa monien ohjelmointikielten toteutusta. Tämä Benjamin C. Piercen teoksen Types and Programming Languages ([Pie02]) luvun 21 loppuosaan pohjautuva teksti esittelee rekursiivisten tyyppien alityypitykseen käytännöllisen algoritmin.

2 Jäsenyyden tarkistus

Tarkastelemme seuraavaa ongelmaa: Annettuna generoiva funktio $F \in \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ ja alkio $x \in \mathcal{U}$. Onko tällöin $x \in \nu F$?

F voi generoida alkion $x \in \mathcal{U}$ monella tapaa, eli on mahdollisesti useita joukkoja $X \subseteq \mathcal{U}$, joilla $x \in F(X)$. Kutsutaan tälläistä joukkoa X alkion x generoivaksi joukoksi. F:n monotonisuuden vuoksi mikä tahansa x:n generoivan joukun ylijoukko on x:n generoiva joukko, joten on järkevää rajoittaa tarkastelu minimaalisiin generoiviin joukkoihin. Lisäksi oletamme generoivien funktioiden olevan kääntyviä.

Määritelmä 1 *Generoiva funktio F on kääntyvä, jos jokaisella $x \in \mathcal{U}$ joukko*

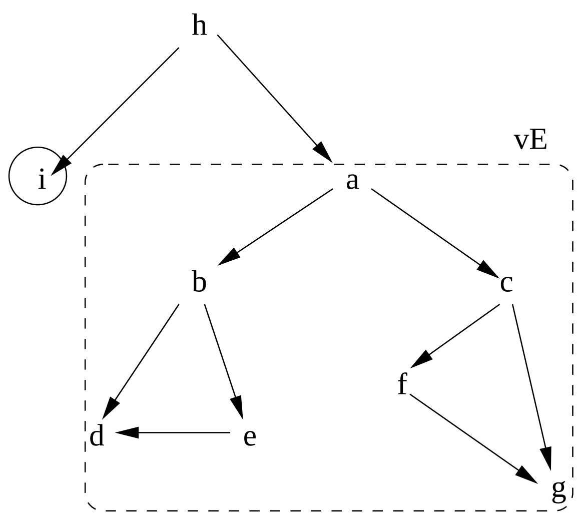
$$G_x = \{X \subseteq \mathcal{U} \mid x \in F(X)\}$$

on joko tyhjä tai sisältää pienimmän alkion. Kääntyvälle F määritellään tukifunktio $support_F \in \mathcal{U} \rightharpoonup \mathcal{P}(\mathcal{U})$ seuraavasti:

$$support_F(x) = \begin{cases} X & jos\ X \in G_x\ ja\ \forall X' \in G_x.\ X \subseteq X' \\ \uparrow & jos\ G_x = \emptyset \end{cases}$$

Funktio yleistyy joukoille seuraavasti:

$$support_F(X) = \begin{cases} \bigcup_{x \in X} support_F(x) & jos\ \forall x \in X.\ support_F(x) \downarrow \\ \uparrow & muuten \end{cases}$$



Kuva 1: Esimerkki tukifunktiosta

Kun F on asiayhteydestään selvä, jätetään alaindeksi pois funktion $support_F$ nimestä.

Määritelmä 2 *Alkio x on* ***tuettu*** *(eng. supported) jos $support_F(x)$ on määritelty, muuten x on* ***tukematon*** *(eng. unsupported).*

Määritelmästä 2 seuraa, ettei tukematon alkio voi kuulua joukkoon $F(X)$ millään X.

Kääntyvä generoiva funktio voidaan visualisoida tukigraafina. Kuva 1 määrittelee funktion E universumissa {a, b, c, d, e, f, g, h, i} näyttämällä kunkin alkion tukialkiot: alkion x $support_E(x)$ sisältää kaikki alkiot y, joihin on nuoli x:stä. Tukematon alkio i on merkitty ympyrällä.

Tarkastelemalla kuvan 1 esimerkkiä voimme päätellä alkion x kuuluvan suurimpaan kiintopisteeseen jos ja vain jos mikään tukematon alkio ei ole tavoitettavissa x:stä. Tästä saadaan algoritmiehdokas ongelmalle.

Määritelmä 3 *Olkoon F kääntyvä. Funktio gfp_F (tai vain gfp) määritellään seu-*

raavasti:

$$\begin{aligned} gfp(X) = {} & \textit{if } support(X) \uparrow\textit{, then false} \\ & \textit{else if } support(X) \subseteq X\textit{, then true} \\ & \textit{else } gfp(support(X) \cup X). \end{aligned}$$

Voidaan ajatella, että *gfp* laajentaa joukkoa X (käyttäen tukifunktiota) kunnes X:stä tulee konsistentti tai tukematon alkio löytyy. Alkiolle $x \in \mathcal{U}$ määritellään $gfp(x) = gfp(\{x\})$.

Seuraavaksi todistetaan *gfp*:n oikeellisuus. Ensin kuitenkin esitellään muutama tukifunktion ominaisuus ilman todistuksia.

Lemma 4 *$X \subseteq F(Y)$ joss $support(X) \subseteq Y$.*

Lemma 5 *Olkoon P F:n kiintopiste. Tällöin $X \subseteq P$ joss $support(X) \subseteq P$.*

Lause 6

1. *Jos $gfp(X) = true$, niin $X \subseteq \nu F$.*
2. *Jos $gfp(X) = false$, niin $X \not\subseteq \nu F$.*

Todistus. Kummankin lauseen todistus suoritetaan induktiolla algoritmin ajon rekursiivisen rakenteen suhteen.

1. $gfp(X)$ voi palauttaa arvon *true* kahdessa tapauksessa. Jos $gfp(X) = true$ koska $support(X) \subseteq X$, niin lemman 4 perusteella $X \subseteq F(X)$, eli X on F-konsistentti, joten $X \subseteq \nu F$ koinduktioperiaatteen mukaan. Jos $gfp(X) = true$, koska $gfp(support(X) \cup X) = true$, niin käyttämällä induktio-oletusta saadaan $support(X) \cup X \subseteq \nu F$.

2. Olkoon $gfp(X) = false$ koska $support(X)$ ei ole määritelty. Tällöin $X \not\subseteq \nu F$ lemman 5 perusteella. Olkoon $gfp(X) = false$ koska $gfp(support(X) \cup X) =$

false. Induktio-oletuksen nojalla $support(X) \cup X \not\subseteq \nu F$. Yhtäpitävästi $X \not\subseteq \nu F$ tai $support(X) \not\subseteq \nu F$, joten $X \not\subseteq \nu F$ (lemma 5).

Seuraavaksi määritellään riittävä pysähtymisehto *gfp*:lle supistamalla mahdollisten generoivien funktioiden joukkoa. Tarvitaan seuraavat määritelmät:

Määritelmä 7 *Alkion $x \in \mathcal{U}$ edeltäjien joukko on*

$$pred_F(x) = \begin{cases} \emptyset & jos\ support_F(x) \uparrow \\ support_F(x) & jos\ support_F(x) \downarrow \end{cases}$$

ja sen laajennos joukoille $X \subseteq \mathcal{U}$ on

$$pred_F(X) = \bigcup_{x \in X} pred_F(X).$$

Joukolta X tavoitettavissa olevat alkiot määritellään seuraavasti:

$$reachable_F(X) = \bigcup_{n \geq 0} pred_F^n(X).$$

Tämän laajennos yhdelle alkiolle $x \in \mathcal{U}$ on

$$reachable_F(x) = reachable_F(\{x\}).$$

Alkion $y \in \mathcal{U}$ sanotaan olevan F-saavutettavissa alkiolta x, jos $y \in reachable_F(x)$.

Määritelmä 8 *Generoivaa funktiota sanotaan äärellistilaiseksi, jos $reachable_F(x)$ on äärellinen jokaisella $x \in \mathcal{U}$.*

Osoittautuu että äärellistilaisilla generoivilla funktioilla *gfp* pysähtyy.

Lause 9 *Jos $reachable(X)$ on äärellinen, niin $gfp(X)$ on määritelty. Siis F:n ollessa äärellistilainen $gfp(X)$ pysähtyy kaikilla äärellisillä $X \subseteq \mathcal{U}$.*

Todistus. Jokaisella rekursiivisella kutsulla $\mathit{gfp}(Y)$ pätee $Y \subseteq \mathit{reachable}(X)$, jossa X on ensimmäisen gfp:n kutsun parametri. Lisäksi Y kasvaa jokaisella kutsulla. Koska $\mathit{reachable}(X)$ on äärellinen, niin arvo $m(Y) = |\mathit{reachable}(X)| - |Y|$ kuvaa sitä milloin gfp pysähtyy.

Esitellään vielä tehokkaampi versio algoritmista todistamatta sen oikeellisuutta.

Määritelmä 10 *Olkoon F kääntyvä funktio. Määritellään funktio gfp^t seuraavasti.*

$\mathit{gfp}^t(A, x) =$

if $x \in A$, *then* A

else if $\mathit{support}(x) \uparrow$, *then* $\uparrow$

else

let $x_1, \ldots, x_n = \mathit{support}(x)$ *in*

let $A_0 = A \cup \{x\}$ *in*

let $A_1 = \mathit{gfp}^t(A_0, x_1)$ *in*

$\ldots$

let $A_n = \mathit{gfp}^t(A_{n-1}, x_n)$ *in*

A_n.

Jos halutaan tarkistaa x:n kuuluvuus joukkoon νF, niin lasketaan $\mathit{gfp}^t(\emptyset, x)$: $x \in \nu F$ jos tulos on joukko. Jos taas tulos on määrittelemätön, niin $x \notin \nu F$.

3 Säännölliset puut

Geneerinen algoritmi alkion suurimpaan kiintopisteeseen kuulumisen tarkistamiseen on nyt olemassa. On myös näytetty, miten alityypitys määritellään äärettömille puille generoivan funktion S suurimpana kiintopisteenä. Nyt voitaisiin soveltaa algoritmia *gfp* funktiolle S. Algoritmi ei kuitenkaan pysähtyisi kaikilla syötteillä, koska parilta äärettömiä tyyppejä voidaan saavuttaa ääretön määrä tiloja. Seuraavaksi rajoitetaan äärettömät tyypit erääseen hyvin käyttäytyvään muotoon. Nämä säännölliset tyypit takaavat äärellisen määrän saavutettavia tiloja.

Määritelmä 11 *Puutyyppi S on puutyypin T* **alipuu** *jos S voidaan esittää muodossa $S = \lambda\sigma.T(\pi, \sigma)$ jollakin π. T:n kaikkien alipuiden joukko on $subtrees(T)$.*

Määritelmä 12 *Puutyyppi $T \in \mathcal{T}$ on* **säännöllinen**, *jos $subtrees(T)$ on äärellinen, eli T:llä on vain äärellinen määrä erilaisia alipuita. Säännöllisten puiden joukkoa merkitään symbolilla $\mathcal{T}_r$.*

Esimerkki 13

1. *Kaikki äärelliset puut ovat säännöllisiä, koska solmujen määrä asettaa rajan alipuiden määrälle.*

2. *Erilaisten alipuiden määrä voi olla paljon solmujen määrää pienempi. Esimerkiksi puulla $T = Top {\rightarrow} (Top {\times} Top)$ on viisi solmua mutta vain kolme erilaista alipuuta.*

3. *Ääretön puu voi olla säännöllinen. Esimerkiksi puulla*

$$T = Top \times (Top \times (Top \times \ldots))$$

on vain kaksi erilaista alipuuta (T ja Top).

Havainto 14 *S:n rajoittuma säännöllisiin puihin, jota merkitään S_r, on äärellistilainen.*

Tämä tarkoittaa sitä, että gfp_{S_r} pysähtyy kaikilla syötteillä. Käytännön toteutusta varten tarvitaan vielä kuitenkin äärellinen esitys säännöllisille puille. Eräs tälläinen esitys on μ-esitysmuoto.

4 μ-tyypit

Formalisoidaan säännollisten tyyppien μ-esitysmuoto.

Määritelmä 15 *Olkoon X mikä tahansa joukon $\{X_1, X_2, \ldots\}$ tyyppimuuttuja. Joukko $\mathcal{T}_m^{raw}$ on raakojen μ-tyyppien (eng. raw μ-types) joukko, joka määritellään seuraavasti:*

$$
\begin{aligned}
T ::= {} & X \\
& Top \\
& T \times T \\
& T \to T \\
& \mu X.T
\end{aligned}
$$

Rajoitetaan raakoja μ-tyyppejä hiukan, jotta vastaavuus säännöllisten tyyppien kanssa saavutetaan.

Määritelmä 16 *Sanomme raakaa μ-tyyppiä T vain μ-tyypiksi, jos jokaisessa muotoa $\mu X.\mu X_1 \ldots X_n.S$ olevassa T:n alilauseessa S ei ole X. Kaikkien μ-tyyppien joukko on $\mathcal{T}_m$.*

Seuraavana esiteltävän $treeof$-funktion avulla μ-tyypit voidaan ymmärtää äärellisenä esitysmuotona äärettömille säännöllisille puille.

Määritelmä 17 *Määritellään induktiivisesti funktio treeof, joka kuvaa suljettuja μ-tyyppejä puutyypeiksi:*

$$
\begin{array}{lcl}
\mathit{treeof(Top)(\bullet)} & = & \mathit{Top} \\
\mathit{treeof(T_1{\rightarrow}T_2)(\bullet)} & = & \rightarrow \\
\mathit{treeof(T_1{\rightarrow}T_2)(i,\pi)} & = & \mathit{treeof(T_i)(\pi)} \\
\mathit{treeof(T_1{\times}T_2)(\bullet)} & = & \times \\
\mathit{treeof(T_1{\times}T_2)(i,\pi)} & = & \mathit{treeof(T_i)(\pi)} \\
\mathit{treeof(\mu X.T)(\pi)} & = & \mathit{treeof(\{X{\mapsto}\mu X.T\}T)(\pi)}
\end{array}
$$

Kuvaus yleistyy pareille tavalliseen tapaan: $treeof(S,T) = (treeof(S), treeof(T))$. Määritellään alityypitys μ-tyypeille.

Määritelmä 18 *Kaksi μ-tyyppiä S ja T ovat alityyppisuhteessa jos $(S,T) \in \nu F$, jossa monotoninen funktio $S_m \in \mathcal{P}(\mathcal{T}_m{\times}\mathcal{T}_m) \rightarrow \mathcal{P}(\mathcal{T}_m{\times}\mathcal{T}_m)$ määritellään seuraavasti:*

$$
\begin{array}{rcl}
S_m(R) & = & \{(S, Top) \mid S \in \mathcal{T}_m\} \\
& \cup & \{(S_1 \times S_2, T_1 \times T_2 \mid (S_1,T_1),(S_2,T_2) \in R\} \\
& \cup & \{(S_1 \rightarrow S_2, T_1 \rightarrow T_2 \mid (T_1,S_1),(S_2,T_2) \in R\} \\
& \cup & \{(S, \mu X.T) \mid (S, \{X \mapsto \mu X.T\}T) \in R\} \\
& \cup & \{(\mu X.S, T) \mid (\{X \mapsto \mu X.S\}S, T) \in R, T \neq \mathit{Top},\ \mathit{ja}\ T \neq \mu Y.T_1\}.
\end{array}
$$

S_m:aa vastaava tukifunktio on:

$$
support_{S_m}(S,T) = \begin{cases}
\emptyset & \text{jos } T = Top \\
\{(S_1,T_1),(S_2,T_2))\} & \text{jos } S = S_1 \times S_2 \text{ ja } T = T_1 \times T_2 \\
\{(T_1,S_1),(S_2,T_2))\} & \text{jos } S = S_1 \rightarrow S_2 \text{ ja } T = T_1 \rightarrow T_2 \\
\{(S, \{X{\mapsto}\mu X.T_1\}T_1)\} & \text{jos } T = \mu X.T_1 \\
\{(\{X{\mapsto}\mu X.S_1\}S_1, T)\} & \text{jos } S = \mu X.S_1 \text{ ja } T \neq \mu X.T_1, T \neq Top \\
\uparrow & \text{muuten.}
\end{cases}
$$

Puutyyppejen ja μ-tyyppejen alityypitys vastaavat toisiaan, kuten seuraava lause toteaa.

Lause 19 *Olkoon* $(S,T) \in \mathcal{T}_m \times \mathcal{T}_m$*.* $(S,T) \in \nu S_m$ *joss* $treeof(S,T) \in \nu S$*.*

5 Konkreettinen algoritmi

gfp^t tukifunktiolla $support_{S_m}$ on seuraavanlainen algoritmi:

$subtype(A, S, T) =$

if $(S,T) \in A$, then

A

else let $A_0 = A \cup (S,T)$ in

if $T = Top$, then

A_0

else if $S = S_1 \times S_2$ and $T = T_1 \times T_2$, then

let $A_1 = subtype(A_0, S_1, T_1)$ in

$subtype(A_1, S_2, T_2)$

else if $S = S_1 \to S_2$ and $T = T_1 \to T_2$, then

let $A_1 = subtype(A_0, T_1, S_1)$ in

$subtype(A_1, S_2, T_2)$

else if $S = \mu X.S_1$, then

$subtype(A_0, \{X \mapsto \mu X.S_1\}S_1, T)$

else if $T = \mu X.T_1$, then

$subtype(A_0, S, \{X \mapsto \mu X.T_1\}T_1)$

else

$\uparrow$

Tämän algoritmin pysähtyminen on taattu, jos $reachables_{S_m}(S,T)$ on äärellinen kaikilla pareilla (S, T). Osoittautuu, että näin todella on, mutta tämän todistaminen sivuutetaan tässä.

Viitteet

Pie02 Benjamin C. Pierce. *Types and Programming Languages.* The MIT Press, Cambridge, Massachusetts, 2002.