

Պրոցեսների այլընտրանքային համակցման գործողությունը և նրա իրականացումը

Լևոն Հ. Հայրապետյան

Երևանի Պետական Համալսարան
e-mail levon.hayrapetyan@gmail.com

Ամփոփում

Աշխատանքում դիտարկված է պրոցեսների այլընտրանքային համակցման գործողությունը որոշակի օբյեկտը կոդմոդուլված մոդելում: Այն տրված պրոցեսները միավորում է և կառուցում է մեկ մոդ պրոցես, որի կատարումը համարվում է աղբյուր պրոցեսների սինթեզում կատարմանը: Աշխատանքում սահմանված է ալգորիթ, որը տրված պրոցեսների համար կառուցում է նրանց՝ ժամանակի տեսակետից օպտիմալ համակցումը: Գնահատված է ալգորիթի բարդությունը և մկարագրված են իրականացման մանրամասները:

1. Ներածություն

Վերտուալ Գործարանը (ՎԳ) սահմանվել էր [4]-ում որպես ծրագրային մոդելների և կիրառությունների համակարգ, որը աշխատեցնում է ֆիզիկական գործարանը: Այն թույլ է տալիս զուգահեռ և բաշխված միջավայրերում լուծել այնպիսի խնդիրներ, ինչպիսիք են պաշարների բաշխումը, պրոցեսների դեկավարումը և օպտիմիզացիան: ՎԳ մոդելի մկարագրման տարրերն են օբյեկտները և պրոցեսները: Պրոցեսը մկարագրում է ժամանակի մեջ օբյեկտների գործունեության տիպալը: Պրոցեսի մկարագրման տարրերն են իրավիճակները և գործողությունները: Իրավիճակը, ընդհանրապես ասած, սահմանափակում է օբյեկտի վիճակը ինչ-որ ժամանակահատվածում: Այն բաղկացած է վիճակի մասին պնդումից, որ կարող է փոխել իր ճշտությունը ժամանակի մեջ և ժամանակային սահմանափակումից, որը մշտն է այն ժամանակահատվածները, որոնց պրոցեսում պնդումը տեղի ունի: Գործողությունը մկարագրում է մի մեխանիզմ, որը կատարվելիս առաջացնում է պատահար:

Պրոցեսը կարող է մկարագրվել իրավիճակների և գործողությունների ցանցի միջոցով: [2, 6]-ում մկարագրված է պրոցեսի սեմանտիկան և պրոցեսների այլընտրանքային համակցման գաղափարը: Ներկայումս պատրաստվել և տպագրության է հանձնվել մի աշխատանք, որն ընդլայնում է զմային պրոցեսների դասի վրա սահմանված այլընտրանքային համակցման գործողությունը զուգահեռ պրոցեսների որոշ ենթադասի վրա և մկարագրում է (ժամանակի տեսակետից) օպտիմալ համակցման կառուցման ալգորիթը: Այս աշխատանքի մպատակն է

Լ. Հ. Հայրապետյան

նկարագրել այդ ալգորիթմի իրականացման և [5]-ում նկարագրված ProcessBuilder համակարգի մեջ նրա իմպլեմենտացիան մանրամասները:

2. Մոդելի նկարագրությունը

Պրոցեսների և համակցման գործողության նկարագրման համար օգտագործվում է [2, 4]-ում նկարագրված մոդելը: Ամբողջականության և ընթերցողականության նկատառումներից ելնելով ստորև քերվում է այդ մոդելի նկարագրությունը:

- $object = (ID, \{state\}, \{operation\})$
- $state = \{variable\} \rightarrow \{value\}$
- $operation = \{state\} \times \{msg\} \rightarrow \{state\} \times 2^{(msg)}$
- $environment = \{object\}$

Դիցուք $\varphi : N \rightarrow environment$ որևէ քվարկում է: $\varphi(i)$ -ն կամ կնշանակվի o_i : Բաղադրյալ կառուցվածքի բաղկացուցիչ տարրերը կնշանակվեն իրենց անունով՝ նախաձանցված «.»-ով: Օրինակ՝ o օբյեկտի $state$ տարրը կնշանակվի $o.state$: Դիցուք Θ -ն որևէ միջավայր է:

- $\Theta.StateSet = o_1.StateSet \times \dots \times o_n.StateSet$, որտեղ $o_i \in \Theta$;
- $Time = \{0, 1, 2, \dots\}$;
- $TimeInterval = \{t - t_0 \mid t, t_0 \in Time\}$;
- $situation = (object, assertion, TimeConstraint)$;
- $assertion : \{state\} \rightarrow \{TRUE, FALSE\}$;
- $TimeConstraint = \{t_0 + n * \Delta t \mid n \in N, t_0 \in Time, \Delta t \in TimeInterval\}$;
- $duration : \{operation\} \rightarrow TimeInterval$;
- $SituationBatch = \{situation\}$;
- $Scene = (t, state, \{msg\})$, որտեղ $state \in \Theta.StateSet$:

Միջավայրի վիճակը նշանակվում է $\Theta.state$ -ով: $\Theta.state \in \Theta.StateSet$:

2.1. Պրոցեսի սահմանումը

Պրոցեսները սահմանված են [2, 4]-ում: Ստորև քերվում է համառոտ սահմանումը:

Սահմանում Պրոցես է կոչվում հետևյալ վեցյակը $P = (\Theta, \Sigma, B, \Gamma, b_s, b_e)$, որտեղ

- Θ – պրոցեսի միջավայրն է;
- Σ – իրավիճակների վերջավոր բազմություն է, $\forall \sigma \in \Sigma \sigma.Object \in \Theta$;
- B – իրավիճակների-խմբերի վերջավոր բազմություն է $B \subseteq \Sigma$;
- $\Gamma = \{(\sigma, b, \omega) \mid \sigma \in \Sigma, b \in B, \omega \in \sigma.Object.Operations\}$ անցումների բազմություն է;
- b_s – սկզբնական իրավիճակների-խումբն է;
- b_e – վերջնական իրավիճակների-խումբն է:

Սահմանում $p = (\sigma, b, \omega) \in \Gamma$ կոչվում է տարրական ճանապարհ: $p_1 p_2 \dots p_n$ կոչվում է ճանապարհ, եթե $p_{i+1}.\sigma \in p_i.b$, որտեղ $1 \leq i < n$: n -ը կոչվում է ճանապարհի երկարություն: Ճանապարհը կոչվում է լրիվ, եթե $p_1.\sigma \in b_s$ և $p_n.b = b_e$: $p_1 p_2 \dots p_n$ ճանապարհը կոչվում է ցիկլ, եթե $p_1.\sigma \in p_n.b$:

Սահմանում Եթե $\exists (\sigma, b, \omega) \in \Gamma$ ապա կասենք ω գործողությունը հաջորդում է σ իրավիճակին:

Դիտարկվում են հետևյալ սահմանափակումներով պրոցեսները.

1. $\forall \sigma \in \Sigma \exists! (\sigma, b, \omega) \in \Gamma$;
2. ցիկլեր չկան:

Արդյախի պրոցեսների բազմությունը նշանակվում է Π -ով: Պրոցեսը կոչվում է գծային, եթե այն պարունակում է ճիշտ մեկ լրիվ ճանապարհ: Գծային պրոցեսների բազմությունը նշանակվում է Λ -ով:

Դիցուք $P \in \Pi$ և $b \in P.B$: b_i -ից b բոլոր ճանապարհների բազմությունը նշանակվում է p_i -ով: Դիցուք $k = \max_{p \in p_i} (\text{length of } p)$: Այդ դեպքում կասենք b -ն *պատկանում է k -րդ մակարդակին*:

k -րդ մակարդակին պատկանող իրավիճակների-խմբերի բազմությունը նշանակվում է L_k -ով:

2.2 Պրոցեսների համարժեքությունը

Դիցուք $P \in \Pi$; $q, q' \in P.\Theta.StateSet$: Դիցուք $I = (t, q, MsgSet)$ և $I' = (t, q', MsgSet)$ որևէ մուտքային պատկերներ են: q և q' կոչվում են համարժեք, եթե $P(I) = P(I')$: Կնշանակվի $q \equiv q'$: Ակնհայտ է, որ $\equiv$ համարժեքության հարաբերություն է և ծնում է համարժեքության դասեր $P.\Theta.StateSet$ -ում: Այդ դասերի բազմությունը կնշանակվի $E(P)$ -ով:

Դիցուք $P_1, P_2 \in \Pi$, $I_1 = (t_0, P_1.\Theta.state, MsgSet_0)$ և $I_2 = (t_0, P_2.\Theta.state, MsgSet_0)$:

Սահմանում P_1 և P_2 պրոցեսները կոչվում են *համարժեք*, եթե $|E(P_1)| = |E(P_2)|$ և կա՛մ միաժամանակ $P(I_1)$ և $P(I_2) = \perp$, կա՛մ $\exists \psi : E(P_1) \rightarrow E(P_2)$ այնպիսի փոխմիարժեք արտապատկերում, որ այն բանից, որ $\exists Q_0 \in E(P_1)$ $I_1.state \in Q_0$ և $I_2.state \in \psi(Q_0)$ հետևում է, որ $\exists Q \in E(P_1)$ այնպիսին, որ $P_1(I_1).state \in Q$, $P_2(I_2).state \in \psi(Q)$ և $P_1(I_1).MsgSet = P_2(I_2).MsgSet$. Կնշանակվի $P_1 \equiv P_2$:

2.3 Պրոցեսի ազդեցությունների գրաֆը

Պրոցեսի տարրեր իրավիճակների միջև ազդեցությունները, ինչպես մաս ազդեցությունների գրաֆը սահմանվել են [5]-ում: Ստորև բերվում են համառոտ սահմանումները:

Դիցուք $P \in \Pi$ և $\sigma, \sigma' \in P.\Sigma$: $\exists! \omega \in \sigma.\text{object.OperationSet}$ այնպես որ ω -ն հաջորդում է σ -ին:

Կասենք տեղի ունի σ իրավիճակի *ազդեցություն* $\sigma' \rightarrow \sigma$ իրավիճակի վրա ($\sigma \rightarrow \sigma'$) եթե ω գործողության կատարումից ճիշտ է հետևյալից որևէ մեկը.

1. $\sigma'.$ object-ը ստանում է հաղորդագրություն (տեղի ունի $\sigma \rightarrow^m \sigma'$ ազդեցություն);
2. $\sigma'.$ object-ը փոխում է իր վիճակը (տեղի ունի $\sigma \rightarrow^s \sigma'$ ազդեցություն):

Կասենք տեղի ունի I մուտքային պատկերի *ազդեցություն* σ իրավիճակի վրա ($I \rightarrow \sigma$), եթե չկա այնպիսի σ' , որ $\exists \sigma' \rightarrow \sigma$ և հետևյալից որևէ մեկը ճիշտ է.

1. $\sigma.$ object-ը ստանում է հաղորդագրություն $I.MsgSet$ -ից ($I \rightarrow^m \sigma$);
2. $\sigma.$ object $\in P.\Theta$ ($I \rightarrow^s \sigma$):

Կասենք տեղի ունի σ իրավիճակի *ազդեցություն* O նյքային պատկերի վրա ($\sigma \rightarrow O$) եթե չկա այնպիսի σ' , որ $\exists \sigma \rightarrow \sigma'$ և հետևյալից որևէ մեկը ճիշտ է.

1. ω գործողությունը առաջացնում է m հաղորդագրություն և $m \in O.MsgSet$ ($\sigma \rightarrow^m O$);
2. $\sigma.$ object $\in P.\Theta$ ($\sigma \rightarrow^s O$):

Սահմանում $\sigma \rightarrow \sigma'$ ազդեցությունը կոչվում է *անմիջական* եթե $\exists p_1 \dots p_n$ այնպիսի ճանապարհ, որ $\sigma = p_1.\sigma$ և $\sigma' \in p_n.b$: Հակառակ դեպքում այն կոչվում է *ամուղղակի*:

$V(P) = \{I, O\} \cup P.\Sigma$ կոչվում է ազդեցության զագաթների բազմություն: P պրոցեսի բոլոր ազդեցությունների բազմությունը նշանակվում է $W(P)$ -ով: Ակնհայտ է, որ $W(P) \subseteq V(P) \times V(P)$: $IG(P) = (V(P), W(P))$ կոչվում է P պրոցեսի ազդեցությունների գրաֆ: $N(\sigma) = \{\sigma' \mid \exists \sigma \rightarrow \sigma'\}$ σ իրավիճակից ազդվող իրավիճակների բազմությունն է:

3. Պրոցեսների այլընտրանքային համակցումը

3.1. Պրոցեսի բառերը և նրանց այլընտրանքային համակցումը

Դիցուք $P \in \Pi$ և $p = p_1 \dots p_n$ լրիվ ճանապարհ է P -ում, որտեղ $p_i = (\sigma_i, b_i, \omega_i)$: p ճանապարհին համապատասխանեցվում է $w_p = \omega_1 \dots \omega_n$: n -ը կոչվում է w_p -ի

L. 4. Հայրապետյան

երկարություն և նշանակվում է $w_k = w^{(k)} = \omega_1 \dots \omega_k$ ($k \leq n$) կոչվում է w -ի ենթաբառ, իսկ w -ի k -րդ տառը նշանակվում է w_k -ով: $w_p = \{w_p \mid p \in P\}$: նշանակվում է P պրոցեսի բառերի բազմությունը:

Մահմանում $w = \omega_1 \dots \omega_n$ և $w' = v_1 \dots v_m$ բառերը կաշվում են անկախ, եթե $\omega_i \neq v_j$ ($1 \leq i \leq n, 1 \leq j \leq m$): Նշանակվում է $w \parallel w'$:

Դիցուք $w_1 = \omega_1 \dots \omega_n$ և $w_2 = v_1 \dots v_m$ նրան անկախ բառեր են: $w = \eta_1 \dots \eta_{m+n}$ բառը կոչվում է w_1 և w_2 բառերի *այլընտրանքային համակցում*, եթե $\exists 1 \leq r_1 < r_2 < \dots < r_n \leq n+m$ և $1 \leq k_1 < k_2 < \dots < k_m \leq n+m$ թվեր, այնպիսիք, որ $r_i \neq k_j$, $\eta_{r_i} = \omega_i$ և $\eta_{k_j} = v_j$, որտեղ $1 \leq i \leq n$ և $1 \leq j \leq m$: w_1 և w_2 բառերի բոլոր այլընտրանքային համակցումների բազմությունը նշանակվում է $w_1 \oplus w_2$: Այնուհայտ է, որ $|w_1 \oplus w_2| = C_{n+m}^{n+m}$: Դիցուք $P, Q \in \Pi$ և $\forall w \in w_P \forall v \in w_Q w \parallel v$: w_P -ի և w_Q -ի բոլոր այլընտրանքային համակցումների բազմությունը նշանակվում է $w_P \oplus w_Q = \bigcup_{w_1 \in w_P, w_2 \in w_Q} w_1 \oplus w_2$:

Դիցուք $w = \omega_1 \dots \omega_n$ որևէ բառ է և $\tau = (t_1, t_2, \dots, t_n)$ որևէ տեղադրություն է [3, 7]: Այդ դեպքում τ -ի կիրառությունը w -ի վրա նշանակվում է $w\tau = v_1 \dots v_n$, որտեղ $v_i = \omega_{\tau_i}$:

Մահմանում w_1 բառը կոչվում է w_2 -ի տեղափոխություն, եթե $\exists \tau$ տեղադրություն, այնպիսին, որ $w_1\tau = w_2$: Նշանակվում է $w_1 \approx w_2$: W բառերի բազմությունը կոչվում է *տեղափոխությունների բազմություն*, եթե $\forall w_1, w_2 \in W w_1 \approx w_2$: Դիցուք W_1 -ը և W_2 -ը այնպիսի բազմություններ են, որ $\forall w_1 \in W_1 \forall w_2 \in W_2 w_1 \approx w_2$: Այդ դեպքում կգրվի $W_1 \approx W_2$:

Դիցուք $b \in P.B$: Կնշանակենք $w_b = \{w_p \mid p \in p_b\}$:

3.2. A-պրոցեսների բազմությունը

Դիցուք $P \in \Pi$ և $L_0, \dots, L_n$ P -ի մակարդակներն են:

Մահմանում L_k մակարդակը կոչվում է մինիմալ, եթե $\forall b, b' \in L_k w_b \approx w_{b'} \Rightarrow b = b'$:

Մահմանում P -ն կոչվում է ալտերանտիվների պրոցես կամ A-պրոցես, եթե

- $\exists I \geq 0$ թիվ այնպիսին, որ $\forall w \in w_P$ համար $|w| = I$;
- $\forall b \in L_k w_b$ -ն տեղափոխությունների բազմություն է ($0 \leq k \leq I$);
- L_k -ն մինիմալ է ($0 \leq k \leq I$):

I -ը կոչվում է P պրոցեսի երկարություն և նշանակվում է $|P| = I$: Բոլոր A-պրոցեսների բազմությունը նշանակվում է A-ով: Այնուհայտ է, որ $\Lambda \subset A \subset \Pi$: Այլընտրանքային համակցման գործողությունը կսահմանվի A-պրոցեսների դասի վրա:

Մահմանում P' պրոցեսը կոչվում է PQ -լրիվ, եթե $w_{P'} = w_P \oplus w_Q$:

3.3. A-պրոցեսների այլընտրանքային համակցումը

Դիցուք P, P_1 և $P_2 \in A$: Դիցուք $I_1 = (t_1^0, \text{state}_1^0, \text{msgs}_1^0)$ և $I_2 = (t_2^0, \text{state}_2^0, \text{msgs}_2^0)$ որևէ մուտքի պատկերներ են, որտեղ $\text{state}_i^0 \in Q_i^0 \in E(P_i)$ և $\text{state}_2^0 \in Q_2^0 \in E(P_2)$: Դիցուք $P_1(I_1) = O_1$ և $P_2(I_2) = O_2$, որտեղ $O_1 = (t_1, \text{state}_1, \text{msgs}_1)$ և $O_2 = (t_2, \text{state}_2, \text{msgs}_2)$: Դիցուք $I(I_1, I_2) = (t_0, \text{state}_0, \text{msgs}_0)$ -ն որևէ մուտքի պատկեր է, որտեղ $\text{state}_0 \in Q_0 \in E(P)$ և $\text{msgs}_0 = \text{msgs}_1^0 \setminus \text{msgs}_2 \cup \text{msgs}_2^0 \setminus \text{msgs}_1$:

Մահմանում P -ն կոչվում է P_1 -ի և P_2 -ի այլընտրանքային համակցում, եթե

- $w_P \subseteq w_{P_1} \oplus w_{P_2}$;
- $\exists \psi : E(P_1) \times E(P_2) \rightarrow E(P)$ փոխմիարժեք արտապատկերում այնպես, որ $\psi(Q_1^0, Q_2^0) = Q_0$;
- $P(I(I_1, I_2)) = (t, \text{state}, \text{msgs})$, որտեղ
 - $\text{state} \in Q = \psi(Q_1, Q_2)$

$$b. \text{msg}_s = \text{msg}_1 \setminus \text{msg}_2^* \cup \text{msg}_2 \setminus \text{msg}_1^* :$$

Սահմանում P_1 -ի և P_2 -ի բոլոր համակցումների բազմությունը նշանակվում է $P_1 \oplus P_2$ -ով:

Լեմմա 3.3.1 Դիցուք $P_1, P_2 \in A: \forall P, P' \in P_1 \oplus P_2, P \equiv P'$:

Սահմանում $P \in P_1 \oplus P_2$ կոչվում է P_1 -ի և P_2 -ի *օպտիմալ* համակցում, եթե $\forall I, \forall Q \in P_1 \oplus P_2, \text{dur}(P, I) \leq \text{dur}(Q, I)$:

4. Համակցման ալգորիթմը

Ալգորիթմը բաղկացած է հետևյալ քայլերից.

1. $P_1 P_2$ -լրիվ պրոցեսի կառուցում;
2. P_1 բույլատրելի ծանապարհների հեռացում;
3. Մնուղակի ազդեցությունների հեռացում:

4.1. $P_1 P_2$ -լրիվ պրոցեսի կառուցում

Այս քայլը բաղկացած է մի քանի գործողություններից, որոնք կիրառելով P_1 և P_2 պրոցեսների վրա կառուցվում է $P_1 P_2$ -լրիվ P պրոցեսը: Պրոցեսության սկզբում $P, \Theta, P, \Sigma, P, B$ և P, Γ բազմությունները դատարկ են:

Ձևափոխություն: Սկզբնաղբերակորոշում

ՄՈՒՏՔ: աղբյուր պրոցեսներ P_1 և P_2 ; ԵԼՔ: պրոցես P

- $\Theta = P_1, \Theta \cup P_2, \Theta$;
- $\Sigma = P_1, \Sigma \cup P_2, \Sigma$;
- $b_s = P_1, b_s \cup P_2, b_s$;
- $b_e = \emptyset$;
- $B = \{b_s, b_e\} \cup P_1, B \cup P_2, B \setminus \{P_1, b_s, P_2, b_s\}$;
- $\Gamma = P_1, \Gamma \cup P_2, \Gamma$;

Սկզբնաղբերակորոշման պրոցեսում աղբյուր պրոցեսների սկզբնական իրավիճակների խմբերը միավորվում են (Նկար 1):

Ձևափոխություն: Կոմպակտացում

ՄՈՒՏՔ: պրոցես P ; ԵԼՔ: պրոցես P'

$\forall b \in P, B$ համար, եթե $\exists \sigma \in b$ և $\exists p_1, \dots, p_n (n > 1)$ այնպիսիք, որ $\forall i \in \{1, \dots, n\} p_i, \sigma = \sigma$ և $\forall j \in \{1, \dots, n\} p_j, \omega = p_j, \omega$ ապա կատարվում են հետևյալ ձևափոխությունները.

- $b_{\text{new}} = \bigcup_{i=1}^n p_i, b$ ավելացվում է P, B -ին;
- $(\sigma, b_{\text{new}}, p_1, \omega)$ ավելացվում է P, Γ -ին;
- p_i և p_j, b ($i \in \{1, \dots, n\}$) հեռացվում են P, Γ -ից և P, B -ից

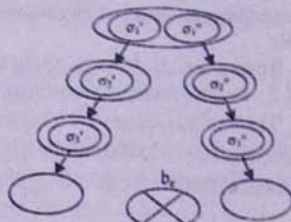
Նկար 2-ը ցույց է տալիս այս ձևափոխությունը:

Ձևափոխություն: Ամփագում

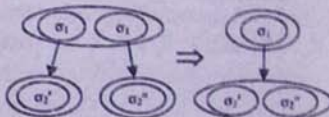
ՄՈՒՏՔ: աղբյուր պրոցեսներ P_1 և P_2 և պրոցես P ; ԵԼՔ: պրոցես P'

$\forall b \in P, B$ համար, եթե $\exists \sigma_1 \in P_1, \Sigma$ և $\exists \sigma_2 \in P_2, \Sigma$ այնպիսիք, որ $\sigma_1, \sigma_2 \in b$ ապա կատարվում են հետևյալ ձևափոխությունները (դիցուք $(\sigma_1, b_1, \omega_1), (\sigma_2, b_2, \omega_2) \in P, \Gamma$)

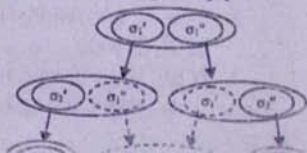
- նոր $b_{\text{new}} = \emptyset$ ավելացվում է P, B -ին;



Նկար 1: Սկզբնաղբերակորոշում



Նկար 2: Կոմպակտացում



Լ. Լ. Հայրապետյան

- σ_1 -ի պատճենն σ' -ը ավելացվում է b_2 -ին և σ_2 -ի պատճենն σ' -ը ավելացվում է b_1 -ին, σ' -ը և σ'' -ը ավելացվում են P, Σ ;
- $(\sigma', b_{new}, \omega_1)$ -ը և $(\sigma'', b_{new}, \omega_2)$ -ը ավելացվում են P, Γ -ին:

Ավելացման ճնափոխությունը ցուցադրված է նկար 3-ում:

Չնափոխություն: Ավարտում

ՄոՒՏԶ: պրացես P ; ԵԼԶ: պրոցես P'

P, b_i -ն հեռացվում է P, B -ից և նոր P, b_i է նշանակվում P -ի վերջին մակարդակի միակ իրավիճակների խումբը:

Պրոցեսորա: Փակում

Սկզբնարժեքավորում;

$k=0$;

while(L_k -ի վրա Ավելացում կամ Կոմպակտացում կիրառելի է) do
begin

$\forall b \in L_k$ կոմպակտացվում է, եթե կիրառելի է;

$\forall b \in L_k$ ավելացվում է, եթե կիրառելի է;

$k = k + 1$;

end

Ավարտում;

Փակում պրացեսորայի արդյունքը նշանակվում է $P=C(P_1, P_2)$ -ով:

Src: $P, \Sigma \rightarrow P_1, \Sigma \cup P_2, \Sigma$ արտապատկերումը սահմանվում է հետևյալ կերպ. $\text{Src}(\sigma)=\sigma$, եթե $\sigma \in P_1, \Sigma \cup P_2, \Sigma$ (ավելացվել է Սկզբնարժեքավորման ժամանակ) և $\text{Src}(\sigma)=\sigma'$, եթե σ' -ը σ -ի պատճենն է (ավելացված Ավելացման ժամանակ): $\text{Src}(\sigma)$ -ը σ իրավիճակի արդյունքն է:

Թեորեմ 4.1.1 Փակումը վերջավոր է և եթե $P = C(P_1, P_2)$, ապա $P \in A$ և P -ն $P_1 P_2$ -լրիվ է:

Այսպիսով, Փակում պրոցեսորայի արդյունքում ստացվում է պրոցես P -ն, որը պարունակում է բոլոր ճանապարհները, որ P_1 -ի և P_2 -ի կամայական այլընտրանքային համակցում կարող է պարունակել: Ինչպես կտեսնենք հաջորդ բաժնում, P -ի որոշ լրիվ ճանապարհներ կարող են լինել ոչ թույլատրելի այլընտրանքային համակցման տեսակետից: Այդ ճանապարհները պետք է հեռացվեն:

4.2. Ոչ թույլատրելի ճանապարհների հեռացում

Դիցուք $P_1, P_2 \in A$: Ընդհանրապես ասած, P_1 -ը և P_2 -ը կարող են փոխադրել միմյանց հետ և հետևաբար լինել կախված միմյանցից: Բոլոր այդպիսի կախվածությունները պետք է արտացոլվեն կամայական այլընտրանքային համակցումում: Ստորև այդ կախվածությունները սահմանվում են որպես գծային անհավասարությունների համակարգ, որ կապում է աղբյուր պրոցեսները:

Դիցուք m -ը որևէ հաղորդագրություն է և $P \in A$ որևէ պրոցես է: $P, \Sigma_k(m)$ -ով կնշանակվեն $\{\sigma \mid \sigma \in P, \Sigma \text{ and } \exists l \rightarrow^m \sigma \in W(P)\}$ և $\{\sigma \mid \sigma \in P, \Sigma \text{ and } \exists \sigma' \rightarrow^m \sigma \in W(P)\}$ իրավիճակների բազմությունները, որոնք համապատասխանաբար ազդվում են մտադրային և ազդում են ելքային պատկերների վրա:

Սահմանում $S_{P_1 P_2} = \{\sigma_1 < \sigma_2 \mid \exists \text{ հաղորդագրություն } m \text{ այնպիսին, որ } \sigma_1 \in P_1, \Sigma_0(m) \text{ և } \sigma_2 \in P_2, \Sigma_1(m) \text{ կամ } \sigma_1 \in P_2, \Sigma_0(m) \text{ և } \sigma_2 \in P_1, \Sigma_1(m)\}$ անհավասարությունների համակարգը կոչվում է P_1 և P_2 պրոցեսները կապող համակարգ:

Սահմանում Դիցուք P -ն $P_1 P_2$ -լրիվ պրոցես է և $p = p_1 \dots p_n$ որևէ ճանապարհ է P -ում, որտեղ $p_i = (\sigma_i, b_i, \omega_i)$: Այդ դեպքում $\{\sigma < \sigma' \mid \sigma = \text{Src}(\sigma_i), \sigma' = \text{Src}(\sigma_{i+1}) \mid 1 \leq i \leq n-1\}$ գծային անհավասարությունների համակարգը նշանակվում է S_p -ով:

Մահմանում Γ -իցուր $P_1, P_2 \in A$, P -ն $P_1 P_2$ -լինի պրոցես է: P պրոցեսի p ճանապարհը կոչվում է $P_1 P_2$ -բուլլատրեյի կամ պարզապես բուլլատրեյի, եթե $S_{P_1 P_2} \cup S_p$ համակարգը ունի լուծում:

Մահմանում $P \in A$ կոչվում է $P_1 P_2$ -մաքսիմալ, եթե $w_p \subseteq w_{p_1} \oplus w_{p_2}$ և p -ն բուլլատրեյի է $\Leftrightarrow w_p \in W_P$:

Ձևափոխություն: Պրոցեսի մշում

ՄՈՒՏՏՔ: աղբյուր պրոցեսներ P_1 և P_2 և $P_1 P_2$ -լինի պրոցես P ; ԵԼՔ: նշված պրոցես P' b_i նշվում է;

foreach($p = (\sigma, b', \omega) \in P \cdot \Gamma$)

begin

if($\exists p_1 \dots p_i p_{i+2} \dots p_{n+m}$ բուլլատրեյի լինի ճանապարհ) then

begin

σ, p and b' նշվում են;

end

end

Γ -իցուր $P, P' \in \Pi$ և $p = (\sigma, b', \omega) \in P \cdot \Gamma$, որտեղ $\sigma \in b$ և $b \in P \cdot B$: Γ -իցուր $P' \cdot \Theta = P \cdot \Theta$, $P' \cdot \Sigma = P \cdot \Sigma \setminus \{\sigma\}$, $P' \cdot B = P \cdot B \cup \{b''\} \setminus \{b\}$, $b'' = b' \setminus \{\sigma\}$, $P' \cdot \Gamma = P \cdot \Gamma \setminus \{p\}$, $P' \cdot b_i = P \cdot b_i$ and $P' \cdot b_i = P \cdot b_i$: Այսպիսով P -ը կառուցվում է P -ից՝ հետացնելով σ -ն և p -ն: Այն նշանակվում է $P' = r(P, p)$

Ձևափոխություն: Ու բուլլատրեյի ճանապարհների հեռացում

ՄՈՒՏՏՔ: նշված պրոցես P ; ԵԼՔ: $P_1 P_2$ -մաքսիմալ պրոցես P'

$P' := P$;

foreach($p \in P \cdot \Gamma$ և p -ն նշված չէ)

$P' := r(P', p)$;

foreach($b \in P'$ և b -ն նշված չէ)

$P' \cdot B := P' \cdot B \setminus \{b\}$;

Արդյունքում ստացված պրոցեսը նշանակվում է $P' = R(P_1, P_2, P)$ -ով:

Թեորեմ 4.2.1 $P' = R(P_1, P_2, P)$ պրոցեսը $P_1 P_2$ -մաքսիմալ A -պրոցես է:

4.3. Անուղղակի ազդեցությունների հեռացում

Այստեղևոր կոդիտարկներն այնպիսի աղբյուր պրոցեսներ, որոնք չեն պարունակում անուղղակի ազդեցություններ: Γ -իցուր $P_1, P_2 \in A$ այդպիսի պրոցեսներ են: Ընդհանրապես ասած, $P = R(P_1, P_2, C(P_1, P_2))$ պրոցեսը կարող է ունենալ անուղղակի ազդեցություններ, որոնք կարող էին առաջանալ մախորդ երկու պրոցեսից: Այդ ազդեցությունները պետք է հեռացվեն, որպեսզի համակցման ամեն մի ճանապարհ կարողանա անկախ կատարվել: Հետևյալ պրոցեսից հեռացվում է աղբյուր A -պրոցեսը այպես, որ արդյունքում ստացված P' A -պրոցեսը չպարունակի անուղղակի ազդեցություններ: Առաջարկվող պրոցեսից հեռացվում է P -ի և P' -ի խոնկյիցումալ համարձեցությունը:

Պրոցեսից հեռացվում է մի քանի ձևափոխություններից որոնք կառուցում են արդյունք պրոցեսը քայլ առ քայլ: Γ -իցուր $P = (\Theta, \Sigma, B, \Gamma, b_o, b_e)$ աղբյուր A -պրոցեսն է, իսկ $P' = (\Theta', \Sigma', B', \Gamma', b'_o, b'_e)$ ՝ արդյունք: Պրոցեսից հեռացվում է Θ', Σ', B' and Γ' բազմությունները դատարկ են, իսկ b'_o -ը և b'_e -ը՝ անորոշ: Հետևյալ լրացուցիչ սահմանումները օգտագործվում են ձևափոխության պրոցեսում: $\Sigma(o) = \{\sigma \in \Sigma \mid \sigma.\text{object} = o\}$ o օբյեկտին վերաբերվող իրավիճակների բազմությունն է, $\Phi_o : N \rightarrow \Sigma(o)$ որևէ բվարկում է: Որպես $\Phi_o(i)$ -ի կարճ ձև կօգտագործվի σ_i :

1. Հ. Հայրապետյան

Չևափոխություն: Միջավայրի ընդլայնում

Դիցուք $\alpha = (ID, Q, W) \in \Theta$ որևէ օբյեկտ է: Ստեղծվում է նոր $\alpha(o) = (ID, Q', W')$ օբյեկտ, որտեղ $Q' = Q^{(2(o)+1)}$ և $W' = \emptyset$: $\alpha(o)$ -ն կարելի է դիտարկել որպես օբյեկտների զանգված: Դիցուք $q' = (q_0, \dots, q_n)$: q' -ի $q_1, \dots, q_n$ բաղադրիչները կօգտագործվեն համապատասխանաբար $\sigma'_1 \dots \sigma'_n$ իրավիճակներում, իսկ q_0 -ն կօգտագործվի որպես օբյեկտի վիճակ մուտքային և ելքային պատկերներում: Այսպիսով, ամեն $\alpha \in \Theta$ օբյեկտի համար ավելացվում է $\alpha(o)$ օբյեկտը Θ' -ին: Նկատենք, որ այս պահին $\alpha(o)$ օբյեկտի գործողությունների բազմությունը դատարկ է:

Չևափոխություն: Իրավիճակների փոխարինում

Ամեն մի $b \in B$ իրավիճակների-խմբի համար նոր $\beta(b)$ իրավիճակների խումբ է ավելացվում B' -ին: Ամեն մի $\sigma \in \Sigma$ իրավիճակի համար նոր $\delta(\sigma)$ իրավիճակ է ավելացվում Σ' -ին և $\beta(b)$ -ին, որտեղ $\delta(\sigma).object = \alpha(\sigma.object)$ և $\delta(\sigma).TimeConstraint = \sigma.TimeConstraint$: $\delta(\sigma).assertion$ սահմանվում է հետևյալ կերպ: Դիցուք ϕ'_0 . Σ' -ի այնպիսի թվարկում է, որ $\phi'_0(i) = \delta(\phi_0(i))$. $\phi'_0(i)$ -ի փոխարեն կգրվի σ'_i : Այդ դեպքում $\sigma'_i.assertion(q_0, \dots, q_n) = \sigma_i.assertion(q_i)$: Այսպիսով, σ'_i իրավիճակի պնդումը կախված է միայն $\sigma'_i.object.state$ -ի i -րդ բաղադրիչից:

b'_i և b''_i սահմանվում են որպես $\beta(b_i)$ և $\beta(b_o)$:

Չևափոխություն: Գործողությունների փոխարինում

Ամեն մի $(\sigma, b, \omega) \in \Gamma$ տարրական ճանապարհի համար նոր $(\delta(\sigma), \beta(b), \gamma(\omega))$ տարրական ճանապարհ է ավելացվում Γ' -ին, որտեղ $\gamma(\omega)$ գործողությունը ավելացվում է $\delta(\sigma).object$ -ի գործողությունների բազմությանը: Այն սահմանվում է հետևյալ կերպ: Դիցուք $\delta(\sigma) = \sigma'_i$: Այդ դեպքում $\gamma(\omega) = (q_0, q_1, \dots, q_n, m) = ((q'_0, q'_1, \dots, q'_n), mSet)$, որտեղ

$$q'_0 = \begin{cases} \omega(q_0, m).q, & \text{if } \exists \sigma_i \rightarrow O \\ q_0, & \text{otherwise} \end{cases} \quad \text{և} \quad q'_j = \begin{cases} \omega(q_0, m).q, & \text{if } \exists i \rightarrow \sigma \\ \omega(q_i, m).q, & \text{if } \sigma_j \in N(\sigma_i), 1 \leq j \leq n \\ q_j, & \text{otherwise} \end{cases}$$

Այս ձևափոխությունների հաջորդական կիրառումը մշակակվում է $T(P)$ -ով: Ակնհայտ է, որ $T(P)$ -ի բարդությունը $O(n)$ է, որտեղ n -ը աղբյուր պրոցեսում իրավիճակների քանակն է: [2]-ում ցույց էր տրվել, որ եթե L_1 -ը և L_2 -ը գծային պրոցեսներ են, ապա $P = T(R(L_1, L_2, C(L_1, L_2)))$ հանդիսանում է L_1 -ի և L_2 -ի օպտիմալ համակցում:

Թեորեմ 4.3.1 $\forall P_1, P_2 \in A \quad P = T(R(P_1, P_2, C(P_1, P_2))) \Rightarrow P \in A$ և P -ն P_1 և P_2 պրոցեսների օպտիմալ համակցում է:

5. Գործողության բարդությունը

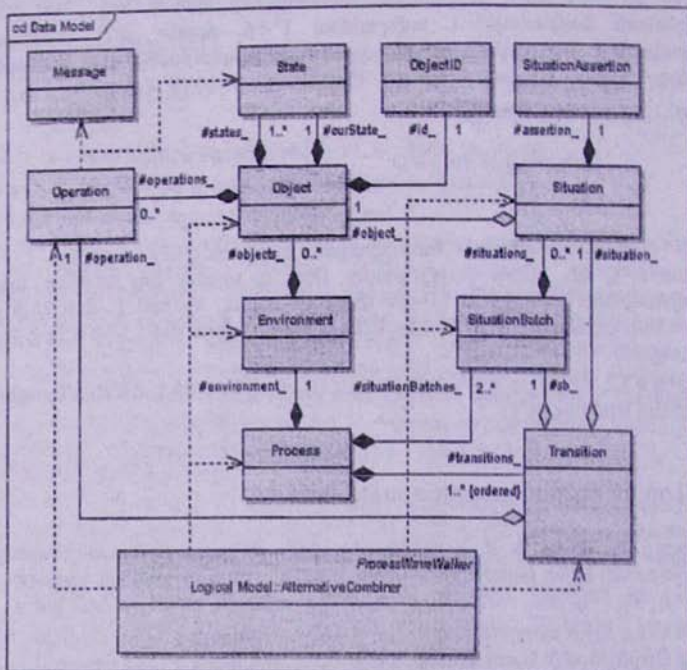
Դիցուք $P_1, P_2 \in A$ և $P = T(R(P_1, P_2, C(P_1, P_2)))$: Համակցման գործողության բարդությունը կլինի նկարագրված երեք քայլերի բարդությունների գումարը: Դիցուք $|P_1| = l_1$, $|P_2| = l_2$, $|w_{P_1}| = r_1$ և $|w_{P_2}| = r_2$: Ինչպես ցույց է տրվել [2]-ում, եթե $L_1, L_2 \in \Lambda$ ապա $C(L_1, L_2)$ -ի բարդությունը $O(|L_1| \times |L_2|)$ է: Հետևաբար $C(P_1, P_2)$ -ի բարդությունը կլինի $O(r_1 \times r_2 \times l_1 \times l_2)$: Մյուս կողմից, եթե $P' = C(P_1, P_2)$, ապա $|P'.\Sigma| = O(r_1 \times r_2 \times l_1 \times l_2)$ և $|P'.\Gamma| = O(r_1 \times r_2 \times l_1 \times l_2)$: Հետևաբար $R(P')$ և $T(R(P'))$ գործողությունների բարդությունը կլինի

$O(r_1 \times r_2 \times l_1 \times l_2)$: Այսպիսով, ամբողջ գործողության բարդությունը նույնպես կլինի $O(r_1 \times r_2 \times l_1 \times l_2)$:

6. Գործողության իրականացումը

Պրոցեսների այլընտրանքային համակցման գործողության ալգորիթը իրականացվել է ProcessBuilder [5] համակարգի շրջանակում: Այդ համակարգը բույլ է տալիս վիրտուալ բաշխված միջավայրերում պրոցեսների ֆորմալ նկարագրումը, ինչպես նաև դրանց մեկնաբանումը (interpretation), ստուգումը (verification), օպտիմիզացիան և այլն:

ProcessBuilder համակարգը բաղկացած է միջուկից, գրաֆիկական ինտերֆեյսից և գործիքներից: Միջուկը տրամադրում է տվյալների կառուցվածքներ՝ պրոցեսների և օբյեկտների նկերկայացման համար: Նկար 4-ում պատկերված է տվյալների մոդելի UML դիագրաման [1], որը նկարագրում է օգտագործված դասերը և նրանց փոխհարաբերությունները: Գրաֆիկական ինտերֆեյսը մախատեսված է այդ տվյալների կառուցվածքները օգտվողին ցույց տալու, ինչպես նաև նոր պրոցեսներ և օբյեկտներ ստեղծելու համար: Միջուկը նաև տրամադրում է ծրագրային ինտերֆեյս գործիքների համար: Ամեն գործիք մախատեսված է պրոցեսների հետ որոշակի գործողություն կատարելու համար: Պրոցեսների այլընտրանքային համակցումը իրականացված է որպես այդպիսի գործիք՝ AlternativeCombiner դասի միջոցով: Նկար 4-ում պատկերված է այդ դասի կախվածությունը միջուկի դասերից: Ինչպես ProcessBuilder համակարգը, այնպես էլ այլընտրանքային համակցման գործողությունը իրականացված են C++ լեզվով:



Նկար 4: Տվյալների

7. Եզրակացություն

Առաջարկվող ալգորիթմը թույլ է տալիս միավորել գծային պրոցեսների տրված բազմությունը մեկ ընդհանուր պրոցեսի մեջ, որը զուգահեռաբար կատարում է այդ պրոցեսները և սինխրոնիզացնում է նրանց փոխգործակցությունը: Ալգորիթմը ապահովում է մաև ժամանակի տեսակետից օպտիմալ համակցման կառուցումը: Ալգորիթմի իրականացումը ինտեգրված է գոյություն ունեցող ProcessBuilder համակարգի մեջ:

Գրականություն

- [1]. Grady Booch et.al., Unified Modeling Language User Guide, Pearson Education, 1999.
- [2]. L. Hayrapetyan. "Alternative combination of linear processes". In *Proceedings of CSIT'2005, Armenia*, September 2005, pp 65-69.
- [3]. Donald Knuth. "The Art of Computer Programming, Volume 3: Sorting and Searching", Third Edition. Addison-Wesley, 1997. ISBN 0-201-89685-0.
- [4]. P. Raulefs. "The Virtual Factory", IFIP World Computer Congress'94, v.2, pp.18-30, 1994.
- [5]. P. Raulefs; S. Shoukourian; A. Grigoryan. "Transformation of hammock type processes", In *Proceedings of HPC'2002, SCS International Advanced Simulation Technologies Conference ASTC'2002, USA*, April 2002, pp. 288-293.
- [6]. S. Shoukourian, A. Avagyan, D. Tavangarian, "Combination of separate processes in a distributed environment. A case of study." In *Proceedings of HPC'2000, SCS International Advanced Simulation Technologies Conference ASTC'2000, USA*, April 2000, pp. 280-285.
- [7]. B. L. van der Waerden, Algebra, Springer-Verlag, 1971

Operation of alternative combination of processes and its implementation

L. Hayrapetyan

Abstract

In this paper an object oriented model for processes is used to describe the operation of alternative combination of processes. For specified processes this operation constructs a new process, execution of which is equivalent to the synchronized execution of source processes. An algorithm is proposed that constructs the optimal (by time) combination of given source processes. Also, complexity of the algorithm is evaluated and implementation details are described.