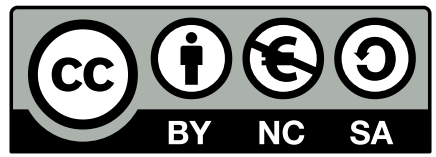
π-λογισμός

Γεώργιος Πιτσιλαδής

Εργασία για το μάθημα «Προχωρημένα Θέματα Λογικής»
ΔΠΜΣ Αλγόριθμοι, Λογική και Διακριτά Μαθηματικά

16 Μαΐου 2017



Ενότητα 1

Γενικά στοιχεία

Τι είναι ο π-λογισμός

Ο π-λογισμός (π-calculus) είναι ένα Turing-complete υπολογιστικό μοντέλο στο οποίο *διεργασίες (processes)* επικοινωνούν μέσω *καναλιών (channels)*.
Η ειδοποιός διαφορά του από παρόμοια συστήματα (άλγεβρες διεργασιών, process algebras) είναι ότι οι διεργασίες μπορούν να ανταλλάξουν «δείκτες» σε κανάλια.

Ο π-λογισμός μοντελοποιεί (καλύτερα) κατανεμημένα συστήματα και είναι εν γένει μη-ντετερμινιστικός.

Υπάρχουν δύο κύριες παραλλαγές: ο σύγχρονος και ο ασύγχρονος· ο πρώτος είναι συνηθέστερος και ισχυρότερος.

Γενικά, ανάλογα με τις ανάγκες υιοθετούνται διάφορες (μικρο)παραλλαγές ή επεκτάσεις του π-λογισμού.

Διαισθητικό παράδειγμα

Ένας εξυπηρετητής S διαθέτει μια πληθώρα καναλιών $\{ x_i \mid i = 1, \ldots, n \}$ προς έναν εκτυπωτή P, καθένα από τα οποία αντιστοιχεί σε διαφορετική συμπεριφορά P_i του εκτυπωτή.

Ένας πελάτης C, που συνδέεται με τον εξυπηρετητή με το κανάλι a, θέλει να χρησιμοποιήσει μία συγκεκριμένη λειτουργία του εκτυπωτή (πχ απλή εκτύπωση).

Ο S στέλνει το αντίστοιχο κανάλι x_i στον C μέσω του a και ο C μπορεί πλέον να χρησιμοποιήσει το x_i για να «μιλήσει» κατ' ευθείαν με τον εκτυπωτή.

(Πολύ) σύντομο ιστορικό

Ο π-λογισμός εισήχθη το 1992 από τους Robin Milner, Joachim Parrow και David Walker, ως βελτίωση του Calculus of Concurrent Systems (CCS) του Milner.

Έχει χρησιμοποιηθεί σε αρκετές εφαρμογές, με κυριότερες

- την ανάλυση διάφορων ειδών πρωτοκόλλων
- τη μοντελοποίηση κυτταρικών μηχανισμών.

Συγκεκριμένα για κρυπτογραφικές εφαρμογές:

1997	Spi-calculus	Martin Abadi & Andrew Gordon
2001	applied π-calculus	Martin Abadi & Cedric Fournet

Ενότητα 2

Συντακτικό

Η γλώσσα (μιας παραλλαγής) του π-λογισμού

Χρησιμοποιούμε ένα αριθμήσιμο σύνολο ονομάτων $\mathcal{N}$, που αντιστοιχούν ταυτόχρονα σε κανάλια και σε μεταβλητές.

Όρος	Διαίσθηση
$\mathbf{0}$	η κενή διεργασία· δεν κάνει τίποτα
$x(y).P$	λαμβάνει ένα z μέσω του x και συνεχίζει ως $P\{z/y\}$
$\overline{x}\langle y\rangle.P$	στέλνει το y μέσω του x και συνεχίζει ως P
$(\nu\, x)P$	ορίζει ένα καινούριο «ιδιωτικό» κανάλι x στην P
$P_1 + P_2$	επιλέγεται (μη ντετερμινιστικά) μία από τις P_1 και P_2
$P_1 \mid P_2$	οι P_1 και P_2 τρέχουν παράλληλα
$!P$	η P τρέχει δυνητικά άπειρες φορές $(P \mid P \mid P \mid \ldots)$
$[x = y]\, P$	αν τα ονόματα x, y είναι ίδια, συνεχίζει ως P
$[x \neq y]\, P$	αν τα ονόματα x, y είναι διαφορετικά, συνεχίζει ως P

Παράδειγμα

Το παράδειγμα με τον εκτυπωτή μπορεί να μοντελοποιηθεί ως εξής:

- Ο εκτυπωτής έχει μια σειρά από λειτουργίες $P_1, \ldots, P_n$, η καθεμιά από τις οποίες ενεργοποιείται μέσω του αντίστοιχου καναλιού x_i. Τα x_i μπορούν να έχουν περιοριστεί σε ένα περιβάλλον που περιλαμβάνει μόνο τον εξυπηρετητή και τον εκτυπωτή.
- Ο εξυπηρετητής S και ο πελάτης C μοιράζονται ένα κανάλι a.
- Έστω ότι η απλή εκτύπωση αντιστοιχεί στο δείκτη 1.
- Όταν ο εξυπηρετητής λάβει ένα σήμα στο a, στέλνει μέσω του a μία αναφορά στο κανάλι x_1.
- Ο πελάτης, που γνωρίζει πλέον το x_1, μπορεί να στείλει εκεί το αρχείο y που θέλει να εκτυπώσει.

Παράδειγμα (συνέχεια)

Συμβολικά, έχουμε

$$S = !(a(\epsilon).\bar{a}\langle x_1\rangle.\mathbf{0})$$
$$P = !(x_1(y_1).P_1 + \ldots + x_n(y_n).P_n)$$
$$C = \bar{a}\langle\epsilon\rangle.a(z).\bar{z}\langle y\rangle.\mathbf{0}$$

και όλο το σύστημα γράφεται

$$((\nu\, x_1)\cdots(\nu\, x_n)(S \mid P)) \mid C$$

Παράδειγμα (συνέχεια)

Συμβολικά, έχουμε

$$S = !(a(\epsilon).\bar{a}\langle x_1\rangle.\mathbf{0})$$
$$P = !(x_1(y_1).P_1 + \ldots + x_n(y_n).P_n)$$
$$C = \bar{a}\langle\epsilon\rangle.a(z).\bar{z}\langle y\rangle.\mathbf{0}$$

και όλο το σύστημα γράφεται

$$((\nu\, x_1)\cdots(\nu\, x_n)(S \mid P)) \mid C$$

Παρατήρηση

Συμπεριφορές δεύτερης τάξης (δηλαδή αποστολή ή λήψη διεργασιών) μπορούν –και είναι σωστότερο– να μοντελοποιούνται με τη χρήση καναλιών-triggers, όπως περίπου κάναμε εδώ.

Ελεύθερα και δεσμευμένα ονόματα

Στις διεργασίες $x(y).P$ και $(\nu\, y)P$, δεσμεύονται οι ελεύθερες εμφανίσεις του y στο P.

$\mathrm{fn}(P)$: τα ονόματα που εμφανίζονται ελεύθερα στην P
$\mathrm{bn}(P)$: τα ονόματα που εμφανίζονται δεσμευμένα στην P

Η μετονομασία ενός δεσμευμένου ονόματος ονομάζεται α-μετατροπή (α-conversion).
Δύο διεργασίες P, Q που μπορούν να ταυτιστούν με α-μετατροπές ονομάζονται α-ισοδύναμες (α-equivalent)· γράφουμε $P \equiv_\alpha Q$.

Αντικατάσταση ονομάτων

Συμβολισμός

$P\{y/x\}$: αντικατάσταση* των ελεύθερων εμφανίσεων του x με y στην P

*σε περίπτωση που το y εμφανίζεται δεσμευμένο στην P, πρέπει να προηγηθεί μία α-μετατροπή

Για παράδειγμα,

$$x(z).(\nu\, y)\overline{x}\langle y\rangle.\mathbf{0}\,\{y/x\} = y(z).(\nu\, v)\overline{y}\langle v\rangle.\mathbf{0}$$

Ενότητα 3

Σημασιολογία

Βασικά είδη σημασιολογίας

Γενικά, υπάρχουν διάφορα είδη σημασιολογίας γλωσσών προγραμματισμού:

- λειτουργική/μηχανιστική (operational· πώς συμπεριφέρεται η γλώσσα)
- δηλωτική/ενδεικτική (denotational· από ποια μαθηματικά αντικείμενα μοντελοποιείται)
- αξιωματική (axiomatic· ποια λογικά αξιώματα ικανοποιεί κάθε εντολή)

Η πιο συχνά χρησιμοποιούμενη για τον π-λογισμό είναι (μάλλον) η λειτουργική.

Βασικές έννοιες σημασιολογίας για τον π-λογισμό

Οι διεργασίες του π-λογισμού έχουν δύο ειδών σημασιολογικά χαρακτηριστικά:

1. Τι κάνουν: μία διεργασία μπορεί να εξελίσσεται σε μία άλλη.
2. Τι είναι: δυο διεργασίες μπορεί να έχουν ακριβώς την ίδια συμπεριφορά παρά τις συντακτικές διαφορές.

Βασικές έννοιες σημασιολογίας για τον π-λογισμό

Οι διεργασίες του π-λογισμού έχουν δύο ειδών σημασιολογικά χαρακτηριστικά:

1. Μία διεργασία μπορεί να εξελίσσεται σε μία άλλη (μεταβάσεις ή αναγωγές).
2. Τι είναι: δυο διεργασίες μπορεί να έχουν ακριβώς την ίδια συμπεριφορά παρά τις συντακτικές διαφορές.

Για την πρώτη περίπτωση (τι κάνουν οι διεργασίες), χρησιμοποιούμε

- είτε τη σημασιολογία με αναγωγές (reduction semantics), που βλέπει τις διεργασίες αφ' εαυτές
- είτε την (εκφραστικότερη) σημασιολογία μεταβάσεων με ετικέτες (labeled transition semantics), που βλέπει τις διεργασίες ως στοιχεία μέσα σε ένα περιβάλλον.

Εδώ θα εστιάσουμε στη δεύτερη.

Βασικές έννοιες σημασιολογίας για τον π-λογισμό

Οι διεργασίες του π-λογισμού έχουν δύο ειδών σημασιολογικά χαρακτηριστικά:

1. Μία διεργασία μπορεί να εξελίσσεται σε μία άλλη (μεταβάσεις ή αναγωγές).
2. Δυο διεργασίες μπορεί να έχουν ακριβώς την ίδια συμπεριφορά παρά τις συντακτικές διαφορές (δομική ομοιότητα).

Για τη δεύτερη περίπτωση (τι είναι οι διεργασίες), χρησιμοποιούμε την έννοια της δομικής ομοιότητας (structural congruence).

Βασικές έννοιες σημασιολογίας για τον π-λογισμό

Οι διεργασίες του π-λογισμού έχουν δύο ειδών σημασιολογικά χαρακτηριστικά:

1. Μία διεργασία μπορεί να εξελίσσεται σε μία άλλη (μεταβάσεις ή αναγωγές).
2. Δυο διεργασίες μπορεί να έχουν ακριβώς την ίδια συμπεριφορά παρά τις συντακτικές διαφορές (δομική ομοιότητα).

Η χρήση της δομικής ομοιότητας σε σημασιολογία μεταβάσεων με ετικέτες είναι εν μέρει θέμα προτίμησης· κάθε ειδική περίπτωση ομοιότητας μπορεί να αντικατασταθεί από επιπλέον κανόνες μετάβασης, αλλά και κάποιες μεταβάσεις μπορούν να εκφραστούν ως σχέσεις ομοιότητας.

Δομική ομοιότητα

Ορισμός (Δομική ομοιότητα)

Η $\equiv$ είναι η ελάχιστη σχέση ισοδυναμίας που σέβεται τη δομή των διεργασιών, γενικεύει την α-ισοδυναμία και επιπλέον:

1. κάνει τους τελεστές $\mid$ και $+$ να έχουν ουδέτερο στοιχείο το $\mathbf{0}$ και να ικανοποιούν την προσεταιριστική και την αντιμεταθετική ιδιότητα (αντιμεταθετικά μονοειδή)
2. $(\nu\, x)\mathbf{0} \equiv \mathbf{0}$
3. $(\nu\, y)(\nu\, x)P \equiv (\nu\, x)(\nu\, y)P$
4. αν $z \notin \{ x, y \}$, τότε $(\nu\, z)\, [x = y]\, P \equiv [x = y]\, (\nu\, z)P$
5. αν $z \notin \{ x, y \}$, τότε $(\nu\, z)\, [x \neq y]\, P \equiv [x \neq y]\, (\nu\, z)P$
6. αν $x \notin \mathrm{fn}(Q)$, τότε $(\nu\, x)(P \mid Q) \equiv (\nu\, x)P \mid Q$
7. αν $x \notin \mathrm{fn}(Q)$, τότε $(\nu\, x)(P + Q) \equiv (\nu\, x)P + Q$

Μεταβάσεις με ετικέτες

Ορίζουμε τις ετικέτες ως εξής:

Όρος	Διαίσθηση
τ	η σιωπηρή (silent) ή εσωτερική (internal) μετάβαση· δεν υπάρχει αλληλεπίδραση με το περιβάλλον
$x(y)$	λήψη του y μέσω του x
$\overline{x}\langle y\rangle$	αποστολή του y μέσω του x
$(\nu\, y)\overline{x}\,\langle y\rangle$	αποστολή του δεσμευμένου y μέσω του x

Στις ετικέτες $x(y)$ και $(\nu\, y)\overline{x}\,\langle y\rangle$, το y είναι δεσμευμένο.
$\mathrm{bn}(l)$: τα ονόματα που εμφανίζονται δεσμευμένα στην l
$\mathrm{n}(l)$: όλα τα ονόματα που εμφανίζονται στην l

Μπορούμε τώρα να ορίσουμε τη σημασιολογία με βάση τις σχέσεις μετάβασης $P \xrightarrow{l} Q$.

Κανόνες μετάβασης I

$$x(y).P \xrightarrow{x(y)} P \qquad \text{(In)}$$

$$\overline{x}\langle y\rangle.P \xrightarrow{\overline{x}\langle y\rangle} P \qquad \text{(Out)}$$

$$\frac{P \equiv P' \qquad P \xrightarrow{l} Q \qquad Q \equiv Q'}{P' \xrightarrow{l} Q'} \qquad \text{(Struct)}$$

$$\frac{P \xrightarrow{l} P'}{P + Q \xrightarrow{l} P'} \qquad \text{(Sum)}$$

$$\frac{P \xrightarrow{l} P'}{[x = x]\, P \xrightarrow{l} P'} \qquad \text{(Match)}$$

$$\frac{P \xrightarrow{l} P' \qquad x \neq y}{[x \neq y]\, P \xrightarrow{l} P'} \qquad \text{(Mismatch)}$$

Κανόνες μετάβασης II

$$\frac{P \xrightarrow{l} P'}{!P \xrightarrow{l} P' \mid !P} \quad \text{(Rep)}$$

$$\frac{P \xrightarrow{l} P' \qquad \mathrm{bn}(l) \cap \mathrm{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{l} P' \mid Q} \quad \text{(Par)}$$

$$\frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}\langle z \rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'} \quad \text{(Com)}$$

$$\frac{P \xrightarrow{l} P' \qquad x \notin \mathrm{n}(l)}{(\nu\, x)P \xrightarrow{l} (\nu\, x)P'} \quad \text{(Res)}$$

$$\frac{P \xrightarrow{\overline{a}\langle x \rangle} P' \qquad a \neq x}{(\nu\, x)P \xrightarrow{(\nu\, x)\overline{a}\langle x \rangle} P'} \quad \text{(Open)}$$

Ίχνη εκτέλεσης (Traces)

Συμβολισμός

Αν το P μετά από μια σειρά σιωπηλών μεταβάσεων γίνεται Q, αν δηλαδή $P \xrightarrow{\tau} P_1 \xrightarrow{\tau} P_2 \xrightarrow{\tau} \ldots \xrightarrow{\tau} Q$, τότε γράφουμε $P \Rightarrow Q$.

Αν $l \neq \tau$ και $P \Rightarrow P' \xrightarrow{l} Q' \Rightarrow Q$, γράφουμε $P \stackrel{l}{\Rightarrow} Q$.

Τέλος, γράφουμε $P \stackrel{\hat{l}}{\Rightarrow} Q$ στη θέση του $\begin{cases} P \Rightarrow Q & \text{αν } l = \tau \\ P \stackrel{l}{\Rightarrow} Q & \text{αλλιώς} \end{cases}$

Ορισμός

Αν $P \stackrel{l^n}{\Rightarrow} Q$, το l^n καλείται *ίχνος εκτέλεσης* του P.

Ενότητα 4

Παραδείγματα

Παράδειγμα: Αποστολή δεσμευμένου ονόματος

Παράδειγμα

$x(y).P \mid (\nu\, z)\overline{x}\langle z\rangle.Q$, με $z \notin \mathrm{fn}(P)$

$$\dfrac{\dfrac{z \notin \mathrm{n}(\tau) \qquad \dfrac{x(y).P \xrightarrow{x(y)} P \qquad \overline{x}\langle z\rangle.Q \xrightarrow{\overline{x}\langle z\rangle} Q}{x(y).P \mid \overline{x}\langle z\rangle.Q \xrightarrow{\tau} P\{z/y\} \mid Q}\,(\text{Com})}{(\nu\, z)(x(y).P \mid \overline{x}\langle z\rangle.Q) \xrightarrow{\tau} (\nu\, z)(P\{z/y\} \mid Q)}\,(\text{Res}) \qquad (*)$$

$$\dfrac{x(y).P \mid (\nu\, z)\overline{x}\langle z\rangle.Q \equiv (\nu\, z)(x(y).P \mid \overline{x}\langle z\rangle.Q) \qquad (*)}{x(y).P \mid (\nu\, z)\overline{x}\langle z\rangle.Q \xrightarrow{\tau} (\nu\, z)(P\{z/y\} \mid Q)}\,(\text{Struct})$$

Υπενθύμιση:

$$(\text{Com})\ \dfrac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'} \qquad \dfrac{P \xrightarrow{l} P' \qquad x \notin \mathrm{n}(l)}{(\nu\, x)P \xrightarrow{l} (\nu\, x)P'}\,(\text{Res})$$

Παράδειγμα: Αποστολή δεσμευμένου ονόματος

Παράδειγμα

$x(y).P \mid (\nu\, z)\overline{x}\langle z\rangle.Q$, με $z \notin \mathrm{fn}(P)$

Παρατήρηση

$(\nu\, z)\overline{x}\langle z\rangle.Q \xrightarrow{(\nu\, z)\overline{x}\langle z\rangle} Q$

Υπενθύμιση:

$$\text{(Open)}\ \frac{P \xrightarrow{\overline{a}\langle x\rangle} P' \qquad a \neq x}{(\nu\, x)P \xrightarrow{(\nu\, x)\overline{a}\langle x\rangle} P'}$$

Παράδειγμα: Ιδιωτικά κανάλια

Παράδειγμα

$x(a).P_{\text{Eve}} \mid (\nu\, x)(\overline{x}\langle z\rangle.P_{\text{Alice}} \mid x(y).P_{\text{Bob}})$
Υπάρχει τρόπος η Eve να μάθει το z μέσω του καναλιού x;

Υπενθύμιση:

$$\text{(Open)}\ \frac{P \xrightarrow{\overline{a}\langle x\rangle} P' \qquad a \neq x}{(\nu\, x)P \xrightarrow{(\nu\, x)\overline{a}\langle x\rangle} P'} \qquad \frac{P \xrightarrow{l} P' \qquad x \notin \mathrm{n}(l)}{(\nu\, x)P \xrightarrow{l} (\nu\, x)P'}\ \text{(Res)}$$

$$P \mid (\nu\, x)Q \equiv (\nu\, x)(P \mid Q),\ \text{αν}\ x \notin \mathrm{fn}(P)$$

Παράδειγμα: Ιδιωτικά κανάλια

Παράδειγμα

$x(a).P_{\text{Eve}} \mid (\nu\, x)(\overline{x}\langle z\rangle.P_{\text{Alice}} \mid x(y).P_{\text{Bob}})$
Υπάρχει τρόπος η Eve να μάθει το z μέσω του καναλιού x;

Όχι, διότι για να εφαρμόσουμε τη διαδικασία του προηγούμενου παραδείγματος (δηλαδή τον κανόνα (Struct)), θα έπρεπε $x \notin \text{fn}(x(a).P_{\text{Eve}})$.

Παρατήρηση

$(\nu\, x)\overline{x}\langle z\rangle.P_{\text{Alice}} \xrightarrow{(\nu\, z)\overline{x}\langle z\rangle} P_{\text{Alice}}$

Υπενθύμιση:

$$\text{(Open)}\ \frac{P \xrightarrow{\overline{a}\langle x\rangle} P' \qquad a \neq x}{(\nu\, x)P \xrightarrow{(\nu\, x)\overline{a}\langle x\rangle} P'} \qquad \frac{P \xrightarrow{l} P' \qquad x \notin \text{n}(l)}{(\nu\, x)P \xrightarrow{l} (\nu\, x)P'}\ \text{(Res)}$$

$$P \mid (\nu\, x)Q \equiv (\nu\, x)(P \mid Q),\ \text{αν}\ x \notin \text{fn}(P)$$

Παράδειγμα: Μη ντετερμινισμός I

Παράδειγμα

$Q = x(y).P \mid \overline{x}\langle a\rangle.\mathbf{0} \mid \overline{x}\langle b\rangle.\mathbf{0}$

$$\dfrac{\dfrac{x(y).P \xrightarrow{x(y)} P \qquad \overline{x}\langle a\rangle.\mathbf{0} \xrightarrow{\overline{x}\langle a\rangle} \mathbf{0}}{x(y).P \mid \overline{x}\langle a\rangle.\mathbf{0} \xrightarrow{\tau} P\{a/y\} \mid \mathbf{0}}\,(\text{Com}) \qquad \mathrm{bn}(\tau) \cap \{x, b\} = \emptyset}{x(y).P \mid \overline{x}\langle a\rangle.\mathbf{0} \mid \overline{x}\langle b\rangle.\mathbf{0} \xrightarrow{\tau} P\{a/y\} \mid \mathbf{0} \mid \overline{x}\langle b\rangle.\mathbf{0}}\,(\text{Par})$$

Επομένως, $Q \xrightarrow{\tau} P\{a/y\} \mid \overline{x}\langle b\rangle.\mathbf{0}$, αλλά τελείως όμοια ισχύει και η μετάβαση $Q \xrightarrow{\tau} P\{b/y\} \mid \overline{x}\langle a\rangle.\mathbf{0}$. Η επιλογή ανάμεσα στις δύο είναι μη ντετερμινιστική.

Υπενθύμιση:

$$(\text{Com})\ \frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'} \qquad \frac{P \xrightarrow{l} P' \qquad \mathrm{bn}(l) \cap \mathrm{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{l} P' \mid Q}\ (\text{Par})$$

Παράδειγμα: Μη ντετερμινισμός II

Παράδειγμα

$Q = (x(y).P_1 + x(y).P_2) \mid \overline{x}\langle a\rangle.\mathbf{0}$

$$\dfrac{\dfrac{x(y).P_1 \xrightarrow{x(y)} P_1}{(x(y).P_1 + x(y).P_2) \xrightarrow{x(y)} P_1}\,(\text{Sum}) \qquad \overline{x}\langle a\rangle.\mathbf{0} \xrightarrow{\overline{x}\langle a\rangle} \mathbf{0}}{(x(y).P_1 + x(y).P_2) \mid \overline{x}\langle a\rangle.\mathbf{0} \xrightarrow{\tau} P_1\{a/y\} \mid \mathbf{0}}\,(\text{Com})$$

Επομένως, $Q \xrightarrow{\tau} P_1\{a/y\}$, αλλά και (όμοια) $Q \xrightarrow{\tau} P_2\{a/y\}$. Η επιλογή ανάμεσα στα δύο είναι και πάλι μη ντετερμινιστική.

Υπενθύμιση:

$$(\text{Com})\ \frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'} \qquad \frac{P \xrightarrow{l} P'}{P + Q \xrightarrow{l} P'}\,(\text{Sum})$$

Παράδειγμα: Επαναλαμβανόμενες διεργασίες

Παράδειγμα

$!(x(y).P) \mid \overline{x}\langle z\rangle.\mathbf{0} \mid \overline{x}\langle b\rangle.\mathbf{0}$

$$\text{(Rep)}\ \frac{x(y).P \xrightarrow{x(y)} P}{\text{(Com)}\ \dfrac{!(x(y).P) \xrightarrow{x(y)} P \mid !(x(y).P) \qquad \overline{x}\langle z\rangle.\mathbf{0} \xrightarrow{\overline{x}\langle z\rangle} \mathbf{0}}{\text{(Par)}\ \dfrac{!(x(y).P) \mid \overline{x}\langle z\rangle.\mathbf{0} \xrightarrow{\tau} P\{z/y\} \mid !(x(y).P) \qquad \mathrm{bn}(\tau) = \emptyset}{!(x(y).P) \mid \overline{x}\langle z\rangle.\mathbf{0} \mid \overline{x}\langle b\rangle.\mathbf{0} \xrightarrow{\tau} !(x(y).P) \mid P\{z/y\} \mid \overline{x}\langle b\rangle.\mathbf{0}}}$$

Όμοια, παίρνουμε και

$$!(x(y).P) \mid P\{z/y\} \mid \overline{x}\langle b\rangle.\mathbf{0} \xrightarrow{\tau} !(x(y).P) \mid P\{z/y\} \mid P\{b/y\}$$

Υπενθύμιση:

$$\text{(Rep)}\ \frac{P \xrightarrow{l} P'}{!P \xrightarrow{l} P' \mid !P} \qquad \frac{P \xrightarrow{l} P' \qquad \mathrm{bn}(l) \cap \mathrm{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{l} P' \mid Q}\ \text{(Par)}$$

Ενότητα 5

Παραλλαγές

Polyadic π-calculus

Σε όσα είδαμε μέχρι τώρα, κάθε κανάλι μπορεί να μεταφέρει ακριβώς ένα όνομα τη φορά (monadic π-calculus).

Ερώτηση

Πώς μπορούν να γενικευτούν τα $x(y).P$ και $\overline{x}\langle y\rangle.P$ σε $x(y_1,\ldots,y_n).P$ (με τα y_i διαφορετικά ανά δύο) και $\overline{x}\langle y_1,\ldots,y_n\rangle.P$ αντίστοιχα;

- Με προσομοίωση, χρησιμοποιώντας ένα επιπλέον ιδιωτικό κανάλι όπου θα αποστέλλονται ένα-ένα τα y_i.

$$\overline{x}\langle y_1,\ldots,y_n\rangle.P \overset{\text{op}}{=} (\nu\, u)\overline{x}\langle u\rangle.\overline{u}\langle y_1\rangle.\cdots.\overline{u}\langle y_n\rangle.P, u \notin \text{fn}(P)$$

$$x(y_1,\ldots,y_n).P \overset{\text{op}}{=} x(v).v(y_1).\cdots.v(y_n).P, v \notin \text{fn}(P)$$

- Επεκτείνοντας τη γλώσσα (polyadic π-calculus).

Εμφανίζεται η ανάγκη να ελέγχουμε (πχ με κάποιο σύστημα τύπων) αν προκύπτουν επικοινωνίες όπου οι δύο πλευρές έχουν διαφορετικό αριθμό ορισμάτων.

Ο κανόνας (Close)

Στο παράδειγμα της αποστολής δεσμευμένου ονόματος, $x(y).\mathbf{0} \mid (\nu\, z)\overline{x}\langle z\rangle.\mathbf{0}$, η απόδειξη της ύπαρξης μετάβασης χρησιμοποιεί με ουσιαστικό τρόπο τη δομική ομοιότητα για να επεκτείνει τη δέσμευση του z. Θα μπορούσαμε να αφαιρέσουμε τον αντίστοιχο κανόνα από τον ορισμό της δομικής ομοιότητας και να εισάγουμε στις μεταβάσεις τον κανόνα:

$$\frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{(\nu\, z)\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} (\nu\, z)(P'\,\{z/y\} \mid Q')} \text{(Close)}$$

Τότε, η μετάβαση του παραδείγματος προκύπτει ως εξής:

$$\frac{x(y).\mathbf{0} \xrightarrow{x(y)} \mathbf{0} \qquad \dfrac{\overline{x}\langle z\rangle.\mathbf{0} \xrightarrow{\overline{x}\langle z\rangle} \mathbf{0} \qquad x \neq z}{(\nu\, z)\overline{x}\langle z\rangle.\mathbf{0} \xrightarrow{(\nu\, z)\overline{x}\langle z\rangle} \mathbf{0}}\,\text{(Open)}}{x(y).\mathbf{0} \mid (\nu\, z)\overline{x}\langle z\rangle.\mathbf{0} \xrightarrow{\tau} (\nu\, z)\mathbf{0} \mid \mathbf{0}}\,\text{(Close)}$$

Για τους κανόνες (Rep) και (Match)

Ο κανόνας μετάβασης

$$\frac{P \xrightarrow{l} P'}{!P \xrightarrow{l} P' \mid !P} \qquad \text{(Rep)}$$

μπορεί να αντικατασταθεί από τον κανόνα δομικής ισοδυναμίας

$$!P \equiv !P \mid P.$$

Επίσης, ο κανόνας μετάβασης

$$\frac{P \xrightarrow{l} P'}{[x = x]\, P \xrightarrow{l} P'} \qquad \text{(Match)}$$

μπορεί να αντικατασταθεί από τον κανόνα δομικής ισοδυναμίας

$$[x = x]\, P \equiv P.$$

Πρότερη και ύστερη σημασιολογία[1] I

Ας παρατηρήσουμε τους κανόνες για τη λήψη μηνυμάτων.

$$x(y).P \xrightarrow{x(y)} P \qquad \text{(In)} \qquad \frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'} \text{(Com)}$$

Η πραγματική λήψη του μηνύματος γίνεται όταν υπάρξει κάποια συγκεκριμένη αποστολή· στην ετικέτα βρίσκεται μία μεταβλητή που θα αντικατασταθεί από την τιμή που θα ληφθεί.
Η σημασιολογία αυτού του είδους καλείται ύστερη (late).

[1] Οι όροι «πρότερη» και «ύστερη» είναι μη δόκιμες αποδόσεις των αντίστοιχων αγγλικών όρων.

Πρότερη και ύστερη σημασιολογία[1] I

Ας παρατηρήσουμε τους κανόνες για τη λήψη μηνυμάτων.

$$x(y).P \xrightarrow{x(y)} P \qquad \text{(In)} \qquad \frac{P \xrightarrow{x(y)} P' \qquad Q \xrightarrow{\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{z/y\} \mid Q'} \text{(Com)}$$

Η πραγματική λήψη του μηνύματος γίνεται όταν υπάρξει κάποια συγκεκριμένη αποστολή· στην ετικέτα βρίσκεται μία μεταβλητή που θα αντικατασταθεί από την τιμή που θα ληφθεί.
Η σημασιολογία αυτού του είδους καλείται ύστερη (late).

Θα μπορούσαμε να χρησιμοποιούμε τους εξής κανόνες:

$$x(y).P \xrightarrow{x(z)} P\{z/y\} \text{ (E-In)} \qquad \frac{P \xrightarrow{x(z)} P' \qquad Q \xrightarrow{\overline{x}\langle z\rangle} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \text{(E-Com)}$$

Εδώ, η λήψη του μηνύματος γίνεται κατ' ευθείαν· στην ετικέτα βρίσκεται η τιμή που λήφθηκε.
Η σημασιολογία αυτού του είδους καλείται πρότερη (early)· θα συμβολίζουμε τις μεταβάσεις της με $\xrightarrow{l}_{\mathrm{E}}$.

[1] Οι όροι «πρότερη» και «ύστερη» είναι μη δόκιμες αποδόσεις των αντίστοιχων αγγλικών όρων.

Πρότερη και ύστερη σημασιολογία II

Πρόταση

1. *Αν το* l *δεν είναι της μορφής* $x(y)$, *τότε* $P \xrightarrow{l} Q \Leftrightarrow P \xrightarrow{l}_E Q$.
2. $P \xrightarrow{x(y)}_E Q \Leftrightarrow \exists P', w : P \xrightarrow{x(w)} P' \wedge Q \equiv P'\{y/w\}$

Πόρισμα

$P \xrightarrow{x(w)} P' \Rightarrow \forall u : P \xrightarrow{x(u)}_{\mathrm{E}} P'\{u/w\}$

Σημασιολογία με αναγωγές I

Ας θεωρήσουμε το υποσύνολο του π-λογισμού όπου κάθε φορά που χρησιμοποιούμε τον τελεστή $+$, οι τελεστέοι είναι της μορφής $\alpha.P$, όπου α_i είναι της μορφής $x(y)$ ή $\overline{x}\langle y\rangle$. Για αυτή τη γλώσσα, μπορούμε να ορίσουμε μια απλή σημασιολογία με αναγωγές που σχετίζεται στενά με τη σημασιολογία των μεταβάσεων.

Επεκτείνουμε τη σχέση της δομικής ομοιότητας με το $!P \equiv !P \mid P$ και τους ακόλουθους κανόνες:

- Αν το $[x = y]\, P$ δεν βρίσκεται στην εμβέλεια ενός input ή ενός output action, τότε $[x = y]\, P \equiv \begin{cases} P & \text{αν } x = y \\ \mathbf{0} & \text{αν } x \neq y \end{cases}$.
- Αν το $[x \neq y]\, P$ δεν βρίσκεται στην εμβέλεια ενός input ή ενός output action, τότε $[x \neq y]\, P \equiv \begin{cases} P & \text{αν } x \neq y \\ \mathbf{0} & \text{αν } x = y \end{cases}$.

Σημασιολογία με αναγωγές II

Έπειτα, ορίζουμε τους παρακάτω κανόνες αναγωγής:

$$(\ldots + x(y).P) \mid (\ldots + \overline{x}\langle z\rangle.Q) \to P\{z/y\} \mid Q \qquad \text{(R-Com)}$$

$$\frac{P \to P'}{P \mid Q \to P' \mid Q} \qquad \text{(R-Par)}$$

$$\frac{P \to P'}{(\nu\, x)P \to (\nu\, x)P'} \qquad \text{(R-Res)}$$

$$\frac{P \equiv P' \qquad P \to Q \qquad Q \equiv Q'}{P' \to Q'} \qquad \text{(R-Struct)}$$

Πρόταση

$P \to Q$ στη σημασιολογία με αναγωγές αν και μόνο αν $P \xrightarrow{\tau} Q$ στη σημασιολογία των μεταβάσεων

Ενότητα 6

Αλληλοπροσομοιωσιμότητα

Η χρησιμότητα της αλληλοπροσομοιωσιμότητας[2] (bisimilarity)

Μας ενδιαφέρει πότε δύο διεργασίες έχουν την ίδια συμπεριφορά βάσει της σημασιολογίας που έχουμε ορίσει.

Ιδέα

Δύο διεργασίες έχουν την ίδια συμπεριφορά αν κάθε μετάβαση της μίας αντικατοπτρίζεται («προσομοιώνεται») από την άλλη.

- Ανάλογα με το αν τα βήματα της «προσομοίωσης» γίνονται «ένα προς ένα» ή «ένα προς πολλά», έχουμε ισχυρή (strong) και ασθενή (weak) αλληλοπροσομοίωση.
- Η πρότερη και η ύστερη σημασιολογία οδηγούν σε αντίστοιχα είδη αλληλοπροσομοίωσης.
- Άλλα είδη είναι η barbed και η open.

Για κάθε αλληλοπροσομοίωση, μας ενδιαφέρει επίσης το μέγιστο υποσύνολό του που αποτελεί σχέση ομοιότητας.

[2] Η απόδοση του όρου στα ελληνικά είναι μη δόκιμη.

Ισχυρή αλληλοπροσομοίωση: Ορισμοί

Ορισμός (Ισχυρή πρότερη αλληλοπροσομοίωση)

Μια συμμετρική δυαδική σχέση $\mathcal{R}$ είναι ισχυρή πρότερη αλληλοπροσομοίωση αν για κάθε $P\mathcal{R}Q$ με $P \xrightarrow{l}_{\mathrm{E}} P'$ όπου $\mathrm{bn}(l) \cap \mathrm{fn}(P, Q) = \emptyset$ υπάρχει Q' ώστε $Q \xrightarrow{l}_{\mathrm{E}} Q'$ και $P'\mathcal{R}Q'$.
Αν δύο διεργασίες P, Q σχετίζονται με μια τέτοια $\mathcal{R}$, γράφουμε $P \dot{\sim}_{\mathrm{E}} Q$.

Ορισμός (Ισχυρή ύστερη αλληλοπροσομοίωση)

Μια συμμετρική δυαδική σχέση $\mathcal{R}$ είναι ισχυρή ύστερη αλληλοπροσομοίωση αν για κάθε $P\mathcal{R}Q$ με $P \xrightarrow{l} P'$ όπου $\mathrm{bn}(l) \cap \mathrm{fn}(P, Q) = \emptyset$ υπάρχει Q' ώστε $Q \xrightarrow{l} Q'$ και

1. αν το l δεν είναι της μορφής $x(y)$, τότε $P'\mathcal{R}Q'$,
2. αν το l είναι της μορφής $x(y)$, τότε $\forall u : P'\{u/y\}\,\mathcal{R}\,Q'\{u/y\}$.

Αν δύο διεργασίες P, Q σχετίζονται με μια τέτοια $\mathcal{R}$, γράφουμε $P \dot{\sim} Q$.

Ισχυρή αλληλοπροσομοίωση: Ιδιότητες

Παραδείγματα

1. Αν $a \neq b$, τότε $a(c).\mathbf{0} \mid \overline{b}\langle d\rangle.\mathbf{0} \dot{\sim} a(c).\overline{b}\langle d\rangle.\mathbf{0} + \overline{b}\langle d\rangle.a(c).\mathbf{0}$.
2. $x(a).(a(c).\mathbf{0} \mid \overline{b}\langle d\rangle.\mathbf{0}) \dot{\not\sim}_{\mathrm{E}} x(a).(a(c).\overline{b}\langle d\rangle.\mathbf{0} + \overline{b}\langle d\rangle.a(c).\mathbf{0})$
3. $a(x).P + a(x).\mathbf{0} \dot{\not\sim} a(x).P + a(x).\mathbf{0} + [x = u]\,P$
4. $a(x).P + a(x).\mathbf{0} \dot{\sim}_{\mathrm{E}} a(x).P + a(x).\mathbf{0} + [x = u]\,P$

Πρόταση

Οι $\dot{\sim}$, $\dot{\sim}_E$ είναι σχέσεις ισοδυναμίας.
Επιπλέον, σέβονται όλους τους τελεστές πλην της εισόδου.

Πρόταση

$\equiv \subset \dot{\sim} \subset \dot{\sim}_E$

Ισχυρή ομοιότητα

Ορισμός (Ισχυρή ύστερη ομοιότητα)

Η ισχυρή ύστερη ομοιότητα $\sim$ είναι η μεγαλύτερη σχέση ομοιότητας που περιέχεται στην $\dot{\sim}$.

Πρόταση

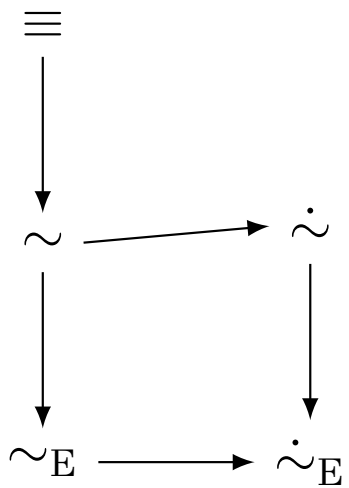
$P \sim Q$ αν και μόνο αν $P\sigma \dot{\sim} Q\sigma$ για κάθε αντικατάσταση ονομάτων σ

Τα αντίστοιχα ισχύουν για την ισχυρή πρότερη ομοιότητα.

Παρατήρηση

$\sim \neq \sim_{\mathrm{E}}$

Ανακεφαλαίωση



Σχήμα: Οι συμπεριλήψεις ανάμεσα στις σχέσεις που είδαμε· όλες είναι γνήσιες.

Ενότητα 7

Πληρότητα

Η πληρότητα του π-λογισμού

Ο λ-λογισμός (με lazy evaluation) μπορεί να μοντελοποιηθεί από το υποσύνολο του π-λογισμού που περιγράφεται από τη γραμματική

$$P \quad = \quad \mathbf{0} \quad | \quad x(y).P \quad | \quad \overline{x}\langle y\rangle.P \quad | \quad (\nu\, x)P \quad | \quad P \,|\, P \quad | \quad !P$$

Επομένως, ο π-λογισμός είναι Turing-complete.

Ενότητα 8

Αναφορές


Joachim Parrow. «An Introduction to the π-calculus». Στο: *Handbook of Process Algebra*. Elsevier, 2001. Κεφ. 8, σσ. 479–543.


Wikipedia. *π-calculus*. 2017. URL: https://en.wikipedia.org/w/index.php?title=%C3%8E%C2%A0-calculus&oldid=773689778 (επίσκεψη 25/04/2017).


Robin Milner, Joachim Parrow και David Walker. «A calculus of mobile processes, I». Στο: *Information and Computation* 100 (1992), σσ. 1–40.


Robin Milner, Joachim Parrow και David Walker. «A calculus of mobile processes, II». Στο: *Information and Computation* 100 (1992), σσ. 41–77.


Robin Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, 1999.


Robin Milner. «Functions as processes». Στο: *Mathematical Structures in Computer Science* 2 (1992), σσ. 119–141.

Σας ευχαριστώ