Koncepcja współbieżności w CCS

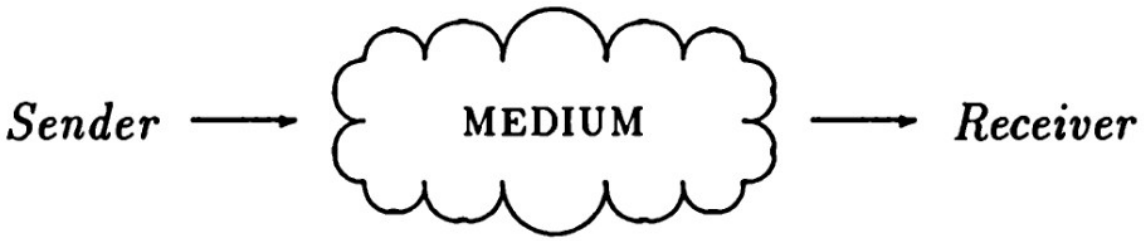
Teoria współbieżności

Tomasz Szewczyk

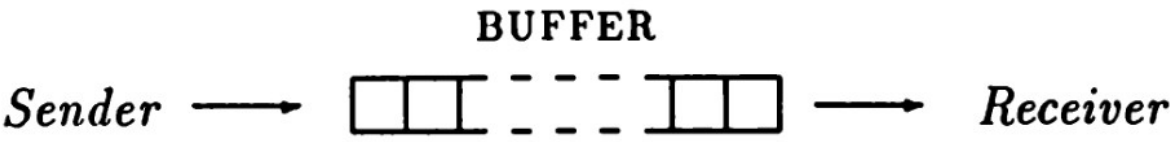
Współbieżność pozostaje jednym z największych wyzwań z którymi mierzy się teoria informatyki, zarówno w teorii jak i w praktyce. Współczesne komputery wyposażane są w coraz większą liczbę procesorów, a rosnący cyfrowy świat sprawia, że wymagania stają się coraz bardziej wygórowane. Systemy muszą być szybkie, niezawodne oraz skalowalne. W dzisiejszych czasach nie da się już zaspokoić wymagań bez zastosowania zaawansowanych technik współbieżności. Tempo postępu w tej dziedzinie technologii informacyjnych wprowadza zamęt i niepewność. Na szczęście istnieją formalne sposoby opisu tych problemów, których zastosowanie daje pewność i porządek.

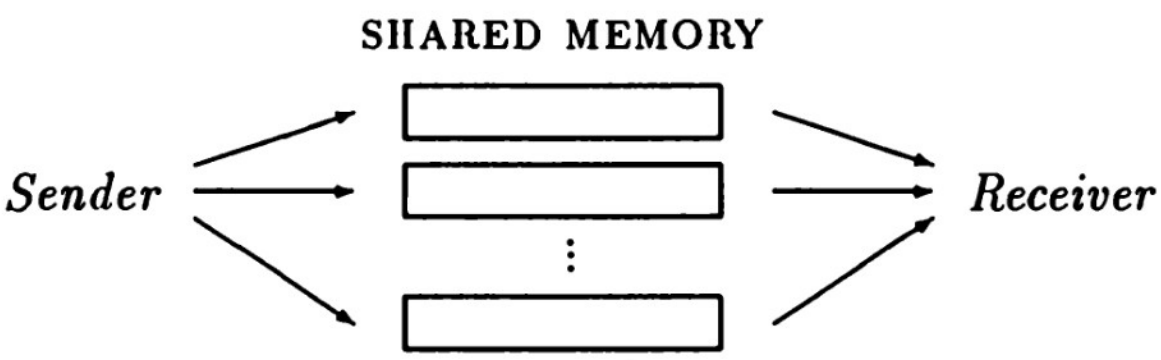
Na przestrzenie lat powstało wiele technik opisu działania procesów współbieżnych. Do najpopularniejszych należą: Communicating Sequential Processes w skrócie CSP C.A.R. Hoare'a, rachunek pi oraz Calculus of Communicating Systems w skrócie CSS, który został wprowadzony przez Robina Milnera około roku 1980. W tej pracy skupię na tym ostatnim.

CCS modeluje system współbieżny jako sieć agentów, komunikatów które są między nimi wymieniane, akcji oraz zachowań. Agent to uogólnienie jednostki która wykonuje operację, na przykład procesu. CCS abstachuje od szczegółów implementacyjnych agenta, ważna jest jedynie komunikacja z nim. Model współbieżności CCS opiera się na komunikacji.

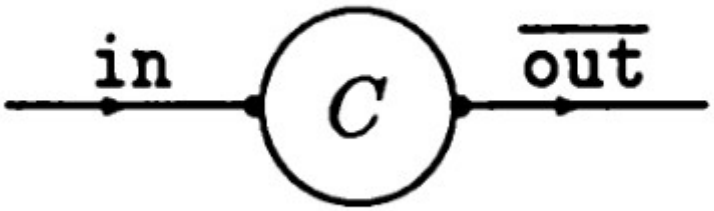


Najprostrzym przykładem komunikacji jest sytuacji w której dwóch agentów: Nadawca oraz Odbiorca komunikują się przez jakieś medium. Tym medium może być na przykład pamięć współdzielona komputera lub bufor w pamięci.





Podstawowy akt komunikacji dwóch agentów nazywamy handshake, czyli uścisk dłoni. Handshake jest graficznie reprezentowany jako strzałka. Każda komunikacja jest synchroniczna i niebuforowana. Każdy agent ma porty wejściowe oraz porty wyjściowe na których agent może akceptować lub inicjować komunikację. Porty wyjściowe zapisywane są z kreską ponad nazwą.



Zachowania agena definiowane są specjalnym językiem opisującym akcje oraz zmiany stanu agenta. Poniższy przykład pokazuje opis agenta C, który przyjmuje wartość x z kanału in, przechodzi do stanu C', a następnie wysyła niezmienioną wartość x na kanał out, i wraca do stanu C. Zachowania opisane są jako rekursywne równania.

$$C \stackrel{\text{def}}{=} \mathtt{in}(x).C'(x)$$

$$C'(x) \stackrel{\text{def}}{=} \overline{\mathtt{out}}(x).C$$

SCC używa pięciu podstawowych kombinatorów do stworzenia algebry opisującej agentów i ich zachowania. Są to:

- **prefix .** - służy do łączenia operacji w sekwencje. a.P oznacza, wykonaj a i kontynuuj według P, jest to jedyny sposób szeregowania zdarzeń w języku,
- **suma** + - informuje, że agent ma dwa równoważne zachowania. Sumę P + Q można przeczytać jako „taki jak P lub taki jak Q"
- **kompozycja |** - opisuje równoległość w systemie,
- **zakaz** \ - ogranicza ilość kanałów,
- **przemianowanie []** - służy do zmieniania nazwy kanałów

Ponadto w języku zdenifiowane są również operacje warunkowe: **If** b **then** E, skoki **skip**, bloki **begin** … **end**.

SCC definiuje pięc zasad tranzycji: Sum, Com, Res, Rel oraz Con. Zasady opisują zachowanie systemu do którego zostały zaaplikowane kombinatory. Na przykład zasada sumy mówi, że jeżeli jakiś składnik ma pewną akcję to suma tego składnika z innym również będzie posiadała tą akcję. Formalny opis tych zasad znajduje się na rysunku.

$$\textbf{Act}\ \frac{}{\alpha.E \xrightarrow{\alpha} E} \qquad \textbf{Sum}_j\ \frac{E_j \xrightarrow{\alpha} E'_j}{\sum_{i\in I} E_i \xrightarrow{\alpha} E'_j}\ (j \in I)$$

$$\textbf{Com}_1\ \frac{E \xrightarrow{\alpha} E'}{E|F \xrightarrow{\alpha} E'|F} \qquad \textbf{Com}_2\ \frac{F \xrightarrow{\alpha} F'}{E|F \xrightarrow{\alpha} E|F'}$$

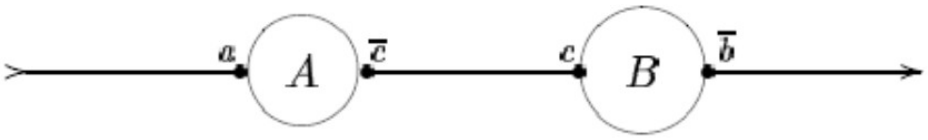
$$\textbf{Com}_3\ \frac{E \xrightarrow{l} E' \quad F \xrightarrow{\bar{l}} F'}{E|F \xrightarrow{\tau} E'|F'}$$

$$\textbf{Res}\ \frac{E \xrightarrow{\alpha} E'}{E\backslash L \xrightarrow{\alpha} E'\backslash L}\ (\alpha, \bar{\alpha} \notin L) \qquad \textbf{Rel}\ \frac{E \xrightarrow{\alpha} E'}{E[f] \xrightarrow{f(\alpha)} E'[f]}$$

$$\textbf{Con}\ \frac{P \xrightarrow{\alpha} P'}{A \xrightarrow{\alpha} P'}\ (A \overset{\text{def}}{=} P)$$

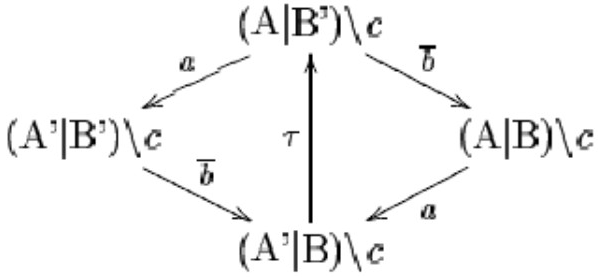
SCC definiuje kilka wyjątkowych obiektów, których zachowanie różni się od zachowania pozostałych. Są to proces ‘0’, tak zwany proces nieaktywny, który powoduje zakleszczenie gdy dotrze do niego komunikacja oraz zdarzenie τ, które oznacza „cichą komunikację”, czyli taką, której nie da się obserwować z zewnątrz.

Używając zaprezentowanych metod opisu zachowań można budować grafy tranzycji w celu przedstawienia możliwych przejść i zachowań systemu. Na przykład proces (A | B) \ c daje następujący graf.



$A \overset{\text{def}}{=} a.A'$ $\qquad$ $B \overset{\text{def}}{=} c.B'$

$A' \overset{\text{def}}{=} \bar{c}.A$ $\qquad$ $B' \overset{\text{def}}{=} \bar{b}.B$



Kolejnym ważnym pojęciem jest równoważność procesów. Aby zrozumieć to pojęcie warto posłużyć się intuicją. Aby wykazać równoważność A i B, gramy w grę. Graczami są Kowalski i Nowak, Kowalski dowodzi, że A i B są równoważne, a Nowak dowodzi, że się różnią. Obaj wykonują naprzemiennie ruchy. Najpierw Nowak wybiera system i wykonuje dowolne możliwe zdarzenie. Kowalski musi w tym czasie odpowiedzieć w drugim systemie tym samym zdarzeniem. Gracz, który nie ma więcej ruchów, przegrywa. Systemy są równoważne, jeśli Kowalski zawsze wygrywa, tzn. gdy ma strategię wygrywającą. W ten sposób dochodzimy do relacji równoważności ~, którą definiujemy nastepująco:

$P \sim Q$ gdy wtw.

$$\begin{array}{l} \forall \alpha \in \mathrm{Act}, \\ (i)\ \ P \xrightarrow{\alpha} P' \implies \exists Q' \text{ t.że } Q \xrightarrow{\alpha} Q' \text{ i } P' \sim Q' \\ (ii)\ \ Q \xrightarrow{\alpha} Q' \implies \exists P' \text{ t.że } P \xrightarrow{\alpha} P' \text{ i } P' \sim Q' \end{array} \qquad (20)$$

P i Q są równoważne wtedy i tylko wtedy gdy nie można ich odróżnić w jednym kroku eksperymentu oraz otrzymane w tym kroku systemy są równoważne. Koniecznym pojęciem jest również bisymulacja którą definiujemy w następujący sposób:

$$(i)\ \ P \xrightarrow{\alpha} P' \implies \exists Q' \ \ t.że \ \ Q \xrightarrow{\alpha} Q' \ \ i \ \ (P', Q') \in R$$
$$(ii)\ \ Q \xrightarrow{\alpha} Q' \implies \exists P' \ \ t.że \ \ P \xrightarrow{\alpha} P' \ \ i \ \ (P', Q') \in R.$$

Literatura:

1. Robin Milner: A Calculus of Communicating Systems 1980.
2. Robin Milner, Communication and Concurrency 1989.
3. Ali Ahmadzadeh asl, Calculus of Communicating Systems 2014.
4. Grzegorz Maj, CCS w CWB Niezawodność systemów współbieżnych i obiektowych 2009.