ISSN 2303-5366
http://www.bijeljinamc.com

БМЧ
Бијељински методички часопис
Вол. 5 (2018), 28-32
Кратко или претходно саопштење

ОЗНАЧЕНЕ ТРАНЗИЦИОНЕ СЕМАНТИКЕ

Slađana Mitrović[1]
Fakultet tehničkih nauka
Univerzitet u Novom Sadu

Sažetak: *Semantika je oblast koja se bavi strogim matematičkim proučavanjem značenja programskih jezika. Ona opisuje proces izvršavanja programa na nekom određenom programskom jeziku. Tranzicioni sistemi se koriste u različitim oblastima. Imaju veliku primenu u modelovanju procesnih računa, jer daju jedan od najuobičajenijih i najrazumljivijih modela. Označeni tranzicioni sistemi (LTS) predstavljaju osnovni model za softverske i hardverske sisteme.*

Ključne reči: *označeni tranzicioni sistemi (**LTS**), izvršavanje, put, atomičke propozicije, ne-determinizam*

Uvod

U teoriji programskih jezika, semantika je oblast koja se bavi strogim matematičkim proučavanjem značenja programskih jezika. Ona opisuje proces izvršavanja programa na nekom određenom programskom jeziku. Uloga semantike je da:

- programer može da razume kako se program izvršava pre njegovog pokretanja kao i šta mora da obezbedi prilikom kreiranja kompilatora;
- razumevanje karakteristika programskog jezika i njihovih interakcija; i
- dokazivanje svojstava odredjenog programskog jezika.

Tranzicioni sistemi se koriste u različitim oblastima. Imaju veliku primenu u modelovanju procesnih računa, jer daju jedan od najuobičajenijih i najrazumljivijih modela. Oni pružaju osnove operacionih semantika za Milnerov Calculus of Communicating Systems (CCS) i često su osnova za druge pristupe, kao što je Hoareov Communicating Sequential Processes (CSP)(Sampson,2012.).

Označeni tranzicioni sistemi (LTS)

Osnovni model za softverske (hardverske) sisteme je razmatranje da sistem prelazi iz jednog stanja u drugo u zavisnosti od akcije izvršene u okruženju ili od samog sistema. Tranzicioni sistem može biti vizuelizovan kao graf, na kome najviše tačke označavaju stanja a rubovi predstavljaju prelaze među njima. Rubovi mogu imati oznaku. Ova oznaka predstavlja akciju koja pokreće promenu stanja. Stanja mogu biti označena. Označavanje stanja određuje vrednosti nekih promenljivih u njemu. Pojednostavićemo te promenljive samo da budu logičkog tipa što znači da mogu imati vrednosti "tačno" ili "netačno".

Definicija 1: Označeni tranzicioni sistem (**LTS**) je struktura (S, Act,→, I, AP, L), gde je:

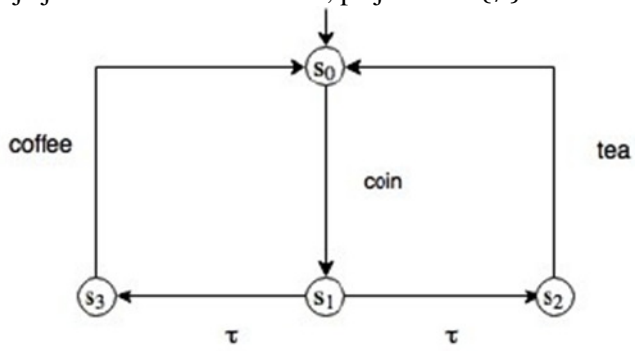
[1] e-mail: sladjanamitrovic2512@gmail.com

- S skup stanja,
- Act skup akcija (ne uključujući internu akciju τ),
- $\rightarrow \subseteq S \times AP \times S$ relacija prelaza,
- I $\subset$ S skup početnih stanja,
- AP skup atomičkih propozicija, i
- L : S $\rightarrow 2^{AP}$ funkcija označavanja.

$(\boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{s'}) \in \rightarrow$ **kraće ćemo pisati** $\boldsymbol{s \xrightarrow{\alpha} s'}$

Da bismo označili unutrašnju promenu čija nam oznaka nije bitna, koristićemo simbol .

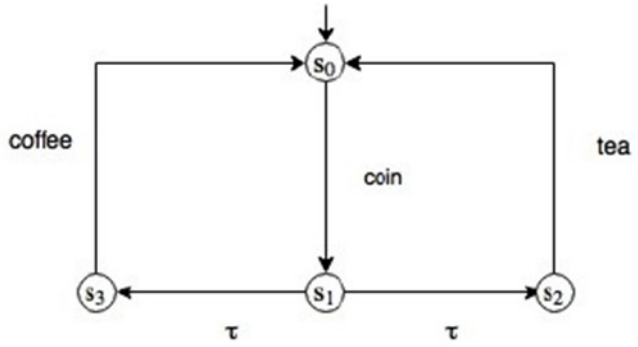
Na slici je prikazan primer **LTS** modelovanja jednostavnog aparata za piće. Skup vrednosti ovog **LTS** je $S = \{s_0, s_1, s_2, s_3\}$. Skup akcija je $Act = \{coin, coffee, tea\}$. Skup početnih stanja je $I = \{s_0\}$. Za sada atomičke propozicije još uvek nisu definisane, pa je $AP = \{\emptyset\}$.



Izvršavanja i putevi

LTS kreće od nekog početnog stanja. Ako tranzicioni sistem ima više od jednog početnog stanja, ono će biti izabrano ne-deterministički. Iz tog izabranog početnog stanja, **LTS** se razvija u skladu sa njegovom relacijom prelaza. Vrlo često, iz datog stanja će biti moguće izabrati više prelaza. U tom slučaju, ono se bira ne-deterministički.

Primer: Posmatrajmo sliku ispod. Iz početnog stanja moguć je samo jedan prelaz. Dakle, **LTS** će uvek prvo izvršiti *coin* akciju i dosegnuti stanje s_1. U s_1 imamo dva interna koraka. **LTS** će ne-deterministički izabati jedan od njih, a zatim primeniti ili *coffee* ili *tea* akciju.



Definicija 2: Izvršavanje je beskonačan niz stanja i akcija. Ima sledeću formu:

$$s_0 \alpha_0 s_1 \alpha_1 s_2 \alpha_2 \ldots$$

gde je $s_i \xrightarrow{\alpha_i} s_{i+1},\ \forall i \geq 0$.

Neke osobine izvršavanja:
• potencijalno beskonačna,
• počinju inicijalnim stanjem, i
• uvek se završavaju krajnjim stanjem (ili su beskonačna).

Primer : Izvršavanje **LTS**-a sa slike prethodnog primera će biti niz jednog ili oba od sledećih konačnih prefiksa:

1. $s_0\ coin\ s_1\ \tau\ s_2\ tea\ s_0$
2. $s_0\ coin\ s_1\ \tau\ s_3\ coffee\ s_0$.

Definicija 3: Put je projekcija izvršavanja. To je beskonačan niz stanja: $s_0 s_1 s_2$ ...

gde je: $\forall i > 0, s_i \in Post(s_{i-1})$.

Uvešćemo par notacija za puteve:

1. $\pi = s_0 s_1 s_2$ ... označava beskonačan put;
2. $\pi[i]$ označava i-ti elemenat puuta;
3. $\pi[..i] = s_0 s_1 s_{2\ldots} s_i$ označava konačan prefiks do pozicije i;
4. $\pi[i..] = s_i s_{i+1} s_{i+2}$ ... označava sufiks od pozicije i.

Primer : Putevi **LTS**-a na slici iz posmatranog primera će biti nizovi jednog ili oba od sledećih prefiksa:

1. $s_0 s_1 s_2 s_0$
2. $s_0 s_1 s_3 s_0$

Definicija 4: Neka je dato stanje s i akcija α, skup nasleđenih stanja od s nakon izvršavanja α je definisan kao:

$$Post(s, \alpha) = \left\{ s' \middle| s \xrightarrow{\alpha} s' \right\}.$$

Dakle, $Post(s, \alpha)$ sadrži sva stanja koja mogu biti dostignuta iz stanja s primenjujući akciju α.
Proširićemo notaciju na skup akcija:

$$Post(s) = \bigcup_{\alpha \in Act} Post(s, \alpha).$$

$Post(s)$ sadrži sva stanja koja mogu biti dostignuta iz stanja s primenjujući bilo koju akciju.

Na sličan način, definisaćemo skup prethodnih stanja:
Definicija 5: Neka je dato stanje s i akcija α, skup prethodnih stanja od s kroz akciju α je definisan kao:

$$Pre(s, \alpha) = \left\{ s' \middle| s' \xrightarrow{\alpha} s \right\}.$$

Dakle, $Pre(s, \alpha)$ sadrži sva stanja koja mogu dostići stanje s primenjujući akciju α.
Proširićemo notaciju na skup akcija:

$$Pre(s) = \bigcup_{\alpha \in Act} Pre(s, \alpha).$$

$Pre(s)$ sadrži sva stanja koja mogu dostići stanje s primenjujući bilo koju akciju.

Definicija 6: Skup ***dostignutih stanja*** sadrži sva stanja dostignuta iz inicijalnog stanja primenjujući bilo koju akciju.

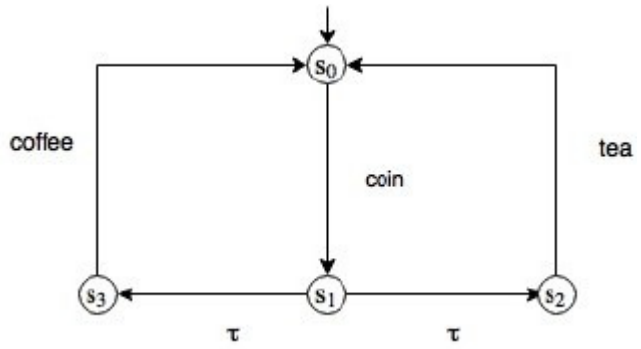
Definicija 7: (***Završno stanje***) Stanje s u tranzicionom sistemu T naziva se ***završno*** ako: $Post(s) = \emptyset$.

Završna stanja mogu se koristiti da izraze završetak računanja. Ona su takođe neka vrsta nepoželjnih zastoja. Ubuduće pretpostavićemo da su **LTS**-i bez završnih stanja. To znači da su putevi i izvršavanja uvek beskonačni.

Atomičke propozicije

Atomičke propozicije koristimo da predstavimo informacije o trenutnom stanju sistema. To je, na primer, stanje registra računara ili trenutna vrednost promenljive programa. Za potrebe verifikacije, atomičke propozicije mogu nam pomoći u formulisanju zahteva za **LTS**.

Primer : Posmatrajmo primer sa slike ispod. Zahtev o stanju ovog aparata bi bio: "Aparat bi uvek trebao primiti novčić pre nego što dostavi kafu". Možemo kreirati atomičku propoziciju *coinInserted* koja dobija vrednost "tačno" kada je novčić ubačen i postaje "netačno" kada je piće dostavljeno.



Iz ovog dobijamo sledeću funkciju označavanja:

- $L(s_1) = L(s_2) = L(s_3) = \{coinInserted\}$
- $L(s_0) = \emptyset$

Bitno je napomenuti da funkcija označavanja vraća samo propozicije koje su sadržane u datom stanju.

Ne-determinizam

LTS se naziva *deterministički* ako:

1. $|I| = 1$, tj. ako ima tačno jedno inicijalno stanje i
2. $\forall s \in S, \forall \alpha \in Act\ |Post(s, \alpha)| \leq 1$ tj. za sve akcije i stanja ima najviše jedno nasleđeno stanje.

U suprotnom **LTS** nazivamo *ne-deterministički*.

Zaključak

Označeni tranzicioni sistemi (**LTS**) predstavljaju osnovni model za softverske i hardverske sisteme. Izvršavanje tranzicionog sistema je menjanje niza stanja i akcija koje počinju u inicijalnom stanju. Putevi predstavljaju projekciju izvršavanja. Stanje je dostignuto ako postoji konačno izvršavanje

prefiksa počevši od inicijalnog stanja i završavajući sa tim stanjem. **LTS** je deterministički ako sadrži tačno jedno inicijalno stanje i ako iz jednog stanja i date akcije najviše jedno stanje može biti dostignuto. U suprotnom je nedeterministički.

Literatura

Parezanović N. (1992). *Metodički aspekti opisa semantike programskih jezika*.Beograd:Samostalno autorsko izdanje.

Sampson, A. (2012). *Programming Languages and Logics*. United States of America, New York.

Slonneger,K.,& Kurtz, B.L. (1995). *Formal Syntax and Semantics of Programming Languages*. United States of America: Addison-Wesley Publishing Company.

Setbi,R.(1989). Programming Languages. United States of America: Addison-Wesley Publishing Company

MARKED TRANSITION SEMANTICS

Summary: Semantics is an area that deals with a rigorous mathematical study of the meaning of programming languages. It describes the process of executing programs in a specific programming language. Transition systems are used in different areas. They have great application in modeling process calculations, because they give one of the most common and most intelligent models. Marked transition systems (LTSs) are the basic model for software and hardware systems.

Key words: marked transition systems (LTS), path, atomic propositions, non-determinism