Logika dla informatyków

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

inf.ug.edu.pl/~amb

Weryfikacja modelowa

Weryfikacja

- czego: software (algorytmy, protokoły), hardware
- jak: środowisko modelowania, język specyfikacji, metoda weryfikacji

Metody weryfikacji:

- teoria dowodów lub modeli – badamy $\Gamma \vdash \varphi$ lub $\mathcal{M} \models \varphi$
- automatyzm – od całkowicie ręcznego dowodu, do pełnej automatyzacji
- zakres – od weryfikacji tylko niektórych własności do pełnej weryfikacji systemu
- dziedzina – software/hardware, systemy reaktywne/programy kończące działanie

Weryfikacja modelowa (ang. *model checking*)

Automatyczna weryfikacja modelu systemu, sprawdza wybrane własności, stosowana do systemów współbieżnych, reaktywnych, a także do weryfikacji istniejących rozwiązań.

- weryfikacja modelowa jest nastawiona na obsługę własności temporalnych (dotyczących czasu i ewolucji systemów w czasie),
- model jest ustalony, ew. kontrprzykładem może być przebieg wykonania procesu.

Logiki temporalne

Logiki temporalne

- jednym z elementów branych pod uwagę jest czas – rozpatrywane są (zmieniające się w czasie) stany systemu
- prawdziwość formuły zależy od stanu
- modelami systemu są systemy tranzycji
- język pozwala na wyrażanie „zmienności w czasie"

Weryfikacja odnosi się do konkretnego modelu: $\mathcal{M} \models \varphi$?
ew. kontrprzykład pokaże *ślad* wykonania, tzn. ciąg niepożądanych kroków

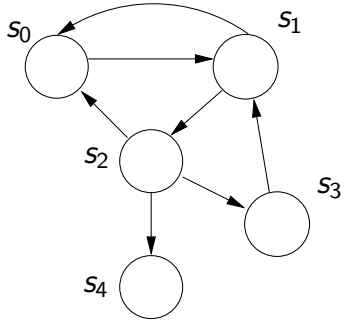
Rodzaje logik temporalnych

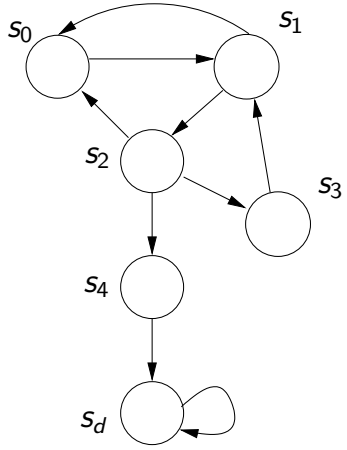
- logika czasu liniowego – przebieg czasu jest ścieżką, rozpatrywane są możliwe ścieżki
- logika czasu rozgałęzionego – od razu rozpatruje się drzewo możliwych wyborów
- czas dyskretny lub ciągły

Logiki mające większą moc wyrażania mają mniejsze szanse na wspomaganie weryfikacji.

Logika czasu liniowego LTL jest często stosowaną logiką.

Systemy tranzycji

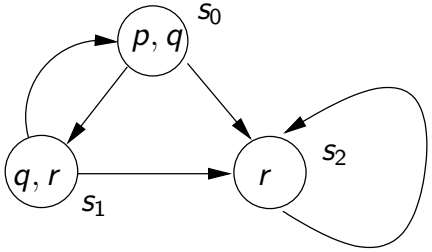




$\mathcal{M} = (S, \rightarrow, L)$

- zbiór stanów S
- relacja przejścia $\rightarrow \subseteq S \times S$
  zakładamy, że nie ma zakleszczeń, tzn. $\forall\, s. \exists\, s'.\, s \rightarrow s'$

Systemy tranzycji c.d.



- etykietowanie $L : S \to \mathcal{P}(atom)$

etykietowanie = wartościowanie – formuły atomowe prawdziwe w danym stanie (pozostałe są fałszywe – niejawnie zakładamy, że znany jest zbiór wszystkich interesujących formuł atomowych)

Ścieżki (przebiegi)

Ścieżka = nieskończony ciąg stanów

$$\pi = s_0 \to s_1 \to s_2 \to \dots$$

- s_0 – teraźniejszość, dalsze stany – przyszłość
- w systemie $\mathcal{M}$ istnieje/może istnieć wiele ścieżek zaczynających się od danego stanu
- na pewno istnieje co najmniej jedna ścieżka (zakleszczenie zostało technicznie wykluczone)
- ozn.: π^i jest to ścieżka π po obcięciu początkowych stanów,

$$\pi^i \mathrel{\widehat{=}} s_i \to s_{i+1} \to s_{i+2} \to \dots$$

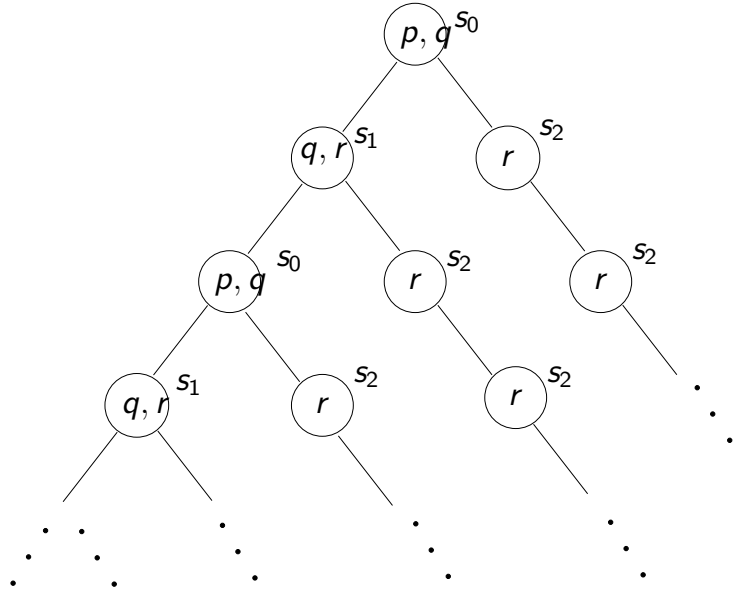
Ślady

- stany nie są rzeczywistością obserwowaną
- system znajduje się w stanie, ale użytkownik widzi tylko własności

Ślad = nieskończony ciąg zbiorów formuł atomowych (prawdziwych)

$$\pi = L(s_0) \to L(s_1) \to L(s_2) \to \dots$$

Rozwinięcie systemu tranzycji



Logika czasu liniowego – LTL

Gramatyka (w postaci Backusa-Naura)

$$\varphi ::= \top \mid \bot \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (X\varphi) \mid (F\varphi) \mid (G\varphi) \mid (\varphi U \varphi) \mid (\varphi W \varphi) \mid (\varphi R \varphi)$$

podobieństwo do logiki zdaniowej, $p \in atom$, nie ma obiektów i termów, są ukryte kwantyfikatory dotyczące czasu

- X: w następnym kroku (*next*)
- F: kiedyś w przyszłości (*finally*)
- G: od teraz na zawsze (*globally*)
- U: do czasu (*until*)
- W, R: podobne do U, szczegóły dalej

X, F, G wiążą najmocniej, (binarne) spójniki zdaniowe najsłabiej

Przykłady formuł

- $Fp \wedge Gq \rightarrow pWr$ (implikacja, poprzednikiem jest koniunkcja, następnikiem jest zdanie pWr)
- $F(p \rightarrow Gr) \vee \neg qUp$ (jest to dyzjunkcja, negacja dotyczy q)
- $pR(qWr)$
- $GFp \rightarrow X(q \vee s)$

Semantyka LTL, ścieżki

Definicja

$\pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$ *ścieżka w modelu* $\mathcal{M}$,
φ – *formuła LTL*

$\pi \models \top$
$\pi \not\models \bot$
$\pi \models p$ *w.t.w.* $p \in L(s_0)$ (*p zachodzi na samym początku*)
$\pi \models \neg\varphi$ *w.t.w.* $\pi \not\models \varphi$
$\pi \models \varphi \wedge \psi$ *w.t.w.* $\pi \models \varphi$ *oraz* $\pi \models \psi$
$\pi \models \varphi \vee \psi$ *w.t.w.* $\pi \models \varphi$ *lub* $\pi \models \psi$
$\pi \models \varphi \rightarrow \psi$ *w.t.w.* $\pi \models \psi$ *jeśli tylko* $\pi \models \varphi$

Uwaga

$\pi \models p$ *w.t.w.* $\{p, \ldots\} \rightarrow \{\ldots\} \rightarrow \{\ldots\} \rightarrow \ldots$

Semantyka LTL c.d.

Definicja

$\pi \models X\varphi$ *w.t.w.* $\pi^1 \models \varphi$
(φ *zachodzi na ścieżce zaczynającej się w* s_1)
$\pi \models F\varphi$ *w.t.w.* $\pi^i \models \varphi$ *dla pewnego* $i \geqslant 0$
$\pi \models G\varphi$ *w.t.w.* $\pi^i \models \varphi$ *dla wszystkich* $i \geqslant 0$
$\pi \models \varphi U \psi$ *w.t.w.* $\exists i\ \pi^i \models \psi$ *oraz* $\pi^j \models \varphi$ *dla* $j{=}0, \ldots, i{-}1$

Uwaga

$\pi \models Xp$ *w.t.w.* $\{\ldots\} \rightarrow \{p, \ldots\} \rightarrow \{\ldots\} \rightarrow \ldots$
$\pi \models Fp$ *w.t.w.* $\{\ldots\} \rightarrow \ldots \rightarrow \{p, \ldots\} \rightarrow \{\ldots\} \rightarrow \ldots$
$\pi \models Gp$ *w.t.w.* $\{p, \ldots\} \rightarrow \ldots \rightarrow \{p, \ldots\} \rightarrow \{p, \ldots\} \rightarrow \ldots$
$\pi \models pUq$ *w.t.w.* $\{p, \ldots\} \rightarrow \ldots \rightarrow \{p, \ldots\} \rightarrow \{q, \ldots\} \rightarrow \{\ldots\} \rightarrow \ldots$

Semantyka LTL c.d.

Uwaga

Znaczenie formuły $\varphi U \psi$:

- *ψ na pewno zajdzie*
- *przedtem φ musi zachodzić*
- *od momentu zajścia ψ nie mówimy co z φ*

Przykłady tautologii: $G\varphi \to \varphi$, $\varphi \to F\varphi$, $\varphi \to \psi U \varphi$

Semantyka LTL c.d.

Definicja

$\pi \models \varphi W \psi$ *w.t.w.* $\pi \models \varphi U \psi$ *lub* $\pi \models G\varphi$ *dla* $j = 0, 1, \ldots$
$\pi \models \varphi R \psi$ *w.t.w.* $\exists i\ \pi^i \models \varphi$ *oraz* $\pi^j \models \psi$ *dla* $j{=}0, \ldots, i$ *lub* $\pi^j \models \psi$ *dla wszystkich* j

Uwaga

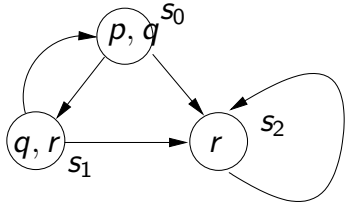
Znaczenie formuły $\varphi W \psi$: może się zdarzyć, że φ będzie prawdziwe zawsze
Znaczenie formuły $\varphi R \psi$:

- *φ, o ile zajdzie, zwalnia konieczność zachodzenia ψ*
- *ψ musi zachodzić do momentu zajścia φ*
- *ψ i φ muszą zachodzić „na zakładkę”*

Semantyka, stany

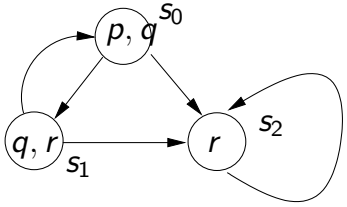
Definicja

$\mathcal{M}$ – model, φ – formuła LTL, $s \in S(\mathcal{M})$
$\mathcal{M}, s \models \varphi$ *w.t.w. jeśli* $\pi \models \varphi$ *dla każdej ścieżki* $\pi = s \to \ldots$



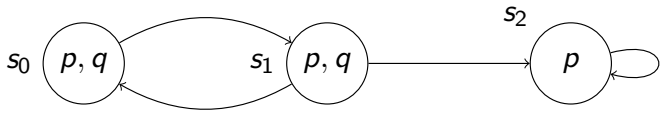
- $\mathcal{M}, s_0 \models p \wedge q$
- $\mathcal{M}, s_0 \models \neg r$
- $\mathcal{M}, s_0 \models \top$
- $\mathcal{M}, s_0 \models Xr$
- $\mathcal{M}, s_0 \not\models X(q \wedge r)$
- $\mathcal{M}, s_0 \models G\neg(p \wedge r)$
- $\mathcal{M}, s_2 \models Gr$

Przykład c.d.



- dla wszystkich stanów s, $\mathcal{M}, s \models F(\neg q \wedge r) \to FGr$
  ścieżka, która spełnia poprzednik musi przechodzić przez s_2, ale wówczas tam już zostanie
- $s_0 \to s_1 \to s_0 \to s_1 \to \ldots \models GFp$
- $s_0 \to s_2 \to s_2 \to s_2 \to \ldots \not\models GFp$
- $\mathcal{M}, s_0 \models GFp \to GFq$
- $\mathcal{M}, s_0 \models GFr$
- $\mathcal{M}, s_0 \not\models GFr \to GFp$

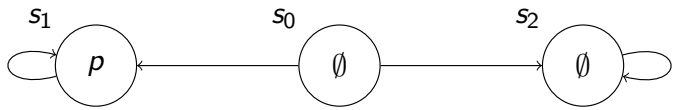
Przykłady dalsze



- $\forall s\ \mathcal{M}, s \models Gp$
- $\mathcal{M}, s_0 \models X(p \wedge q)$
- $\mathcal{M}, s_1 \not\models X(p \wedge q)$
- $\forall s\ \mathcal{M}, s \models G(\neg q \rightarrow G(p \wedge \neg q))$
- $\mathcal{M}, s_0 \not\models pU(p \wedge \neg q)$

Przykłady c.d. – negacja

- zawsze żądamy zachodzenia własności dla *wszystkich* ścieżek
- **nie** jest prawdą, że $\mathcal{M} \models \varphi$ lub $\mathcal{M} \models \neg\varphi$

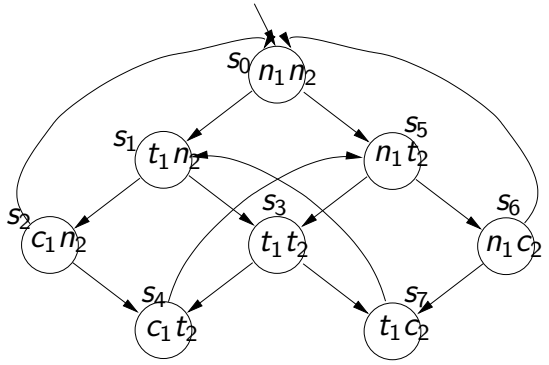


- $\mathcal{M}, s_0 \not\models Fp$ ponieważ na ścieżce $s_0 \rightarrow s_2 \rightarrow \ldots$ nie wystąpi p
- $\mathcal{M}, s_0 \not\models \neg(Fp)$ ponieważ na ścieżce $s_0 \rightarrow s_1 \rightarrow \ldots$ wystąpi p

Przykłady w języku naturalnym

- po włączeniu urządzenia jest ono gotowe do pracy:
  $G(\text{włączone} \rightarrow \text{gotowe})$
- każde żądanie będzie kiedyś potwierdzone:
  $G(\text{żądanie} \rightarrow F\text{potwierdzenie})$
- pewien proces ma gwarancję, że będzie nieskończenie wiele razy umożliwiany: GFumożliwiony
- inny proces na pewno kiedyś zostanie zakleszczony: FGzakleszczenie
- jeśli proces jest możliwy nieskończenie wiele razy, będzie wykonany nieskończenie wiele razy: GFumożliwiony $\rightarrow GF$wykonany
- winda jadąca w górę i mijająca pierwsze piętro nie zatrzyma się, jeśli są wezwania do jazdy na piąte piętro:
  $G(\text{piętro\_1} \wedge \text{w\_górę} \wedge \text{na\_piętro\_5} \rightarrow \text{w\_górę}\ U\ \text{piętro\_5})$

Wzajemne wykluczanie – *mutual exclusion*



Dwa procesy, 1 i 2
stany

- neutralny
- *trying*: zgłasza żądanie wejścia do obszaru krytycznego
- *critical*: znajduje się w obszarze krytycznym

Obszar krytyczny: część kodu wykonywana z wykorzystaniem wspólnych zasobów.
Warunek bezpieczeństwa: oba procesy nie mogą jednocześnie przebywać w obszarze krytycznym.

Mutex c.d.

- drukarka
- komunikacja we współdzielonym kanale
- czas procesora
- zapis pliku/rekordu

Protokół: rejestruje stany procesów, przydziela zasoby procesom tego żądającym

- np. przydziela na stałe cały zasób jednemu procesowi
- czyli, nie dopuszcza do zgłaszania żądań ze strony innych procesów
- albo cyklicznie przydziela zasób kolejnym procesom

Możliwe własności

- bezpieczeństwo (*safety*): w obszarze krytycznym może przebywać najwyżej jeden proces
- żywotność (*liveness*): proces próbujący dostać się do obszaru krytycznego będzie mógł kiedyś wejść do niego
- brak blokowania: każdy może żądać wejścia
- brak kolejności: nie ma z góry ustalonej kolejności

Poprzednie rozwiązania gwarantują głównie bezpieczeństwo,

- stały przydział dla jednego nie jest sprawiedliwy
- cykliczny przydział gwarantuje żywotność i brak blokowania
- ale nie jest ekonomiczny i wymusza cykliczność

Czego nie można wyrazić w LTL?

- w LTL jest wyłącznie kwantyfikator ogólny po wszystkich ścieżkach
- nie ma możliwości żądania, że istnieje ścieżka
- np.: „zawsze można wrócić do stanu początkowego"
- „winda stojąca na pierwszym piętrze może tam stać na zawsze"
- te możliwości ma logika czasu rozgałęzionego CTL

Mutex, własności w LTL wyrażalne i nie

- bezpieczeństwo: $G\neg(c_1 \wedge c_2)$
- żywotność: $G(t_1 \rightarrow Fc_1) \wedge G(t_2 \rightarrow Fc_2)$
- brak wymuszonej kolejności: „istnieje ścieżka z dwoma stanami spełniającymi c_1, pomiędzy którymi nie ma stanów spełniających c_2"
  - nie można wyrazić w LTL
  - przeciwną ideę wyraża $G(c_1 \rightarrow c_1\ W\ (\neg c_1 \wedge \neg c_1\ W\ c_2))$
  - tzn. zawsze po c_1 musi być c_2 nim znowu może zajść c_1
- brak blokowania: „dla każdego procesu w każdym stanie neutralnym istnieje ścieżka, na której proces zgłasza żądanie wejścia do obszaru krytycznego"
  - nie można wyrazić w LTL
  - negacja też nie jest formułą LTL, są dwa kwantyfikatory