ISSN: 2745-3673
http://www.Rcsj.ir

مجلـه‌ی علمی پژوهش در علوم رایانه (سال پنجم)
شماره ۱۷ / بهار ۱۳۹۹ / ص ۵۳-۴۳

مروری بر مشخصه‌های فرمال و زبان Z

حامد بابایی[۱]، جابر کریم پور[۲]، امید جوانشیر[۳]
[۱] مربی، گروه مهندسی کامپیوتر، دانشگاه آزاد واحد اهر، اهر.
[۲] دانشیار، گروه علوم کامپیوتر، دانشکده ریاضی، دانشگاه تبریز، تبریز.
[۳] گروه مهندسی کامپیوتر، دانشگاه آزاد واحد اهر، اهر.

نام نویسنده مسئول:
حامد بابایی

تاریخ دریافت: ۱۳۹۸/۱۱/۲۳
تاریخ پذیرش: ۱۳۹۹/۲/۹

چکیده

یکی از زبان‌های مهم در مشخصه‌سازی متدها زبان Z می‌باشد که امکان مدل-سازی مشخصه‌های سیستم را با قواعد و روش‌های ریاضیات گسسته انجام می-دهد. زبان Z یک زبان رسمی بر پایه‌ی تئوری مجموعه‌ها است. در این مقاله هدف بررسی جایگاه فرمال متدها و زبان Z در تحلیل مهندسی نیازمندی‌های نرم افزار همراه با ارائه نمونه‌هایی با این زبان خواهد بود.

واژگان کلیدی: فرمال متدها، زبان Z، مهندسی نیازمندی‌ها، جبر مجموعه‌ها و حساب محمولات.

مجلـه‌ی علمی پژوهش در علوم رایانه ، شماره ۱۷، بهار ۱۳۹۹، ص ۵۳- ۴۳

ISSN: 2745-3673
http://www.Rcsj.ir

مقدمه

مشخصه‌ی فرمال (فرمال متد) چیست؟ مشخصه‌های فرمال نشانه گذاری ریاضی برای توصیف ویژگی‌ها در یک شیوه و سبک درست و دقیق می‌باشند. در اینجا منظور سیستم چه کاری انجام می‌دهد؟ است نه چگونگی انجام دادن کار توسط سیستم؟ و البته فرمال متدها برای مدل کردن داده‌ها در یک سیستم از مجموعه‌ای توانمند از قوانین ریاضی که رفتار سیستم چگونه خواهد بود را استفاده می‌کنند. که برای این کار از منطق مسندها (منطق مرتبه‌ی اول) – منطق گزاره‌ها – تئوری مجموعه‌ها – دنباله‌ها و... برای توصیف دقیق تأثیرات و نتایج عملیات در سیستم بهره می‌گیرند [1, 3, 4].

نکات قابل توجه در مشخصه‌های فرمال

از تکنیک‌ها و ابزارهای مبتنی بر منطق ریاضی استفاده می‌کنند.
می‌توانند سطوح و قالب‌های گوناگونی از دقت بالا را مورد هدف خود قرار بدهند.
مشخصه‌های سیستم فرمال کامل کننده‌ی تکنیک‌های مشخصه‌ی غیر فرمال می‌باشند.
مشخصه‌های فرمال دقیق، مختصر و غیر مبهم می‌باشند.
توانایی مشخصه‌ی فرمال یک تحلیل از نیازمندی‌های سیستم در مرحله اولیه می‌باشند.
این تکنیک‌ها کاربرد زیادی در توسعه سیستم‌های حساس و استاندارد دارند.
تکنیک‌های جبری برای مشخصه‌ی فرمال به عنوان واسط عمل می‌کنند که به صورت مجموعه‌ای از کلاس‌های اشیاء تعریف شده‌اند.

مفاهیم فرمال متدها

۱- مشخصات فرمال، ۲- اثبات‌های فرمال، ۳- مدل بررسی و ۴- انتزاع

زبان‌های مشخصه فرمال

۱- بر روی منطق وابسته به ریاضی بنا شده‌اند.
۲- زبان‌های مشخصه فرمال غیر اجرایی‌اند (what to do ? را بیان می‌کنند)
۳- بیشتر مبتنی بر اصول موضوعه‌ی منطق ریاضی– منطق گزاره‌ها– حساب محمولات و... می‌باشند.

ویژگی‌های زبان‌های مشخصه‌ی فرمال

۱- داشتن سمانتیک (semantic) صریح و روشن.
۲- رسا بودن خود زبان مانند انعطاف‌پذیری – ملاحظات اقتصادی و... .
۳- پشتیبانی از انواع داده‌ای.
۴- راحتی در بیان گرامر (نحو) زبان.
۵- نمادسازی به صورت دیاگرامی.

مروری بر زبان Z

پس از اندکی مقدمه درباره‌ی مشخصات فرمال حال باید یک زبان را برای بیان توصیف‌های مورد نظرمان در یک سیستم داشته باشیم که ما زبان Z را به عنوان یک زبان مشخصه‌سازی فرمال برای توصیف اهداف‌مان در سیستم انتخاب می‌کنیم [1, 2, 3, 4].

Z یک زبان مشخصه‌ای است که توسط گروه تحقیقاتی برنامه‌نویسی در دانشگاه آکسفورد در سال ۱۹۸۰ میلادی ابداع و توسعه داده شد. این زبان توانایی توصیف و مدل کردن یک دامنه‌ی وسیع از سیستم‌های محاسباتی را دار می‌باشد. خود زبان Z مبتنی بر ریاضیات گسسته (تئوری مجموعه‌ها – منطق مرتبه اول – منطق گزاره‌ها و...) می‌باشد. این زبان ترکیبی از مجموعه‌ی

مجلـه‌ی علمی پژوهش در علوم رایانه ، شماره ۱۷، بهار ۱۳۹۹، ص ۵۳- ۴۳

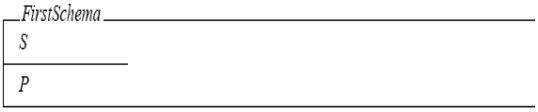
ISSN: 2745-3673
http://www.Rcsj.ir

الگوها (=schema) می‌باشد. یک الگو می‌تواند برای تعریف متغیرهای سراسری و (محلی) – متغیرهای حالت – عملیات‌ها و... مورد استفاده قرار گیرد.

شرح ساختار الگو

در این بخش ما یک ساختار کلی از الگو در زبان Z را بیان می‌کنیم [1, 2, 3, 4].

الگو: یک جفت گزاره (مسند) و امضاء است که به صورت [S|P] می‌باشد:



۱- ساختار یک الگو (مدل) در Z شامل:

الف) . نام الگو (شیء) در بالای الگو.

ب) . در اول بالای خط افقی امضای الگو (منظور شکل مشخصه‌ای که توسط کامپیوتر شناخته و آشکار گردد) قرار می-گیرد.

ج). در پایین خط افقی و در قسمت دوم که الگوی مربوط به گزاره‌ها – محمولات (منطق مسندها) بیان می‌گردند.

۲- الگوی ثابت: هر مدل یا الگوی Z یک قسمت ثابت دارد که تعاریف شرطی که همیشه درست‌اند را در بر می‌گیرد.

۳- سازگاری الگو: اشاره به نیازمندی‌های سیستم در یک مسأله را دارد (باید نیازمندی‌ها با پیاده‌سازی سازگاری داشته باشند).

اصل شمول: در قسمتی از یک الگو می‌توان نام دیگر الگوهای موجود را اعلان نمود.

بیان نیازمندی‌های یک مسأله به وسیله‌ی زبان Z

این قسمت شامل بررسی دو نمونه مسأاله شمارنده‌ی اعداد صحیح و کلوپ اجاره‌ی ویدئو به مشتریان توسط زبان Z می-باشد.

نمونه‌ی ۱: توصیف و پیاده‌سازی زبان Z با نمونه‌ی شمارنده‌ی اعداد صحیح

مسأله: شمارنده‌ای برای شمارش اعداد صحیح $Z=\{\ldots,-1,0,+1,\ldots\}$ (البته در این مسأله مقدار اولیه‌ی شمارنده برابر با صفر(۰) می‌باشد) [3, 4].

Initial counter = 0

در این جا دو عدد ثابت حداقل و حداکثر برای شمارنده وجود دارد که عمل شمارش ما بین این دو عدد را انجام می‌دهد:

Min_value , max_value

در ابتدا تعریف ثابت‌های سراسری را انجام می دهیم.

معرفی جفت ثابت‌های اعداد صحیح

به صورت زیر فرآیند معرفی لیترال‌ها صورت می‌گیرند.

بالای خط افقی اعلان ثابت‌ها و پایین خط شرایط اعلان (مسند (گزاره ی)) می‌باشد.

ISSN: 2745-3673
http://www.Rcsj.ir

توجه: نوع متغیر ریاضی به این صورت است که وقتی مقداری برای این متغیر انتساب شد آن تا آخر مسأله ثابت مانده و تغییر نخواهد کرد. اما در متغیر کامپیوتری چون متغیرها مکان‌های حافظه – ثبات و... می‌باشند پس بنابراین در طول زمان اجرای برنامه از ابتدا تا انتهای برنامه امکان تغییرشان وجود دارد.

نکته: در Z انواع متغیرها در داخل الگوها به صورت ریاضی بیان می‌شوند. که این متغیرها یا مقدار ندارند و اگر مقدار بگیرند همان طور ثابت می‌مانند.

تعریف الگو

(مدل یا یک کلاس (شیء) primary schema) به نام counter که این الگو نشان دهنده‌ی ویژگی‌هایی برای یک عدد صحیح می‌باشد. (این الگو نشان دهنده‌ی مقدار ویژگی است که آن مقدار بزرگتر یا مساوی با min_value و کوچکتر یا مساوی با max_value می‌باشد).

$$
\begin{array}{|l}
\hline
Counter \\
value : \mathbb{Z} \\
\hline
value \geq min\_value \\
value \leq max\_value \\
\hline
\end{array}
$$

در اینجا counter با مقادیر حداقل و حداکثر متناظر با هر مقدار ممکن برای مقدار صفت می‌باشد. (البته با رعایت پیش شرط‌های ورودی شمارنده)

الگوی دو قلو یا پریم (Dashed Schemas)

حال به بررسی متغیر value که برای جداسازی و تفکیک مقدار قبلی ویژگی (تابع شروع قبل ازعملیات = پیش شرط) و استفاده‌ی از متغیر دیگری به نام value' که (پس از انجام عملیات = پس شرط) بر روی تابع (همان شمارنده counter) انجام می‌گیرد، خواهد بود.

$$
\begin{array}{|l}
\hline
Counter' \\
value' : \mathbb{Z} \\
\hline
value' \geq min\_value \\
value' \leq max\_value \\
\hline
\end{array}
$$

در این مرحله مقدار صفت value بعد از انجام عملیات شمارش به value' تبدیل می‌شود. (یعنی value یک دو قلو پیدا می‌کند)

الگوی دلتا (delta schema)

اگر به فرض S یک نامی از یک الگویی که یک شیءای را توصیف می‌کند باشد (مانند counter برای شمارش اعداد صحیح) پس الگوی ΔS که هر دو الگوی s' و S را با هم و بدون هیچ اعلان یا گزاره (شرایطی برای عملیات) دیگری در بر گرفته و شامل می‌گردد.

ISSN: 2745-3673
http://www.Rcsj.ir

$$
\begin{array}{|l}
\hline
\Delta Counter \\
value : \mathbb{Z} \\
value' : \mathbb{Z} \\
\hline
value \geq min\_value \\
value' \geq min\_value \\
value \leq max\_value \\
value' \leq max\_value \\
\hline
\end{array}
$$

الگوی Xi (Xi schema)

اگر S نام یک الگویی که یک شیءای را تعریف می‌کند باشد پس الگوی Xi schema برای الگویی که به واسطه‌ی الگوی ورودی 's وS بعلاوه‌ی یک گزاره‌ای (شرایطی برای عملیات) که درک شده، خواهد بود.

توجه: x = x' و x = x' (برای هر صفت x (مثلا" اعداد صحیح شمارش شده در شمارنده) که در الگوی S (مثلا" الگوی counter برای شمارش اعداد صحیح) تعریف شده باشد است و البته کاراکتر "=" به عنوان عملگر انتساب نیست بلکه به عنوان یک تابع بولین عمل می‌کند که دو خروجی true و false را دارد {0, 1} ← Boolean :"=").

$$
\begin{array}{|l}
\hline
\Xi Counter \\
value : \mathbb{Z} \\
value' : \mathbb{Z} \\
\hline
value \geq min\_value \\
value' \geq min\_value \\
value \leq max\_value \\
value' \leq max\_value \\
value' = value \\
\hline
\end{array}
$$

الگوی Xi که برای بازبینی حالتی از یک شیء (مثلا" شمارشی از اعداد صحیح توسط شمارنده) بدون تغییر آن شیء (مثلا" عدد صحیح شمارش شده) می‌باشد.

نکته: چگونگی نام گذاری ورودی‌ها و خروجی‌های تابع (همان شمارنده‌ی اعداد صحیح) در انتهای اسم ورودی؟ (علامت سوال قرار می‌دهیم) و در انتهای اسم خروجی! (علامت تعجب را قرار می‌دهیم). : ? Input، : ! output

الگوی set-counter

الگویی است که نیازمندی‌هایی را برای یک تابع تعیین می‌کند. مثلا" برای شمارنده‌ی اعداد صحیح نیازمندی‌هایی را تعیین می‌کند. این تابع یک عدد صحیح را از ورودی دریافت و یک عدد صحیح را در خروجی بازگشت می‌دهد. حال اگر ورودی مقدار قانونی برای شمارنده باشد پس مقدار شمارنده مجموعه‌ای برای این مقدار ورودی است و مقدار خروجی مقدار جدیدی می‌شود و گرنه در غیر اینصورت مقدار شمارنده همان مقدار قبلی خودش می‌باشد، اگر مقدار ورودی قانونی نباشد.

ISSN: 2745-3673
http://www.Rcsj.ir

Set_Counter

$\Delta Counter$

$value\_received? : \mathbb{Z}$

$value\_used! : \mathbb{Z}$

$((value\_received? \geq min\_value) \land (value\_received? \leq max\_value)) \Rightarrow (value' = value\_received?)$

$(\neg (value\_received? \geq min\_value) \land (value\_received? \leq max\_value))) \Rightarrow (value' = value)$

$value' = value\_used!$

الگوی گزارش‌دهی تابع شمارنده (report- counter)

گزارش‌های داده شده، مقدار شمارنده بدون تغییر مقدار شمارنده و خروجی مجموعه‌ای از مقادیر مجازی که به شمارنده داده می‌شود.

Report_Counter

$\Xi Counter$

$value\_returned! : \mathbb{Z}$

$value = value\_returned!$

نمونه‌ی ۲: یک کلوپ برای اجاره ویدئو برای مشتریان

توصیف: در ابتدا باید حالت الگو (مدل) و سپس عملیات الگو و در انتها الگوی پرس و جوی از توصیف‌ها صورت پذیرد [3, 4].

حالت الگو

یک مجموعه‌ای به نام all_videos را داریم – که این (مجموعه) اشیایی از نوع VIDEO می‌باشد. یک زیر مجموعه‌ای از VIDEOs که in_stock می‌باشد و این مجموعه لیست کل ویدئوها را نگه می‌دارد. یک زیر مجموعه‌ای که آن booked_out می‌باشد و آن مجموعه اطلاعات ثبت شده‌ی ویدئوهایی است که از کلوپ خارج شده‌اند (به اجاره گرفته شده‌اند) را درخود دارد.

نکته: یک ویدئو همواره در یکی از دو مجموعه‌ی in_stock یا booked_out قرار می‌گیرد.

باید ابتدا گزاره‌ها (مسندها) و محدودیت‌ها بیان شوند:

max video: N

#all videos <= max videos

حال به بیان الگوی حالت می‌پردازیم.

ISSN: 2745-3673
http://www.Rcsj.ir

$$[VIDEO]$$

Video_shop

$$all\_videos, in\_stock, booked\_out : P\ VIDEO$$

$$in\_stock \cup booked\_out = all\_videos$$

Video_shop′

$$all\_videos', in\_stock', booked\_out' : P\ VIDEO$$

$$in\_stock' \cup booked\_out' = all\_videos'$$

الگوی عملیات

پس دادن (بازگشت دادن) ویدئوهای اجاره گرفته شده. در این جا هر ویدئویی دارای شماره‌ای است (هر ویدئو نام دارد) که آن ویدئویی که پس داده می‌شود نام آن ویدئو از مجموعه‌ی booked_out حذف می‌شود و به مجموعه‌ی in_stock افزوده می‌شود.

Video_returned

$$\Delta Video\_shop$$
$$video? : VIDEO$$

$$video? \in booked\_out$$
$$booked\_out' = booked\_out - \{video?\}$$
$$in\_stock' = in\_stock \cup \{video?\}$$
$$all\_videos' = all\_videos$$

و اگر اعلان Δ Video_shop را نادیده بگیریم پس الگوی مربوط به عملیات عودت دادن ویدئو به صورت زیر خواهد بود.

Video_returned

$$all\_videos, in\_stock, booked\_out : P\ VIDEO$$
$$all\_videos', in\_stock', booked\_out' : P\ VIDEO$$
$$video? : VIDEO$$

$$in\_stock \cup booked\_out = all\_videos$$
$$in\_stock' \cup booked\_out' = all\_videos'$$
$$video? \in booked\_out$$
$$booked\_out' = booked\_out - \{video?\}$$
$$in\_stock' = in\_stock \cup \{video?\}$$
$$all\_videos' = all\_videos$$

حال به بررسی عملیات مربوط به حذف ویدئو می‌رسیم که اگر یک ویدئویی حذف شود پس بنابراین باید از لیست‌های all_videos، in_stock وbooked_out حذف گردد.

الگوی پایین حالت کلی‌تر حذف ویدئوها را از لیست‌ها نشان می‌دهد.

ISSN: 2745-3673
http://www.Rcsj.ir

RemoveVideo

$\Delta Video\_shop$
$video? : VIDEO$

$video? \in all\_videos$
$all\_videos' = all\_videos - \{video?\}$
$video? \in in\_stock \Rightarrow in\_stock' = in\_stock - \{video?\}$
$video? \notin in\_stock \Rightarrow in\_stock' = in\_stock$
$video? \in booked\_out \Rightarrow booked\_out' = booked\_out - \{video?\}$
$video? \notin booked\_out \Rightarrow booked\_out' = booked\_out$

RemoveVideo

$\Delta Video\_shop$
$video? : VIDEO$

$video? \in all\_videos$
$all\_videos' = all\_videos - \{video?\}$
$in\_stock' = in\_stock - \{video?\}$
$booked\_out' = booked\_out - \{video?\}$

الگوی پرس وجو (مشاهدات)

به عنوان مثال جستجوی یک ویدئو (بر اساس نام یا شماره‌ی ویدئو) که آن نیازمند بیان یک پرس و جو می‌باشد. پس الگوی مربوط به جستجوی یک ویدئو به شکل زیر می‌باشد.

FindVideo

$\Xi Video\_shop$
$video? : VIDEO$
$message! : MESSAGE$

$video? \in all\_videos$
$video? \in in\_stock \Rightarrow message! = is\_in\_stock$
$video? \notin in\_stock \Rightarrow message! = is\_booked\_out$

این پرس وجو- نام (یا شماره‌ی) ویدئوی مورد جستجو در صورت موجود بودن در خروجی نشان می‌دهد.

نکته: all_videos، لیست کلیه‌ی ویدئوها را در خود دارد. الگوی مربوط به پرس وجوی لیست کلیه‌ی ویدئوها به صورت زیر می‌باشد.

ListVideos

$\Xi Video\_shop$
$list! : P\ VIDEO$

$list! = all\_videos$

ISSN: 2745-3673
http://www.Rcsj.ir

نتیجه‌گیری

در این مقاله یک بررسی مختصر و اجمالی از فرمال متدها و زبان Z ارائه نمودیم که البته کلیه‌ی قواعد حاکم بر زبان Z همراه با دو نمونه مثال کامل با کلیه‌ی الگوها، شروط و محدودیت‌ها مورد بررسی قرار گرفت.

- فرمال متدها برای بیان عملکرد و تحلیل سیستم‌ها استفاده می‌شوند.
- یکی از زبان‌های مشخصه‌سازی فرمال زبان Z است که برای اعمال فرمال متدها به کار می‌رود.
- زبان Z بر مبنای ریاضی و شامل عملیات منطقی و گزاره‌ای است.

پیوست

در انتهای این مقاله به نشانه گذاری‌هایی که در زبان Z الزامی‌اند، می‌پردازیم:

۱-عملگرهای منطقی:

معادل	تفسیر	عملگر
not	نقیض	¬
and	ترکیب عطفی	∧
or	ترکیب فصلی	∨
اگر...آنگاه...	دلالت	⇒
اگر و فقط اگر	هم ارزی	⇔

۲- مجموعه‌ها و عملیات:

نماد	عملیات
∈	عضویت
⊆	زیرمجموعه
∩	اشتراک
∪	اجتماع
\	تفاضل
ℙ	مجموعه توانی
×	ضرب کارتزین

ISSN: 2745-3673
http://www.Rcsj.ir

۳- خواص توابع:

Name

$\pfun$ – Partial functions
$\fun$ – Total functions
$\pinj$ – Partial injections
$\inj$ – Total injections
$\psurj$ – Partial surjections
$\surj$ – Total surjections
$\bij$ – Bijections

Definition

$$X \pfun Y == \{ f : X \rel Y \mid (\forall x : X; y_1, y_2 : Y \bullet (x \mapsto y_1) \in f \land (x \mapsto y_2) \in f \Rightarrow y_1 = y_2) \}$$

$$X \fun Y == \{ f : X \pfun Y \mid \dom f = X \}$$

$$X \pinj Y == \{ f : X \pfun Y \mid (\forall x_1, x_2 : \dom f \bullet f(x_1) = f(x_2) \Rightarrow x_1 = x_2) \}$$

$$X \inj Y == (X \pinj Y) \cap (X \fun Y)$$

$$X \psurj Y == \{ f : X \pfun Y \mid \ran f = Y \}$$

$$X \surj Y == (X \psurj Y) \cap (X \fun Y)$$

$$X \bij Y == (X \surj Y) \cap (X \inj Y)$$

ISSN: 2745-3673
http://www.Rcsj.ir

منابع و مراجع

[1] Beckert, B. (2009) *Formal Specification of Software The Z Specification Language*, Lecture Note, UNIVERSIT˜T KOBLENZ-LANDAU.

[2] Saiedian, H. (2006) *Developing Formal Specifications in Z, Software Requirements Engineering*, Electrical Engineering & Computer Science University of Kansas.

[3] Davies, J. (2006) *Using Z Specification, Refinement, and Proof*, Oxford press, University of Oxford.

[4] Spivey, J. M., Abrial, J. R., Hayes, I. J., Hoare, C. A. R., Morgan, C. C.,Sanders, J. W., Sufrin, B. A. (2006) *The Z Notation: A Reference Manual Second Edition*, Oxford press,University of Oxford.