



معرفی یک مدل برای نرم افزارهای آموزشی مبتنی بر اکتشاف

نرگس السادات بطحائیان

دانشگاه بوعلی سینا، دانشکده مهندسی

bathaeian@basu.ac.ir

چکیده:

یکی از روشهای آموزش روش اکتشافی است که بکارگیری آن برای آموزشهای کلاسیک کمتر استفاده شده است. در این مقاله این روش آموزشی معرفی می شود و مزایای بکارگیری نرم افزارهای کامپیوتری در این نوع آموزش مورد بررسی قرار می گیرد. سپس یک مدل ترکیبی و یک معماری مرجع برای این نوع افزارها پیشنهاد می شود و نشان داده می شود این مدل ترکیبی پیشنهادی در آموزش دروس مهندسی بسیار کاراست. در آخر به عنوان تجربه انجام شده یک بسته نرم افزاری با نام CDLP که برای آموزش درس اصول طراحی کامپایلر در گروه کامپیوتر دانشگاه بوعلی سینا توسعه یافته است، معرفی می شود. روش آموزش در CDLP مطابق مدل پیشنهادی مقاله می باشد.

واژه های کلیدی: نرم افزارهای آموزشی، آموزش اکتشافی، مدل حل مساله، مدل هدایت شده، مدل شبیه سازی، آموزش طراحی کامپایلر

1- مقدمه

استفاده از نرم افزار برای آموزش، سالهاست که مورد بررسی قرار گرفته است و نرم افزارهای مختلفی برای دروس مختلف پیاده سازی شده است. نرم افزارهای متداول آموزشی معمولا از روش بیان صریح محتوا استفاده می کنند. در واقع نوعی معلم مجازی هستند که مطالب را بصورت گرافیکی یا متنی نمایش می دهند. ساخت انیمیشن نیز به هرچه بهتر شدن محتوای این دروس کمک می کند {1}. اما نوع دیگری از یادگیری که کمتر مورد توجه قرار گرفته است، یادگیری مبتنی بر اکتشاف است. یادگیری مبتنی بر اکتشاف یک روش مبتنی بر جستجو است {2}. در این نوع یادگیری معلم نقش کمرنگ یا بیرنگی را به عهده دارد اما یادگیرنده نقش بسیار فعالی را به عهده دارد. یادگیرنده باید با استفاده از مفروضاتی که می داند و با جستجو در فضای مفروضات به راه حل یا اثبات مساله برسد.

ثابت شده است مطالبی که به این روش آموزش داده می شود در پایگاه دانش یادگیرنده پایدارتر است و پیوندهای محکمتری با دانشهای قبلی یادگیرنده پیدا می کند. به علاوه تقویت مهارت کشف کردن یکی دیگر از مزایای این روش آموزش می باشد {3}. البته این روش از آموزش منتقدانی نیز دارد. اشکالاتی مانند سربار زیاد فرضیات، احتمال فهم غلط از مفاهیم آموزشی و بی اطلاع بودن معلم از دانش غلطی که توسط یادگیرنده آموخته شده است {2}، از اشکالات وارد به این روش می باشد.

در این مقاله، در بخش 2 چند نرم افزار مبتنی بر آموزش اکتشافی معرفی می شوند. در بخش 3 انواع مدل‌های این روش آموزش معرفی می شوند. در بخش 4 یک معماری برای نرم افزارهای آموزشی مبتنی بر اکتشاف بر اساس مدل‌های مناسب برای آموزش مهندسی پیشنهاد می شود. در بخش 5 براساس معماری پیشنهادی در بخش 4 سیستم CDLP که نرم افزاری برای آموزش درس طراحی کامپایلرهاست معرفی می شود. در بخش 6 نتیجه گیری و کارهای پیش رو ارائه می شود.

2- کارهای مرتبط

تعداد نرم افزارهای آموزشی که براساس روش آموزش اکتشافی توسعه یافته اند اندک است. از این بین می توان به این موارد اشاره کرد. بسته نرم افزاری JFLAP {4} برای آموزش درس نظریه زبانها و ماشینها پیاده سازی شده است و یک نرم افزار محاوره ای است که می توان گفت به نوعی از روش یادگیری با اکتشاف نیز استفاده کرده است.

بسته نرم افزاری GraphBench که در قسمتی از آن به یادگیرنده اجازه داده می شود الگوریتم خود را طراحی کند و سیستم نتیجه الگوریتم را برای یادگیرنده شبیه سازی می کند، به نوعی از روش یادگیری مبتنی بر اکتشاف برای آموزش مسائل مربوط به گرافها استفاده می کند {5}.

در {6} یک سیستم نرم افزاری برای آموزش مهندسی نرم افزار معرفی شده است که بر پایه یادگیری با اکتشاف است.

{7} سیستمی را معرفی کرده که به روش آموزش مبتنی بر اکتشاف، فرضیه های علمی مانند فیزیک را آموزش می دهد.

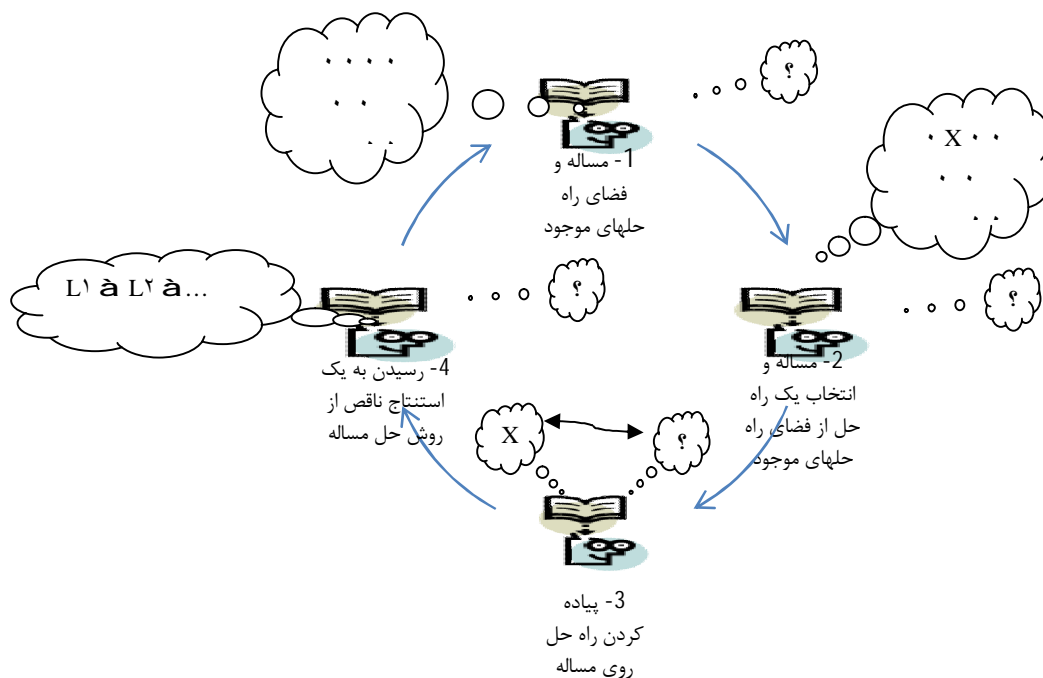
3- آموزش مبتنی بر اکتشاف

شکل 1 مراحل روش آموزش اکتشافی را نشان می دهد. در مرحله اول یادگیرنده با فرضیه یا مساله ای روبرو می شود. در این حالت تمام راه حلهای ممکن، فضای جستجو را تشکیل می دهند. در مرحله دوم از بین تمام راه حلهایی که به نظرش می رسد (فضای جستجو) یکی را انتخاب می کند. در مرحله سوم یادگیرنده راه حل انتخابی را روی مساله امتحان می کند اگر به جواب دست نیافت دوباره به مرحله اول بر می گردد تا راه حل دیگری را انتخاب کند. این چرخه آنقدر ادامه دارد تا یادگیرنده به روش درست حل آن مساله پی ببرد.

همانطور که گفته شد معمولا یادگیرنده ها در محیط آموزشی مبتنی بر اکتشاف با 2 مشکل اصلی مواجهند {2}. یک مشکل سربار زیاد فرضیات است که باعث می شود یادگیرنده نتواند فرضیه مناسبی را برای امتحان انتخاب کند. مشکل دیگر اینکه فرضیه های غلط می توانند برای نمونه های خاص یا کوچک، پاسخ درست داشته باشند و به این ترتیب یادگیرنده ممکن است نتواند به حقیقت دست یابد. برای رفع اشکالات وارد بر این روش، مدل‌های مختلفی برای این نوع یادگیری پیشنهاد شده است که از آن جمله می توان به مدل‌های هدایت شده، حل مساله و شبیه سازی اشاره کرد {2}. مدل هدایت شده مدلیست که معلم نقش یک هدایت کننده را دارد و در مواقعی که یادگیرنده نمی تواند انتخاب کند، او را راهنمایی می کند. در مدل حل مساله، مطالبی به این روش آموزش داده می شوند که بتوان با حل

چند مساله به تئوری آن مطالب پی برد. مدل شبیه سازی راه حل غلطی که توسط یادگیرنده انتخاب شده است را برای مثالهای مختلف شبیه سازی می کند تا یادگیرنده به اشتباه بودن روش خود پی ببرد.

حتی با وجود این مدلها، در روشهای معمول آموزش از آموزش مبتنی بر اکتشاف بسیار کم استفاده شده است. علاوه بر مشکلاتی که در مورد این نوع آموزش نام برده شد، مشکل زمان طولانی آموزش نیز می باشد. فرض کنید که یادگیرنده یک روند طولانی را انجام می دهد تا فرضیه خود را امتحان کند. بعد متوجه اشتباه فرضیه خود می شود و دوباره باید این روند را برای فرضیه جدید خود تکرار کند که کار پرهزمت و وقتگیری است. مساله دیگر، مساله امکان شبیه سازی در آموزشهای واقعی می باشد. معمولاً چنین کاری باید توسط راهنما و یا معلم و بوسیله قلم و کاغذ انجام شود که پر واضح است کارایی یک انیمیشن کامپیوتری را نخواهد داشت. همچنین در خیلی از موارد، راهنما می داند که فرضیه یادگیرنده غلط است اما پیدا کردن یک مثال برای اثبات غلط بودن آن کار وقت گیر و سختی است.



شکل 1- مراحل روش آموزش اکتشافی

4- پیشنهاد یک سیستم آموزشی مبتنی بر اکتشاف

در اینجا یک سیستم آموزشی مبتنی بر اکتشاف ارائه می شود که براساس ترکیبی از مدلهای حل مساله، شبیه سازی و هدایت شده می باشد. این سیستم برای دروسی که محتوای آنها روش حل مسائل با الگوریتمی خاص می باشد بسیار کارآمد است.

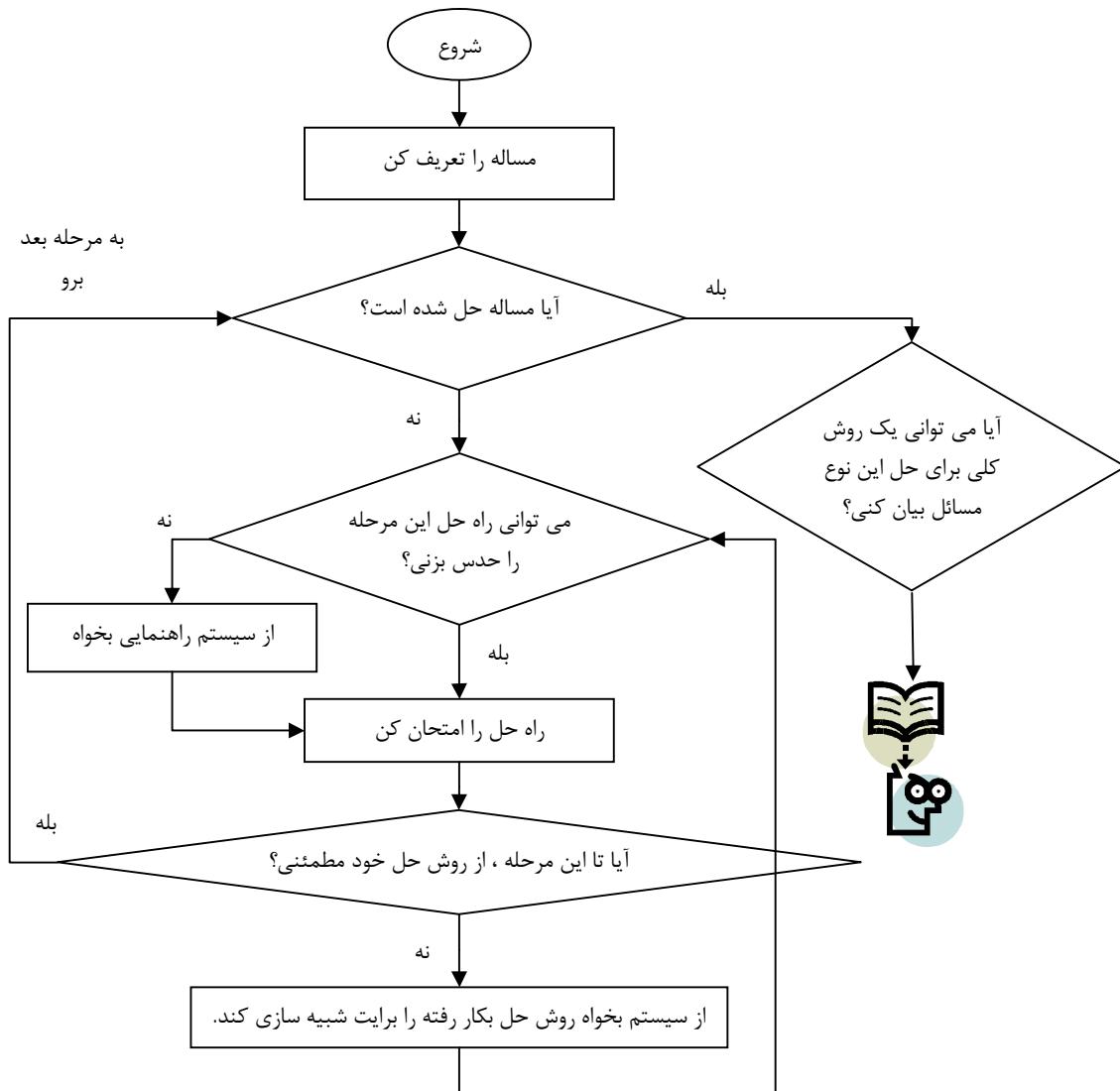
4-1- خصوصیات سیستم پیشنهادی

شکل 2 فلوچارت چنین سیستمی را در تقابل با یادگیرنده نمایش می دهد. همانطور که در فلوچارت مشخص است این سیستم مبتنی بر مدل حل مساله می باشد. یعنی یادگیرنده روش حل نوعی از مسائل را کشف می کند. این نوع مسائل بهتر است از نوع الگوریتمی باشند. یعنی قدم به قدم و طبق روشی خاص حل شوند. یادگیرنده مساله را برای سیستم تعریف می کند و با استفاده از امکاناتی که سیستم در اختیار او قرار می دهد، آن را حل می کند. در واقع روش حل مساله را کشف می کند. مدل هدایت شده در دو قسمت از این سیستم

استفاده شده است. یکی وقتی که یادگیرنده با انبوهی از فرضیات و روشهای مختلف حل مساله روبرو می شود و برای رسیدن به راه حل درست باید یکی یکی آنها را امتحان کند، در این حالت یادگیرنده می تواند از سیستم راهنمایی بخواهد و پیغامی که نشان دهنده قدم بعدی برای حل مساله است به او ارائه می شود. دوم شبیه سازی نتایج است. در این بخش روش حل یادگیرنده با روش درست حل مقایسه می شود و بطور غیر مستقیم عدم درستی آن به او اطلاع داده می شود. این قسمت که به نوعی پیاده سازی مدل شبیه سازی نیز می باشد، بسته به نوع مساله و روش حل آن متفاوت پیاده سازی می شود.

پیاده سازی قسمت شبیه سازی

قسمت شبیه سازی پاسخ یادگیرنده، می تواند به شکلهای مختلفی پیاده سازی شود. فرض کنید که پاسخ یادگیرنده به مساله X بوده است و پاسخ اصلی مساله Y است. در این حالت باید بدانیم که یادگیرنده برای رسیدن به پاسخ X از چه مسیرهایی گذشته است. بنابراین مسیره را در پایگاه دانش سیستم ذخیره می کنیم. فرض کنید یادگیرنده از مسیره های $L_1 \rightarrow L_2 \rightarrow \dots \rightarrow L_n$ گذشته است تا به پاسخ X رسیده است و در قسمتی از مسیر مثلاً L_i اشتباه کرده است. به منظور اطلاع رسانی غیر مستقیم، مسیر اشتباه را روی یک مثال اعمال می کنیم و نتایج غلط را به یادگیرنده نشان می دهیم. بهتر است مثال طوری انتخاب شود که اشتباه بودن نتایج پایانی برای یادگیرنده محرز باشد. بسته به نوع مساله، این مثال می تواند از بین چند مثال ثابت انتخاب شود یا با استفاده از روشهای مبتنی بر ریاضی و یا هوش مصنوعی انتخاب شود. اگر نتوان مثالی پیدا کرد، تنها بیان این نکته که چرا انتخاب این مسیر اشتباه است (اگرچه یک آموزش مستقیم می شود) کمک خوبی برای ادامه درست مسیر می باشد. در نوعی از مسائل که هدف از مساله طراحی یک سیستم بصورت انتزاعی باشد، قسمت شبیه سازی می تواند راحتتر پیاده سازی شود. در این حالت X و Y دو طراحی متفاوت برای آن سیستم خواهند بود. اگر بتوانیم خصوصیتی را بیابیم که در طراحی Y رعایت شده است اما در طراحی X رعایت نشده است، آنگاه می توان به نوعی مشکلات پیش رو بخاطر عدم رعایت آن ویژگی را به یادگیرنده گوشزد کنیم.



شکل 2- فلوچارت سیستم پیشنهادی در تقابل با یادگیرنده

2-4- بکارگیری سیستم پیشنهادی برای آموزش علوم مهندسی

همانطور که گفته شد بخاطر مشکلات موجود در روش اکتشافی، این روش، یک روش مرسوم در هیچ یک از علوم نمی باشد. اما سیستم پیشنهادی می تواند برای علوم مهندسی کارآمد عمل کند، زیرا:

- ü اگر چه مبنای علوم مهندسی، علوم پایه می باشد. اما علوم مهندسی کاربردی اند و به چگونگی حل مسائل (الگوریتم حل مساله) بیشتر توجه دارند. از اینرو مدل حل مساله در روش آموزش اکتشافی مناسب به نظر می رسد.
- ü پایه علوم مهندسی بر ریاضیات است و این کمک می کند هنگام شبیه سازی پاسخ غلط، یافتن مثال ساده تر شود. به علاوه محور اصلی بسیاری از دروس بر طراحی استوار است و همانطور که گفته شد و در سیستم نمونه خواهیم دید، پیاده سازی قسمت شبیه سازی برای اینگونه کاربردها راحتتر است.
- ü نرم افزارهای کامپیوتری می توانند از انجام بسیاری از کارهای تکراری جلوگیری کنند. در چرخه یادگیری اکتشافی، یادگیرنده لازم نیست راه حلهای انتخابی را هر بار از ابتدا شروع کند بلکه می تواند تنها تغییرات حل جدید نسبت به قبلی را اعمال کند و نرم افزارهای کامپیوتری با فراهم کردن امکان ویرایش این روند را ساده می کنند.

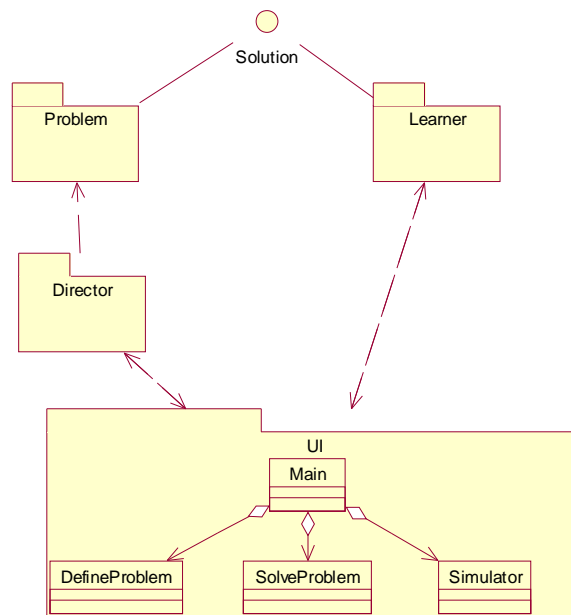
نرم افزارهای کامپیوتری می توانند قسمت شبیه سازی را بصورت انیمیشن ارائه کنند. در اینصورت تاثیر آموزش افزایش می یابد.

عبور از چرخه یادگیری اکتشافی و رسیدن از چگونگی حل مساله به چرایی آن (theory discovery) آنچنان مساله را در ذهن یادگیرنده پایدار می کند که بعدها او بتواند در مسائل عملی بهتر با مسائل روبرو شود.

3-4 - معماری مرجع برای سیستم پیشنهادی

با توجه به مشترکاتی که این نوع سیستمهای معرفی شده دارند می توان یک معماری مرجع {8} برای این نوع سیستمهای آموزشی در نظر گرفت. البته برای سیستمهای آموزشی متداول چنین معماری پیشنهاد شده است {9} اما برای سیستمهای آموزشی مبتنی بر اکتشاف چنین موردی پیدا نشد.

شکل 3 بسته ها و ارتباطات بینشان را در این معماری پیشنهادی نشان می دهد.



شکل 3 - معماری مرجع پیشنهادی

بسته UI واسط کاربری برنامه است که از 3 نوع واسط مختلف تشکیل شده است. یکی مربوط به تعریف مساله (DefineProblem)، دیگری حل مساله (SolveProblem) و سومی (Simulator) مربوط به شبیه سازی پاسخ یادگیرنده می باشد.

بسته Problem تعاریف مساله و قدمهای حل آن مساله را نگهداری می کند. الگوریتمهای مربوط به حل مساله نیز در این بسته تعریف شده است.

بسته Learner قدمهای روش حل مساله را که یادگیرنده اجرا کرده است نگهداری می کند.

بسته Director الگوریتمهای مقایسه پاسخ اصلی و پاسخ یادگیرنده را در بر دارد. به علاوه الگوریتمهای مربوط به شبیه سازی پاسخ غلط یادگیرنده در این بسته پیاده سازی می شود.

ویژگی مهم این نوع سیستمها این است که در طول مدت آموزش، تعریف مساله ثابت است و یک روند حل ثابت نیز دارد. اما با توجه به اینکه روش آموزش اکتشافی است روش حلی که توسط یادگیرنده اعمال می شود در طی زمان مرتب تغییر می کند. بنابراین بسته مربوط به حل مساله توسط یادگیرنده (Learner) مستقل از بسته مربوط به راهنما (Director) است. تنها هنگام درخواست کاربر برای راهنمایی یا پاسخ شبیه سازی شده اطلاعات لازم از طریق UI به Director فرستاده می شود. اما بسته Problem تنها یکبار تعریف می شود ولی

به تعداد درخواستهای کاربر، به منظور مقایسه پاسخ یادگیرنده با پاسخ درست، توسط بسته Director استفاده می شود. بنابراین این بسته با Director رابطه مستقیم دارد.

با توجه به اینکه هر دو بسته Learner و Problem در قسمت ساختارهای نگهداری و دنبال کردن حل مساله اشتراکهای زیادی دارند کلاسهای مربوط به حل مساله در این دو بسته واسط Solution را realize می کنند.

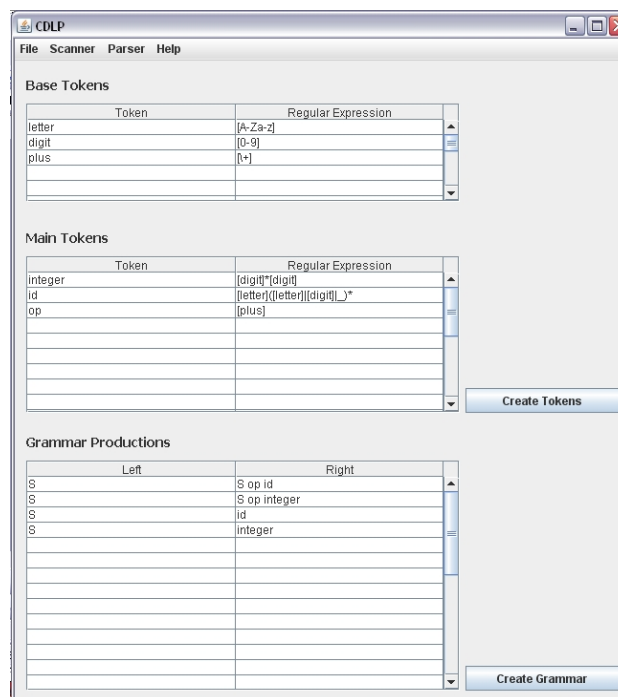
5- CDLP : یک تجربه

در درس طراحی کامپایلر که یکی از دروس مهندسی نرم افزار است، طراحی فازهای مختلف کامپایلر یک زبان بررسی می شود. در فاز اول که اسکندر نام دارد نوع لغات بکار رفته در یک زبان تعریف می شوند. براین اساس اسکندر می تواند یک لغت را گرفته و نوع آن را بررسی کند. یکی از راههای مرسوم طراحی اسکندر، طراحی ماشینی موسوم به ماشین حالت متناهی (DFA) است که طراحی آن در چندین مرحله مرتبط به هم صورت می گیرد.

در فاز دوم که پارسر نام دارد، چینش بین لغات، براساس گرامر زبان بررسی می شود. روشهای مرسوم طراحی پارسر، ساخت جداولی موسوم به $LR(1)$, $LL(1)$, $LR(0)$, $SLR(1)$, $LALR(1)$ می باشد. طراحی این جداول الگوریتمی است.

CDLP یا Compiler Design Lab Package یک بسته نرم افزاری برای آموزش درس اصول طراحی کامپایلر است که در دانشگاه بوعلی سینا برای درس اصول طراحی کامپایلر و براساس مدل پیشنهادی این مقاله طراحی و پیاده سازی شده است.

شکل 4 فرم اولیه این سیستم را نشان می دهد. با استفاده از این فرم، کاربر مساله مورد نظر خود را که همان خصوصیات یک زبان برنامه نویسی و گرامر مربوطه اش می باشد را تعریف می کند.



The screenshot shows the CDLP software interface with three main sections:

- Base Tokens:** A table with columns 'Token' and 'Regular Expression'. It contains:

Token	Regular Expression
letter	[A-Za-z]
digit	[0-9]
plus	[+]
- Main Tokens:** A table with columns 'Token' and 'Regular Expression'. It contains:

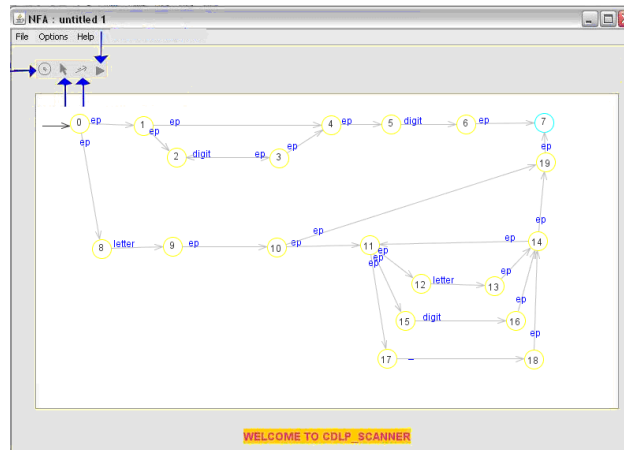
Token	Regular Expression
integer	[digit]*[digit]
id	[letter]([letter] digit _)*
op	[plus]
- Grammar Productions:** A table with columns 'Left' and 'Right'. It contains:

Left	Right
S	S op id
S	S op integer
S	id
S	integer

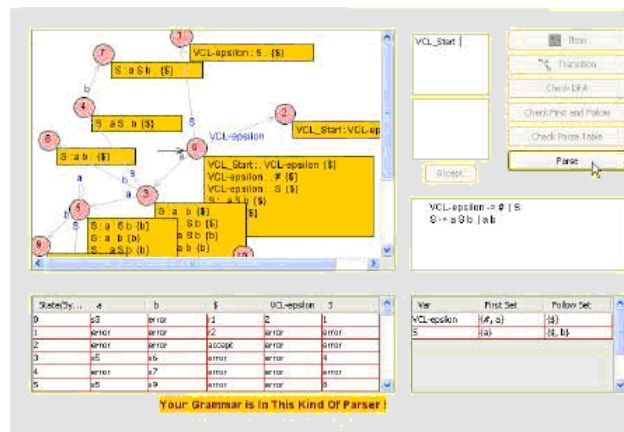
شکل 4- فرم اولیه سیستم CDLP

همانطور که مشخص است دو فاز اصلی کامپایلرها یعنی فازهای اسکندر و پارسر در این سیستم پیاده سازی شده است {10}. فاز اسکندر امکاناتی برای ساختن NFA، DFA و minimized DFA را به یادگیرنده می دهد. فاز پارسر امکاناتی برای ساخت یک پارسر برای گرامر تعریف شده را به کاربر می دهد. انواع پارسی که در این سیستم پوشش داده می شود، پارسرهای $LL(1)$ ، $LR(0)$ ، $SLR(1)$ ،

$LR(1)$ و $LALR(1)$ است. شکل 5 نمایی از محیط ساخت اسکنر و شکل 6 نمایی از محیط ساخت پارسر را نشان می دهد. نرم افزار امکانات لازم برای طراحی اسکنر و پارسر را در اختیار یادگیرنده قرار می دهد. در هر قدم از ساخت، کاربر می تواند از سیستم راهنمایی بخواهد و سیستم راهنمایی لازم را به کاربر می دهد. به این ترتیب به کاربر در مورد انتخاب درست راه حل راهنمایی داده می شود.



شکل 5- نمایی از محیط و ابزارهای لازم برای ساخت اسکنر

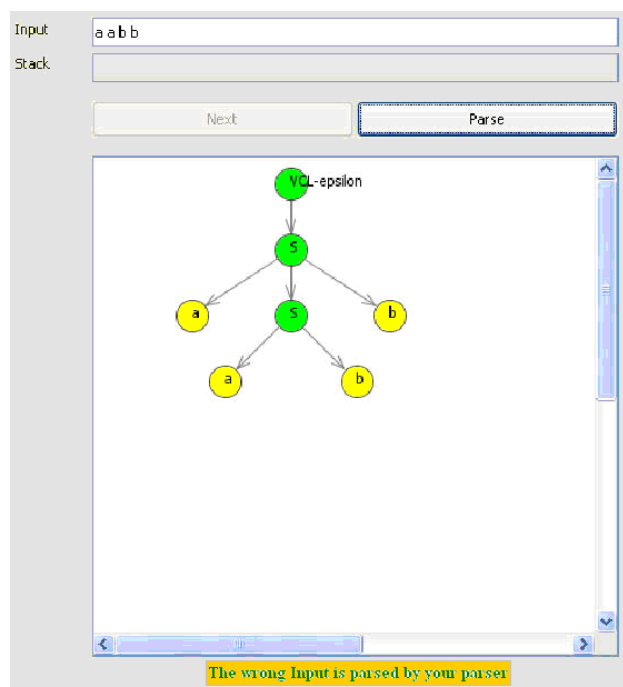


شکل 6- نمایی از محیط ساخت پارسر $LR(1)$ و پیغام راهنمایی آن

در قسمت شبیه سازی پاسخ از رویه اختلاف دو جواب استفاده می شود. یعنی در قسمت اسکنر رشته ای از حروف را پیدا می کنیم که در جواب باید باشد، مثلا توسط DFA، اسکن شود ولی نمی شود. و این روند را با انیمیشن به یادگیرنده نشان می دهیم. شکل 7 الگوریتم یافتن این مثال را نشان می دهد که بر پایه ریاضیات "زبانها و ماشینها" است. برای جزئیات بیشتر به [11] رجوع شود. در قسمت پارسر هم مشابه حالت قبل رشته ای از توکنها را پیدا می کنیم که در زبان موجود است ولی با پارسر تعریف شده توسط یادگیرنده پارسر نمی شود که این را بوسیله یک درخت شبیه سازی شده موسوم به "درخت پارس" نشان می دهیم. شکل 8 نمایی از ساخت درخت پارس را نشان می دهد.

R: Regular Expression
M, N, P: DFA
W: String
R= Learner's defined regular expression
M= Learner's defined DFA
N= DFA for R
P= (M-N) Union (N-M)
IF P is Null THEN M is a correct DFA for R
ELSE Create W accepted by P and simulate it on M

شکل 7- الگوریتم شبیه سازی مثال برای اسکندر



شکل 8- نمایی از شبیه سازی ساخت درخت پارس

6- نتیجه گیری

در این مقاله سیستمهای آموزشی که از طریق اکتشاف آموزش می دهند، معرفی و مورد بررسی قرار گرفتند. اگر چه چنین روش آموزشی متداول نیست اما با توجه به ظرفیتهای خوبی که فناوری اطلاعات و سیستمهای نرم افزاری دارند و مزایای منحصر به فرد این روش آموزش در علوم مهندسی، می توان امیدوار بود که در آینده جایگاه خوبی در آموزش مهندسی پیدا کند. مدل و معماری مرجعی که ارائه شد را می توان برای بسیاری از دروس با محتواهای مختلف بسط داد. نمونه ارائه شده با نام CDLP نیز نشان دهنده کاربرد این نوع نرم افزارها می باشد.

کارهای پیش رو شامل موارد زیر می باشد:

§ گسترش CDLP و اضافه کردن فازهای دیگر مربوط به طراحی کامپایلرها.

§ دو سیستم نرم افزاری دیگر نیز در دست اجراست. یکی برای آموزش درس سیستمهای عامل از رشته مهندسی کامپیوتر و دیگری برای آموزش درس مدارهای الکتریکی از رشته مهندسی برق.

تشکر و قدردانی

در اینجا لازم است از خانمها فرزانه زارعی و سوسن همایی که سهم مهمی در پیاده سازی سیستم CDLP داشتند، تشکر نمایم.

مراجع

- [۱] Diehl, S. (Ed.), *Software Visualization*, LNCS ۲۲۶۹, pp. ۱-۱۵, ۲۰۰۲.
- [۲] Bruner, J., *Discovery Learning*, URL: <http://www.learning-theories.com/discovery-learning-bruner.html>
- [۳] Kerren, A., "Learning by Generation in Computer Science Education", *JCS&T*, Vol ۴, No. ۲, pp. ۸۴-۹۰, August ۲۰۰۴.
- [۴] S. Rodger and T. Finley, *JFLAP - An Interactive Formal Languages and Automata Package*, ISBN ۰۷۶۳۷۳۸۳۴۴, Jones and Bartlett, ۲۰۰۶.
- [۵] Braendle, M., Nievergelt, J., "Tackling Complexity: A Case Study on Educational Software", *Swiss Federal Institute of Technology. Zurich, Switzerland* URL: www.inf.ethz.ch/personal/braendle/graphbench.pdf
- [۶] Hanakawa, N., "Tools for Discovery Learning and Generating Educational Sequences for Software Engineering", ۱۹th Conference on *Software Engineering Education and Training Workshops*, ۲۰۰۶. CSEETW '۰۶.
- [۷] Jong, T., et al, "SMISLE: System for Multimedia Integrated Simulation of Learning Environment", *Klumer Academic Publisher*, pp. ۱۳۳-۱۶۵, ۱۹۹۴.
- [۸] Hofmeister, C., et al., *Applied Software Architecture*, USA Addison-Wesley, ۲۰۰۰.
- [۹] Pollard, J., & and Roger Duke, R., "A Reference Architecture for Instructional Educational Software", *South East Asia Regional Computer Confederation (SEARCC)* ۲۰۰۵:
- [۱۰] A.V. Aho, R. Sethi & J.D.Ullman, "*Compilers: Principles, Techniques and Tools*", Addison-Wesley, ۱۹۸۶.
- [۱۱] P. Linz, "Formal Languages and Automata", Jones and Bartlett, ۲۰۰۶.