

**موضوع تحقیق:**

**کاربرد و استفاده اتوماتا در هوش  
مصنوعی و طراحی کامپایر**

**گرد آوری: امیر محمد خالقی فرید**

**منبع: [www.geeksforgeeks.org](http://www.geeksforgeeks.org)**

## مقدمه

مهندسان کامپیوتر در گذشته در راه سهولت برنامه نویسی مجبور به طراحی سیستم های پردازش زبان شدند که اتوماتا به عبارتی محصل قواعد و تجربیات آنها در راستای تحقق بخشیدن به این امر میباشد. اتوماتا به عبارت ساده ماشینی است که رشته هایی از یک زبان را به صورت ورودی از الفبای آن زبان میپذیرد که به کاربرد نوع های آن در کامپایلر و هوش مصنوعی خواهیم پرداخت.

## 1. اتوماتای متناهی یا (Finite Automata/FA):

این اتوماتا به صورت گسترده در برنامه های زیادی استفاده میشود مانند:

- آنالیز کردن کلمات در کامپایلر (Lexical Analysis): FA به شناسایی کلمات کلیدی و اپراتور ها و توکن های داخل کد کمک میکند.
- شناسایی الگو با عبارات عادی: FA عبارات عادی برای جستجو و تطابق الگو ها در فایل های متنی را مدلسازی میکند.

## 2. اتوماتای پشته ای یا (Push Down Automata/PDA):

- پردازش قواعد نوشتاری در کامپایلر ها: در تجزیه ساختار های زبانی از پشته برای ذخیره عبارات تو در تو استفاده میکند.

- پردازش زبان های طبیعی: برای کارهایی مانند تجزیه جملات و درخت های نوشتاری استفاده میشود.
- کریپتوگرافی: برای طراحی الگوریتم برای رمز نگاری و رمز گشایی استفاده میشود.
- اثبات خودکار قضیه: برای سنجش درستی مدل های نرم افزاری و سیستم ها.

### 3. اتوماتای خطی کران دار یا (Linear Bounded Automata/LBA):

- آنالیز مفهومی در کامپایلر ها: برای ساخت درخت های تجزیه نوشتار به منظور آنالیز و بررسی مفهوم کد.
- شناسایی زبان های حساس به متن: برای زبان هایی که قدرت پردازشی بیشتری نسبت به زبان های مستقل از متن نیاز دارند.
- تئوری بازی: تماس های بین عوامل هوشمند را مدلسازی و رفتار های استراتژیک آنها را تحقیق میکنند.

#### 4. ماشین تورینگ (Turing Machine):

هوش مصنوعی: اساس الگوریتم های هوش مصنوعی را فرم میدهد که تصمیم گیری و machine learning را شامل میشود.

رباتیک: برای مدلسازی فعالیت های ربات و کنترل سیستم ها استفاده میشود.

نظریه پیچیدگی: برای آنالیز و بررسی پیچیدگی محاسباتی الگوریتم ها مورد استفاده قرار میگیرد.

## کاربرد اتوماتای متناهی در فعالیت کامپایلر ها:

مبنای اتوماتای متناهی در دو مرحله از سه مرحله فرایند کامپایل شدن انجام میشود

- **اولی (Lexical Analysis):** بخش پردازش کلمات است که عبارت های عادی را به توکن ها (مفاهیم کوچک نقش دار) تبدیل میکند. و کار آن عملاً تک به تک خواندن کارکتر ها از کد منبع و تبدیل آنها به شکل ساده تر اما معنا دار بر اساس مقدار و نقش آنها میباشد که به آن **Lexeme** میگویند تا سپس مفاهیم آنها را در جدولی ذخیره کند. در پروسه پردازش کلماتی، کلمات به گروهبندی هایی نظیر کلمات کلیدی، اپراتورها، محدود کننده ها و علائم نگارشی

تقسیم میشوند. هرتوکن به شکل دو مقدار در نظر گرفته میشود که مانند فرمت زیر نوشته میشود:  
<شماره، شناسه> و به عبارتی نقش گرمر در اتوماتای متناهی را دارند.

- **دومی (Abstract Syntax Tree):** در این بخش که در مراحل کامپایلر تحت مرحله تجزیه (Parsing) نام دارد یک درخت انتزاع تشکیل میشود که ساختار نوشتاری را شرح میدهد و دو نوع تکنیک برای پردازش آن استفاده میشود که یکی از بالا به پایین و دیگری از پایین به بالا است. در نوع بالا به پایین گرمر باید از فرم بازگشت پذیر چپ به گرمر مستقل از متن معادل، برسد. در حالیکه در تجزیه پایین به بالا برای هر گرمر مستقل از متن اتوماتای تعریف شده چندین وضعیت دارد تا یا محصل وضعیت غیر پایانه ای را نمایش دهد و یا وضعیت هر پایانه را.