

# شرح برنامه تبدیل فرمول منطقی به فرم نرمال فصلی

## 1. ورودی و پارسینگ آن:

هر فرمول در یک خط و با فرمت تعریف شده وارد میشود، کاراکترهای فاصله و پرانتزهای خالی و جفت نقیضهای بلافاصله پشت سر هم، در ورودی لحاظ نمیشوند.

سپس این رشته کاراکتری به تابع build از کلاس formula تحویل داده میشود تا یک نمونه از این کلاس بسازد، این تابع در هر مرحله رشته دریافتی را به صورت  $P \vee Q$  تجزیه میکند (c یک رابط دומکانی فرض شده) و به صورت بازگشتی دو نمونه جدید از نوع خود میسازد و رشته های P و Q را به آنها تحویل میدهد تا کار ساخت درخت باینریای که برای ذخیره فرمول به کار میرود را ادامه دهند.

در نهایت یک درخت باینری در دست خواهیم داشت که برگهای آن متشکل از لیترالها (گزاره های اتمیک یا نقیض آنها) و سایر نودها متشکل از رابطهای "و" و "یا" هستند (سایر رابطها در طول پارسینگ به این دو تبدیل میشوند)

## 2. تولید DNF:

این مرحله توسط تابع convertToDNF انجام میشود که یک نمونه از نوع formula میگیرد و یک آرایه دو بعدی از لیترالها تحویل میدهد. این تابع یک تابع بازگشتی است که ابتدا زیردرخت چپ و راست فرمول ورودی را تبدیل به فرم نرمال فصلی میکند یعنی آرایه دو بعدی ذکر شده را برای این دو تولید میکند و سپس در صورتی که رابط ریشه درخت "یا" باشد این دو آرایه را با هم ادغام میکند و برمیگرداند و اگر رابط "و" باشد دو آرایه را در هم پخش میکند و یک آرایه جدید میسازد یعنی هر ردیف از آرایه اولی را با هر ردیف از آرایه دومی ادغام میکند و به این ترتیب یکی از ردیف های آرایه جدید را تولید میکند (توجه داشته باشید که در این نوع نمایش، ردیف های آرایه متناظر با کلاوزهای فرم نرمال فصلی مربوطه هستند)

## اضافات:

### 1. تشخیص مشکلات سینتکسی:

این قابلیت در دل تابع build گنجانده شده و در حین پارسینگ انجام میشود. در هر مرحله از تولید درخت مذکور اگر ترکیب نامتعارفی از کاراکترها رویت شود (مثل خطاهای گرامری یا پرانتزهایی که بالانس نیستند) کار پارسینگ متوقف میشود و کاربر از این ارور مطلع میشود.

### 2. تشخیص عدم استفاده کافی از پرانتزها:

پس از اینکه معلوم شد فرمول از لحاظ سینتکس مشکلی ندارد. با توجه به اینکه هیچ اولویتی برای رابطها در نظر گرفته نشده بررسی میشود که در هر سطح از پرانتزها، بیش از یک نوع رابط استفاده نشده باشد یا اینکه از رابطهایی که شرکت پذیر نیستند بیش از یک بار استفاده نشده باشد. در این مرحله هم در صورت وجود مشکل، کاربر مطلع میشود.

### 3. مرتب کردن DNF تولید شده:

در این مرحله لیترالهای هر کلاوز سورت میشوند و تکراریها حذف میشوند و در صورتی که دو لیترال متناقض رویت شود کلاوز مورد نظر کلا حذف میشود چون وجود کلاوزهایی که تناقض هستند بی مورد است.

سپس خود کلاوزها نیز بر اساس روابطی که در ابتدای کد تعریف شده سورت میشوند و در صورتی که کلاوزهای تکراری وجود داشته باشند تکرارها حذف میشوند.

#### **4. حذف کلاوزهایی که شرط قویتر برای یک کلاوز دیگر هستند:**

وجود یا عدم این نوع کلاوزها نیز تفاوتی در ارزش فرم نرمال فصلی ایجاد نمیکند پس اینها نیز حذف میشوند.

به عنوان مثال فرم  $(p \wedge q) \mid (p)$  تبدیل به فرم  $(p)$  میشود.