

گزارش پروژه‌ی درس هوش محاسباتی - منطق فازی

۱. دریافت ورودی‌های مسئله

طی متد `read_fuzzy_problem` در `FuzzyController` ورودی‌های مسئله که به دو دسته‌ی متغیرهای فازی (مانند `cp`, `cv`, `pv`) و قانون‌ها می‌شوند از طریق یک فایل `yaml` دریافت می‌شوند. در برنامه فرض شده این فایل، با نام `input.yml` در `src/` قرار می‌گیرد.

۲. مدل‌سازی ورودی‌ها

ورودی‌های دریافت شده در بخش قبل، به صورت نمونه‌هایی از کلاس‌های `FuzzyVar` و `FuzzyRule` (پیاده شده در `model.py`) مدل‌سازی می‌شوند.

هر نمونه از کلاس `FuzzyVar` شامل نام متغیر فازی و مجموعه‌های آن متغیر است. هر مجموعه (کلاس `FuzzySet`) شامل نام صفت مجموعه، و مقادیر تعلقشان است. این مقادیر تعلق می‌توانند به صورت مثلثی (کلاس `Triangle`) یا خط صاف (کلاس `LinearLine`) باشند.

همچنین هر نمونه از کلاس `FuzzyRule`، شامل مقادیر فرض و حکم یک قاعده است. در نهایت کلاس کنترلر ما (`FuzzyController`)، یک دیکشنری از `FuzzyVar`‌ها و یک لیست از `FuzzyRule`‌ها خواهد داشت.

۳. استنتاج

طی متد `inference` کلاس کنترلر، هر یک از قاعده‌های مسئله با ورودی‌های داده شده چک می‌شوند. در نهایت در قالب یک دیکشنری، برای هر یک از حالت‌های `Force` خروجی، یک میزان تعلق برگردانده می‌شود.

۳. غیرفازی‌سازی

طی متد `defuzzify` کلاس کنترلر، دیکشنری ساخته شده در بخش استنتاج را به یک میزان قطعی از نیرو تبدیل می‌کنیم. برای این کار، ابتدا برای هر مجموعه از کلاس `FuzzyVar` مختص نیرو، شکل مرتبط با میزان تعلقش را به کمک متد `(set_max_y)` کلاس `Triangle` می‌سازیم (عموماً نتیجه به شکل دوزنقه است). در ادامه با کمک `shapely.ops.cascade_union` شکل‌های مجموعه‌ها را ترکیب کرده، و سپس به کمک `shapely.Polygon.centroid` مرکز جرم اشکال را حساب می‌کنیم. مرکز جرم محاسبه شده همان نیروی مورد نیاز برای نگه داشتن تعادل آونگ معکوس است.