

Mini Project #1

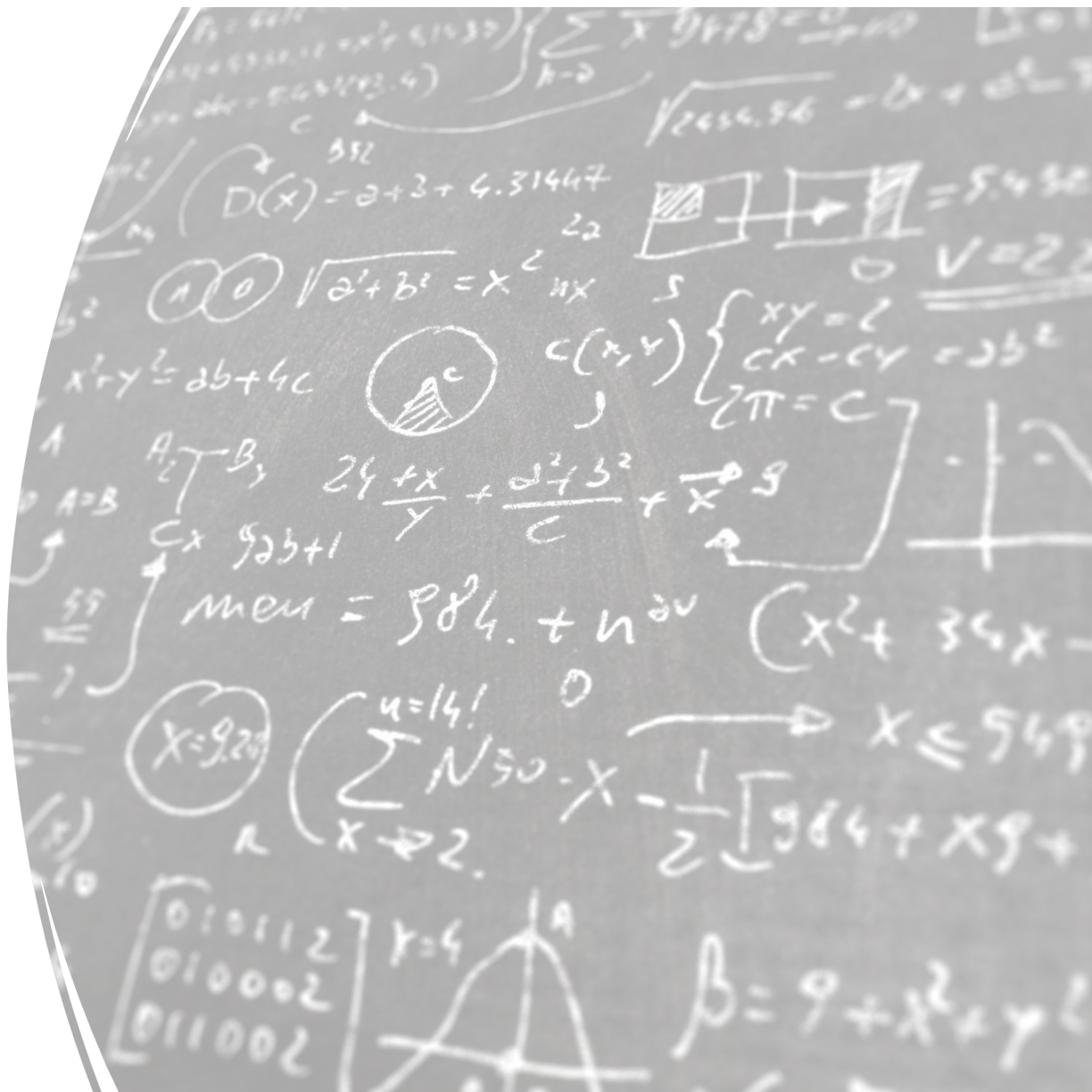
Rule-based System

만들기

2024254003 정용석

규칙 내 용

1. 식품 제조 현장에서 발생하는 **이물검침 상황**을 규칙으로 정리
2. 공통 이물 규칙: 제품 유형에 상관없이 공통적으로 감지된 이물질에 대한 규칙입니다. 예를 들어, 'wood'가 감지되면 'wood_detected' 이슈를 추가합니다.
3. 수산물 원료의 이물 검출 규칙: 수산물 원료에 대한 특정 이물질 감지를 처리합니다. 예를 들어, 'mackerel'에 대한 'fish_bone' 이물질 감지가 발생하면 'fish_bone_detected' 이슈를 추가합니다.
4. 과채류 원료의 이물물질 검출 규칙: 과채류 원료에 대한 특정 이물질 감지를 처리합니다. 예를 들어, 'yellow_peach'에 대한 'seed' 이물질 감지가 발생하면 'seed_detected' 이슈를 추가합니다.
5. 이물질 미검출 규칙: 이물질이 감지되지 않은 상황을 처리합니다. 이 경우 'no_foreign_object_detected' 이슈를 추가합니다.



코드 작성

1.

```
from durable.lang import *

with ruleset('foreign_object'):

    # 공통 이물 규칙
    # 나무 이물 규칙
    @when_all((m.product_type == 'N/A') & (m.detection_type == 'wood'))
    def detection_wood(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'wood_detected'})

    # sus 이물 규칙
    @when_all((m.product_type == 'N/A') & (m.detection_type == 'stainless_steel'))
    def detection_steel(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'stainless_steel_detected'})

    # 플라스틱 이물 규칙
    @when_all((m.product_type == 'N/A') & (m.detection_type == 'plastic'))
    def detection_plastic(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'plastic_detected'})

    # 돌맹이 이물 규칙
    @when_all((m.product_type == 'N/A') & (m.detection_type == 'stone'))
    def detection_stone(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'stone_detected'})

    # 담배 이물 규칙
    @when_all((m.product_type == 'N/A') & (m.detection_type == 'tobacco'))
    def detection_tobacco(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'tobacco_detected'})

    # 수산물의 이물 검출 규칙

    # 고등어 가시 이물 규칙
    @when_all((m.product_type == 'seafood') & (m.product == 'mackerel') & (m.detection_type == 'fish_bone'))
    def mackerel_detection_fish_bone(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'fish_bone_detected'})

    # 고등어 낚시바늘 이물 규칙
    @when_all((m.product_type == 'seafood') & (m.product == 'mackerel') & (m.detection_type == 'fishing_hook'))
    def mackerel_detection_fishing_hook(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'fishing_hook_detected'})

    # 굴껍데기 파카 이물 규칙
    @when_all((m.product_type == 'seafood') & (m.product == 'whelk') & (m.detection_type == 'shell_piece'))
    def whelk_detection_shell_piece(c):
        c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'fragmented_shell_detected'})

    # 참치 로인 가시 이물 규칙
    @when_all((m.product_type == 'seafood') & (m.product == 'skipjack_loin') & (m.detection_type == 'fish_bone'))
```

2.

```
# 참치 로인 가시 이물 규칙
@when_all((m.product_type == 'seafood') & (m.product == 'skipjack_loin') & (m.detection_type == 'fish_bone'))
def skipjack_detection_fish_bone(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'fish_bone_detected'})

# 과채류의 이물물질 검출 규칙

# 황도 씨앗 이물 규칙
@when_all((m.product_type == 'fruit') & (m.product == 'yellow_peach') & (m.detection_type == 'seed'))
def peach_detection_seed(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'seed_detected'})

# 황도 벌레 이물 규칙
@when_all((m.product_type == 'fruit') & (m.product == 'yellow_peach') & (m.detection_type == 'worm'))
def peach_detection_worm(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'worm_detected'})

# 백도 씨앗 이물 규칙
@when_all((m.product_type == 'fruit') & (m.product == 'white_peach') & (m.detection_type == 'seed'))
def peach_detection_seed(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'seed_detected'})

# 백도 벌레 이물 규칙
@when_all((m.product_type == 'fruit') & (m.product == 'white_peach') & (m.detection_type == 'worm'))
def peach_detection_worm(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'worm_detected'})

# 깽포도 씨앗 이물 규칙
@when_all((m.product_type == 'fruit') & (m.product == 'peeled_grape') & (m.detection_type == 'seed'))
def grape_detection_seed(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'seed_detected'})

# 깽포도 씨앗 이물 규칙
@when_all((m.product_type == 'fruit') & (m.product == 'peeled_grape') & (m.detection_type == 'worm'))
def grape_detection_worm(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'worm_detected'})

# 이물질 미검출 규칙
@when_all((m.detection_type == 'none'))
def no_detection(c):
    c.assert_fact({'product': c.m.product, 'detection': c.m.detection_type, 'issue': 'no_foreign_object_detected'})

@when_all((m.detection != None) & (m.issue != None))
def output_result(c):
    print('Product: {0}, Detection: {1}, Issue: {2}'.format(c.m.product, c.m.detection, c.m.issue))
```


고수 실행 3.

규칙 엔진 실행

규칙 엔진 실행 후 사실 추가

```
assert_fact('foreign_object', {'product_type': 'seafood', 'product': 'mackerel', 'detection_type': 'fish_bone'})
assert_fact('foreign_object', {'product_type': 'fruit', 'product': 'yellow_peach', 'detection_type': 'seed'})
assert_fact('foreign_object', {'product_type': 'fruit', 'product': 'white_peach', 'detection_type': 'worm'})
assert_fact('foreign_object', {'product_type': 'seafood', 'product': 'whelk', 'detection_type': 'shell_piece'})
assert_fact('foreign_object', {'product_type': 'N/A', 'product': 'N/A', 'detection_type': 'plastic'})
assert_fact('foreign_object', {'product_type': 'fruit', 'product': 'peeled_grape', 'detection_type': 'worm'})
assert_fact('foreign_object', {'product_type': 'N/A', 'product': 'N/A', 'detection_type': 'stone'})
assert_fact('foreign_object', {'product_type': 'seafood', 'product': 'mackerel', 'detection_type': 'fishing_hook'})
assert_fact('foreign_object', {'product_type': 'fruit', 'product': 'yellow_peach', 'detection_type': 'none'})
assert_fact('foreign_object', {'product_type': 'seafood', 'product': 'whelk', 'detection_type': 'none'})
assert_fact('foreign_object', {'product_type': 'N/A', 'product': 'N/A', 'detection_type': 'none'})
```



```
Product: mackerel, Detection: fish_bone, Issue: fish_bone_detected
Product: yellow_peach, Detection: seed, Issue: seed_detected
Product: white_peach, Detection: worm, Issue: worm_detected
Product: whelk, Detection: shell_piece, Issue: fragmented_shell_detected
Product: N/A, Detection: plastic, Issue: plastic_detected
Product: peeled_grape, Detection: worm, Issue: worm_detected
Product: N/A, Detection: stone, Issue: stone_detected
Product: mackerel, Detection: fishing_hook, Issue: fishing_hook_detected
Product: yellow_peach, Detection: none, Issue: no_foreign_object_detected
Product: whelk, Detection: none, Issue: no_foreign_object_detected
Product: N/A, Detection: none, Issue: no_foreign_object_detected
{'sid': '0', 'id': 'sid-0', '$s': 1}
```

- 고등어(mackerel)에서는 가시(fish_bone)가 감지되어 'fish_bone_detected' 이슈가 발생했습니다.
- 황도(yellow_peach)에서는 씨앗(seed)이 감지되어 'seed_detected' 이슈가 발생했습니다.
- 백도(white_peach)에서는 벌레(worm)가 감지되어 'worm_detected' 이슈가 발생했습니다.
- 굴뱅이(whelk)에서는 패각(shell_piece)이 감지되어 'fragmented_shell_detected' 이슈가 발생했습니다.
- 플라스틱(plastic)이 감지되어 'plastic_detected' 이슈가 발생했습니다.
- 돌맹이가(stone)이 감지되어 'stone_detected' 이슈가 발생했습니다.
- 고등어(mackerel)에서 낚시바늘(fishing_hook)이 감지되어 'fishing_hook_detected' 이슈가 발생했습니다.
- 황도(yellow_peach)와 굴뱅이(whelk)에서는 이물질이 감지되지 않아 미검출 'no_foreign_object_detected' 이슈가 발생했습니다.