SAR

XHCOME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOLD | GPT4 | GPT3.5 | GEMINI | CLAUDE |
| Location(?loc) ^ Fire(?f) ^ hasFireLocation(?f, ?loc) ^ Humidity(?h) ^ hasHumidity(?loc, ?h) ^ swrlb:lessThan(?h, 0.3) ^ WindSpeed(?ws) ^ hasWindSpeed(?loc, ?ws) ^ swrlb:greaterThan(?ws, 20) ^ Mission(?m) ^ Alert(?a) -> activates(?a, ?m) | Wildfire(?w) ^ hasLocation(?w, ?l) ^ humidity(?l, ?h) ^ windSpeed(?l, ?ws) ^ swrlb:lessThan(?h, 30) ^ swrlb:greaterThan(?ws, 20) -> Alert(?a) ^ hasAlertForMission(?a, true) | (hasWeatherConditions(?location, ?weather) ∧ hasHumidity(?weather, ?humidity) ∧ hasWindSpeed(?weather, ?windSpeed) ∧ lessThan(?humidity, 30) ∧ greaterThan(?windSpeed, 20)) → hasAlertForMission(?location) | Location(?location) ^  hasFire(?location) ^  hasHumidity(?location, ?humidity) ^  lessThan(?humidity, 30) ^  hasWindSpeed(?location, ?windSpeed) ^  greaterThan(?windSpeed, 20) ->  activateAlertForMission("missionX") | WildfireIncident(?incident) ^ hasLocation(?incident, ?location) ^ hasWeatherCondition(?incident, ?weather) ^ WeatherCondition(?weather) ^ hasWindSpeed(?weather, ?windSpeed) ^ swrlb:greaterThan(?windSpeed, 20.0) ^ hasTemperature(?weather, ?temperature) ^ swrlb:lessThan(?temperature, 30.0) -> SAR\_Mission(?mission) |

SIMXHCOME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOLD | GPT4 | GPT3.5 | GEMINI | CLAUDE |
| Location(?loc) ^ Fire(?f) ^ hasFireLocation(?f, ?loc) ^ Humidity(?h) ^ hasHumidity(?loc, ?h) ^ swrlb:lessThan(?h, 0.3) ^ WindSpeed(?ws) ^ hasWindSpeed(?loc, ?ws) ^ swrlb:greaterThan(?ws, 20) ^ Mission(?m) ^ Alert(?a) -> activates(?a, ?m) | FireIncident(?fi) ^ hasLocation(?fi, ?loc) ^ WeatherCondition(?wc) ^ hasWeatherCondition(?fi, ?wc) ^ humidity(?wc, ?hum) ^ windSpeed(?wc, ?ws) ^ swrlb:lessThan(?hum, 30) ^ swrlb:greaterThan(?ws, 20) -> AlertForMission(?fi) | ex:WildfireIncident(?incident) ∧  ex:hasLocation(?incident, ?location) ∧  ex:Humidity(?humidity) ∧  ex:hasWeatherCondition(?incident, ?humidity) ∧  ex:WindSpeed(?windSpeed) ∧  ex:hasWeatherCondition(?incident, ?windSpeed) ∧  ex:hasStatus(?incident, "on fire") ∧  swrlb:lessThan(?humidity, 30) ∧  swrlb:greaterThan(?windSpeed, 20) →  ex:hasMissionType(?mission, ex:Alert) ∧  ex:hasMission(?incident, ?mission) | hasAlert(?mission) :-  hasLocation(?fireIncident, ?location),  FireIncident(?fireIncident), hasWeatherConditions(?fireIncident, ?weatherCondition),  hasHumidity(?weatherCondition, ?humidityValue), xsd:float(?humidityValue) < 30, hasWindSpeed(?weatherCondition, ?windSpeedValue),  xsd:float(?windSpeedValue) > 20,  SARMission(?mission),  targets(?mission, ?fireIncident) | FireIncident(?fi) ^ locatedInArea(?fi, ?loc) ^ Humidity(?humidity) ^ WeatherForecast(?wf) ^ hasWeatherForecast(?fi, ?wf) ^ forecastedValues(?wf, ?humidity) ^ swrlb:lessThan(?humidity, 30.0) ^ WindSpeed(?windSpeed) ^ forecastedValues(?wf, ?windSpeed) ^ swrlb:greaterThan(?windSpeed, 20.0) -> providesService(?emergencyService, ?mission) ^ missionType(?mission, "Alert for Mission"^^xsd:string) ^ missionLocation(?mission, ?loc) |

simHCOME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOLD | GPT4 | GPT3.5 | GEMINI | CLAUDE |
| Location(?loc) ^ Fire(?f) ^ hasFireLocation(?f, ?loc) ^ Humidity(?h) ^ hasHumidity(?loc, ?h) ^ swrlb:lessThan(?h, 0.3) ^ WindSpeed(?ws) ^ hasWindSpeed(?loc, ?ws) ^ swrlb:greaterThan(?ws, 20) ^ Mission(?m) ^ Alert(?a) -> activates(?a, ?m) | Location(?l) ^ hasWeatherCondition(?l, ?wc) ^ humidity(?wc, ?h) ^ swrlb:lessThan(?h, 30)  ^ windspeed(?wc, ?ws) ^ swrlb:greaterThan(?ws, 20) -> Alert(?a) | FireLocation(?location) ∧ WeatherConditions(?weather) ∧  humidity(?weather, ?humidity) ∧ windSpeed(?weather, ?speed) ∧  ?humidity < 30 ∧ ?speed > 20 → AlertActivation(?location) | FireAlertRule( ?location ?hazard ) :-  Wildfire(?fire) relatesTo(?location) .  hasSeverity(?fire, "severe") .  Hazard(?hazard) relatesTo(?location) .  ofType(?hazard, "LowHumidity") .  hasMeasurement(?hazard, ?humidityValue) .  filter(?humidityValue < 30) .  ofType(?hazard, "HighWindSpeed") .  hasMeasurement(?hazard, ?windSpeedValue) .  filter(?windSpeedValue > 20) . | Wildfire(?w) ^ FireLocation(?l) ^ hasLocation(?w, ?l) ^  EnvironmentalCondition(?env) ^ hasEnvironmentalCondition(?l, ?env) ^  swrlb:lessThan(?env, 30.0, ?humidity) ^ swrlb:greaterThan(?env, 20.0, ?windSpeed) ->  SARMission(?mission) ^ EmergencyResponse(?response) ^ isCommandedBy(?mission, ?command) ^  isCommandedBy(?response, ?command) ^ activateAlert(?command) |

PD

XHCOME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOLD | GPT4 | GPT3.5 | GEMINI | CLAUDE |
| Observation(?obs) ^ onto:hasBradykinisiaOfUpperLimp(?obs, true) ^ sosa:observedProperty(?obs, onto:BradykinisiaUpperLimpForPDpatient) ^ onto:obsAfterDosing(?obs, true) -> onto:sendNotification(onto:MissingDoseNotification, true) ^ onto:PDpatientMissingDoseEventObservation(?obs) | PD\_Patient(?patient) ^ hasObservation(?patient, ?observation) ^ BradykinesiaObservation(?observation) ^ occursAfterMedicationTime(?observation, true) -> classifiedAs(?observation, ?missedDoseEvent) ^ MissedDoseEvent(?missedDoseEvent) ^ sendsNotification(?missedDoseEvent, ?notification) ^ Notification(?notification) | ParkinsonsDiseasePatient(?patient) ∧  hasDiagnosis(?patient, ?diagnosis) ∧  exhibitsSymptom(?patient, ?bradykinesia) ∧  hasSeverityLevel(?bradykinesia, ?severity) ∧  hasTimestamp(?bradykinesia, ?bradykinesiaTimestamp) ∧  hasDosagePlan(?patient, ?dosagePlan) ∧  hasMedication(?diagnosis, ?medication) ∧  hasTimestamp(?dosagePlan, ?dosageTime) ∧  greaterThan(?bradykinesiaTimestamp, ?dosageTime) ∧  hasSeverityLevel(?bradykinesia, ?severity), swrlb:greaterThan(?severity, 0) →  MissedDoseEventObservation(?event) ∧  hasRecord(?patient, ?healthRecord) ∧  hasMissingDoseEvent(?patient, ?event) ∧  notifiesDoctor(?notification) ∧  hasAlert(?patient, ?notification) ∧  hasSeverityLevel(?event, 3) | parkinson:MissedDoseRule(?patient ?observation) <-  # Observation indicates upper limb bradykinesia  sosa:hasResult(?observation ?result) ^  parkinson:hasSymptom(?result ?symptom) ^  parkinson:Bradykinesia(?symptom) ^  # Observation specifically targets upper limb  [ parkinson:hasLaterality ?symptom parkinson:UpperLimb ] OR  [ parkinson:hasBodyPart ?symptom parkinson:UpperLimb ]^  # Patient has Parkinson's Disease  parkinson:hasPatient(?observation ?patient) ^  parkinson:hasDisease(?patient parkinson:ParkinsonDisease) ^  # Missed medication timeline  EXISTS (?medicationSchedule ?medicationTime) [  parkinson:hasTreatmentPlan(?patient ?medicationSchedule) ^  parkinson:hasMedication(?medicationSchedule ?medication) ^  parkinson:intendedIntakeTime(?medicationSchedule ?medicationTime) ^  xsd:dateTime(?observation) > ?medicationTime | Observation(?o), Patient(?p), Assessment(?a), hasAssessment(?p, ?a), hasObservation(?a, ?o),  observesSymptom(?o, UpperLimbBradykinesia), hasRecommendedTreatment(?o, ?t), Treatment(?t),  hasMedicationPrescription(?t, ?m), MedicationPrescription(?m), observationDate(?o, ?odate),  lastUpdate(?a, ?ldate), swrlb:greaterThan(?odate, ?ldate) ->  triggersAlert(?o, ?alert), Alert(?alert), alertMessage(?alert, "Missed medication dose"),  alertSeverity(?alert, High) |

SIM-XHCOME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOLD | GPT4 | GPT3.5 | GEMINI | CLAUDE |
| Observation(?obs) ^ onto:hasBradykinisiaOfUpperLimp(?obs, true) ^ sosa:observedProperty(?obs, onto:BradykinisiaUpperLimpForPDpatient) ^ onto:obsAfterDosing(?obs, true) -> onto:sendNotification(onto:MissingDoseNotification, true) ^ onto:PDpatientMissingDoseEventObservation(?obs) | Patient(?p) ^ hasObservation(?p, ?o) ^ BradykinesiaObservation(?o) ^ hasBodyPart(?o, UpperLimb) ^ hasDisease(?p, ParkinsonDisease) ^ supposedToTakeMedicationBefore(?p, ?time) ^ hasObservationTime(?o, ?obsTime) ^ swrlb:greaterThan(?obsTime, ?time) -> SendNotification(?p, MissedDose) ^ ClassifiedAs(?o, MissedDoseEvent) | Patient(?p) ^ Observation(?o) ^ observedIn(?o, ?p) ^ observedSymptom(?o, :Bradykinesia) ^ hasObservationTime(?o, ?time) ^ Medication(?m) ^ takingMedication(?p, ?m) ^ after(?time, ?medTime) →  MissedDoseNotification(?o) | parkinson:hasObservation(?patient, ?obs) ^  parkinson:hasSymptom(?obs, ?symptom) ^  parkinson:ParkinsonSymptom(?symptom) ^  parkinson:affectsBodyPart(?symptom, parkinson:UpperLimb) ^  parkinson:hasPatient(?obs, ?patient) ^  parkinson:Patient(?patient) ^  parkinson:hasMedicationSchedule(?patient, ?schedule) ^  (?obsTime > ?scheduleTime) ^ # obsTime after scheduleTime  xsd:dateTime(?obsTime) ^ xsd:dateTime(?scheduleTime) ^  parkinson:alertType(?obs, parkinson:Phone) ->  parkinson:hasMissedDose(?patient) ^  parkinson:isMissedDoseObservation(?obs) | BradykinesiaObservation(?ob) ^     Observation.forSymptom(?ob, Bradykinesia) ^     Observation.forBodyRegion(?ob, UpperLimb) ^     Patient.hasObservation(?pat, ?ob) ^     Patient.takenDose(?pat, ?dose) ^     Observation.observationTime(?ob, ?obTime) ^     DoseEvent.doseScheduledTime(?dose, ?doseTime) ^     swrlb:greaterThan(?obTime, ?doseTime)    -> MissedDoseEvent(?ob) ^     triggersNotification(?ob, true) |

SIM-HCOME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOLD | GPT4 | GPT3.5 | GEMINI | CLAUDE |
| Observation(?obs) ^ onto:hasBradykinisiaOfUpperLimp(?obs, true) ^ sosa:observedProperty(?obs, onto:BradykinisiaUpperLimpForPDpatient) ^ onto:obsAfterDosing(?obs, true) -> onto:sendNotification(onto:MissingDoseNotification, true) ^ onto:PDpatientMissingDoseEventObservation(?obs) | Patient(?p) ^ Observation(?o) ^ observes(?p, ?o) ^ Bradykinesia(?b) ^ hasSymptom(?p, ?b) ^ observationTime(?o, ?time) ^ MedicationIntakeObservation(?mo) ^ observationTime(?mo, ?medTime) ^ swrlb:greaterThan(?time, ?medTime) -> Notification(?n) ^ sendsNotification(?o, ?n) | Patient(?p) ^ Observation(?o) ^ observedIn(?o, ?p) ^ observedSymptom(?o, :Bradykinesia) ^ hasObservationTime(?o, ?time) ^ Medication(?m) ^ takingMedication(?p, ?m) ^ after(?time, ?medTime) → MissedDoseNotification(?o) | parkinson:hasObservation(?patient, ?obs) ^  parkinson:hasSymptom(?obs, ?symptom) ^  parkinson:ParkinsonSymptom(?symptom) ^  parkinson:affectsBodyPart(?symptom, parkinson:UpperLimb) ^  parkinson:hasPatient(?obs, ?patient) ^  parkinson:Patient(?patient) ^  parkinson:hasMedicationSchedule(?patient, ?schedule) ^  (?obsTime > ?scheduleTime) ^ # obsTime after scheduleTime  xsd:dateTime(?obsTime) ^ xsd:dateTime(?scheduleTime) ^  parkinson:alertType(?obs, parkinson:Phone) ->  parkinson:hasMissedDose(?patient) ^  parkinson:isMissedDoseObservation(?obs) | Patient(?p), hasObservation(?p, ?o), Bradykinesia(?o), involvesBodyPart(?o, UpperLimb),  hasObservationTime(?o, ?time), hasMedication(?p, ?med), hasSchedule(?med, ?schedule),  after(?time, ?schedule)  ->  Alert(?a), classifiedAs(?o, MissedDose), sendNotification(?a, ?p) |