

Wissensverarbeitung in der Landwirtschaft mit regelbasierten Inferenzsystemen und Begründungsverwaltung

Nils Niemann¹, Christoph Tieben¹, Kai Lingemann¹ und Joachim Hertzberg¹

Abstract: In der modernen landwirtschaftlichen Praxis existiert eine große Menge an Expertenwissen sowie Informationen und Regularien zu komplexen Prozessen, welche ein Landwirt zur Entscheidungsfindung berücksichtigen muss. Durch ein regelbasiertes Inferenzsystem, welches die Möglichkeit bietet, Wissen zu modellieren und automatisch Schlussfolgerungen zu ziehen und bei Bedarf zurückzuziehen, sollen die Landwirte bei dessen Anwendung unterstützt werden.

Keywords: Wissensverarbeitung, Entscheidungsunterstützungssystem, RETE, automatische Schlussfolgerung

1 Einleitung

Durch die Verknüpfung von sowohl semantischen als auch geometrischen/geographischen Daten in einem regelbasierten Inferenzsystem soll die Entwicklung von Entscheidungsunterstützungssystemen ermöglicht werden, in denen diese vielfältigen Wissensquellen zusammenfließen. Dadurch kann der Landwirt dabei unterstützt werden, umfassenden Nutzen aus dem vorhandenen Wissen zu ziehen. Mögliche Anwendungsszenarien finden sich beispielsweise in teilflächenspezifischen Anbaustrategien (Düngen, Pflanzenschutz, ...) auf Basis von Überflugdaten, in der Klassifikation einzelner Spontanvegetation in gewünschte/tolerierbare/schädigende Beikräuter, oder in der Herleitung von Handlungsempfehlungen im einzeltierspezifischen Gesundheitsmonitoring. Für die Akzeptanz solcher Systeme ist zudem eine Begründungsverwaltung von besonderer Bedeutung, durch die die getroffenen Entscheidungen nachvollzogen werden können. Neben der Unterstützung zur Entscheidungsfindung für den menschlichen Benutzer soll es auch möglich sein, Problem-spezifische Applikationskarten zu generieren, die von Landmaschinen und Agrarrobotern verwendet werden können.

¹ DFKI-Labor Niedersachsen, Planbasierte Robotersteuerung, Berghoffstraße 11, 49090 Osnabrück, vorname.nachname@dfki.de, <https://dfki.de/pbr>

Das DFKI Labor Niedersachsen (DFKI NI) wird gefördert im Niedersächsischen Vorab durch das Niedersächsische Ministerium für Wissenschaft und Kultur und die VolkswagenStiftung. Die Arbeit wird gefördert durch das Bundesministerium für Ernährung und Landwirtschaft (FKZ: 28DE103E18) und das Bundesministerium für Bildung und Forschung (FKZ: 031B0729B).

2 Stand der Technik

Die Idee Expertensysteme in der Landwirtschaft zu verwenden ist schon frühzeitig diskutiert worden [ML85] und wurde im Laufe der Zeit in unterschiedlichen landwirtschaftlichen Kontexten angewandt [Ra98]. Eine Vielzahl der Systeme waren rein symbolisch und sehr anwendungsspezifisch. Jüngere Arbeiten beschäftigen sich mit dem Austausch des Wissens durch Methoden vom Semantic Web [BM11] oder der besseren Zugänglichkeit sowie das Einbeziehen von Geometrien innerhalb eines Geo-Information-Systems (GIS) [Ka02].

Ein Ansatz, diese beiden Ausrichtungen zu vereinen, wird im SEMAP Framework [DWH18] verfolgt, indem die geographisch räumliche Datenbank PostGIS mit semantisch annotierten Informationen, unter Verwendung von Apache Jena, verknüpft wurde. In dieser Kombination ist es möglich, räumliche Beziehungen sowie Operationen mit semantischem Wissen zu vereinen. Ein vergleichbares Konzept wird ebenfalls im SOMA-Framework [Ku18] beschrieben.

3 Regelbasiertes Inferenzsystem

In dieser Arbeit stellen wir ein semantisches Umgebungsmodell namens SEMPR (**S**emantic **E**nvironment **M**apping, **P**rocessing and **R**easoning) vor, welches sich durch einen integrierten regelbasierten Reasoner auszeichnet, und mit dessen Hilfe sich Expertensysteme für verschiedenste Anwendungen realisieren lassen.

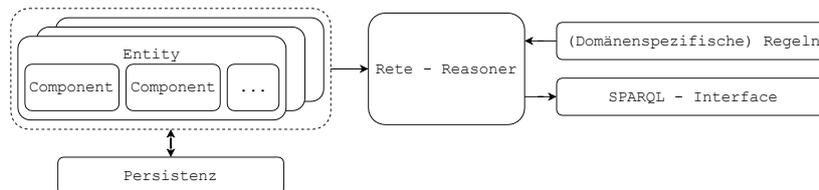


Abb. 1: SEMPR Systemübersicht

Eine Übersicht über SEMPR ist in Abb. 1 dargestellt. Das System setzt auf eine „Entity-Component-System“-Architektur, bei der jegliche Daten in Form von Komponenten implementiert werden. Diese werden dann in Entitäten, die lediglich eine eindeutige ID bereitstellen, gruppiert, um komplexe Objekte zu erzeugen. Auf diese Weise können verschiedene Daten miteinander verknüpft werden, beispielsweise semantische und geometrische Informationen, für die jeweils eine eigene Komponente verwendet wird.

Der „System“-Teil der Architektur beinhaltet die Verarbeitung der vorhandenen Daten und wird hier durch einen regelbasierten Reasoner umgesetzt. Dieser erlaubt den Einsatz von frei spezifizierbaren und damit auf die Anwendungsdomäne angepassten Regeln, wodurch eine große Flexibilität des Systems erreicht wird.

Für den Reasoner geschriebene Regeln folgen einem einfachen Format:

```
[Regelname: Vorbedingungen -> Konsequenzen]
```

Neben einem optionalen Namen der Regel ist eine Liste von Bedingungen sowie eine Liste von Konsequenzen zu spezifizieren. Die Bedingungen verstehen sich dabei als ein Muster, das in der Wissensbasis gefunden werden muss, um daraus die gegebenen Konsequenzen zu ziehen. Die Aufgabe des Reasoners ist es, alle möglichen Kombinationen von Daten zu finden, die auf die Bedingungen einer Regel passen, die entsprechenden Konsequenzen umzusetzen und die dadurch ggf. neu inferierten Daten in die Suche mit einzubeziehen.

Dazu wird ein RETE-Pattern-Matching-Algorithmus [Fo89] eingesetzt. Dieser stellt aus den gegebenen Mustern einen gerichteten Graphen zusammen, in dem jeder Knoten nur einen kleinen Teil des Musters überprüft. Die Daten werden über den Wurzelknoten in das Netzwerk gegeben. Fällt der Test eines Knoten auf den ihm gegebenen Daten positiv aus, leitet er sie an seine Nachfolger weiter und hält sie anderenfalls zurück. In den äußersten Randknoten finden sich dann die gefundenen Kombinationen zur Aktivierung der Regeln. Vorteile dieser Vorgehensweise finden sich darin, dass Teile des Graphen für verschiedene Regeln mit denselben Mustern wiederverwendet werden können. Zudem ermöglicht der Einsatz von Speicherknoten, welche Zwischenergebnisse im Graphen sammeln, inkrementelle Aktualisierungen: Werden Daten hinzugefügt oder entfernt, ist es nicht nötig, alle Regeln neu zu evaluieren. Stattdessen werden nur die relevanten Bereiche des Rete-Netzwerks aktualisiert. Der Reasoner arbeitet dadurch allerdings auch nach dem Prinzip des „Forward-Chaining“, d.h. dass beginnend bei den bekannten Fakten alle Regeln ausgewertet werden und neu hergeleitetes Wissen dem hinzugefügt wird, bis alle möglichen Schlussfolgerungen gezogen wurden – unabhängig davon, ob das Wissen gerade benötigt wird. Dies ermöglicht es, Regeln zum Monitoring einzusetzen und automatisch externe Systeme zu informieren, wenn bestimmte Bedingungen erfüllt sind.

Bei der Aktivierung einer Regel werden nicht nur die angegebenen Konsequenzen gezogen, sondern auch der Zusammenhang zwischen den Daten, mit denen ihre Vorbedingungen erfüllt wurden, der Regel selbst, und dem neu inferierten Wissen in einem Inferenzgraphen hinterlegt. Dieser wird verwendet, um nicht-monotone Änderungen effizient umsetzen zu können: Soll ein zuvor zugesicherter Fakt aus der Wissensbasis entfernt werden, wird zuvor geprüft, ob dieser nicht durch andere Fakten und Regeln hergeleitet werden kann und dadurch seine Gültigkeit behält. Weiter werden selbsterhaltene Zyklen im Inferenzgraphen gesucht und entfernt. Dadurch wird sichergestellt, dass jegliches abgeleitetes Wissen vollständig auf zugesicherten Fakten basiert.

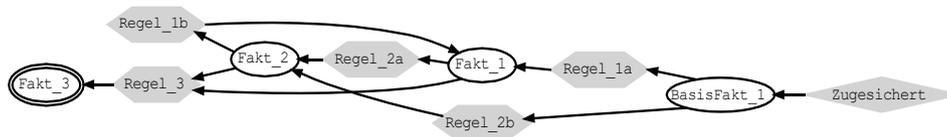


Abb. 2: Beispiel eines Inferenzgraphen. Fakten (Ellipsen) führen durch die Verknüpfung mit Regeln (Hexagone) zu weiteren Fakten.

Die Verwaltung des Inferenzgraphen bedeutet einerseits einen zusätzlichen Aufwand, bringt aber andererseits auch nützliche Vorteile mit sich: Neben den effizienteren Aktualisierungen nach dem Entfernen von Daten bietet er auch die Möglichkeit, die Herkunft des abgeleiteten Wissens nachzuverfolgen, um dieses zu erklären. Ein Teil eines Inferenzgraphen ist in Abb. 2 zu sehen. Die ovalen Knoten repräsentieren dabei einzelne Datenelemente, aus denen durch Regel-Effekte (graue Hexagone) neue Daten abgeleitet wurden. Der doppelt umrandete Knoten ist jener, für dessen Erklärung diese Darstellung generiert wurde. Der Weg seiner Herleitung lässt sich bis zu den externen Zusicherungen, abgebildet als grauer Diamant auf der rechten Seite des Graphen, verfolgen.

```
[ (?k <type> ?typ), (?typ <subClassOf> <Kraut>),  
  (?typ <hatUnkrautPotenzial> "mittel"),  
  noValue { (?osl <type> <OSLPotenzial>),  
            (?osl <vonKraut> ?typ) },  
  noValue { (?k <ignore> "ja") }  
->  
  (?k <istMassnahmeErforderlich> "ja") ]  
  
[ (?k <type> ?typ), (?typ <subClassOf> <Kraut>),  
  (?typ <hatUnkrautPotenzial> "mittel"),  
  (?k <inBBCH> ?bbch), ge(?bbch 70),  
  noValue { (?k <ignore> "ja") }  
->  
  (?k <istMassnahmeErforderlich> "ja") ]
```

Abb. 3: Beispiel für symbolische Inferenz. Pflanzen mit mittlerem Unkrautpotential werden bekämpft, wenn sie keinen ökologischen Nutzen haben oder bereits verblüht sind ($BBCH \geq 70$).

4 Exemplarische Anwendung

In der Landwirtschaft gibt es viele Bereiche, in denen eine Unterstützung durch Expertensysteme wünschenswert ist. So könnte beispielsweise veterinär-medizinisches Wissen modelliert werden, um es systematisch in einem einzeltierspezifischen Gesundheitsmonitoring anzuwenden und Handlungsempfehlungen zur Verbesserung der Tiergesundheit zu erzeugen. Aber auch auf pflanzenkundlicher Seite lassen sich Anwendungsgebiete finden. So könnte etwa die Modellierung von Wissen über Un- und Beikräuter helfen, ein angemessenes und ökologisches Unkrautmanagement umzusetzen. Angenommen, durch die sensorische Erfassung eines Feldes, zum Beispiel durch mobile Roboter oder Drohnen, werden einzelne Kräuter oder Gebiete mit vermehrtem Aufkommen eines Un-/Beikrauts bestimmt. Mit in Regeln formalisiertem, pflanzenkundlichem Wissen können dann aus zusätzlichen Daten wie der Jahreszeit, Wetterdaten, der angebauten Nutzpflanze, der Art des Betriebs, der Verfügbaren Gerätschaften, oder

auch geographischen Gegebenheiten abgeleitet werden, ob und welche Art von Unkrautmanagement eingesetzt werden soll.

In einem einfachen Beispiel dazu werden die Kräuter zunächst nach ihrem Unkrautpotenzial klassifiziert, d.h. nach ihrer schädlichen Wirkung auf die Nutzpflanze und den Ertrag, ihrer Reproduktionsrate, etc. Ob bereits gegen die jeweilige Pflanze vorgegangen werden muss, kann dann von ihrem Unkrautpotenzial, ihrem Wachstumsstadium und ihrem ökologischen Nutzen abhängig gemacht werden. Mit den Regeln in Abb. 3 werden Kräuter mit mittlerem Unkrautpotenzial und ökologischem Nutzen erst nach ihrer Blüte bekämpft. Die Struktur der Implementierung dieser Regeln in einem Rete-Netzwerk ist in Abb. 4 zu sehen: Ellipsen repräsentieren darin einzelne Tests auf den Daten, während Rechtecke für die eingesetzten Speicher-knoten stehen. Der Start-Knoten auf der linken sowie die Effekte der Regeln auf der rechten Seite sind dunkelgrau hervorgehoben. Viele Knoten können für beide Regeln wiederverwendet werden, was dadurch zu erkennen ist, dass von ihnen Pfade zu beiden Effekten führen.

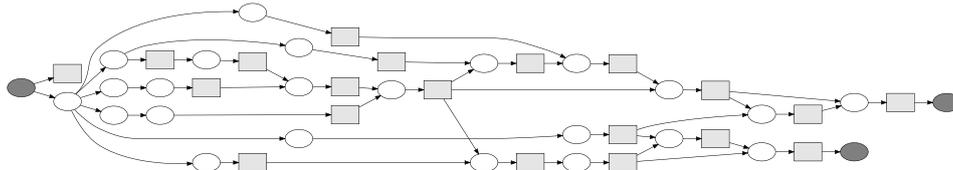


Abb. 4: Beispiel für die Struktur eines Rete-Netzwerkes, generiert aus den Regeln in Abb. 3.

Durch ähnliche Regeln lassen sich auf diese Weise auch die Art der Bekämpfung anhand verschiedener Parameter bestimmen und beispielsweise noch sehr junge Pflanzen durch wenig intensives mechanisches Striegeln behandeln, während bei älteren und schädlicheren Kräutern auf rigorosere Maßnahmen wie Hacken oder Herbizideinsatz zurückgegriffen wird.

```
[
  (?g <type> <Gewaesser>), EC<Geo>(?g ?gg),
  geo:UTMFromWGS(?ggUTM ?gg 32),
  (?k <type> <Kraut>),      EC<Geo>(?k ?kg),
  geo:UTMFromWGS(?kgUTM ?kg 32),
  geo:distance(?dist ?kgUTM ?ggUTM), le(?dist 20)
->
  (?k <ignore> "ja") ]
```

Abb. 5: Beispiel für den Einbezug von Geometrien. Un-/Beikräuter in ≤ 20 m Entfernung zu Gewässern werden bei der Bekämpfung ignoriert. EC<Geo> referenziert Geometrien, über geo:UTMFromWGS in UTM transformiert, mit Distanzbestimmung zueinander über geo:distance.

Da in dem hier vorgestellten System die erkannten Pflanzen neben ihrer semantischen Klassifizierung auch mit geometrischen Daten behaftet sein können, lassen sich nach der Ableitung der auszuführenden Handlungen Applikationskarten für die verschiedenen Maßnahmen erzeugen. Dadurch können Unkräuter gezielt und angemessen bekämpft werden, wodurch der Einsatz von Herbiziden reduziert und gleichzeitig der ökologische

Nutzen der Beikräuter erhalten bleibt, solange diese gemäß dem modellierten Wissen tolerierbar sind. Dabei kann auch auf die geographischen Gegebenheiten Rücksicht genommen werden, indem etwa mit der Regel in Abb. 5 alle Kräuter in der Nähe von Gewässern aus der Bearbeitung ausgeschlossen werden, da dort zum Schutz des Gewässers auf Pflanzenschutzmittel verzichtet werden soll und zudem die Gerätschaften zur mechanischen Unkrautbekämpfung in unmittelbarer Nähe zum Ufer schwer einsetzbar sind.

5 Ausblick

Wir haben gezeigt, dass ein regelbasiertes Inferenzsystem eine gute Grundlage zur Entwicklung eines Expertensystems im landwirtschaftlichen Bereich sein kann. Die gezeigte Verknüpfung mit geometrischen Daten und Operationen trägt zu einer flexiblen und ausdrucksstarken Wissensmodellierung bei. Die durch den Inferenzgraphen realisierte Begründungsverwaltung liefert dabei die wichtige Möglichkeit, dem Anwender die Herkunft des inferierten Wissens zu erläutern und Handlungsempfehlungen zu begründen. Für weitere Anwendungsgebiete und Szenarien, die über das im vorherigen Abschnitt dargestellte Beispiel hinaus gehen, kann es zudem notwendig sein, zusätzlich Optimierungsalgorithmen einzusetzen, um „die beste“ Lösung/Handlungsempfehlung für eine Situation zu finden. Die dadurch ermöglichte automatisierte Anwendung von Expertenwissen und die vielfältigen Einsatzmöglichkeiten des Systems eröffnen Wege, die moderne landwirtschaftliche Praxis digital voranzutreiben.

Literaturverzeichnis

- [BM11] Bansal, N.; Malik, S. K.: A framework for agriculture ontology development in semantic web: ICCNT, 2011; S. 283–286.
- [DWH18] Deeken, H.; Wiemann, T.; Hertzberg, J.: Grounding semantic maps in spatial databases. In *Robotics and Autonomous Systems*, 2018, 105; S. 146–165.
- [Fo89] Forgy, C. L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. In (Mylopoulos, J. Hrsg.): *Readings in artificial intelligence and databases*. Kaufmann, San Mateo, Calif., 1989; S. 547–559.
- [Ka02] Kalogirou, S.: Expert systems and GIS: an application of land suitability evaluation. In *Computers, environment and urban systems*, 2002, 26; S. 89–112.
- [Ku18] Kunze, L. et al.: Soma: A framework for understanding change in everyday environments using semantic object maps. In *AAAI Fall Symposium on Reasoning and Learning in Real-World Systems on Long-Term Autonomy*, 2018; S. 47–54.
- [ML85] McKinion, J. M.; Lemmon, H. E.: Expert systems for agriculture. In *Computers and Electronics in Agriculture*, 1985, 1; S. 31–40.
- [Ra98] Rafea, A.: Expert system applications: Agriculture. In *Central Laboratory for Agricultural Expert Systems*, PO Box 100, 1998.