

Constraint Satisfaction Problem

Aus Zusammenfassung

- Auf den ersten Blick würde man denken es hat nichts mit den ersten beiden Punkten zu tun. Das CSP ist primär ein Problem über nur einzige Entscheidung, und zwar "eine Karte anzufertigen oder alle Variablen zu beziffern".
- Das CSP ist im Grunde genommen nicht sequenziell, sondern es ist nach einer Komplettlösung gefragt.
- Um eine Entscheidung zu finden ist der Standardansatz das Sequenzialisieren - **sequential assignment**.
- Es gäbe auch andere Methodiken, die eher auf (parallelen) Optimierungs- oder Interferenzprozessen basieren, aber der Standardansatz ist, das Problem sequenziell zu zerlegen.
- Das bedeutet, dass man zum Beispiel Werte der Karte sequenziell zuweist. Hier ist die Lösung **Backtracking** und **Treesearch**.
- Methodisch macht man dann Constraint Propagation, was sehr eng mit dynamischem Programmieren verwandt ist, weil bei CP die Funktion, mit der wir jetzt die Variablen festlegen, die Indikatorfunktion ist, die aussagt welche Belegungen überhaupt noch erlaubt sind.
- Die Constraint Satisfaction nutzt die Constraints der benachbarten Variablen um die Belegungen aus dem Raum der möglichen Werte herauszustreichen, um herauszufinden welche Variablen für dieses Feld denn noch möglich sind.
- Diese Propagation ist sehr ähnlich zum dynamischen Programmieren.

Inferenz

Given some pieces of information on some things (observed variables, prior, knowledge base) what is the implication (the implied information, the posterior) on other things (non-observed variables, sentence)

Entscheidungen können als Inferenz betrachtet werden: Das Wissen, das wir über das Spiel haben ist die Prior, unsere Entscheidung die Posterior.

Wir behandeln in dieser Vorlesung:

- "Deterministische" Inferenz in CSPs
- Probabilistische Inferenz in [Graphischen Modellen](#)
- Logische Inferenz in "propositional & FO logic"

Generelles zu CSP

CSPs sind keine sequentiellen Probleme

Aufbau:

- Es gibt n Variablen x_i jeweils mit einer Domain D_i , $x_i \in D_i$
- Es gibt außerdem k Constraints C_k , die jeweils erlaubte Kombinationen von manchen der Variablen konfigurieren.
- Gesucht ist eine Lösung $C = (x_1, \dots, x_n)$ alle Variablen definiert und alle Constraints erfüllt.

Constraint Graph

- Zeichne jede Variable als Kreis.
- Zeichne alle Constraints zwischen den Variablen als Boxen.

Lösungsansätze

Sequential Assignment

"Let's start with the straight-forward, stupid approach, then fix it"

- Beginne mit allen Variablen unassigned
- Ordne einer Variable einen erlaubten Wert zu
- Wenn du durchkommst, ist alles gut, wenn nicht, fange nochmal an.

Backtracking Sequential Assignment

- Die Pfadzahl wird verringert, indem eine Reihenfolge zur Belegung festgelegt wird.

Rekursives Backtracking

- Bekommt eine Liste aller bereits zugewiesenen Variablen
- wähle eine noch nicht zugewiesene variable
- Probiere alle möglichen Variablen durch, indem du rek. Backtracking durchführst
- gebe die Variable zurück, wenn "es aufgeht", gebe **failure** zurück, wenn es keine Lösung gibt.

Vergleiche: 4 Queens Ballett

Heuristiken

Name	Strategie
Minimum Remaining Values	choose the variable with the fewest legal values
Degree Heuristik	choose the variable with the most constraints on remaining values
Least constraining values	given variable, choose the least constraining value
Constraint Propagation	see below

Constraint Propagation: Rekursiv

- Nach jeder Entscheidung berechnen wir, welche Werte für die Variablen noch möglich sind.
- So lange wiederholen, bis sich nichts mehr verändert.

Prüfungsrelevant ist vor Allem

- Backtracking und die Heuristiken, die wir gelernt haben.
- "Sie sind in einer gegebenen Situation, die sie mit Backtracking/CSP lösen. Welche Variablen picken sie sich raus?" --> testet Variable Order (Min. Remaining Value etc.)
- Constraint Propagation: "Hier sind die Variablen, die sie kennen und die Constraints, die sie erfüllen, angenommen Sie führen jetzt Constraint Propagation aus ohne Backtracking. Wohin konvergiert das? Was sind die verbleibenden Domains und Wertelängen?"