And–or tree

From Wikipedia, the free encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
فارسی
Српски / srpski
Edit links

An **and–or tree** is a graphical representation of the reduction of problems (or goals) to conjunctions and disjunctions of subproblems (or subgoals).

Contents

Example [edit]

The and-or tree:

Andortree.png

represents the search space for solving the problem P, using the goal-reduction methods:

P if Q and R

P if S

Q if T

Q if U

Definitions [edit]

Given an initial problem P0 and set of problem solving methods of the form:

P if P1 and … and Pn

the associated and-or tree is a set of labelled nodes such that:

1. The root of the tree is a node labelled by P0.
2. For every node N labelled by a problem or sub-problem P and for every method of the form P if P1 and … and Pn, there exists a set of children nodes N1, …, Nn of the node N, such that each node Ni is labelled by Pi. The nodes are conjoined by an arc, to distinguish them from children of N that might be associated with other methods.

A node N, labelled by a problem P, is a success node if there is a method of the form P if nothing (i.e., P is a "fact"). The node is a failure node if there is no method for solving P.

If all of the children of a node N, conjoined by the same arc, are success nodes, then the node N is also a success node. Otherwise the node is a failure node.

Search strategies [edit]

An and-or tree specifies only the search space for solving a problem. Different search strategies for searching the space are possible. These include searching the tree depth-first, breadth-first, or best-first using some measure of desirability of solutions. The search strategy can be sequential, searching or generating one node at a time, or parallel, searching or generating several nodes in parallel.

Relationship with logic programming [edit]

The methods used for generating and-or trees are propositional logic programs (without variables). In the case of logic programs containing variables, the solutions of conjoint sub-problems must be compatible. Subject to this complication, sequential and parallel search strategies for and-or trees provide a computational model for executing logic programs.

Relationship with two-player games [edit]

And–or trees can also be used to represent the search spaces for two-person games. The root node of such a tree represents the problem of one of the players winning the game, starting from the initial state of the game. Given a node N, labelled by the problem P of the player winning the game from a particular state of play, there exists a single set of conjoint children nodes, corresponding to all of the opponents responding moves. For each of these children nodes, there exists a set of non-conjoint children nodes, corresponding to all of the player's defending moves.

For solving game trees with proof-number search family of algorithms, game trees are to be mapped to And/Or trees. MAX-nodes (i.e. maximizing player to move) are represented as OR nodes, MIN-nodes map to AND nodes. The mapping is possible, when the search is done with only a binary goal, which usually is "player to move wins the game".

Bibliography [edit]

- Luger, George F.; Stubblefield, William A. (1993). *Artificial intelligence: structures and strategies for complex problem solving* (2 ed.). The Benjamin/Cummings. ISBN 978-0-8053-4785-2. Retrieved 28 February 2013.
- Nilsson, Nils J. (1998). *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann. ISBN 978-1-55860-467-4. Retrieved 28 February 2013.

Categories: Trees (data structures) | Artificial intelligence