APPENDIX A

The references to the material listed here are:

[Craig 1988], [Dieterrich 1986],
[DeJong 1988], [Falkanhainer and Rajamooney 1988],
[Hall 1988], [Harmelen and Bundy 1988],
[Hunter 1989], [Kaplan and Simon 1988],
[Kulkarni and Simon 1986], [Laird et.al. 1986],
[Langley 1985], [Langley and Jones 1986],
[Lebowitz 1986], [Lenat 1982],
[Lenat 1983], [Michalski 1980],
[Michalski 1983], [Punch et.al 1990],
[Rose and Langley 1988]
[Star 1989], [Utgoff 1986],


(clear)
(set-cursor 1 35)
(princ "SRISHTI")
(terpri)
(setq property-list '(what-is-it evaluation-criteria types-of requirements general-principles basics-of features stages-in heading applicability-conditions method-alg-heuristic operators fundamental-operators interesting-subset-of-category constraints-problems representation links-between-relationship solutions-to-a-problem tasks-of important-examples-of subcomponents research-paradigms equal-namewise generates general-paradigm cause superior-case-of ))

;;;; concept ****
(putprop 'concept-description 'consists-of-relevant-variable-bindings 'basics-of-1)
(putprop 'concept 'corresponds-to-descriptive-predicates 'features-1)
(putprop 'concept 'a-new-concept-may-arise-because new-predicates-are-required-to-specify-theory 'features-2)

;;;; heuristic ****
(putprop 'more-specific-heuristic '(applicability-conditions-are-matched identical-recommendations-in-all-applicable-cases)
 'what-is-it-1)
(putprop 'bad-heuristics '(not-interpretable-at-all but-never-potentially-relevant but-antecedent-never-satisfied triggers-but-never-selected-for-execution fires-but-consequent-has-no-effect-on-situation produces-bad-effects-on-situation produces-good-effects-but-not-cost-effective-for-implementation) 'evaluation-criteria-1)
(putprop 'new-heuristics '( specialisation-nh generalisation-nh analogy-nh ) 'method-alg-heuristic-1)
(putprop 'specialisation-nh '(adapt bind-variables match-template-to-data find-exceptions) 'method-alg-heuristic-1)
(putprop 'generalisation-nh 'abstraction 'types-of-1)
(putprop 'analogy 'favoured-to-generate-new-heuristics 'features-1)
(putprop 'heuristic 'should-specify-a-situation-or-context-in which-the-suggested-actions-are-especially-appropriate-and-also-those-contexts-where-the-actions-are-inappropriate 'features-1)
predictive features

features-likely-to-be-causes-in-a-causal-explanation

features-likely-to-be-causes-in-a-causal-explanation

provides-a-focus-for-application-of-general-explanatory-rules

features-of-a-generalisation-that-are-most-nearly-unique-to-the-generalisation

indicate-applicability

predictive-rules

rules-that-indicate-presence-of-one-feature-cause-presence-of-another

rules-that-indicate-presence-of-one-feature-cause-presence-of-another

presence-of-one-feature-used-to-explain-presence-of-another-feature

known-absence-of-one-feature-used-to-explain-absence-of-another-whose-presence-would-have-forced-presence-of-first-feature

closed-world-domain

counter example

although-assumed-conditions-were-satisfied-and-actions-were-successfully-executed-the-predicted-effects-failed-to-materialise-in-example

understand

when-one-shows-proper-usage-of-the-concept-basics-of-1

situation

world-state-spanning-an-interval-of-time-in-which-some-observable-event-occurs

human learning

human-learning

events causative-actions basic-concepts advanced-concepts

events causative-actions

causal-corpus

basic-concepts

naive-physics

advanced-concepts

expert-models
(putprop 'naive-physics 'processes 'equal-namewise-1)
(putprop 'expert-models 'quantitative-representations 'equal-namewise-1)
(putprop 'human-learning '(literal-similarity mere-appearance-analogy abstraction) 'stages-in-2)
(putprop 'human-learning '(Chomsky-claims-that-there-are-no-general-principles-of-human-learning-Each-domain-has-its-own-particular-constraints-The-search-therefore-is-for-a-domain-independent-theory-consisting-of-general-principles-supplementing-the-constraints-of-each-domain) 'general-principles-of-1)
(putprop 'human-learning 'proof-tree-is-not-explicit 'basics-of-1)
(putprop 'proof-tree 'plan 'what-is-it)
(putprop 'human-learning 'uses-both-backward-and-forward-search 'basics-of-2)
(putprop 'human-learning 'analogy-in-students-is-superficial-and-is-restricted-to-problems-in-the-same-section-of-a-chapter 'basics-of-2)

****** machine learning ******
(putprop 'learning-strategies "learning is constructing or modifying representations of what is being experienced" 'what-is-it-1)
(putprop 'learning-strategies al 'types-of-1)
(putprop 'learning-strategies '( validity effectiveness abstraction-level) 'evaluation-criteria-1)
(putprop 'learning-strategies '(neural-modelling symbolic-concept-acquisition knowledge-intensive-domain-specific-learning) 'research-paradigms-1)
(putprop 'learning-strategies '( require-link subset-principle ) 'general-principles-1)
(putprop 'require-link 'a-descriptor-modification-introduced-when-a-negative-example-establishes-that-a-particular-feature-must-be-present 'what-is-it-1)
(putprop 'subset-principle 'if-possible-guesses-can-be-arranged-
in-a-subset-relationship-then-the-learner-should-
make-the-smallest-possible-guess-about-what-he-should-learn
'what-is-it-1)
(putprop 'incremental-learning 'many-positive-and-negative
-examples-to-converge 'requirements-1)
(putprop 'incremental-learning 'many-partially-learned-heuristics
-represented-by-all-possible-descriptions-of-the-heuristic-are-
maintained 'requirements-2)
(putprop 'incremental-learning 'after-each-example-the-best-
possible-generalisation-is-made 'what-is-it-1)

;***** learning-problem ******
(putprop 'learning-problem 'major-problem 'constraints-problems-
1)
(putprop 'major-problem 'as-each-input-is-a-complex-combination-
of-instances-of-many-different-rules-the-major-problem-is-
determining-boundaries-of-rule-applicabilities 'what-is-it-1)
(putprop 'major-problem 'use-current-knowledge-to-partially-
understand-a-precedent-before-conjecturing-a-new-rule-pruning-
away-the-portion-of-the-precedent-already-understood 'solutions-
to-a-problem-1)

;***** structural learning *****
(putprop 'structural-learning '( adequacy-of-representation-
language generalisation-rules-employed computational-efficiency
flexibility-&-extensibility ) 'evaluation-criteria-1)
(putprop 'structural-description 'portray-objects-as-composite-
structures-consisting-of-various-components 'what-is-it-1)
(putprop 'structural-description 'predicate-logic-representation
'requirements-1)

;***** experiential learning ******
(putprop 'experiential-learning 'comparing-current-situation-with-stored-descriptions
'what-is-it-1)

;***** symbolic concept acquisition ******
(putprop 'symbolic-concept-acquisition 'learn-from-examples-
-through-analysis-of-positive-and-negative-examples 'important-examples-
of-1)
(putprop 'symbolic-concept-acquisition 'learns-a-symbolic-
representation-of-a-given-set-of-concepts-through-analysis-of-
positive-and-negative-examples 'general-principles-1)
(putprop 'symbolic-concept-acquisition 'process-of-pruning-away-
possibilities-rather-than-building-up-from-primitives 'what-is-it-1)
(putprop 'symbolic-concept-acquisition 'SCA 'equal-namewise-1)

****** knowledge intensive domain specific learning *****
(putprop 'knowledge-intensive-domain-specific-learning '( learn-
fromexamples-corresponding-to-a-background-knowledge learning-by-analogy learning-by-discovery)
'important-examples-of-1)
(putprop 'knowledge-intensive-domain-specific-learning 'background-knowledge 'requirements-1)
(setq background-knowledge '( predefined-concepts knowledge-
structures domain-constraints heuristics transformation-rules) 'what-is-it-1)
(putprop 'knowledge-intensive-domain-specific-learning 'does-
constructive-induction 'features-1)
(putprop 'knowledge-intensive-domain-specific-learning 'KDL 'equal-namewise-1)

;**** combination of SCA & KDL ****
(putprop 'combination-of-SCA-&-KDL 'exchangeable-knowledge-
modules 'requirements-1)

;***** rote learning *****
(putprop 'rote-learning '(no-transformation, indexes-knowledge-
for-retrieval) 'general-principles-1)

;****** learning by instruction & learning by deduction ************
(putprop 'learning-by-instruction 'draw-deductive-truth-
preserving-inferences-from-the-knowledge-and-store-useful-
conclusions 'general-principles-1)
(putprop 'learning-by-instruction '(knowledge-reformulation knowledge-compilation creation-of-macro-operators caching chunking equivalence-preserving-operations)
'important-examples-of-1 )

;****** learning by induction ************
(putprop 'learning-by-induction '(classifier-system generalisation pattern-recognition) 'types-of-2)
(putprop 'learning-by-induction '(sequence-extrapolation concept-formation theory-formation) 'types-of-3)
(putprop 'learning-by-induction '(similarity-based-learning discriminatory-based-learning explanation-based-learning) 'types-of-4)
(putprop 'descriptive-generalisation 'description-specifying-properties-of-objects-that-belong-to-a-class 'what-is-it-1)
(putprop 'learning-by-induction 'goes-beyond-data-to-develop-general-beliefs-from-specific-data 'what-is-it-1)
(putprop 'learning-by-induction 'search-for-plausible-general-descriptions-that-explain-given-input-and-are-useful-for-predicting-new-data 'what-is-it-2)
(putprop 'learning-by-induction '(completeness consistency) 'evaluation-criteria-1)
(putprop 'completeness 'for-every-i-E-implies-D 'what-is-it-1)
(putprop 'consistency 'for-every-i-j-i-not-equal-to-j-D-i-implies-not-E-j 'what-is-it-1)
(putprop 'learning-by-induction '(given '(observation-statements tentative-inductive-assertion background-knowledge) find inductive-assertion) 'general-principles-1)
(putprop 'inductive-assertion 'more-fulfilling-criteria 'what-is-it-1)
(putprop 'learning-by-induction 'transform-involves-generalisation-of-input-and-selection-of-most-plausible-or-desirable-result 'features-1)
(putprop 'learning-by-induction 'falsity-preserving 'features-2)
(putprop 'brittleness '(recombination-and-parallelism
103
categorisation-and-default-hierarchies association-between-categories diachronic-pointing-models-and-prediction competition-confirmation-and-gracefulness 'solution-to-a-problem-1')

-It-is-incumbent-for-induction-and-learning-to-search-for-rules-that-are-useful-building-blocks-in-a-variety-of-contexts 'what-is-it-1)

(putprop 'categorisation-and-default-hierarchies 'categories-generated-should-spontaneously-arrange-themselves-in-a-default-hierarchy 'what-is-it-1)


(putprop 'diachronic-pointing-models-and-prediction 'system-using-a-model-to-generate-expectation-driven-prediction-can-use-feedback-to-guide-revisions-of-model 'what-is-it-1)

(putprop 'diachronic-pointers 'indicates-temporal-sequence-of-categories 'what-is-it-1)

(putprop 'synchronic-pointers 'indicate-only-association-and-not-temporal-sequence-of-the-categories 'what-is-it-1)

(putprop 'competition-confirmation-and-gracefulness 'various-rules-held-by-the-system-are-treated-as-competing-hypotheses 'what-is-it-1)

(putprop 'learning-by-induction 'version-space 'requirements-1)

(putprop 'version-space 'partially-ordered-set-of-current-best-descriptions-and-hypotheses 'what-is-it-1)

(putprop 'learning-by-induction 'rating-usefulness-of-existing-knowledge generating-new-knowledge 'tasks-of-1)

(putprop 'bucket-brigade-algorithm 'method-alg-heuristic-1)

(putprop 'bucket-brigade-algorithm 'survival-of-a-rule-depends-on-its-making-a-profit-in-its-local-interactions 'basics-of-1)

(putprop 'generating-new-knowledge 'genetic-algorithm 'method-alg-heuristic-1)

(putprop 'genetic-algorithm 'high-strength-classifiers-used-as-parent-of-new-classifiers 'basics-of-1)

(putprop 'bucket-brigade-algorithm 'each-satisfied-classifier-makes-a-bid-based-on-its-strength the-highest-bidder-gets-its-message-on-the-list the-strength-of-this-classifier-is-reduced-by-the-amount-of-the-bid the-classifiers-that-contributed-to-this-winner-being-satisfied-increase-their-strength 'method-alg-
heuristic-1)
(putprop 'bucket-brigade-algorithm 'the-method-ensures-that-
strength-increases-if-actions-are-profitable 'basics-of-1)
(putprop 'learning-by-induction 'the-large-number-of-inductive-
assertions-that-can-be-formulated-for-a-given-input-
form-a-set-of-descriptions-that-are-partially-ordered-by-the-
relation
-of-relative-generality 'general-principles-2 )
(putprop 'learning-by-induction '( representation type-of-
description-sought rules-of-generalisation constructive-induction
control-strategy general-vs-problem-oriented-approach )
'features-1)
(putprop 'type-of-description-sought 'criteria-i-required-to-
define
-goal-description 'what-is-it-1)
(putprop 'criteria-i '( characteristic-criteria discriminant-
criteria taxonomic-criteria ) 'types-of-1)
(putprop 'characteristic-criteria 'stores-facts-true-of-all-
objects 
in-the-class 'what-is-it-1)
(putprop 'discriminant-criteria 'stores-facts-required-to-
distinguish-objects-of-a-class-from-those-of-another-class 'what-
is-it-1)
(putprop 'discriminant-criteria 'learning-from-training-
instances 'important-examples-of-1 )
(putprop 'taxonomic-criteria 'subdivides-class-into-sub-classes
'what-is-it-1)
(putprop 'taxonomic-criteria 'fundamentally-disjunctive
'features-1)
(putprop 'taxonomic-criteria 'learning-from-observation-and-
discovery 'important-examples-of-1 )
(putprop 'rules-of-generalisation 'inductive-generalisation
'equal-namewise-1 )
(putprop 'characteristic-induction 'conjunctive-generalisation
'equal-namewise-1 )
(putprop 'characteristic-induction 'satisfies-completeness-
condition 'features-1)
(putprop 'characteristic-induction 'conjunction-of-simple-
properties-common-to-all-objects 'what-is-it-1)
(putprop 'characteristic-induction 'maximal-characteristic-
descriptions-that-are-most-specific-logical-products
-characterising-all-objects 'interesting-subset-of-category-1)
(putprop 'discriminant-induction 'satisfies-completeness-and-
consistency 'features-1)
(putprop 'discriminant-induction 'minimal-discriminant-descriptions
|that-are-the-shortest-expressions-distinguishing-all-objects
|'interesting-subset-of-category-1)
(putprop 'learning-a-single-concept '(
|learning-a-single-concept-from-positive-examples-only
|learning-a-single-concept-from-both-positive-and-negative-examples)
types-of-1)
(putprop 'learning-a-single-concept-from-positive-examples-only
|'consistency-condition-not-ensured 'features-1)
(putprop 'learning-a-single-concept-from-positive-examples-only
|'limits-have-to-be-imposed-on-generalisation-of-descriptions
|'constraints-problems-1)
(putprop 'learning-a-single-concept-from-both-positive
|and-negative-examples 'consistency-achieved 'features-1)
(putprop 'learning-a-single-concept-from-both-positive
|and-negative-examples 'most-useful-negative-examples-are-near-misses
|'interesting-subset-of-category-1)

(putprop 'learning-multiple-concepts '(
|descriptions-of-different-classes-are-mutually-disjoint
|descriptions-of-different-classes-are-overlapping) 'types-of-1)
(putprop 'descriptions-of-different-classes-are-overlapping
|'an-event-may-satisfy-more-than-one-description 'what-is-it-1)
(putprop 'background-knowledge '(information-about-descriptors
|form-of-observational-and-inductive-assertions
|preference-criterion
|inference-rules-and-heuristics-to-generate-logical-consequences)
types-of-1)
(putprop 'degree-of-relevance-of-initial-descriptors-to-learning-
|problem '(
|complete-relevance partial-relevance indirect-relevance) 'types-of-1)
(putprop 'complete-relevance 'all-descriptors-are-relevant 'what-
|is-it-1)
(putprop 'complete-relevance 'learning-by-being-told-and-asking-
|questions 'method-alg-heuristic-1)
(putprop 'partial-relevance 'irrelevant-as-well-as-relevant-
|descriptors-exist 'what-is-it-1)
(putprop 'partial-relevance 'selective-induction 'method-alg-
|heuristic-1)
(putprop 'selective-induction 'select-most-relevant-descriptors-
|to-construct-inductive-assertion 'method-alg-heuristic-1)
(putprop 'indirect-relevance 'derived-descriptors-constructed-
|from-initial-descriptors-are-directly-relevant 'what-is-it-1)
(putprop 'indirect-relevance 'constructive-induction 'method-alg-
|heuristic-1)
(putprop 'constructive-induction 'construct-derived-descriptors-from-initial-descriptors 'method-alg-heuristic-1)

;;; induction -- subset principle ****
(putprop 'subset-principle 'dictates-that-the-next-refined-predication-tree-should-be-a-minimal-refinement-of-existing-trees 'what-is-it-2)
(putprop 'subset-principle 'generally-applicable 'applicability-conditions-1)

;;; induction -- classifier system *****
(putprop 'classifier-system '(message-passing-rule-based-system-wherein-a-large-number-of-rules-can-be-active-simultaneously-order-in-which-classifiers-are-stored-is-independent-of-the-order-in-which-they-are-executed) 'basics-of-1)

;;; induction - theory formation ****

;;; induction --- accretion model of theory formation *****
(putprop 'accretion-model-of-theory-formation 'uses-either-a-deep-model-of-the-domain-or-a-growing-body-of-heuristics
'requirements-1)
(putprop 'accretion-model-of-theory-formation 'step1-step2-step3-step4-step5-step6-step7 'method-alg-heuristic )
(putprop 'step1 'gather-empirical-data-find-examples-find-applications-etc 'what-is-it-1)
(putprop 'step1 'new-area 'applicability-conditions-1)
(putprop 'step2 'note-regularities-patterns-and-exceptions-to-patterns 'what-is-it-1)
(putprop 'step3 (' from-observations-form-new-hypotheses-and-or-modify-hypotheses design-and-carry-out-experiments-to-test-hypotheses ) 'what-is-it-1)
(putprop 'step4 'optimise-definitions-and-hypotheses-by-repeating-from-step1 'what-is-it-1)
(putprop 'step5 'occasionally-abstract-new-heuristics-by-compiling-the-learners-hindsight 'what-is-it-1)
(putprop 'step5 'known-area 'applicability-conditions-1)
(putprop 'step6 'on-rare-occasions-augment-a-shift-in-representation-and-in-bias 'what-is-it-1)
(putprop 'step6 'lack-of-consistent-hypotheses 'applicability-conditions-1)
(putprop 'step7 'for-steps1-to-6-it-suffices-to-collect-and-use-a-body-of-heuristics 'what-is-it-1)

;***** induction -- pattern recognition *****
(putprop 'pattern-recognition '( decision-theoretic-approach structured-approach ) 'types-of-1)
(putprop 'decision-theoretic-approach 'variables-are-assumed-to-be-measured-on-at-least-an-interval-scale-and-they-are-relevant-and-independent-characteristics-of-the-object 'applicability-conditions-1)
(putprop 'structured-approach 'syntactic-approach 'equal-namewise-1)
(putprop 'structured-approach 'variables-strongly-interconnected-and-or-relevant-objects-characteristics-are-not-numbers-but-the-variables-relations-among-other-variables-or-among-parts-or-sub-parts-of-objects 'applicability-conditions-1)
(putprop 'structured-approach 'overlooks-semantics-of-problem 'constraints-problems-1)

;***** induction -- generalisation *****
(putprop 'inductive-generalisation 'discovery-of-important-links-between-components-of-examples 'what-is-it-1)
(putprop 'inductive-generalisation ' (main-goal-of-a-character-is-achieved the-goal-is-a-general-one resources-required-by-the-character-are-normally-available method-is-at-least-as-effective-as-previous-ones method-is-a-new-one input-matches-one-of-the-known-generalisable-patterns ) 'applicability-conditions-1 )
(putprop 'inductive-generalisation ' (schema-composition schema-alteration volitionalisation secondary-effect-evaluation ) 'applicability-conditions-2 )
(putprop 'inductive-generalisation ' conjunctive-description-of-least-generality 'types-of-1)
(putprop ' conjunctive-description-of-least-generality ' (dropping-a-condition turning-constants-into-variables climbing-generalisation-tree ) 'method-alg-heuristic-1)
(putprop 'inductive-generalisation ' produces-rules-that-have-unknown-truth-value 'features-1)
(putprop 'inductive-generalisation ' ( non-constructive-induction constructive-induction ) 'types-of-2)
(putprop 'non-constructive-induction ' selective-induction 'equal-namewise-1 )
(putprop 'non-constructive-induction 'the-generalised-heuristic-got-by-application-of-rule-contains-no-new-descriptors 'what-is-it-1)
(putprop 'non-constructive-induction 'dropping-condition 'method-alg-heuristic-1 )
(putprop 'non-constructive-induction ' converting-constants-to-variables 'method-alg-heuristic-2 )
(putprop 'non-constructive-induction ' by-internal-disjunction 'method-alg-heuristic-3 )
(putprop 'non-constructive-induction ' by-closing-interval 'method-alg-heuristic-4 )
(putprop 'non-constructive-induction ' by-climbing-generalisation-tree 'method-alg-heuristic-5 )
(putprop 'non-constructive-induction ' extension-against-rule 'method-alg-heuristic-6 )
(putprop 'by-internal-disjunction ' (dropping-condition converting-constants-to-variable) 'superior-case-of-1 )
(putprop 'generating-chain-properties-rule ' ( by-internal-disjunction by-closing-interval by-climbing-generalisation-tree) 'superior-case-of-1 )
(putprop 'by-closing-interval ' any-linear-description 'applicability-conditions-1)
(putprop 'by-climbing-generalisation-tree ' only-for-structural-descriptions 'applicability-conditions-1)
(putprop 'extension-against-rule 'requires-positive-and-negative-
(putprop 'constructive-induction 'generates-new-descriptors-in-new-heuristic 'what-is-it-1)
(putprop 'constructive-induction 'many-rules-possible 'features-1)
(putprop 'constructive-induction 'counting-rule 'method-alg-heuristic-1)
(putprop 'constructive-induction 'generating-chain-properties-rule 'method-alg-heuristic-2)
(putprop 'counting-rule 'find-the-number-of-attributes-with-particular-values 'method-alg-heuristic-1)
(putprop 'generating-chain-properties-rule 'pick-characteristic-object-of-chain 'method-alg-heuristic-1)
(putprop 'generating-chain-properties-rule 'on-linearly-ordered-applicability-conditions-1)
(putprop 'inductive-generalisation '(disjunctive-concepts-are-not-handled cannot-handle-dynamic-environments cannot-handle-errorfull-data inadequate-treatment-of-time does-not-handle-variable-constraints) 'constraints-problems-1)
(putprop 'inductive-generalisation '(similarity-based constraint-based) 'types-of-1)
(putprop 'similarity-based-inductive-generalisation 'finds-inter-example-relationships 'basics-of-1)
(putprop 'constraint-based-inductive-generalisation 'finds-intra-example-relationships 'basics-of-1)
(putprop 'similarity-based-inductive-generalisation 'searches-for-common-features 'method-alg-heuristic-1)
(putprop 'constraint-based-inductive-generalisation 'constraints-that-form-the-concept-are-searched-for 'method-alg-heuristic-1)
(putprop 'detect-structural-matchings-among-examples 'each-variable-of-generalisation-variable-must-be-at-the-same-occurrence-of-the-same-predicate-in-all-examples 'requirements-1)
(putprop 'schema-composition 'preconditions-satisfied-in-a-novel-way-by-other-schemata 'what-is-it-1)
(putprop 'schema-alteration 'modification-guided-by-world-model-to-fit-need-of-new-situation 'what-is-it-1)
(putprop 'volitionalisation 'introduce-human-planner 'what-is-it-1)
(putprop 'volitionalisation 'schema-usable-by-planner-to-reach-goal-is-generated 'basics-of-1)
(putprop 'secondary-effect-evaluation 'using-schema-in-a-new-way-
that ignores the main effect and focuses on the secondary effect of the schema 'what is it-1)

(putprop 'inductive-generalisation 'replacement-of-specific-item
by-general-variable 'fundamental-operators-1)

(putprop 'learning-a-new-clause 'example 'requirements-1)

(putprop 'learning-a-new-clause '(find-list-of-replacement-operators use-these-operators-to-generalise-trial qualify-trial specialise-trial construct-example) 'stages-in-1)

(putprop 'pragmatic-inductive-generalisation 'concepts-describe what-is-usually-and-not-necessarily-always-true 'what-is-it-1)

(putprop 'pragmatic-inductive-generalisation 'major-effect-on-learning-process 'features-1)

(putprop 'maximally-specific-conjunctive-generalisation 'conjunction-of-all-positive-examples 'what-is-it-1)

(putprop 'maximally-specific-conjunctive-generalisation 'simplest-learning-strategy 'features-1)

;*** induction -- bias ***


(putprop 'bias '(used-to-prune-search-tree criteria-used-to-make-inductive-leap) 'what-is-it-1)

(putprop 'bias '(occams-razor restricted-language conjunctive-description maximally-general-description maximally-specific-description least-disjunctive-description one-disjunct-per-lesson) 'types-of-1)

(putprop 'occams-razor 'prefer-simpler-concept-descriptions 'what-is-it-1)

(putprop 'restricted-language 'prefer-descriptions-that-can-be-expressed-in-a-restricted-logically-incomplete-language 'what-is-it-1)

(putprop 'conjunctive-description 'prefer-descriptions-that-can-be-expressed-as-a-conjunction-of-literals 'what-is-it-1)

(putprop 'maximally-general-description 'prefer-description-as-generalised-as-possible 'what-is-it-1)

(putprop 'maximally-general-description 'results-in-selection-of-disjunction-of-negation-of-all-negative-training-instances 'features-1)

(putprop 'maximally-general-description 'restricting-language-to-
exclude-negation-makes-this-more-powerful 'features-2 )
(putprop 'maximally-specific-description 'prefer-as-specific-as-
possible-description 'what-is-it-1)
(putprop 
'maximally-specific-description 
'results-in-selection-of-disjunction-of-positive-instances 
'features-1 )
(putprop 'maximally-specific-description 'used-in-combination-
with-restricted-language-or-conjunctive-bias 'features-2 )
(putprop 'least-disjunctive-description 'prefer-description-with-
fewest-number-of-disjuncts 'what-is-it-1)
(putprop 'one-disjunct-per-lesson 'exactly-one-disjunct-
introduced-per-lesson 'what-is-it-1)
(putprop 'one-disjunct-per-lesson 'partitioning-of-training-
instances 'requirements-1)
(putprop 'bias '( total-preference-ordering-of-all-hypotheses 
partial-preference-ordering-of-hypotheses restricted-hypotheses-
space combination) 'types-of-2 )
(putprop 'partial-preference-ordering-of-hypotheses 'only-most-
preferred 'basics-of-1)
(putprop 'partial-preference-ordering-of-hypotheses 'random-if-
best-is-unavailable 'features-1)
(putprop 'restricted-hypotheses-space 'searching-for-a-better-
description-language 'basics-of-1)
(putprop 'shifting-bias 'learning-how-to-learn 'what-is-it-1)
(putprop 'shifting-bias 'when-space-of-possible-hypotheses-does-
not-include-consistent-hypotheses 'applicability-conditions )
(putprop 'shifting-bias '( recommend-new-concept-description-or-
learning-strategy translate-new-recommendation-to-concept-
description assimilate ) 'method-alg-heuristic-1)
(putprop 'shifting-bias 'shifting-from-a-strong-bias-to-a-weaker-
one-is-the-process-of-shifting-from-deduction-through-analogy-
towards-induction 'general-principles-1)
(putprop 'multiple-biases 'required-because-of-the-many-
dimensions-on-which-to-evaluate-hypotheses 'basics-of-1)
(putprop 'bias '( simplicity scope cost-of-testing ) 'evaluation-
criteria-1)

;**** induction -discrimination *****
(putprop 'inductive-discrimination 'applicability-
conditions-1)
induction -- Similarity Based Learning

(similarity-based-learning 'learns-by-comparing-instances-more-or-less-syntactically-using-little-high-level-knowledge-of-domain 'what-is-it-1)

(similarity-based-learning (observed-regularities-in-data-have-to-reflect-underlying-causal-laws-of-domain target-concepts-have-to-be-expressed-in-a-limited-biased-language) 'constraints-problems-1)


(similarity-based-learning (acquiring-generalised-plans updating-similarity-metric) 'applicability-conditions-1)

induction -- by observation and discovery

(learning-by-induction-from-observation-and-discovery 'primarily-acquiring-descriptive-generalisation 'what-is-it-1)


(given-class-find-properties) 'general-principles-1)

(search-for-regularities-and-general-rules-explaining-observations 'what-is-it-2)

(no-teacher-is-available 'basics-of-1)

(learning-by-induction-from-observation-and-discovery)
'(genetic-algorithm empirical-prediction conceptual-clustering)
'types-of-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'(correlational-approach explanatory-schema-acquisition-approach
'types-of-2)
(putprop 'correlational-approach 'construct-concept-by-noting-
similarities-and-dissimilarities-between-a-large-number-of-events
'basics-of-1)
(putprop 'correlational-approach 'many-examples 'requirements-1)
(putprop 'explanatory-schema-acquisition-approach 'feature-
significance-is-judged-through-use-of-background-
knowledge-rather-than-similarity 'basics-of-1)
(putprop 'explanatory-schema-acquisition-approach
'one-example-is-sufficient 'requirements-1)
(putprop 'explanatory-schema-acquisition-approach 'new-concepts
-may-be-incomplete-and-may-be-refined-later 'features-1)
(putprop 'verifying-solution-to-transform-instance-to-goal
'(is-sequence-of-operators-viable does-resulting-state-after-
application-of-last-operator-contain-goal-state constraint-based-
generalisation) 'method-alg-heuristic-1)
(putprop 'constraint-based-generalisation 'preservation-of-
underlying-verification-steps-composing-explanation-of-why-the-
solution-works 'basics-of-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'learning-multiple-concepts-through-observation-and-discovery
'types-of-3)
(putprop 'learning-multiple-concepts-through-observation-and-
discovery 'no-prespecification-of-concepts-to-generalise
'general-principles-1)
(putprop 'learning-multiple-concepts-through-observation-and-
discovery 'sets-of-concepts-are-acquired-by-deciding-what-
instances-to-compare 'features-1)
(putprop 'learning-multiple-concepts-through-observation-and-
discovery 'derived-concepts-are-often-overlapping 'features-2)

;***** heuristics for discovery *****
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-1 'method-alg-heuristic-1)
(putprop 'Rule-1 'if-F-is-an-interesting-function-which-
take-a-pair-of-A-as-input-then-define-and-study-the-coalesced-
function-G-of-AF-of-a-and-a 'what-is-it-1)
(putprop 'Rule-1 'mathematics-domain 'applicability-conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery
 'Rule-2 'method-alg-heuristic-2)
(putprop 'Rule-2 'if-a-predicate-P-rarely-returns-True-then-define-a-new-one-which-is-similar-to-but-more-general-than-P 'what-is-it-1)
(putprop 'Rule-2 'heuristics-and-specific-domains 'applicability-conditions)
(putprop 'Rule-2 'rule-for-generalisation 'features-1)
(putprop 'learning-by-induction-from-observation-and-discovery
 'Rule-3 'method-alg-heuristic-3)
(putprop 'Rule-3 'if-F-is-a-known-interesting-function
 -and-B-is-a-known-interesting-extrema-subset-of-its-range-then-
 -define-and-
 -study-F-inverse-of-B 'what-is-it-1)
(putprop 'Rule-3 'mathematics-domain 'applicability-conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery
 'Rule-4 'method-alg-heuristic-4)
(putprop 'Rule-4 'if-some-but-not-most-examples-of-X-are-also-
 -examples-of-Y-and-some-but-not-most-examples-of-Y-are-also-
 -examples-of-X-then-define-and-study-intersection-of-X-and-Y
 'what-is-it-1)
(putprop 'Rule-4 'new-concept-is-a-specialisation-of-both-X-and-Y
 'features-1)
(putprop 'learning-by-induction-from-observation-and-discovery
 'Rule-5 'method-alg-heuristic-5)
(putprop 'Rule-5 'define-and-study-the-concepts-one-gets-by-
 -seperating-the-conjunctions-if-any-in-a-definition 'what-is-it-1)
(putprop 'Rule-5 'converse-of-Rule-4 'features-1)
(putprop 'learning-by-induction-from-observation-and-discovery
 'Rule-6 'method-alg-heuristic-6)
(putprop 'Rule-6 'if-a-few-examples-of-a-concept-are-found-then-
 -examine-a-typical-one-and-see-what-properties-it-satisfies-and-
 -see-if-any-of-these-properties-are-satisfied-by-all-examples-of-
 -the-concept 'what-is-it-1)
(putprop 'Rule-6 'heuristics-and-specific-domains 'applicability-
 -conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery
 'Rule-7 'method-alg-heuristic-7)
(putprop 'Rule-7 'if-an-analogy-is-strong-between-A-and-B-
 -but-there-is-a-conjecture-whose-analogy-is-false-then-define-the-
 -subset-of-A-for-which-the-analogous-conjecture-holds-good 'what-
 -is-it-1)
(putprop 'Rule-7 'heuristics-and-specific-domains 'applicability-
 -conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery
Rule-8 'method-alg-heuristic-8 )
(putprop 'Rule-8 'if-a-concept-has-a-complement-or-a-negation
-which-is-much-smaller-or-rarer-then-explicitly-define-and-name-
that-concept 'what-is-it-1)
(putprop 'Rule-8 'mathematics-domain 'applicability-conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-9 'method-alg-heuristic-9 )
(putprop 'Rule-9 'if-you-have-just-generalised-C-into-a-
new-concept-G-then-beware-that-G-is-not-really-the-same-as-C.
-check-are-there-any-other-specialisations-of-G-and-if-there-are-
look-for-examples-for-them.-If-yes-at-least-a-conjecture-about-C-
exists 'what-is-it-1)
(putprop 'Rule-9 'heuristics-and-specific-domains 'applicability-
conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-10 'method-alg-heuristic-10 )
(putprop 'Rule-10 'if-an-operator-L-is-useful-and-is-to-be-
generalised-then-consider-just-widening-domain-of-C 'what-is-it-
1)
(putprop 'Rule-10 'mathematics-domain 'applicability-conditions-
1)
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-11 'method-alg-heuristic-11 )
(putprop 'Rule-11 'if-slots-S1-and-S2-of-frame-F-can-have-the-
same-type-then-define-a-new-heuristic-attached-to-F-that-says-
If-F-is-an-interesting-frame-and-its-S1-and-S2-slots-are-of-the-
same-type-then-define-and-study-the-situation-in-which-S1-and-S2-
have-values-that-are-equal 'what-is-it-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-12 'method-alg-heuristic-12 )
(putprop 'Rule-12 'if-function-F-of-extreme-of-A-and-extreme
-of-B-is-nearly-extreme-then-combine-definition-of-A-and-
definition-of-B-to-yield-a-new-concept-definition-Prefer-
combining-functions-analogous-to-F 'what-is-it-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-13 'method-alg-heuristic-13 )
(putprop 'Rule-13 'finding-heuristics-specific-to-concept-C
'applicability-conditions-1)
(putprop 'Rule-13 'try-to-analogise-heuristics-specific-to-
concepts-that-were-discovered-the-same-way-that-C-was 'what-is-it-1)
(putprop 'learning-by-induction-from-observation-and-discovery
'Rule-14 'method-alg-heuristic-14 )
(putprop 'Rule-14 'if-the-results-of-performing-act-F-have-
always-been-numerous-and-worthless-then-lower-expected-worth-of-F
116
'what-is-it-1)
(putprop 'Rule-15 'if-results-of-performing-act-F-are-only-
occasionally-useful-then-consider-creating-new-specialisation-of-
F 'what-is-it )
(putprop 'learning-by-induction-from-observation-and-discovery 'Rule-16 'method-alg-heuristic-16 )
(putprop 'Rule-16 'take-a-slot-and-ask-questions-s-on-it 'what-is-it-1)
(putprop 'questions-s '( evolution-over-time relationship-to-
other-entities use ) 'types-of-1)
(putprop 'Rule-16 'finding-new-slots 'applicability-conditions-1)
(putprop 'learning-by-induction-from-observation-and-discovery 'Rule-17 'method-alg-heuristic-17 )
(putprop 'Rule-17 'take-a-pair-of-slots-and-a-relation-and-
define-a-new-unary-function-on-heuristics 'what-is-it-1)
(putprop 'Rule-17 'finding-new-slots 'applicability-conditions-1)
(putprop 'Rule-18 'generalise-a-rarely-true-predicate 'what-is-it-1)

;**** data driven heuristics ****
(putprop 'Rule-19 'if-you-see-a-number-of-descriptions-at-
level-L-in-which-the-dependent-variable-D-has-the-same-value-V-
then-create-a-new-description-at-level-L+1-in-which-the-value-of-
D-is-also-V-and-which-has-all-conditions-common-to-the-observed-
description 'what-is-it-1)
(putprop 'Rule-19 'looks-for-recurring-values-of-a-dependent-
variable 'applicability-conditions-1)
(putprop 'Rule-19 'does-constructive-induction 'features-1)
(putprop 'learning-by-induction-from-observation-and-discovery 'Rule-20 'method-alg-heuristic-20 )
(putprop 'Rule-20 'if-the-value-of-dependent-variable-A1-
increases-as-the-corresponding-values-of-variable-A2 -decreases-in-a-number-of-descriptions-at-level-L-then-note-a-
(putprop 'Rule-20 'for-noticing-decreasing-relations

117
'applicability-conditions-1)
(putprop 'Rule-20 'relates-numeric-variables 'features-1)
(putprop 'Rule-21 'for-postulating-an-intrinsic-property-of-a-nominal-variable 'applicability-conditions-1)

;***** operators for discovery *****
(putprop 'learning-by-induction-from-observation-and-discovery '
'(empirical-laws qualitative-structure components-of-substance structural-models-of-chemical-reactions ) 'types-of-1)
(putprop 'empirical-laws 'specify-value 'subcomponents-1)
(putprop 'specify-value 'specifies-value-of-an-undetermined-independent-value 'what-is-it-1)
(putprop 'qualitative-structure '(data-gathering law-finding ) 'subcomponents-1 )
(putprop 'data-gathering '(form-law determine-quantifier ) 'operators-1)
(putprop 'form-law 'define-class-and-substitute-it-into-facts 'what-is-it-1)
(putprop 'determine-quantifier 'specifies-universal-or-existential-quantifier 'what-is-it-1)
(putprop 'form-law 'select-that-object-occurring-in-most-analogous-facts 'method-alg-heuristic-1)
(putprop 'determine-quantifier 'quantify-universally-if-data-justify 'method-alg-heuristic-1)
(putprop 'law-finding '(add subtract) 'operators-1)
(putprop 'add 'add-one-parameter-value 'what-is-it-1)
(putprop 'subtract 'subtract-one-parameter-value 'what-is-it-1)
(putprop 'add 'select-states-that-lead-to-higher-correlation-thus-predicting-data 'method-alg-heuristic-1)
(putprop 'subtract 'select-states-that-lead-to-higher-correlation-thus-predicting-data 'method-alg-heuristic-1)
(putprop 'components-of-substance '(infer-composition reduce substitute identify-component ) 'operators-1)
(putprop 'infer-composition 'decide-on-components-of-substance 'what-is-it-1)
(putprop 'substitute 'replace-a-substance-with-its-components 'what-is-it-1)
(putprop 'identify-component 'identifies-two-components-as-same 'what-is-it-1)
(putprop 'reduce 'if-A-occurs-on-both-sides-of-a-reaction-then-remove-A-from-reaction 'method-alg-heuristic-1)
(putprop 'structural-models-of-chemical-reactions '( determine-molecules determine-atom conserve-particles ) 'operators-1)
(putprop 'determine-molecules 'determine-the-number-of-times-a-compound-occurs-in-a-reaction 'what-is-it-1)
(putprop 'determine-atom 'determine-number-of-atoms-of-given-type 'what-is-it-1)
(putprop 'conserve-particles 'remaining-numbers-determined-based-on-conservation-principle 'what-is-it-1)
(putprop 'determine-molecules 'consider-only-multiples-of-combining-volumes 'method-alg-heuristic-1)
(putprop 'determine-atom 'select-simplest-models-first 'method-alg-heuristic-1)

;;;; conceptual clustering ****
(putprop 'classification 'conceptual-clustering 'types-of-1)
(putprop 'conceptual-clustering 'background-knowledge-cc 'requirements-1)
(putprop 'background-knowledge-cc '(domain-specific-b-k-cc general-purpose-b-k-cc ) 'types-of-1)
(putprop 'domain-specific-b-k-cc '(deductive-and-inductive-inference-rules-for-deciding-values-of-new-descriptors goal-dependency-network-to-infer-which-descriptors-are-relevant-to-the-classification-goal ) 'what-is-it-1)
(putprop 'general-purpose-b-k-cc 'goal-of-classification classification-evaluation-criteria definitions-of-attributes-used-in-object-descriptions domain-of-
attributes-used-in-object-descriptions types-of-attributes-used-in-object-descriptions) 'what-is-it-1)
(putprop 'conceptual-clustering '( generate-conceptual-description-of-the-classes classify-objects-according-to-the-description) 'stages-in-1)
(putprop 'class 'represents-a-simple-concept-rather-than-defining-a-predefined-similarity-among-its-members 'what-is-it-1)
(putprop 'conceptual-clustering 'attribute-based-conjunctive-conceptual-clustering 'types-of-1)
(putprop 'conceptual-cohesiveness 'opposed-to-a-non-gestaltic-similarity-function 'features-1)
(putprop 'conceptual-cohesiveness 'function-of-Acc-Bcc-Ecc-Ccc 'what-is-it-1)
(putprop 'Acc 'object 'what-is-it-1)
(putprop 'Bcc 'object 'what-is-it-1)
(putprop 'Ecc 'environment 'what-is-it-1)
(putprop 'Ccc 'set-of-concepts-available-for-describing-A-and-B 'what-is-it-1)
(putprop 'conceptual-cohesiveness 'conceptual-clustering 'method-alg-heuristic-1)
(putprop 'event-descriptors '( initial-event-descriptors derived-event-descriptors) 'types-of-1)
(putprop 'derived-event-descriptors '( by-special-computation-experiment-or-device by-logical-inference) 'types-of-1)
(putprop 'classification '(simplicity-of-descriptors good-fit-of-categories-to-examples) 'evaluation-criteria-1)
(putprop 'people-create-disjoint-classes 'simplicity-of-descriptors 'cause-1)
(putprop 'classification '(repeated-discrimination classifying-attributes) 'method-alg-heuristic-1)
(putprop 'repeated-discrimination 'attempts-to-find-one-or-more-
classifying-attributes-whose-value-sets-can-be-split-into-ranges-
that-define-individual-classes 'what-is-it-1)
(putprop 'classifying-attributes 'generates-new-descriptors-
using-a-chain-of-inferences-and-tests-these-as-candidate-
classifying-criteria 'what-is-it-1)

;**** induction - explanation based learning *****
(putprop 'explanation-based-learning 'knowledge-intensive-method-
for-analysing-instances-to-build-a-generalised-schemata 'what-is-
it-1)
(putprop 'explanation-based-learning 'works-well-in-functional-
domains-but-not-so-in-abstract-areas 'applicability-conditions-1)
(putprop 'explanation-based-learning '(given 'training-example)
explain '(finding-out-relevant-feature-values) generalise
'(constrain-feature-value-to-achieve-sufficiency-in-general) )
'general-principles-1)
(putprop 'explanation-based-learning 'operationality-criteria-
achieved-for-goal-concept 'features-1)
(putprop 'explanation-based-learning 'truth-preserving 'features-
2)
(putprop 'explanation-based-learning 'non-amplicative 'features-3)
(putprop 'goal-concept 'forms-boundary-for-most-general-version
'what-is-it-1)
(putprop 'training-example 'forms-boundary-for-most-specific-
version 'what-is-it-1)
(putprop 'explanation-based-learning 'cannot-represent-
disjunctions 'constraints-problems-1)
(putprop 'explanation-based-learning '( fails-if-system-cannot-
explain-precedent-due-to-incompleteness-or-intractability-of-
domain-theory can-only-produce-rules-that-are-provably-correct)
'constraints-problems-2)

;*** induction EBL analytical learning ***
(putprop 'analytical-learning 'precondition-analysis 'types-of-1)
(putprop 'precondition-analysis 'learns-from-worked-example-of-a-
correctly-executed-task 'requirements-1)
(putprop 'precondition-analysis 'can-learn-strategies 'features-1)
(putprop 'precondition-analysis 'learns-from-a-single-example
'features-2)
(putprop 'precondition-analysis ' operator-identification
precondition-analysis method-creation schema-creation ) 'stages-
(putprop 'operator-identification 'done-in-the-forward-scan 'features-1)
(putprop 'operator-identification 'analyses-what-method-was-used-in-each-step 'what-is-it-1)
(putprop 'operator-identification 'can-find-new-methods 'features-2)
(putprop 'precondition-analysis 'done-in-the-backward-scan 'features-3)
(putprop 'precondition-analysis 'find-the-purpose-of-each-step 'what-is-it-1)
(putprop 'method-creation 'new-problem-solving-methods-are-created 'features-1)
(putprop 'schema-creation 'schema-records-sequence-of-methods-invoked-for-example 'what-is-it-1)
(putprop 'schema-creation 'records-strategic-reasons-for-each-step 'features-1)
(putprop 'operator-identification 'given-two-consecutive-lines-of-an-example-delete-all-terms-common-to-both-and-conjecture-that-remaining-expressions-are-equal-or-transferrable 'method-alg-heuristic-1)

;;;; induction — learning by experimentation ****
(putprop 'learning-by-induction-from-experimentation '( problem-proposers decision-makers experiment-proposers expectation-setters experimenters hypothesis-generators hypothesis-modifiers hypothesis-removers confidence-modifiers ) 'method-alg-heuristic-1)
(putprop 'experimentation-based-theory-revision 'types-of-1)
(putprop 'experimentation-based-theory-revision '( contradiction-detection hypothesis-formation experiment-design ) 'stages-in-1)
(putprop 'hypothesis-formation 'explanation-construction 'equal-namewise-1)
(putprop 'experimentation-based-theory-revision '(provides-empirical-tests-of-relevance-of-aspects-of-situation discriminates-among-hypotheses-or-refutes-one-hypothesis provides-experiment-to-test-for-unnoticed-or-conjectured-object-or-test-for-conjectured-properties probes-for-unknown-values-of-a-quantity ) 'features-1)
(putprop 'queries '( indirect-queries direct-queries ) 'types-of-

;**** heuristics ****
(putprop 'learning-by-induction-from-experimentation 'Rule-22 'method-alg-heuristic-1)
(putprop 'Rule-22 'if-F-is-a-predicate-and-definition-of-F-exists-then-apply-definition-of-F-to-random-entries-on-examples-of-domain-of-F 'what-is-it-1)
(putprop 'Rule-22 'to-find-examples 'applicability-conditions-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-23 'method-alg-heuristic-2)
(putprop 'Rule-23 'if-a-few-examples-of-X-are-found-but-over-90%-are-non-examples-then-define-and-study-various-generalisations-of-X 'what-is-it-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-24 'method-alg-heuristic-3)
(putprop 'Rule-24 'if-a-heuristic-is-relevant-to-concept-C-then-it-is-also-relevant-to-all-specialisations-of-C 'what-is-it-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-25 'method-alg-heuristic-4)
(putprop 'Rule-25 'if-action-A-is-appropriate-in-situation-S-then-A-is-applicable-in-most-situations-similar-to-S 'what-is-it-1)
(putprop 'Rule-25 'justifies-analogy 'features-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-26 'method-alg-heuristic-5)
(putprop 'Rule-26 'if-action-A-is-applicable-in-situation-S-then-so-are-most-actions-similar-to-A 'what-is-it-1)
(putprop 'Rule-26 'justifies-use-of-inexact-reasoning 'features-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-27 'method-alg-heuristic-1)
(putprop 'Rule-27 'if-action-A-would-have-been
Applicable in past situation S then the rule
If similar to S try A will be useful in future 'what-is-it-1)
(putprop 'Rule-27 'justifies-use-of-memory-to-remember 'features-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-28
'method-alg-heuristic-1)
(putprop 'Rule-28 'if-modifying-an-if-part-of-a-heuristic
then-dont-replace-and-by-any-other-predicate 'what-is-it-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-29
'method-alg-heuristic-1)
(putprop 'Rule-29 'if-a-newly-synthesised-concept-has-criterial-slots-that-coincide-in-value-with-those-of-an-already-existing-concept-then-discard-new-concept-as-it-is-redundant 'what-is-it-1)
(putprop 'learning-by-induction-from-experimentation 'Rule-30
'method-alg-heuristic-1)
(putprop 'Rule-30 'ssl ss2 'what-is-it-1)
(putprop 'ssl 'select-initial-state-sl-and-goal-state-gl
(putprop 'ssl 'to-create-new-problems 'applicability-conditions)
(putprop 'ss2 'to-create-problem-to-propose-new-heuristics 'applicability-conditions-1)
(putprop 'ss2 'look-for-pairs-of-operators-whose-preconditions-intersect-but-do-not-have-current-heuristics 'what-is-it-1)

;**** induction --- discriminatory learning *****
(putprop 'learning-by-discrimination 'learns-the-if-parts 'what-is-it-1)
(putprop 'learning-by-discrimination '(step1-d step2-d ) 'method-alg-heuristic-1)
(putprop 'step1-d 'look-for-differences-between-positive-and-negative-examples 'what-is-it-1)
(putprop 'step2-d 'differences-used-to-specify-conditions-of-application-of-rules 'what-is-it-1)
(putprop 'learning-by-discrimination 'add-conditions-to-overly-general-rules 'what-is-it-2)
(putprop 'discrimination-net 'used-to-represent-concepts 'what-is-it-1)
(putprop 'discrimination-net 'search-for-attribute-value-pairs-
common-to-all-positive-and-negative-instances this-or-the-pair-occurring-most-frequently-is-chosen-as-the-test-for-a-new-branch-in-the-tree-for-the-concept this-strategy-is-applied-recursively-to-instances-satisfying-this-test-resulting-in-further-discrimination finally-the-tree-would-divide-examples-into-sets-exclusively-positive-or-negative 'method-alg-heuristic-1
(putprop 'discrimination-net 'each-branch-in-the-tree-depends-on-the-value-of-one-of-the-attributes-of-a-concept 'features-1)
(putprop 'learning-by-discrimination 'discrimination-net 'method-alg-heuristic-2)
(putprop 'discrimination-net 'a-fact-present-during-bad-application-of-a-rule-but-absent-during-good-applications-should-be-included-as-a-negated-condition 'method-alg-heuristic-2)
(putprop 'learning-by-discrimination 'at-times-of-ties-discrimination-can-create-more-constrained-version-of-the-competing-rules-with-additional-conditions 'features-1)
(putprop 'learning-by-discrimination 'allows-one-to-learn-even-if-the-representation-is-inadequate 'features-2)
(putprop 'learning-by-discrimination 'at-times-of-ties-discrimination-can-create-more-constrained-version-of-the-competing-rules-with-additional-conditions 'features-1)
(putprop 'learning-by-discrimination 'allows-one-to-learn-even-if-the-representation-is-inadequate 'features-2)
(putprop 'learning-by-discrimination 'at-times-of-ties-discrimination-can-create-more-constrained-version-of-the-competing-rules-with-additional-conditions 'features-1)
(putprop 'learning-by-discrimination 'allows-one-to-learn-even-if-the-representation-is-inadequate 'features-2)
(putprop 'learning-by-discrimination 'at-times-of-ties-discrimination-can-create-more-constrained-version-of-the-competing-rules-with-additional-conditions 'features-1)
(putprop 'learning-by-discrimination 'allows-one-to-learn-even-if-the-representation-is-inadequate 'features-2)

;**** induction -- learning from training instances (SBL) ****
(putprop 'learning-by-induction-from-training-instances 'primarily-concept-acquisition 'what-is-it-1)
(putprop 'concept-acquisition 'produces-description-for-classifying-objects-into-classes-based-on-properties-of-objects 'what-is-it-1)
(putprop 'learning-by-induction-from-training-instances '(characteristic-description-of-objects discriminant-description-of-objects infer-sequence-of-extrapolation) 'important-examples-of-1)
(putprop 'characteristic-description-of-objects 'determine-common-properties-of-objects-in-a-class 'what-is-it-1)
(putprop 'discriminant-description-of-objects 'determine-distinguishing-features-of-class-from-other-classes 'what-is-it-1)
(putprop 'infer-sequence-of-extrapolation 'determine-next-element-in-sequence 'what-is-it-1)
(putprop 'learning-by-induction-from-training-instances '(given-properties find-class) 'general-principles-1)
(putprop 'learning-by-induction-from-training-instances 'iti-method-1 'method-alg-heuristic-1)
(putprop 'iti-method-1 'maintain-version-space if-a-non-example-is-given-reject-those-descriptions-that-accept-it if-a-positive-example-is-given-generalise-those-descriptions-that-do-not-accept-it 'what-is-it-1)
(putprop 'iti-method-1 'when-generalisation-is-the-only-operator 'applicability-conditions-1)
(putprop 'learning-by-induction-from-training-instances 'iti-method-2 'method-alg-heuristic-2)
(putprop 'iti-method-2 'generalisation-operator-and-specialisation-operator 'what-is-it-1)
(putprop 'iti-method-2 'allows-both-general-to-specific-and-specific-to-general 'features-1)
(putprop 'iti-method-2 'allows-both-disjunctive-and-conjunctive-concept-descriptions 'features-2)
(putprop 'learning-by-induction-from-training-instances '(program-is-to-be-informed-by-a-teacher-whether-given-example-is-positive-or-negative program-knows-the-important-features-of-each-description the-bias-is-similar-to-being-told-an-initial-description-of-the-concept-to-be-learned) 'requirements-1)
(putprop 'learning-by-induction-from-training-instances 'program-knows-the-important-features-of-each-description-by-representing-only-these-features 'method-alg-heuristic-3)
(putprop 'heuristic-learning '(learning-macro-operators learning-analogies-between-states learning-of-heuristic-conditions-that-determine-applicability-of-operator-to-state) 'what-is-it-1)
(putprop 'heuristic-learning 'problem-reduction 'method-alg-heuristic-1)
(putprop 'problem-reduction '(sub-goal-definition formulation-of-heuristic-conditions-for-each-operator recombination-of-rules-to-form-complete-system) 'stages-in-1)
(putprop 'sub-goal-definition 'divide-task-into-a-number-of-sub-problems 'what-is-it-1)
(putprop 'formulation-of-heuristic-conditions-for-each-operator 'are-learned 'features-1)
(putprop 'formulation-of-heuristic-conditions-for-each-operator 'produce-positive-and-negative-examples 'requirements-1)
(putprop 'produce-positive-and-negative-examples 'related-to-credit-assignment-problem 'features-1)
(putprop 'produce-positive-and-negative-examples 'met-1 'method-alg-heuristic-1)
(putprop 'met-1 'solution-path-approach 'what-is-it-1)
(putprop 'met-2 'learning-while-doing 'what-is-it-1)
(putprop 'met-3 'learning-apprentice-approach 'what-is-it-1)
(putprop 'solution-path-approach 'mark-moves-long-solution-as-positive-Mark-moves-off-solution-as-negative 'method-alg-heuristic-1)
(putprop 'solution-path-approach 'a-complete-solution-path 'requirements-1)
(putprop 'solution-path-approach 'does-not-work-well-in-cases-of-long-solution-paths 'constraints-problems-1)
(putprop 'learning-by-induction-from-training-instances 'generate-concept-description-explaining-all-positive-examples-and-excluding-all-negative-examples 'what-is-it-2)
(putprop 'learning-by-induction-from-training-instances '(instance-to-class part-to-whole conceptual-clustering ) 'types-of-1)
(putprop 'instance-to-class 'given-examples-induces-general-description-of-class 'what-is-it-1)
(putprop 'instance-to-class 'learns-diagnostic-rules 'features-1)
(putprop 'instance-to-class 'popularly-used 'features-2)
(putprop 'part-to-whole 'given-selected-parts-of-an-object-hypothesise-the-whole-description 'what-is-it-1)
(putprop 'part-to-whole 'qualitative-process-prediction 'important-examples-of-1)
(putprop 'conceptual-clustering 'invent-class 'what-is-it-1)
(putprop 'near-miss 'very-general-heuristic 'features-1)
(putprop 'learning-by-induction-from-training-instances 'iti-method-3 'method-alg-heuristic-3)
(putprop 'iti-method-3 '(give-precedents give-exercise tell-appropriate-precedent after-applying-learning-by-analogy-the-program-creates-a-principle-capturing-the-IF-THEN-rule by-generalising-over-the-instances if-this-rule-is-an-overgeneralisation-UNLESS-parts-get-included-in-a-consequent-failure-run Later-if-this-UNLESS-part-is-proved-the-rule-is-straightaway-disqualified ) 'what-is-it-1)
(putprop 'blocking-principle 'if-any-relation-in-the-causal-structure-of-the-precedent-is-either-false-or-implausible-then-the-rule-based-on-the-precedent-does-not-apply 'what-is-it-1)
(putprop 'blocking-principle 'constitutes-unless-part 'features-1)
(putprop 'blocking-principle 'only-one-step-inference-allowed-for-checking-unless-part 'features-2 )

(putprop 'if-relevant-augmentation 'the-set-of-goals-which-the-rule
-will-help-in-achieving 'what-is-it-1)

(putprop 'learning-by-induction-from-training-instances 'description-transformation-problem 'constraints-problems-1)

(putprop 'description-transformation-problem 'rules-and-input-do-not-employ-same-description 'what-is-it-1)

(putprop 'description-transformation-problem 'input-is-transformed-to-a-derivation 'solutions-to-a-problem-1)

(putprop 'input-is-transformed-to-a-derivation 'matching-derivation-with-instantiated-rule-models 'method-alg-heuristic-1)

(putprop 'learning-by-induction-from-training-instances 'each-class-is-represented-by-an-abstract-model 'representation-1)

(putprop 'matching-derivation-with-instantiated-rule-models 'dt-method 'method-alg-heuristic-1)

(putprop 'dt-method 'parameterise-abstract-models-and-instantiate-it-to-yield-a-particular-rule 'what-is-it-1)

(putprop 'dt-method ' (selection-of-description-space-transformations-under-heuristic-guidance discovering-new-transformations ) 'method-alg-heuristic-1)

(putprop 'transformation-schema 'instantiated-to-give-set-of-transforms-for-particular-situation 'what-is-it-1)

;;; induction --- specialisation ***

(putprop 'inductive-specialisation 'exception-handling 'method-alg-heuristic-1)

(putprop 'inductive-specialisation 'if-the-results-of-performing-an-action-F-are-only-occasionally-useful-then-consider-creating-new-specialisations-of-F-by-specialising-some-of-the-critical-or-highly-rated-slots-in-the-representation 'method-alg-heuristic-2)

;;; induction from training instances - methods ****

(putprop 'learning-by-induction-from-training-instances ' ( data-driven-methods-iti model-driven-methods-iti ) 'method-alg-heuristic-4)

(putprop 'data-driven-methods-iti ' (near-miss-method sprouter vere ) 'types-of-1)
(putprop 'near-miss-method 'winstons-method 'equal-namewise-1)
(putprop 'near-miss-method '('rapidly-determine-correct-
generalised-concept-description develop-emphatic-conditions)
'features-1)
(putprop 'develop-emphatic-conditions 'results-in-MUST-type-
descriptions 'features-1)
(putprop 'near-miss-method 'semantic-network 'requirements-1)
(putprop 'near-miss-method 'finding-differences-is-accomplished-
by-graph-matching 'method-alg-heuristic-1)
(putprop 'near-miss-method 'does-not-consider-the-problem-of-
mapping-concept-descriptions-in-more-than-one-way 'constraints-
problems-1)
(putprop 'sprouter 'parameterised-structural-descriptions-using-
case-frames 'requirements-1)
(putprop 'sprouter 'cannot-handle-disjunction-or-internal-
disjunction 'constraints-problems-1)
(putprop 'sprouter '('dropping-conditions constants-to-variables)
'method-alg-heuristic-1)
(putprop 'vere 'discovered-literals 'features-1)
(putprop 'literals 'a-list-of-terms 'what-is-it-1)
(putprop 'vere '('dropping-conditions constant-to-variables)
'method-alg-heuristic-1)
(putprop 'vere 'discovers-disjunctions-and-exceptions-in-addition-to-conjunctions'
'features-2)
(putprop 'model-driven-methods-iti '('meta-dendral induce-1.2)
'types-of-1)
(putprop 'meta-dendral 'discovers-clevage-rules-to-explain-
operation-of-a-spectrometer 'what-is-it-1)
(putprop 'meta-dendral '('rulegen rulemod) 'subcomponents-1)
(putprop 'meta-dendral 'tries-to-find-small-set-of-generalised-
clevage-rules-that-cover-most-of-the-highly-specific-training-
rules 'features-1)
(putprop 'rulegen 'gives-a-local-maxima 'what-is-it-1)
(putprop 'rulemod 'improves-and-uses-negative-examples 'what-is-
it-1)
(putprop 'rulemod 'seeks-taxonomic-description-of-input-examples
'features-1)
(putprop 'meta-dendral 'result-predicts-fewer-fragmentation-per-training-module
result-is-sufficiently-general result-makes-at-least-as-many-
predictions-as-parent) 'evaluation-criteria-1)
(putprop 'meta-dendral '(dropping-conditions constants-to-
variables-used-in-reverse) 'method-alg-heuristic-1)
(putprop 'meta-dendral 'has-extensive-constructive-induction-but-
(putprop 'induce-1.2 'most-specific-conjunctive-generalisation-disjunctive-concepts 'method-alg-heuristic-1)

(putprop 'induce-1.2 'two-stage-approach 'stages-in-1)

(putprop 'induce-1.2 'advantages (two-phase-approach 'cause-1)

(putprop 'advantages 'representational-ad computational-ad 'what-is-it-1)

(putprop 'representational-ad 'structure-representation-as-a-graph attribute-space-represented-as-vectors 'what-is-it-1)

(putprop 'comparative-ad 'comparing-two-graph-structures-is-non-polynomial-complete-time-Hence-reduction-in-size-is-very-effective 'what-is-it-1)

(putprop 'disadvantages (defining-plausible-description-in-structure-space-is-a-problem finding-the-corresponding-attribute-description-space-for-the-structure-specification-space-is-a-problem 'what-is-it-1)

(putprop 'induce-1.2 'representational-language-VL21-an-extension-to-first-order-logic 'requirements-1)

(putprop 'induce-1.2 'implements-conjunctive-disjunctive-and-internal-disjunction-operators 'features-2)

(putprop 'induce-1.2 'no-exceptions-handled 'constraints-problems-1)

(putprop 'nominal-representation linear-representation-structured-representation 'representation-1)

(putprop 'nominal-representation 'unordered-value-sets 'what-is-it-1)

(putprop 'linear-representation 'linearly-ordered-sets 'what-is-it-1)

(putprop 'structured-representation 'tree-ordered-sets 'what-is-it-1)

(putprop 'induce-1.2 'uses-all-generalisation-rules 'features-3)

;**** induction - constructive induction in induce-1.2 ****

(putprop 'constructive-induction 'rules-that-count-number-of-objects-possessing-certain-characteristics 'method-alg-heuristic-1)


;****** chunking ******
(putprop 'learning-by-chunking 'learning-knowledge-about-patterns-in-environment 'what-is-it-1)
(putprop 'learning-by-chunking 'learning-solutions-to-impasses 'what-is-it-2)
(putprop 'impasse '(no-change-in-state tie conflict rejection 'types-of-1)
(putprop 'learning-by-chunking '(collection-of-conditions-and-actions variabisation-of-identifiers chunk-optimisation 'stages-in-1)
(putprop 'learning-by-chunking 'cannot-learn-in-new-problem-spaces-or-learn-new-representations 'constraints-problems-1)
(putprop 'learning-by-chunking 'cannot-make-use-of-potential-knowledge-sources-like-examples-and-analogies 'constraints-problems-2)
(putprop 'learning-by-chunking 'chunking-is-ubiquitous-in-human-performance-relating-patterns-of-goal-parameters-to-patterns-of-goal-results 'basics-of-1)
(putprop 'learning-by-chunking 'chunks-remove-the-need-for-intermediate-processing-between-a-response-and-a-first-stimulus 'features-1)
(putprop 'learning-by-chunking '(act-symbol response-symbol 'representation-1)
(putprop 'learning-by-chunking '(f1 f2 f3 f4 f5 f6 f7 'features-2)
(putprop 'features-2 'goal-hierarchies 'applicability-conditions-1)
(putprop 'f1 'each-chunk-represents-a-specific-goal-with-a-specific-set-of-parametric-values-it-relates-parametric-values-to-the-results-of-the-goal 'what-is-it-1)
(putprop 'f2 'chunks-are-created-through-experience-with-the-goals-processed 'what-is-it-1)
(putprop 'f3 'chunks-are-created-bottom-up-in-the-goal-hierarchy

131
'what-is-it-1)
(putprop 'f4 'chunks-consist-of-encoding-decoding-and-connection-
components 'what-is-it-1)
(putprop 'f5 'chunk-encoding-and-decoding-are-parallel-
hierarchical-goal-asynchronous-processes 'what-is-it-1)
(putprop 'f6 'chunk-connection-is-serial-goal-synchronous-and-
generates-results-from-parameters 'what-is-it-1)
(putprop 'f7 'chunk-improves-performance-by-replacing-normal-
processes-of-a-goal-or-sub-goals-with-the-faster-process-of-
encoding-decoding-and-connection 'what-is-it-1)
(putprop 'learning-by-chunking '(immediate-successful-
completion-of-goal all-modifications-are-found-attributable-to-
that-completed-goal-rather-than-one-of-its-subgoals)
'applicability-conditions-1)

(putprop
'learning-centered-on-the-storage-of-composite-
information-rather-than-its-recomputation
'(learning-by-chunking table-look signature-table-creation
macro-operator-creation classical-chunking production-composition)
'types-of-1)

(putprop 'macro-operator-creation 'cp 'constraints-problems-1)
(putprop 'cp '(combinatorics-involved-in-storing-and-searching-
all-possible-subsequences-of-all-solutions-ever-encountered path-
constraints-are-ignored no-provisions-made-for-substituting-
deleting-and-inserting-operators-into-solution-sequence ) 'what-
is-it-1)
(putprop 'cp 'gradually-transfer-existing-solution-into-one-that-
satisfies-requirements-of-new-problem 'solutions-to-a-problem-1)

(putprop 'learning-by-chunking 'skill-refinement 'applicability-
conditions-1)

(putprop 'learning-by-chunking 'learning-which-problem-features-
are-predictive-of-the-success-of-problem-solving-operators
'applicability-conditions-2)
(putprop 'learning-by-chunking 'knowledge-compilation 'equal-
namewise-1)
(putprop 'learning-by-chunking 'create-procedural-representation-
of-knowledge-from-declarative-representation 'tasks-of-1)
(putprop 'learning-by-chunking 'it-is-a-gradual-process-unlike-a-
compiler 'features-1)
(putprop 'learning-by-chunking '(composition proceduralisation)
'stages-in-1)
(putprop 'composition 'do-not-include-conditions-that-are-
consequences-of-earlier-actions 'method-alg-heuristic-1)
(putprop 'composition 'do-not-include-acts-that-set-transitory-goals 'method-alg-heuristic-2)
(putprop 'proceduralisation 'builds-specialised-versions-of-productions 'what-is-it-1)
(putprop 'proceduralisation 'eliminate-retrieval-of-information-from-long-term-memory 'method-alg-heuristic-1)
(putprop 'tuning '( analogy generalising-from-specific-plans-operators-that-capture-what-these-solutions-have-in-common discriminating-applicability-of-operators composition-of-operators ) 'method-alg-heuristic-1)

;***** learning with genetic algorithms *****
(putprop 'learning-with-genetic-algorithms 'a-variant-of-learning-by-induction-from-observation-and-discovery 'what-is-it-1)
(putprop 'learning-with-genetic-algorithms 'large-complex-poorly-understood-search-spaces 'applicability-conditions-1)
(putprop 'learning-with-genetic-algorithms 'large-number-of-examples 'requirements-1)
(putprop 'learning-with-genetic-algorithms '( fitness mating-operator genetic-operator ) 'subcomponents-1)
(putprop 'fitness 'chooses-the-fittest-as-parents-for-reproduction 'what-is-it-1)
(putprop 'mating-operator 'produces-offspring 'what-is-it-1)
(putprop 'genetic-operator '(crossover mutation) 'types-of-1)
(putprop 'genetic-operator 'determines-structure-of-products 'what-is-it-1)
(putprop 'learning-with-genetic-algorithms 'exploits-accumulating-information-to-bias-further-search 'features-1)
(putprop 'learning-with-genetic-algorithms 'provides-online-continuous-learning-and-allows-making-structural-changes-to-produce-behavioural-changes 'features-2)
(putprop 'learning-with-genetic-algorithms 'online-real-time-environment-where-radical-changes-in-behaviour-cannot-be-tolerated 'applicability-conditions-2)
(putprop 'learning-with-genetic-algorithms 'used-to-pick-out-fittest-of-competing-hypotheses 'applicability-conditions-3)
(putprop 'structural-changes '( alter-set-of-parameters change-complex-data-structures-like-agendas changing-the-task-program-
(putprop 'changing-the-task-program-itself 'space-is-large-and-complex 'applicability-conditions-1)
(putprop 'changing-the-task-program-itself 'dictates-choice-of-a-task-program-language-that-is-suited-to-manipulation-by-genetic-operators 'features-1)
(putprop 'learning-with-genetic-algorithms '( michigan-approach pittsburgh-approach ) 'types-of-1)
(putprop 'michigan-approach '( environment-provides-feedback either-reward-or-punishment-is-meted-out-depending-on-state-reached ) 'stages-in-1)
(putprop 'michigan-approach '(detectors-give-detector-messages perturbations rule-firings effector-actions ) 'method-alg-heuristic-1)

;**** learning by abduction ****
(putprop 'learning-by-abduction 'creating-theories-to-explain-new-findings 'what-is-it-1)
(putprop 'learning-by-abduction 'common-features-exist-among-all-abduction-systems-though-different-domains-dictate-different-methods-of-abduction 'general-principles-1)
(putprop 'learning-by-abduction '( plausibility assembly critique ) 'subcomponents-1 )
(putprop 'plausibility 'evaluation-of-hypotheses-using-a-hierarchial-classifier 'what-is-it-1)
(putprop 'assembly 'assemble-a-compound-explanation-from-a-set-of-candidate-hypotheses-that-explains-a-panel-of-findings 'what-is-it-1)
(putprop 'assembly 'the-explanation-should-consist-of-only-compatible-parts 'requirements-1)
(putprop 'learning-by-abduction 'abduction-steps 'method-alg-heuristic-1)
(putprop 'abduction-steps '( ab-step1 ab-step2 ab-step3 ab-step4 ab-step5 ) 'what-is-it-1)
(putprop 'ab-step1 'select-single-finding-to-be-expalined 'what-is-it-1)
(putprop 'ab-step2 'select-those-hypotheses-that-offer-to-explain-the-finding 'what-is-it-1)
(putprop 'ab-step3 'select-most-plausible-hypotheses 'what-is-it-1)
(putprop 'ab-step4 'integrate-selected-hypotheses-into-a-compound-explanation 'what-is-it-1)
(putprop 'ab-step5 'update-other-findings-that-are-now-explained-if-any-remain-unexplained-repeat-method 'what-is-it-1)
(putprop 'abduction-steps 'the-generative-act-is-not-creative 'constraints-problems-1)
(putprop 'the-generative-act-is-not-creative 'the-generation-of-hypotheses-will-have-to-be-through-a-creative-generator-rather-than-from-an-existing-set-of-hypotheses 'solutions-to-a-problem-1)
(putprop 'critique 'assembly-yields-a-complete-consistent-explanation 'applicability-conditions-1)
(putprop 'critique '( parsimony-critique essential-critique ) 'types-of-1)
(putprop 'parsimony-critique 'redundancy-checker 'what-is-it-1)
(putprop 'essential-critique 'identifies-the-important-hypotheses-that-offer-the-only-explanation-for-some-finding 'what-is-it-1)
(putprop 'learning-by-abduction 'top-level-goal 'tasks-of-1)
(putprop 'top-level-goal 'explain-all-findings 'what-is-it-1)
(putprop 'learning-by-abduction 'sub-goals 'tasks-of-2)
(putprop 'sub-goals 'obtain-candidate-set-of-hypotheses-explain-findings-by-making-a-compound-explanation-critique-the-compound-explanation 'subcomponents-1)

;***** failure driven learning *****
(putprop 'failure-driven-learning 'retraction-exclusion avoidance-assurance-inclusion ) 'tasks-of-1)
(putprop 'retraction 'generalise 'tasks-of-1)
(putprop 'retraction 'replace-prediction-with-the-generalisation-of-the-original-prediction-and-the-observed-disconfirming-one 'method-alg-heuristic-1)
(putprop 'exclusion 'specialise 'tasks-of-1)
(putprop 'exclusion 'add-negated-description-of-disconfirming-situation-to-pre-requisites 'method-alg-heuristic-1)
(putprop 'avoidance 'specialise 'tasks-of-1)
(putprop 'avoidance 'from-other-situations-find-antecedents-of-disconfirming-situation add-negation-of-these-to-precondition )
'method-alg-heuristic-1)
(putprop 'assurance 'specialise 'tasks-of-1)
(putprop 'assurance '(from-other-theories-find-antecedents-to-situation-that-will-ensure-success add-to-prerequisite ) 'method-alg-heuristic-1)
(putprop 'inclusion 'most-specific-generalisation 'tasks-of-1)
(putprop 'inclusion 'theory-specialised-to-situations-where-it-makes-a-valid-prediction 'method-alg-heuristic-1)

;****** learning by being told and asking questions ********
(putprop 'learning-by-being-told-and-asking-questions 'creation-of-macro-operators caching chunking equivalence-preserving-operations) 'important-examples-of-1 )
(putprop 'learning-by-being-told-and-asking-questions '(gaps-in-knowledge need-to-explain-observation address-particular-human-needs ) 'applicability-conditions-1)
(putprop 'learning-by-being-told-and-asking-questions 'selection-reformulation ) 'fundamental-operators-1)
(putprop 'learning-by-being-told-and-asking-questions 'ask-questions-while-working-on-assimilation-tasks 'method-alg-heuristic-1)
(putprop 'learning-by-being-told-and-asking-questions 'teacher
'requirements-1)
(putprop 'learning-by-doing 'instructor-watches-and-advises-the-
system-while-it-is-solving-problems-in-its-chosen-domain 'method-
alg-heuristic-1)
(putprop 'learning-by-being-told-and-asking-questions 'operationalisation 'tasks-of-1)
(putprop 'operationalisation 'reformulate-advice-into-a-heuristic-search-procedure 'what-is-it-1)
(putprop 'advice 'source-could-be-external-or-from-experience 'features-1)
(putprop 'operationalisation 'heuristic-1 'method-alg-heuristic-1)
(putprop 'heuristic-1 'unfold-definition-of-a-concept-in-the-problem 'what-is-it-1)
(putprop 'operationalisation '{ identify-sequence-of-choice-points-involved-and-the-set-of-admissible-alternatives-at-each-
one express-solution-criteria-as-a-function-of-the-choice-sequence } 'stages-in-1)
(putprop 'op-transformation-rules 'map-problem-to-the-method-used-for-operationalisation 'what-is-it-1)
(putprop 'generic-heuristic-search-procedure '{ ghs-step1 ghs-
step2 ghs-step3 ghs-step4 ghs-step5 ghs-step6 ghs-step7 } 'method-
alg-heuristic-1)
(putprop 'ghs-step1 'initial-path-typically-null 'what-is-it-1)
(putprop 'ghs-step2 'choice-set-function-which-leads-to-
alternating-extensions-for-path 'what-is-it)
(putprop 'ghs-step3 'step-order-predicate-which-
controls-order-of-generation-of-alternative-choice-elements 'what-is-it-1)
(putprop 'ghs-step4 'step-test-predicate-which-filters-of-choice-
set 'what-is-it-1)
(putprop 'ghs-step5 'path-order-predicate-which-controls-
selection-of-path-for-extension 'what-is-it-1)
(putprop 'ghs-step6 'path-test-which-is-to-be-satisfied-in-order-
for-path-to-be-added-to-active-list-of-paths 'what-is-it-1)
(putprop 'ghs-step7 'solution-path-must-satisfy-solution-
test-and-a-completion-test-which-checks-whether-a-complete-
sequence-of-choice-points-exists 'what-is-it-1)
(putprop 'conversion-of-problem-into-heuristic-search '{ problem-
sequence-of-choices-satisfying-some-conditions finding-choice-
sequence-that-defines-search-space search-criterion-reformulated-
as-a-function-of-this-sequence heuristic-search-method-
instantiated-givrn-choice-sequence-and-search-criterion
procedure-made-executable-on-data) 'method-alg-heuristic-1)
(putprop 'heuristic-search-domain '(observability-of-data
continuity stability) 'requirements-1)
(putprop 'operationalisation 'heuristic-2 'method-alg-heuristic-2)
(putprop 'heuristic-2 'p-derives-possible-p-where-possible p-is-
true-unless-p-is-known-to-be-false 'what-is-it-1)
(putprop 'improving-search '(pruning ordering compiling
modifying-the-data-flow-graph-by-splitting-or-adding-components-
that-improve-it collapsing) 'types-of-1)
(putprop 'operationalisation 'heuristic-3 'method-alg-heuristic-3)
(putprop 'heuristic-3 '(find-a-predicate-q-defined-directly-or-
indirectly-in-terms-of-p find-xl-to-xn-that-reduce-the
-expression-p-of-e1-to-en-implies-q-of-xl-to-xn-to-a-single-
condition-r) 'method-alg-heuristic-1)
(putprop 'heuristic-3 'finding-necessary-condition-for-p-of-e1-to-en
'applicability-conditions-1)

;**** learning by abstraction ****
(putprop 'learning-by-abstraction 'final-stages-of-learning
'applicability-conditions-1)

;***** learning by analogy *****
(putprop 'learning-by-analogy '(transformational-analogy
derivational-analogy) 'types-of-1)
(putprop 'transformational-analogy 'transfer-previous-solution-by-means-end-analysis-or-perturbation
'what-is-it-1)
(putprop 'transformational-analogy 'existence-of-a-measure-of-
similarity-between-past-and-current-problem 'requirements-1)
(putprop 'transformational-analogy '(description-of-past-problem-
solutions-remembered-and-indexed new-problem-is-to-be-matched-
against-a-potentially-large-number-of-past-experiences
-which-further-requires-an-operational-similarity-metric
solution-of-selected-old-problem-solution-is-to-be-transformed
-to-satisfy-needs-of-current-problem) 'requirements-1)
(putprop 'transformational-analogy 'inadequate-in-analogising-
base-concepts 'constraints-problems-1)
(putprop 'transformational-analogy 'solves-problems-inherent-in-macro-operators 'features-1)
(putprop 'derivational-analogy 'walks-through-the-reasoning-steps-in-the-construction-of-the-past-solutions-and-considers-whether-they-are-still-appropriate-or-have-to-be-reconsidered 'what-is-it-1)
(putprop 'derivational-analogy 'the-analogy-process-continues-while-the-justification-holds-for-both-the-problems 'features-1)
(putprop 'da-step1 'store-each-step-in-the-problem-solving 'what-is-it-1)
(putprop 'da-step1 'store-subgoal-structure-of-problem 'subcomponents-1)
(putprop 'da-step1 'store-each-decision-made 'subcomponents-2)
(putprop 'da-step1 'store-knowledge-accessed 'subcomponents-3)
(putprop 'da-step1 'store-resultant-solution 'subcomponents-4)
(putprop 'da-step2 'analyse-new-problem-by-applying-general-laws-and-weak-methods 'what-is-it-1)
(putprop 'da-step3 'if-reasoning-process-parallels-past-problem-situation-retrieve-derivation-tree 'what-is-it-1)
(putprop 'da-step4 'check-if-reasons-for-performing-each-step-are-still-valid 'what-is-it-1)
(putprop 'da-step5 'if-decision-is-not-valid-then-check-alternatives-at-that-situation-not-chosen-previously 'what-is-it-1)
(putprop 'learning-by-analogy '(verification-based-analogy analogy-by-an-extension-of-means-end-analysis) 'types-of-2)
(putprop 'verification-based-analogy 'analogise-then-conduct-experiment-to-verify 'what-is-it-1)
(putprop 'verification-based-analogy '( dynamic-behaviour-match theory-generation theory-completion theory-revision ) 'stages-in-1)
(putprop 'dynamic-behaviour-match 'analogy 'method-alg-heuristic-1)
(putprop 'theory-generation '( map-analogy operationalise-mapping

139
'method-alg-heuristic-1)

(putprop 'theory-completion
'vemethod-alg-heuristic-1)
(putprop 'operator-model-to-consistent-initial-model
'vemethod-alg-heuristic-1)
(putprop 'theory-revision 'new-situation 'applicability-conditions-1)
(putprop 'learning-by-analogy '( ana-recognition ana-elaboration
ana-evaluation ana-consolidation ) 'stages-in-1)
(putprop 'ana-recognition 'analogical-mapping 'basics-of-1)
(putprop 'ana-elaboration 'carry-over-features 'basics-of-1)
(putprop 'ana-consolidation 'check 'basics-of-1)
(putprop 'similarity 'refers-to-correspondence-between-attributes
'what-is-it-1)
(putprop 'analogy 'refers-to-correspondence-between-relations
'what-is-it-1)
(putprop 'hall-framework carbonell-framework winston-framework forbus-framework
'gentner-framework
'types-of-3)
(putprop 'hall-framework 'for-evaluation-and-elaboration-
disregard-attributes-of-objects-and-try-to-preserve-relations-
between-objects-retaining-those-relations-which-maintain-consistency
-among-higher-order-relations 'method-alg-heuristic-1)
(putprop 'systematicity-principle 'retain-those-relations-which-
maintain-consistency-among-higher-order-relations 'what-is-it-1)
(putprop 'carbonell-framework 'for-carry-over-of-reasons-
carry-over-those-structures-that-were-used-in-deriving-the-
source-structure 'method-alg-heuristic-1)
(putprop 'carbonell-framework 'most-elegant 'features-1)
(putprop 'gentner-framework 'structure-mapping-theory
'method-alg-heuristic-1)
(putprop 'structure-mapping-theory 'first-order-predicates-relate-
objects-while-second-order-predicates-relate-other-first-order-
predicates 'what-is-it-1)
(putprop 'gentner-framework 'does-not-address-indexing-and-
retrieval 'constraints-problems-1)
(putprop 'gentner-framework 'does-not-carry-over-attributes
'features-1)
(putprop 'winston-framework '( an-event-or-description-is
-indexed-by-the-type-of-object-or-concept-it-contains organise-
concepts-in-an-is-a-hierarchy
during-retrieval-prefer-connections-that-are-more-discriminating

140
accumulating-evidence-for-each-of-a-number-of-possible-object-to-object-mappings-between-representations-of-two-situations choose-the-match-that-scores-highest 'what-is-it-1)
(putprop 'similarity-matching 'evidence-of-a-match-depended-upon-the-number-of-relation-connections-preserved-for-a-given-alignment-of-objects-between-corresponding-objects 'features-1)
(putprop 'similarity-matching 'similarity-matching-problem 'constraints-problems-1)
(putprop 'similarity-matching-problem '(presupposes-well-defined-bounded-representation-model-of-situations-in-both-base-and-target-domains-available-on-input-which-is-not-usually-available-the-number-of-objects-available-are-many-more-than-those-that-come-into-the-analogy ) 'what-is-it-1)
(putprop 'similarity-matching-problem 'use-heuristics-to-delimit-what-is-to-be-imported 'solutions-to-a-problem-1)
(putprop 'use-heuristics-to-delimit-what-is-to-be-imported '(mapping-previously-formed-abstractions-like-those-embodying-causal-and-planning-rules mapping-constraints ) 'important-examples-of-1)
(putprop 'learning-by-analogy 'multiple-analogies-interact-and-contribute-to-the-formation-of-a-coherent-target-domain 'general-principles-1)
(putprop 'learning-by-analogy '( hidden-analogy incomplete-analogies misleading-analogies overzealous-analogies ) 'constraints-problems-1)
(putprop 'difference-function 'difference-between-initial-state-and-final-state 'what-is-it-1)
(putprop 'similarity-metric-computation 'difference-function 'types-of-1)
(putprop 'difference-function 'augmented-problem-state-difference-function 'types-of-1)
(putprop 'augmented-problem-state-difference-function 'used-to-compare-differences-between-path-constraints 'what-is-it-1)
(putprop 'augmented-problem-state-difference-function 'proportion-of-operator-preconditions-of-retrieved-sequence-that-are-satisfied-in-new-problem-situation-is-considered 'features-1)
(putprop 'analogy-by-an-extension-of-means-end-analysis '(mea-reminding-phase mea-transformation-phase) 'stages-in-1)
(putprop 'analogy-by-an-extension-of-means-end-analysis 'uses-augmented-difference-function 'features-1)
(putprop 'mea-reminding-phase '(mea-initial-state mea-final-state mea-path-constraints mea-operator-preconditions) 'subcomponents-1)
(putprop 'similarity-metric-computation 'relative-invariance-hierarchy 'types-of-1)
(putprop 'relative-invariance-hierarchy 'more-sophisticated-way-than-difference-function 'features-1)
(putprop 'mea-initial-state 'retrieved-solution-of-the-past-similar-problem 'what-is-it-1)
(putprop 'mea-final-state 'solution-satisfying-criteria-of-current-problem 'what-is-it-1)
(putprop 't-space 'analogy-transform-problem-space 'what-is-it-1)
(putprop 'state-in-t-space 'solution-path-in-original-space 'what-is-it-1)
(putprop 't-space '(general-insertion general-deletion subsequence-splicing subgoal-preserving-substitution final-segment-concatenation initial-segment-concatenation sequence-mashing operator-reordering parameter-substitution solution-sequence-truncation sequence-inversion) 'operators-1)
(putprop 'difference-function 'difference-metric 'equal-namewise-1)
(putprop 'subgoal-preserving-substitution 'substitute-operator 'method- alg-heuristic-1)
(putprop 'final-segment-concatenation 'treat-solution-sequence-as-a-macro-operator-and-apply-means-end-analysis-between-old-
(putprop 'initial-segment-concatenation 'means-end-analysis-to-find-a-path-from-new-initial-state-to-old-initial-state 'method-alg-heuristic-1)
(putprop 'sequence-mashing 'merge-operator-sequence-of-two-complimentary-solutions 'method-alg-heuristic-1)
(putprop 'operator-reordering 'to-satisfy-some-path-constraint 'features-1)
(putprop 'sequence-inversion 'if-goal-of-old-problem-matches-initial-state-of-new-problem-and-initial-state-of-old-problem-matches-goal-of-new-problem 'applicability-conditions-1)
(putprop 'analogy-by-an-extension-of-means-end-analysis 'analogy-based-on-fulfilling-goals 'basics-of-1)

;****** control / meta-knowledge / representation for srishti ******
(putprop 'computational-action 'transition-between-one-or-more-data-objects-and-some-other-data-objects 'what-is-it-1)
(putprop 'computational-action 'provides-the-atomic-elements 'features-1)
(putprop 'data-space '( set-of-all-data-objects set-of-all-computational-actions ) 'subcomponents-1)
(putprop 'input-interface 'set-of-feature-detectors 'what-is-it-1)
(putprop 'message 'a-string-of-feature-values 'what-is-it-1)
(putprop 'environment-oriented-approach 'uses-no-symbol-or-grammar 'what-is-it-1)
(putprop 'environment-oriented-approach 'opposed-to-a-language-oriented-one 'features-1)
(putprop 'schemata 'provides-means-for-transferring-knowledge-from-one-situation-to-another 'features-1)
(putprop 'control '(general-learning-mechanism predictive-features interest) 'subcomponents-1)
(putprop 'control 'induction-used-for-primary-immediate-premises-deduction-follows 'stages-in-1)
focus-is-determined-by-predictive-features-and-interest
features-1)
(a-domain-of-knowledge-is-dependent-upon-four-granular-sources-Apparatus-Techniques-Phenomena-PrimitiveConcepts
general-principles-1)
(learning-programs-must-contain-a-representation-of-a-learning-theory-which-facilitates-reasoning-about-the-theory
basics-of-1)
(control 'c-primary-need 'basics-of-2 )
(control-structure-must-evaluate-each-methods-appropriateness-to-the-situation-and-invoke-the-selected-learning-strategy
what-is-it-1)
(control 'sponsor-selector-mechanism 'method-alg-heuristic-1)
(heuristic-has-a-domain-of-relevance 'features-2 )
(discovery-stages 'stages-in-2)
(detect-domain-concepts define-and-study-heuristics augment-the-representation define-new-representations )
(use-knowledge-about-heuristic-search-and-the-meta-knowledge-consisting-of-the-intended-purpose-of-learned-heuristics
basics-of-3 )
(derive-justifiable-generalisations respond-to-deficiencies-in-vocabulary )
(derive-justifiable-generalisations 'analysis-of-instances 'method-alg-heuristic-1)
(analysis-of-instances 'analysis-of-why-an-example-is-a-positive-one 'types-of-1)
(analysis-of-why-an-example-is-a-positive-one definition-of-criterion-by-which-the-instance-is-labelled-positive 'requirements-1)
(definition-of-criterion-by-which-the-instance-is-labelled-positive 'declarative-definition-of-these-allows-system-to-reason-symbolically-about-its-learning-goals 'features-1)
(analysise-of-instances determines-a-logically-sufficient-condition-for-positive-instances 'cause-1)
(determines-a-logically-sufficient-condition-for-positive-instances 'generate-explanation-of-how-example-satisfies extract-from-explanation-a-sufficient-condition-for-
satisfying-positive-example sufficient-condition-is-restated-in-
general-language propogate-restrictions-on-problem
-states-through-solution-tree-and-combine-them-
into-a-generalised-sufficient-condition ) 'method-alg-heuristic-1
)
(putprop 'translation-of-sufficient-condition-into-a-general-
language 'knowledge-about-correspondence-between
-the-representation-language-in-which-analysis-is-being-done-and-
the-general-language-used-to-describe-heuristics 'requirements-1
)
(putprop 'sufficient-condition 'producing-or-proposing-mechanisms
-that-are-powerful-enough-to-produce-the-observed-
learning-of-the-necessary-skills 'what-is-it-1
)
(putprop 'control 'the-constraint-of-psychological-validity-
particularly-in-machine-learning-may-facilitate
-achieving-a-robust-intelligent-system 'basics-of-4
)
(putprop 'control 'recognise-act-computational-paradigm
'types-of-1
)
(putprop 'recognise-act-computational-paradigm 'very-generally-
applicable 'applicability-conditions-1
)
(putprop 'recognise-act-computational-paradigm 'this-along-with-
-global-working-memory-relieves-rule-encoder-of-some
-of-the-burden-of-specifying-control-information 'features-1
)
(putprop 'control 'explicit-steps-reduce-number-and-complexity
-of-conditions-but-reduces-flexibility 'basics-of-5
)
(putprop 'flexibility 'having-the-control-to-be-open-on-each-
cycle-to-global-recognitions-that-can-change-the-direction-of-
processing 'cause-1
)
(putprop 'architecture-of-SRISHTI 'flexibility 'cause-1
)
(putprop 'control '(top-down-approach bottom-up-approach
'types-of-2
)
(putprop 'top-down-approach '(has-better-noise-immunity-than
-bottom-up-approach more-flexible-than-bottom-up-approach
-computationally-more-expensive-than-bottom-up-approach
'features-1
)
(putprop 'top-down-approach 'the-working-hypotheses-must-be
-checked-repeatedly-to-ensure-that-they-subsume-all-input-events
'constraints-problems-1
)
(putprop 'top-down-approach '(the-initial-hypothesis-is-some-
elements-from-the-partially-ordered-set-of-all-possible-
descriptions modify-this-by-either-generalisation-or
-specialisation- check-whether-the-required-criteria-are-satisfied
-if-not-modify-hypothesis-again ) 'method-alg-heuristic-1
)
(putprop 'bottom-up-approach 'generalise-from-input-events-one-at-a-time-until-a-final-conjunctive-generalisation-is-computed

146
'method-alg-heuristic-1)

(putprop 'bottom-up-approach
  'more-noise-sensitive-than-top-down-approach faster-than-top-down-approach less-flexible-than-top-down-approach)

(putprop 'generality 'develop-methods-that-incorporate-various-general-principles-of-induction-like-constructive-induction-together
  -with-mechanisms-for-using-exchangeable-packages-of-problem-specific-knowledge 'method-alg-heuristic-1)

(putprop 'flexibility '(discover-descriptions-with-conjunctive-generalisations-disjunctive-generalisations-exceptions-etc

(putprop 'control '(winstons-control sprouter-control vere-control meta-dendral-control induce-1.2-control) 'types-of-3)

(putprop 'winstons-method 'winstons-control 'subcomponents-1)

(putprop 'sprouter 'sprouter-control 'subcomponents-1)

(putprop 'vere 'vere-control 'subcomponents-1)

(putprop 'winstons-control 'semantic-network-consisting-of-nodes-and-links-with-in-addition-nodes-that-describe-each-type-of-link
  'representation-1)

(putprop 'winstons-control 'c-note 'representation-2)

(putprop 'c-note 'comment-notes 'what-is-it-1)

(putprop 'c-note '(for-nodes for-links) 'types-of-1)

(putprop 'for-nodes 'intersection a-kind-of-merge a-kind-of-chain exit) 'types-of-1)

(putprop 'for-links 'negative-satellite-pair must-be-satellite-pair must-not-be-satellite-pair supplementary-pointer) 'types-of-1)

(putprop 'winstons-control 'the-semantic-network-representation-used-does-not-allow-disjunction-to-be-represented
  'representation-3)

(putprop 'bottom-up-approach 'sprouter-control 'important-examples-of-1)

(putprop 'sprouter-control 'bottom-up-interference-match-after-which-similarities-are-looked-for 'method-alg-heuristic-1)

(putprop 'vere-control 'uses-spreading-activation-to-extract-relations-from-background-knowledge 'method-alg-heuristic-1)

(putprop 'top-down-approach 'meta-dendral-control 'important-examples-of-1)

(putprop 'meta-dendral-control 'uses-top-down-model-driven-approach 'method-alg-heuristic-1)

(putprop 'induce-1.2-control 'represaents-events-as-conjunction-
of-selectors 'representation-1
(putprop 'induce-1.2-control 'descriptors-are-maintained-as
- unary-descriptors-and-non-unary-descriptors 'representation-2
(putprop 'unary-descriptors 'represent-attribute-description 'what-is-it-1
(putprop 'non-unary-descriptors 'represent-structure-specifying-descriptors 'what-is-it-1
(putprop 'induce-1.2-control 'structure-specifying-description-searched-using-beam-search 'method-alg-heuristic-1
(putprop 'induce-1.2-control 'input-events-translated-into-attribute-space 'method-alg-heuristic-2
(putprop 'general-learning-mechanism 'task-generality knowledge-generality aspect-generality transfer-of-learning 'tasks-of-1
(putprop 'task-generality 'symbol-level 'applicability-conditions-1
(putprop 'knowledge-generality 'knowledge-level 'applicability-conditions-1
(putprop 'aspect-generality 'symbol-level 'applicability-conditions-1
(putprop 'aspect-generality 'all-aspects-of-the-system 'features-1
(putprop 'transfer-of-learning 'from-one-situation-to-another 'what-is-it-1
(putprop 'general-learning-mechanism 'multistrategy-learner deliberate-learner simple-experience-learner 'types-of-1
(putprop 'multistrategy-learner 'ACT 'important-examples-of-1
(putprop 'ACT 'six-learning-strategies 'features-1
(putprop 'deliberate-learner 'modify-to-improve 'what-is-it-1
(putprop 'simple-experience-learner 'incorporates-a-single-learning-mechanism-with-the-system-modifying-itself-based-on-experience 'what-is-it-1
(putprop 'deliberate-learner 'modification-done-based-on-analysis-of-task-to-be-done 'features-1
(putprop 'condition-r 'system-hypotheses patterns-in-data 'types-of-1
(putprop 'action-r 'formulate-hypotheses change-confidence suggest-new-experiments 'types-of-1
(putprop 'control 'naive-physics-stage expert-models-stage 'stages-in-3
(putprop 'naive-physics-stage 'replace-cause-states-with-theories-about-the-specific-mechanisms-of-change 'what-is-it-1
(putprop 'naive-physics-stage 'processes-added-to-explain-changes a-vocabulary-of-processes-that-describe-experience-is

148
constructed) 'method-alg-heuristic-1)
(putprop 'expert-models-stage 'resolve-ambiguities 'what-is-it)
(putprop 'expert-models-stage '(construct-domain-independent-
generalisation reformulation-is-invoked) 'method-alg-heuristic-
1)

(putprop 'theory-generation 'process-of-proposing-an-initial-
ty-to-explain-a-novel-phenomenon 'what-is-it-1)
(putprop 'theory-generation 'abduction 'types-of-1)
(putprop 'hypothesised-theory '(primary-process auxiliary-process)
'subcomponents-1)
(putprop 'primary-process 'affects-an-observable-quantity
'features-1)
(putprop 'auxillary-process 'used-in-analogous-cases 'features-1)
(putprop 'auxillary-process 'does-not-affect-observable-quantity
'features-2)

(putprop 'knowledge-acquisition-goals 'classify-distinction-
between-what-is-to-be-learnt-and-how-it-is-to-be-learnt
'features-1)
(putprop 'incremental-learning 'many-partially-learned-
heuristics-represented-by-all-plausible-descriptions-of-the-
heuristic 'requirements-1)

****** Preferred features ******

(putprop 'learning-strategy '(form-of-data-that-it-can-process
background-knowledge-it-requires amount-of-computational-time-
required amount-of-space-required) 'requirements-1)

(putprop 'output-of-a-learning-strategy '(form certainty
breadth-of-coverage) 'evaluation-criteria-1)

(putprop 'problem-solving-approaches '(weak-methods plan-
instantiation hybrid-involving-search-whenever-plan-is-not-
available analogy) 'types-of-1)
(putprop 'weak-methods 'focussed-search-for-suitable-operator-
sequences-involving-analysis-of-states-resulting-
-from-application-of-different-operators 'what-is-it-1)
(putprop 'weak-methods 'no-domain-knowledge-is-available
'applicability-conditions-1)
(putprop 'plan-instantiation 'specific-domain-knowledge-is-
available 'applicability-conditions-1)
(putprop 'reduce-problem-by-using-general-plans 'general-plans-
apply-but-not-specific-ones 'applicability-conditions-1)

149
analogy 'no-plans-apply-but-problem-has-resemblance-to-a-previous-problem 'applicability-conditions-1
control 'if-a-goal-is-to-match-a-set-of-statements-then-match-the-first-statement-in-the-set 'method-alg-heuristic-3
control 'if-the-goal-is-to-match-a-statement-in-a-set-and-the-problem-contains-a-match-to-that-statement-then-go-on-to-match-the-next-statement-in-that-set 'method-alg-heuristic-4
control 'if-the-goal-is-to-match-a-set-of-statements-and-the-last-statement-has-been-matched-then-go-on-to-the-goal-that-follows 'method-alg-heuristic-5

representation
representation 'heuristic-rules-were-represented-as-individual-concepts-in-EURISKO 'important-examples-of-1
representation 'event-is-represented-as-a-sequence-of-assignments 'features-1
representation 'the-set-of-all-possible-events-forms-the-event-space 'features-2
representation 'a-situation-is-a-source-of-information-about-values-of-variables-and-atomic-functions-in-a-description 'features-3
representation 'a-general-structure-of-knowledge-needs-to-be-maintained 'requirements-1
representation 'rules-that-indicate-presence-of-one-feature-cause-presence-of-another-feature 'types-of-1
representation 'causal-chains-can-connect-predictive-features-to-non-predictive-ones 'types-of-2
representation 'representation-in-the-form-of-condition-r-and-action-r-pairs 'types-of-3
representation 'version-space 'types-of-4
version-space 'provides-a-compact-storage-by-storing-only-the-maximally-specific-members-S-and-the-maximally-

150
(putprop 'representation '(production-system semantic-network )
'types-of-5 )
(putprop 'production-system 'used-to-represent-procedural-
knowledge 'features-1 )
(putprop 'semantic-network 'used-to-represent-declarative-
knowledge 'features-1 )
(putprop 'representation 'schema 'types-of-6 )
(putprop 'schema 'postulate-schema 'types-of-1 )
(putprop 'postulate-schema 'initial-declarative-encoding-of-facts
'what-is-it-1 )
(putprop 'postulate-schema ' ( background-knowledge-in-the-form-
of-description-of-situation hypothesis comment conclusion )
'subcomponents-1 )

(putprop 'representation 'hierarchial-description 'types-of-7 )
(putprop 'hierarchial-description 'representing-information-at-
increasing-levels-of-description-with-higher-levels-
representing-or-describing-more-complex-data 'what-is-it-1 )
(putprop 'hierarchial-description 'allows-hypotheses-also-to-be-
treated-as-data 'features-1 )
(putprop 'representation 'canonical-representations-avoid-
duplication 'features-6 )
(putprop 'representation 'if-then-rules-are-represented-as-links-
between-slots 'features-7 )
(putprop 'representation 'changes-in-representation 'features-8 )
(putprop 'representation 'storage-is-done-by-
indexing-on-the-type-of-object-contained-in-the-net-or-
description 'features-9 )
(putprop 'representation 'memory-organised-by-similarities-in-
goal-states-initial-states-means-available-and-path-constraints
'features-10 )
(putprop 'representation 'acheive-P-denotes-goal-of-satisfying-
predicate-P 'basic-of-1 )
(putprop 'representation 'during-S-E-denotes-that-it-is-true-if-
event-E-occurs-during-scenario-S 'basics-of-2 )
(putprop 'scenario 'sequence-of-events 'what-is-it-1 )
(putprop 'representation 'for-every-a-S-P-a-denotes-that-it
-is-true-if-every-element-of-S-satisfies-P 'basics-of-3 )
(putprop 'representation 'the-a-S-P-a-denotes-the-
unique-element-of-S-that-satisfies-P 'basics-of-4 )
(putprop 'representation 'for-some-a-S-Ea-denotes-any-
event-Ea-such-that-a-is-in-S 'basics-of-5 )
(putprop 'representation 'each-a-S-Ea-denotes-event-
sequence-Ea1-to-Ean-where-S-is-equal-to-the-set-a1-to-an 'basics-
of-6
(putprop 'representation 'scenario-el-to-en-denotes-sequence-of-events-el-to-en 'basics-of-7)
(putprop 'representation 'integration-of-new-knowledge 'requirements-2)
(putprop 'integration-of-new-knowledge '{there-are-things there-are-subclasses-of-things there-are-relations-among-things there-are-subclasses-of-relations some-relations-are-functions sometimes-a-given-set-of-constraints-is-sufficient-to-distinguish-an-unique-individual equals-are-interchangeable} 'general-principles-1)
(putprop 'representation 'tangled-tree 'types-of-8)
(putprop 'tangled-tree 'nodes-can-have-many-parents 'features-1)
(putprop 'tangled-tree 'is-a-hierarchies 'subcomponents-1)
(putprop 'is-a-hierarchies 'suited-for-language-acquisition 'features-1)
(putprop 'representation 'theory-representation 'types-of-9)
(putprop 'theory-representation '{for-all-situations-S-there-exists-a-subsequent-situation-S2-where-condition-of-S-intersection-action-of-S-results-in-environment-S2-representation-1}
(putprop 'theory-representation '{assumed-conditions planned-actions predicted-effects} 'subcomponents-1)

(putprop 'changes-in-representation '{if-a-slot-S-is-very-important-and-all-its-values-are-units-then-create-a-new-kind-of-slot-which-contains-all-the-relations-among-the-values-of-slot-S 'method-alg-heuristic-1)
(putprop 'changes-in-representation '{if-some-operation-is-performed-frequently-then-shift-to-a-representation-in-which-it-is-less-expensive-to-perform 'method-alg-heuristic-2})

;**** heuristics to decide on next heuristic to apply ****
(putprop 'control 'non-monotonic-reasoning 'method-alg-heuristic-6)
(putprop 'non-monotonic-reasoning '{assume-some-of-the-uncertain-conditions-of-the-heuristic-hold-good-while-tagging-these-to-enable-undoing-if-assumption-proves-wrong 'what-is-it-1)
(putprop 'control 'deferral 'method-alg-heuristic-7)
(putprop 'deferral '{if-all-the-heuristics-give-rise-to-a-particular-action-then-take-that-action-and-hope-that-soon-more-knowledge-}
will-be-available-to-make-the-choice-among-the-heuristics 'what-is-it-1)
(putprop 'control 'approximate 'method-alg-heuristic-8)
(putprop 'approximate 'weaken-some-of-the-conditions-for-applicability-of-the-heuristic 'what-is-it-1)

; **** interest ******
; rules-31 to 39 are reserved as interest rules
(putprop 'interest 'defines-by-heuristic-measure-of-what-is-likely-to-help-in-learning-knowledge 'what-is-it-1)
(putprop 'interest 'expectations-that-are-not-satisfied-by-wide-margins-provide-interesting-sources 'method-alg-heuristic-1)
(putprop 'interest 'Rule-31 'method-alg-heuristic-2)
(putprop 'Rule-31 'if-some-normally-inefficient-operation-can-be-done-quickly-on-X-then-X-is-a-more-interesting-concept-than-previously-thought 'what-is-it-1)
(putprop 'interest 'Rule-32 'method-alg-heuristic-3)
(putprop 'Rule-32 'if-X-does-not-meet-predictions-then-X-is-more-interesting 'what-is-it-1)
(putprop 'interest 'Rule-33 'method-alg-heuristic-4)
(putprop 'Rule-33 'if-exactly-one-element-of-a-class-satisfies-an-interesting-property-then-that-class-becomes-more-interesting-This-is-especially-true-of-a-function-that-always-produces-an-output-satisfying-this-property 'what-is-it-1)
(putprop 'interest 'Rule-34 'method-alg-heuristic-5)
(putprop 'Rule-34 'if-A-is-similar-to-B-in-a-key-way-and-uses-less-recursion-then-A-is-interesting-and-worth-preserving 'what-is-it-1)
(putprop 'interest 'Rule-35 'method-alg-heuristic-6)
(putprop 'Rule-35 'focus-of-attention-heuristic 'what-is-it-1)
(putprop 'focus-of-attention-heuristic 'work-on-what-you-were-working-on 'what-is-it-1)
(putprop 'interest 'Rule-36 'method-alg-heuristic-7)
(putprop 'Rule-36 'if-the-results-of-performing-a-task-F-have-always-been-numerous-and-worthless-then-lower-the-expected-worth-of-F 'what-is-it)
(putprop 'interest 'Rule-37 'method-alg-heuristic-8)
(putprop 'Rule-37 'notice-invariants 'what-is-it-1)
(putprop 'notice-invariants 'features-that-recurre-or-remain-invariant-are-considered-interesting 'what-is-it-1)
(putprop 'interest 'Rule-38 'method-alg-heuristic-1)
(putprop 'Rule-38 'a-domain-is-interesting-if-a-specialised-method-is-discovered-that-is-effective-to-solve-problems-in-that-domain 'what-is-it-1)

153
(putprop 'interest 'many-coincident-relationships-arise-among-concepts-in-an-interesting-domain 'features-1)