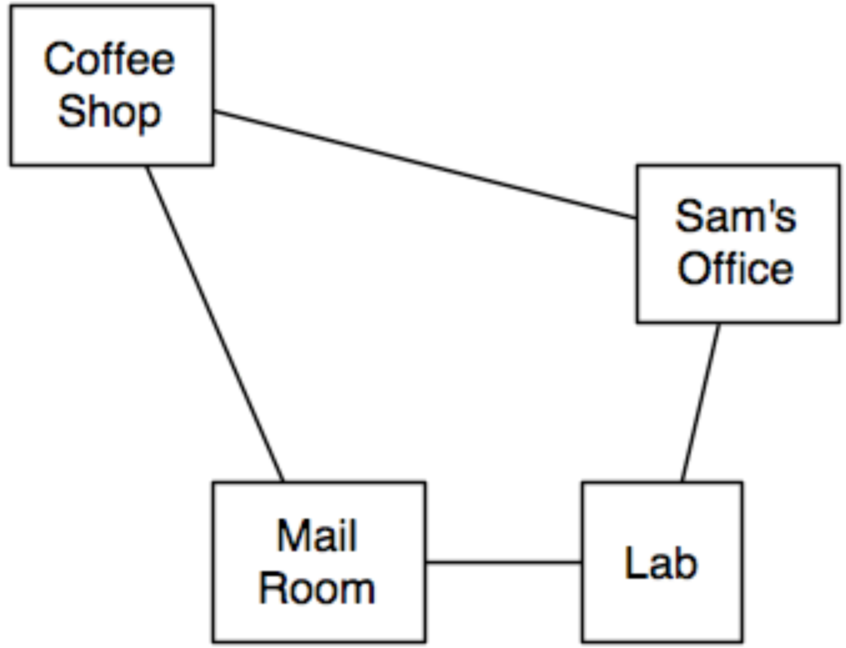
Example 6.1 Consider a delivery robot world with mail and coffee to deliver. Assume a simplified domain with four locations as shown in Figure 6.1



Adapted from ArtInt

Features to describe states

RLoc
– Rob's location
RHC
– Rob has coffee
SWC
– Sam wants coffee
MW
– Mail is waiting
RHM
– Rob has mail

Actions

mc
– move clockwise
mcc
– move counterclockwise
puc
– pickup coffee
dc
– deliver coffee
pum
– pickup mail
dm
– deliver mail

State	Action	Resulting State
$\langle lab, \neg rhc, swc, \neg mw, rhm\rangle$	*mc*	$\langle mr, \neg rhc, swc, \neg mw, rhm\rangle$
$\langle lab, \neg rhc, swc, \neg mw, rhm\rangle$	*mcc*	$\langle off, \neg rhc, swc, \neg mw, rhm\rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm\rangle$	*dm*	$\langle off, \neg rhc, swc, \neg mw, \neg rhm\rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm\rangle$	*mcc*	$\langle cs, \neg rhc, swc, \neg mw, rhm\rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm\rangle$	*mc*	$\langle lab, \neg rhc, swc, \neg mw, rhm\rangle$
...	...	...

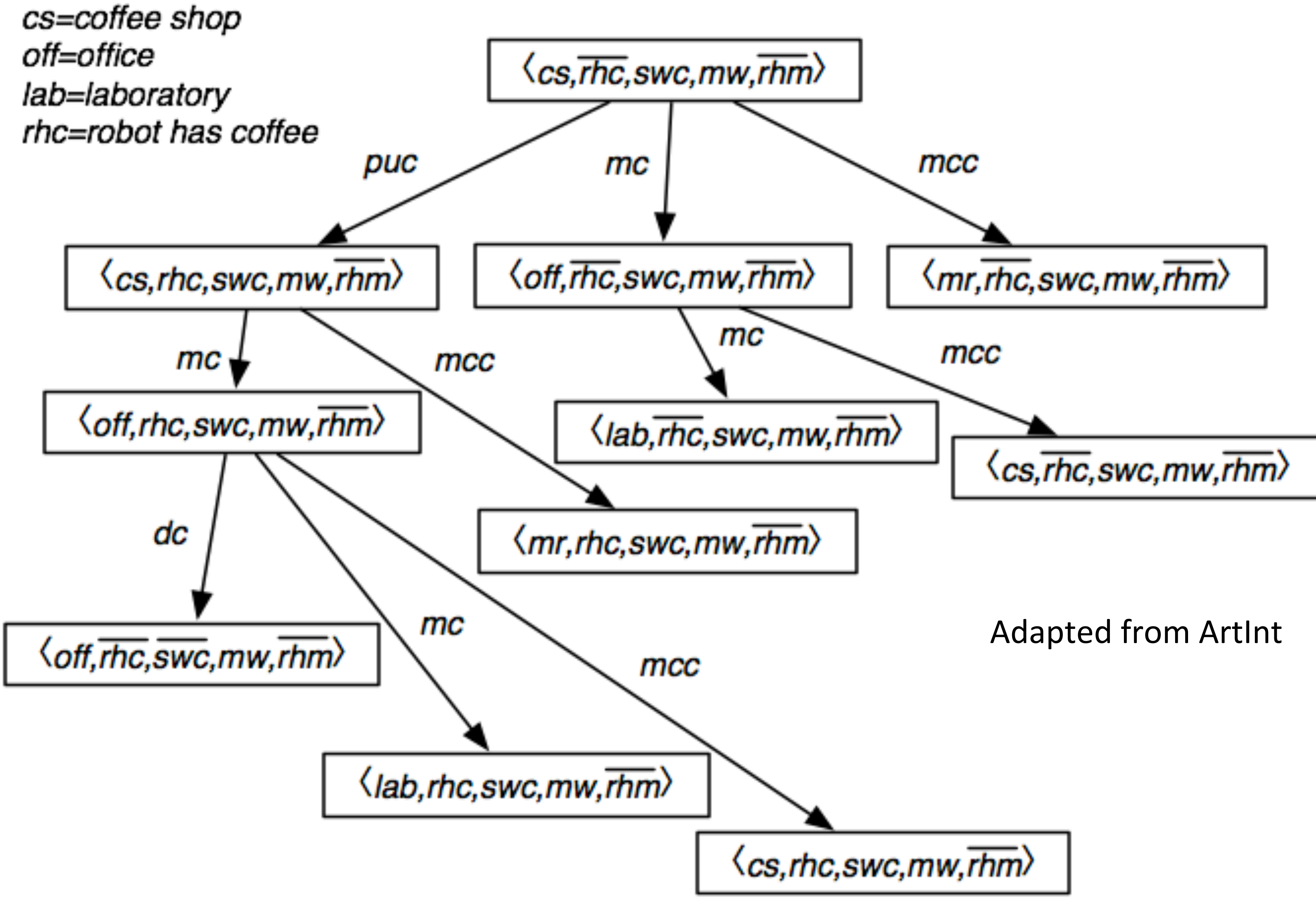
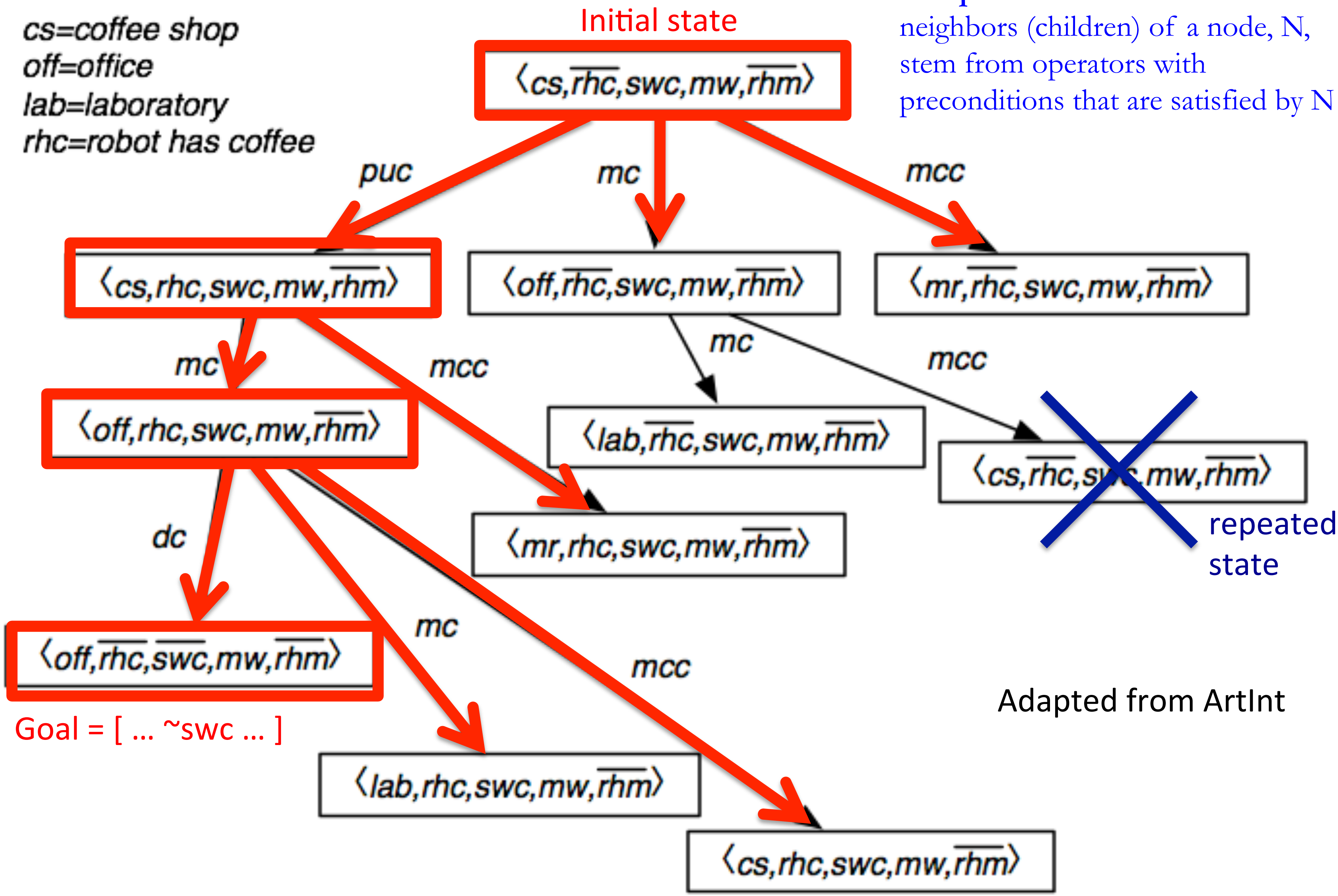


Figure 6.1 : Part of the search space for a state-space planner

A depth-first forward search: neighbors (children) of a node, N, stem from operators with preconditions that are satisfied by N



Adapted from ArtInt

Figure 6.1 : Part of the search space for a state-space planner

STRIPS Operators , which I will write pre(op) ➔ eff(op)

puc: [RHC = ~rhc, RLOC = cs] ➔ [RHC = rhc]

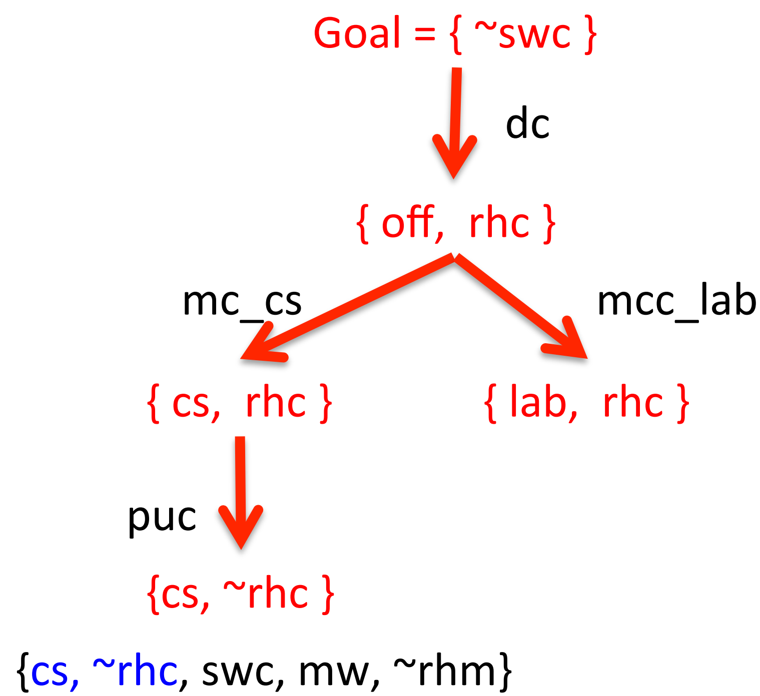
dc: [RHC = rhc, RLOC = off] ➔ [RHC = ~rhc, SWC = ~swc]

mc_cs: [RLOC = cs] ➔ [RLOC = off]

mcc_so = [RLOC = off] ➔ [RLOC = cs]

. . .

Regression or backward planning:
neighbors (children) of a node, N,
stem from operators with
effects that would need to be achieved
to satisfy (sub)goals of N



Exercise 6 from Section 6.8 of text (if we want to use composite (aka macro) operators in search (forward or backward) then we need precondition and effects just like any operator

Existing notation: a is an action; pre(a), eff(a)

C is a set of conditions (variable value assignments)

Conflicts (C1, C2) = set of conditions in C2 that conflict with a condition in C1

$$= \{\ V = v \text{ in } C2 \mid \text{Exists } V = v' \text{ in } C1 \text{ where } v \neq v'\}$$

Define WeakestResult (a) = the minimal set of conditions true after action a 'executed'

$$= eff(a) + \overbrace{(pre(a) - conflicts\ (eff(a), pre(a)))}^{\text{Preconditions unchanged by action a}}$$

Preconditions that are no longer true

(a) What is eff(a1;a2) ?

?

$$eff(a1;a2) = [eff(a2)\ \cancel{- pre(a2)} - pre(a1)] + [eff(a1)\ \cancel{- pre(a1)} - conflicts\ (eff(a2), eff(a1))]$$

If an operator only reflects those conditions that CHANGE, then there will be no conditions shared in eff(a2) and pre(a2), and there should NOT be any shared with pre(a1) in final macro (composite) operator

(b) When is the composite action impossible?

When Conflicts (WeakestResult(a1), pre(a2)) $\neq$ {}

In contrast, composite operator (a1; a2) is consistent if Conflicts (WeakestResult(a1), pre(a2)) = {}

Exercise 6 from section 6.8 of text cont

or WeakestResult(a1)

(c) What is pre(a1;a2) ?

pre(a1;a2) = [pre(a1)] + [pre(a2) – eff(a1)]

(d) puc;mc_cs where puc: [~rhc, cs] → [rhc] and mc_cs: [cs] → [off]

pre(puc; mc_cs) = [~rhc, cs]   eff(puc; mc_cs) = [rhc, off]

~~Typo in example 6.1 (noted on hypthes.is)~~
Fixed in online version of book for F2018

(e) puc; mc_cs; dc where dc: [rhc, off] → [~rhc, ~swc]

pre(puc;mc_cs; dc) = pre((puc;mc_cs); dc) FYI

pre(puc;mc_cs; dc) = [~rhc, cs]   eff(puc;mc_cs;dc) = [~swc, off]

Various typos
Fixed as indicated
They would be in video

(f) mcc_off;puc;mc_cs;dc where mcc_off = [off] → [cs]

pre(mcc_off;puc;mc_cs;dc) = pre(mcc_off; ((puc;mc_cs); dc)) FYI

pre(mcc_off;puc;mc_cs;dc) = [off, ~rhc]   eff(mcc_off;puc;mc_cs;dc) = [~swc]