Planning for a Kitting Station

Zeid Kootbally
National Institute of Standards and Technology
zeid.kootbally@nist.gov

1 The Kitting Domain

The domain for kitting contains some fixed equipment: a robot, a work table, a part gripper, a kit tray gripper, and a changing station. Items that enter the workstation include empty kit trays, boxes in which to put finished kit trays, boxes that contain empty kit trays, and part supplies. Items that leave the workstation may be boxes with finished kits inside, empty part trays, empty boxes. An external agent is responsible of moving the items that leave the workstation. We assume that the workstation has only one work table, one changing station, and one robot.

The goal of the kitting process is to build a kit that contains two parts of type A, one part of type B, and one part of type C.

2 State-Variable Representation

In a state-variable representation, each state is represented by a tuple of values of n state variables $\{x_1, \ldots, x_n\}$, and each action is represented by a partial function that maps this tuple into some other tuple of values of the n state variables.

2.1 Constants Symbols

Constant symbols [**?**], also called *object symbols* are partitioned into disjoint classes corresponding to the objects of the domain. Let D be the finite set of all constant symbols in the kitting planning domain. D is partitioned into the following sets of constant symbols.

- A set of *robots*{r1, r2, ...}: Robotic arms that can move objects in the workstation in order to build kits.

- To build a kit, the different types of kit trays are used as illustrated in the following sequence:
  empty kit tray→*partially full kit tray*→*finished kit tray*.
  The different types of kit trays are the ones described below.
  - A set of *empty kit trays*{ekt1, ekt2, ...}: Empty kit trays do not contain parts yet. Once an empty kit tray contains at least one part, it becomes a *partially full kit tray*.
  - A set of *partially full kit trays*{pfkt1, pfkt2, ...}: A partially full kit tray contains some parts but not all the parts that are necessary to build a kit. Once a partially full kit tray has all the parts required for a kit, the partially full kit tray becomes a *finished kit tray*.
  - A set of *finished kit trays*{fkt1, fkt2, ...}: A finished kit tray contains all the parts that constitute a kit.
  - A set of *part supplies*{ps1, ps2, ...}: Part supplies may be a tray or box with parts inside or a box containing trays with parts. Multiple part supplies can be located in the workstation, and each part supply contains a specific type of part.
- A set of *parts*{pA-1, pA-2,..., pB-1, pB-2,..., pC-1, pC-2,...}: A component that goes in a kit. Parts are delivered to the workstation in part supplies. A kit can have multiple parts of the same type (from the same part supply), thus to differentiate these parts from part supplies A, B, and C for example, parts from part supply A are noted pA-1, pA-2,..., parts from part supply B are noted pB-1, pB-2,..., and parts from part supply C are noted pC-1, pC-2,....
- A set of *grippers*{g1, g2, ...}: Device at the end of the robotic arm used to grasp objects in the workstation. Two types of grippers exist for the kitting domain:
  1. Grippers that hold parts.
  2. Grippers that hold empty and finished kit trays (empty and finished kit trays are the same kit trays).
- A set of *boxes of finished kit trays*{boxfkt1, boxfkt2, ...}: A box that contains only finished kit trays.
- A set of *boxes of empty kit trays*{boxekt1, boxekt2, ...}: A box that contains only empty kit trays.
- A symbol *work table*{wtable}: a surface used to build kits. There is no need to have separate symbols for individual work tables because each work table is uniquely identified by the workstation it is in.
- A symbol *changing station*{chstation}: a surface that contains grippers not in use. There is no need to have separate symbols for individual changing stations

because each changing station is uniquely identified by the workstation it is in. The changing station can be placed on the work table or on a separate surface.

2.2 Object Variable Symbols

Object variable symbols are typed variables [**?**]. Each object variable v has a range D^v which is the union of one or more classes, e.g., $\mathsf{r} \in$ *robots*, $\mathsf{g} \in$ *grippers*, etc.

2.3 State Variable Symbols

A k-ary state variable [**?**] is an expression of the form $x(v_1, \ldots, v_k)$, where x is a state-variable symbol and each v_i is either an object symbol (see Section 2.1) or an object variable (see Section 2.2). A state variable denotes an element of a state-variable function,

$$x : D_1^x \times \cdots \times D_k^x \times S \rightarrow D_{k+1}^x$$

where each $D_i^x \in D$ is the union of one or more classes.
The topology of the kitting domain is denoted using the following unground state variables.

- *gloc*: *grippers* $\times S \rightarrow$ *robots* $\cup$ *changing station* $\cup \{nil\}$: designates the location of a gripper in the workstation. The gripper can be in the changing station or attached to the robot ($\{nil\}$).
- *topworktable*: *work table* $\times S \rightarrow$ *empty kit trays* $\cup$ *partially full kit trays* $\cup$ *finished kit trays* $\cup \{nil\}$: designates the object placed on the work table. It could be an empty kit tray, a partially full kit tray, a finished kit tray, or nothing.
- *ektloc*: *empty kit trays* $\times S \rightarrow$ *boxes of empty kit trays* $\cup$ *work table* $\cup \{nil\}$: designates the different possible locations of an empty kit tray in the workstation. An empty kit tray can be in the box of empty kit trays, on the work table, or being held by the robot ($\{nil\}$) when the robot moves the empty kit tray from the box of empty kit trays to the work table.
- *fktloc*: *finished kit trays* $\times S \rightarrow$ *boxes of finished kit trays* $\cup$ *work table* $\cup \{nil\}$: designates the different possible locations of a finished kit tray in the workstation. A finished kit tray can be in the box of finished kit trays, on the work table, or being held by the robot ($\{nil\}$) when the robot moves the finished kit tray from the work table to the box of finished kit trays.
- *ploc*: *parts* $\times S \rightarrow$ *partially full kit trays* $\cup$ *part supplies* $\cup$ *finished kit trays* $\cup \{nil\}$: designates the different possible locations of a part in the workstation. The part can be in the part supply, in the partially full kit tray, in the finished kit tray, or being held by the robot ($\{nil\}$) when the robot moves the part from the part supply to the empty or partially full kit tray,

- *rhold*: *robots* $\times S \rightarrow$ *finished kit trays* $\cup$ *empty kit trays* $\cup$ *parts* $\cup \{nil\}$: designates the object being held by the robot. It can be a finished kit tray, an empty kit tray, a part, or nothing. We assume that the robot is already equipped with the appropriate gripper.
- *boxempty*: *boxes of finished kit trays* $\times$ *boxes of empty kit trays* $\times$ *part supplies* $\times S \rightarrow \{0\} \cup \{1\}$: designates the status of boxes of empty kit trays, boxes of finished kit trays, and part supplies. 1 if empty, 0 if not empty.
- *boxfull*: *boxes of finished kit trays* $\times$ *boxes of empty kit trays* $\times$ *part supplies* $\times S \rightarrow \{0\} \cup \{1\}$: designates the status of boxes of empty kit trays, boxes of finished kit trays, and part supplies. 1 if full, 0 if not full.
- *topbox*: *box of empty kit trays* $\times S \rightarrow$ *empty kit trays*: designates the empty kit tray at the top of the box of empty kit trays.
- *grippertype*: *grippers* $\times S \rightarrow$ *empty kit trays* $\cup$ *finished kit trays* $\cup$ *parts*: designates the type of object the gripper can hold. One type of gripper can hold empty and finished kit trays. Another type of gripper can only hold parts.

3 Operators and Actions

Let us define planning operators and actions for the state-variable representation [**?**].

Definition 3.1. A planning operator is a triple *o=(name(o), precond(o), effects(o))* where:

- *name(o)* is a syntactic expression of the form $n(u_1, \ldots, u_k)$, where n is a symbol called an operator symbol, $u_1, \ldots, u_k$ are all of the object variable symbols that appear anywhere in *o*, and n is unique (i.e., no two operators can have the same operator symbol).
- *precond(o)* is a set of expressions on state variables and relations.
- *effects(o)* is a set of assignments of values to state variables of the form $x(t_1, \ldots, t_k) \leftarrow t_{k+1}$, where each t_i is a term in the appropriate range.

The kitting domain has eight operators.

- *takeEkt* (r, ekt, boxekt, g)
  - ;; robot r is equipped with gripper g to pick up the empty kit tray ekt which is at the top of the box of empty kit trays boxekt.
  - *precond*: *ektloc*(ekt)=boxekt, *topbox*(boxekt)=ekt, *rhold*(r)=$\{nil\}$, *rgrip*(r)=g, *boxempty*(boxekt)=0
  - *effects*: *ektloc*(ekt)$\leftarrow nil$, *rhold*(r)$\leftarrow$ekt

- *putEkt* (r, ekt, wtable)
  - ;; robot r puts down the empty kit tray ekt on the work table wtable.
  - *precond*: *ektloc*(ekt)=nil, *rhold*(r)=ekt, *topworktable*(wtable)=nil
  - *effects*: *ektloc*(ekt)←wtable, *rhold*(r)← $\{nil\}$, *topworktable*(wtable)←ekt
- *takeFkt* (r, fkt, wtable, g)
  - ;; robot r picks up the finished kit tray fkt from the work table wtable.
  - *precond*: *fktloc*(fkt)=wtable, *rhold*(r)=$\{nil\}$, *topworktable*(wtable)=fkt, *rgrip*(r)=g
  - *effects*: *fktloc*(fkt)← $\{nil\}$, *rhold*(r)←fkt, *topworktable*(wtable)← $\{nil\}$
- *putFkt* (r, fkt, boxfkt)
  - ;; robot r puts down the finished kit tray fkt in the box of finished kit trays boxfkt.
  - *precond*: *fktloc*(fkt)=$\{nil\}$, *rhold*(r)=fkt, *boxfull*(boxfkt)=0
  - *effects*: *fktloc*(fkt)←boxfkt, *rhold*(r)← $\{nil\}$
- *takeP* (r, p, ps, g)
  - ;; robot r uses gripper g to pick up the part p from the part supply ps.
  - *precond*: *ploc*(p)=ps, *rhold*(r)=$\{nil\}$, *rgrip*(r)=g
  - *effects*: *ploc*(p)← $\{nil\}$, *rhold*(r)←p
- *putP* (r, p, kt)
  - ;; robot r puts down the part p in the kit tray kt. Here, kt designates an empty kit tray or a partially full kit tray.
  - *precond*: *ploc*(p)=$\{nil\}$, *rhold*(r)=p
  - *effects*: *ploc*(p)←kt, *rhold*(r)← $\{nil\}$
- *attach* (r, g, chstation)
  - ;; robot r attaches gripper gwhich is originally in the changing station chstation.
  - *precond*: *gloc*(g)=chstation, *rgrip*(r)=$\{nil\}$
  - *effects*: *gloc*(g)← $\{nil\}$, *rgrip*(r)←g,
- *remove* (r, g, chstation)
  - ;; robot r removes gripper g and puts it down in the changing station chstation.
  - *precond*: *gloc*(g)=r, *rgrip*(r)=g
  - *effects*: *gloc*(g)←chstation, *rgrip*(r)=$\{nil\}$

4 States

The kitting problem is quite complex and contains far too many states to represent them all explicitly. This section will only describe the initial and goal states explicitly. The operators detailed in Section 3 are used to generate the other states as needed. The states s_0 and s_g are defined by sets of ground state variables and contain the following equipmets:

- A robot: r1
- A box of empty kit trays: bekt1
- Three empty kit trays: ekt1, ekt2, ekt3
- A box of finished kit trays: bfkt1
- A work table: wtable
- Part supplies A, B, and C with parts: psA, psB, psC
- A changing station: chstation
- Two grippers: g1, g2

Note: *grippertype* is a rigid relation since it does not vary from one state to another state. A rigid relation is invariant for a given planning domain. Hence, the rigid relation *grippertype* does not need to be stated in every state. For the domain presented, *grippertype* is defined as follows:

- *grippertype*(g1)= kit trays
- *grippertype*(g2)= parts

4.1 Initial State

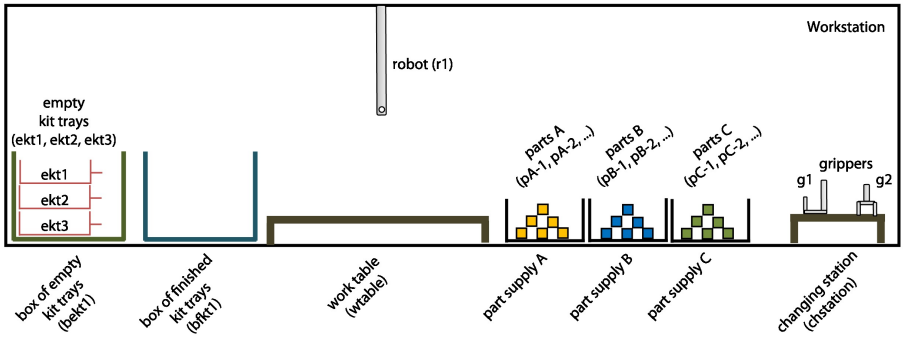

Figure 1: Initial state s_0.

The initial state s_0 (see Figure 1) is defined as follows:

- The robot is not holding anything.
- All the grippers are in the changing station.
- The box of empty kit trays is not empty.
- Part supplies are not empty.
- The box of finished kit trays may contain a number of finished kit tray n with $0 \leq n < maxfkt$, where $maxfkt$ is the maximum finished kit trays that can fit in the box of finished kit trays. If the box of finished kit tray is already full, the robot will not be able to place the current finished kit tray in the box of finished kit trays.
- There is no kit tray on the work table.

s_0 can be defined as the following set of ground state variables:

- $s_0 = \{$*rhold*(r1)=$\{nil\}$, *gloc*(g1)=chstation, *gloc*(g2)=chstation, *boxempty*(bekt1)=0, *boxempty*(psA)=0, *boxempty*(psB)=0, *boxempty*(psC)=0, *boxfull*(bfkt1)=0, *topworktable*(wtable)=$\{nil\}$, *topbox*(bekt1)=ekt1, *ektloc*(ekt1)=bekt1, *ektloc*(ekt2)=bekt1, *ektloc*(ekt3)=bekt1$\}$

4.2 Goal State

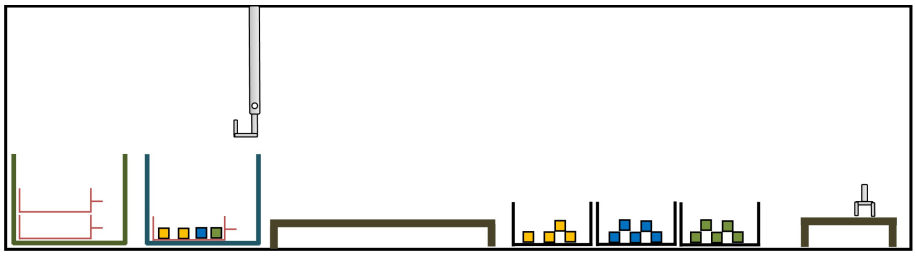

Figure 2: Goal state s_g.

The goal state s_g (see Figure 2) is reached when both of the following statements are achieved:

- The finished kit tray has two parts A, one part B and one part C.
- The finished kit tray is in the box of finished kit trays.

s_g can be defined as the following set of ground state variables:

- $s_g = \{$*rhold*(r1)=$\{nil\}$, *topworktable*(wtable)=$\{nil\}$, *ploc*(pA-1)=fkt1, *ploc*(pA-2)=fkt1, *ploc*(pB-1)=fkt1, *ploc*(pC-1)=fkt1, *fktloc*(fkt1)=bfkt1$\}$