1 Generating the ToMi dataset

The following three pseudo-code algorithms describe how ToMi is created. Alg. 1 describes how a balanced dataset is produced, assuming access to a story generator, by generating stories with rejection until there are the same number for each of the three types considered (true belief, false belief, second order false belief). The oracle that determines the type of the story is obtained by keeping track of all characters’s beliefs as the story is being generated. The next two algorithms describe the story generator. Alg. 2 shows how the core story body is created. Alg. 3 shows how random distractor statements are introduced, with both the comings and goings of a distractor agent, and irrelevant statements about containers and objects.

Algorithm 1: Produces balanced set of stories among the three target types (true belief, false belief, second order false belief), with a story generator (generateStory) and an oracle that knows the type of a story (typeStory)

**Data:** target number of stories of each type $N$

**Result:** set of stories $S$ of size $3 \times N$

$n[\text{tb}], n[\text{fb}], n[\text{sofb}] \leftarrow (0, 0, 0);$

$S \leftarrow \{\};$

**while** $\min(n[\text{tb}], n[\text{fb}], n[\text{sofb}]) < N$ **do**

    $\text{newStory} \leftarrow \text{generateStory};$

    $ts \leftarrow \text{typeStory(newStory)};$

    **if** $n[ts] < N$ **then**

        $n[ts] += 1;$

        $S \leftarrow S + \text{newStory};$

    **end**

**end**
Algorithm 2: Core story generation

Data: world with locations, containers, objects, agents; oracle tracking where things are

Result: story $s$

$s$ \leftarrow \""$;

$(l, l_{distr}) \leftarrow$ two random locations with no replacement;

$(o, c_1, c_2) \leftarrow$ random object in location $l$, container of $o$, container in $l \neq c_1$;

$(a_1, a_2) \leftarrow$ two random agents with no replacement;

(Sally, Anne) \leftarrow$ shuffle $(a_1, a_2)$;

$(a_1, a_2) \leftarrow$ shuffle $(a_1, a_2)$;

$s$ \leftarrow $a_1$ entering $l$;

$s$ \leftarrow $a_2$ entering $l$;

$nLocChanges \leftarrow$ randomChoice $(1, 2)$;

if $nLocChanges == 1$ then

$s_1$ \leftarrow $Anne moves o from c_1 to c_2$;

$s_2$ \leftarrow $Sally exits l$;

$(s_1, s_2) \leftarrow$ shuffle $(s_1, s_2)$;

$s$ \leftarrow $s_1$;

$s$ \leftarrow $s_2$;

return $s$ \quad // Single exit

end

// More than one agent move:

$(act_1, act_2, act_3) \leftarrow$ shuffle $(move_{obj}, change_{loc}, change_{loc})$;

if $act_1 == move_{obj}$ then

$s$ \leftarrow $Anne moves o from c_1 to c_2$;

else

$s$ \leftarrow $Sally exits l$;

end

if $act_2 == move_{obj}$ then

$s$ \leftarrow $Anne moves o from c_1 to c_2$;

else if Sally in $l$ then

$s$ \leftarrow $Sally exits l$;

else

$s$ \leftarrow $Sally enters randomChoice(l, l_{distr})$;

end

if $act_3 == move_{obj}$ then

$s$ \leftarrow $Anne moves o from c_1 to c_2$;

else if $flipCoin == tail$ then

$s$ \leftarrow $Anne exits l$;

end

$s$ \leftarrow $Sally enters randomChoice(l, l_{distr})$;

end
Algorithm 3: Distractor agent entries and distractor statements

**Data:** main story $s$ with location $l$, distractor location $l_{distr}$, agents $(a_1, a_2)$; set of world containers $C$, set of world objects $O$; distractor agent $a_3$

**Result:** updated story $s$

$nDistractorEntrances \leftarrow \text{randomChoice}(0, 1, 2)$;

$\text{insertionIndices} \leftarrow \text{random choice of } nDistractorEntrances \text{ from 1 to length}(s)$;

If $nDistractorEntrances > 0$ then

$s_1 \leftarrow "a_3 \text{ enters randomChoice}(l, l_{distr})"$;

$s \leftarrow \text{insert } s_1 \text{ in } s \text{ at min index of } insert\text{ionIndices}$;

End

If $nDistractorEntrances > 1$ then

$s_2 \leftarrow "a_3 \text{ exits } a_3\text{'s current location}"$;

$s \leftarrow \text{insert } s_2 \text{ in } s \text{ at max index of } insert\text{ionIndices}$;

End

$nDistractorStatements \leftarrow \text{randomChoice}(0, 1, 2)$;

$\text{insertionIndices} \leftarrow \text{random choice of } nDistractorStatements \text{ from 1 to length}(s)$;

For $i \leftarrow 1$ to $nDistractorStatements$ do

$a \leftarrow \text{randomChoice}(a_1, a_2, a_3)$;

$\text{thing} \leftarrow \text{randomChoice}(C \cup O)$;

$s_1 \leftarrow "a (\text{likes} | \text{dislikes} | \text{loves} | \text{hates}) \text{ the thing}"$;

$s \leftarrow \text{insert } s \text{ in } s \text{ at } i\text{-th index of } insert\text{ionIndices}$;

End